Correspondência de Pontos em Formas 3D Baseada em Aprendizagem Profunda Multivisão

Correspondência de Pontos em Formas 3D Baseada em Aprendizagem Profunda Multivisão

Alexandre Soares da Silva

Tese apresentada à Faculdade de Computação (FACOM) da Universidade Federal de Mato Grosso do Sul (UFMS), como parte dos requisitos para obtenção do título de Doutor em Ciência da Computação.

ORIENTADOR: Prof. Dr. Paulo Aristarco Pagliosa

Campo Grande - MS
2022

Agradecimentos

Em primeiro lugar, agradeço à Deus por permitir o privilégio de poder trilhar essa caminhada. Aos familiares pelo apoio em todos os momentos, especialmente quando os desafios encontrados durante o trajeto exigiram um estado constante de resiliência.

Ao meu orientador, Dr. Paulo Pagliosa pela grande ajuda e confiança depositada. Ao meu co-orientador, Dr. Eraldo Luís Rezende Fernandes, pela ajuda e tempo dedicado. A todos que de alguma maneira contribuíram com as discussões que auxiliaram a alcançar os resultados obtidos, em especial ao estudante Juan e professor Edson Takashi Matsubara, do Laboratório de Inteligência Artificial da FACOM.

Agradeço também ao programa Prodoutoral da CAPES, pelo apoio financeiro concedido. A todos os meus profundos agradecimentos.

Resumo

Silva, A. S. *Correspondência de Pontos em Formas 3D Baseada em Aprendizagem Profunda Multivisão.* Tese (Doutorado em Ciência da Computação), Universidade Federal de Mato Grosso do Sul, 2022.

Na área de processamento geométrico, diversas técnicas propostas na literatura requerem que sejam estabelecidos pares de *pontos de correspondência* entre duas ou mais superfícies, isto é, dado um ponto sobre uma superfície *fonte*, é preciso associar qual é o ponto sobre uma superfície *alvo* que corresponde ao ponto dado. As aplicações incluem reconstrução de superfícies, parametrização cruzada, transferência de pose, transferência de texturas ou animações, reconhecimento e busca de formas, entre outras. A definição de uma função de mapeamento entre duas formas, mesmo para um número discreto de pontos característicos, nem sempre envolve somente relações geométricas ou estruturais, mas também relações semânticas. Uma vez que tal mapeamento em geral não pode ser diretamente expresso por abordagens puramente axiomáticas, em vários métodos de processamento geométrico a indicação de um conjunto inicial de pontos de correspondência é efetuada manualmente, através de processos que podem ser laboriosos e sujeitos a erros. De fato, descobrir relações semânticas entre formas quaisquer sem qualquer interação do usuário tratava-se de um problema ainda em aberto. Modelos de aprendizagem de máquina, em especial aprendizagem profunda, têm evoluído por sua capacidade de utilizar grandes conjuntos de dados para estimar a solução de problemas em diversas áreas do conhecimento, inclusive processamento geométrico. Este trabalho apresenta um método que utiliza aprendizagem multivisão profunda como parte do processamento responsável por encontrar *automaticamente*, isto é, sem a intervenção direta do usuário, pontos de correspondência entre superfícies de formas 3D, representadas por malhas de triângulos. O método é dividido em 2 componentes: treinamento e correspondência. O primeiro trata-se de um treinamento multivisão que aprende, com o auxílio de uma CNN, a detectar pontos de interesse em imagens 2D oriundas de malhas de triângulos dos conjuntos de treinamento. O último, utiliza o resultado do treinamento para inferir correspondências semânticas com pontos de interesse (vértices) em formas 3D. A descoberta desses pontos não requer novo treinamento e nem interação humana durante o pipeline de correspondência.

Palavras-chave: *processamento geométrico*, *formas 3D*, *correspondência de pontos*, *aprendizagem profunda multivisão*.

Abstract

Silva, A. S. *Point Correspondence between 3D Shapes Based on Deep Multiview Learning*. Thesis (PhD in Computer Science), Universidade Federal de Mato Grosso do Sul, 2022.

In the field of geometric processing, several techniques proposed in the literature require the establishment of *correspondence points* between two or more surfaces, that is, given a point on a *source* surface, it is necessary to associate which point on a *target* surface corresponds to the given point. Applications include surface reconstruction, cross-parameterization, pose transfer, texture or animation transfer, shape recognition and search, among others. Defining a mapping function between two shapes, even for a discrete number of characteristic points, does not always involve only geometric or structural relationships, but also semantic relationships. Since such a mapping cannot generally be directly expressed by purely axiomatic approaches, in various geometric processing methods, the indication of an initial set of correspondence points is manually performed, through processes that can be laborious and error-prone. In fact, discovering semantic relationships between any shapes without any user interaction is still considered an open problem. Machine learning models, especially deep learning, have evolved due to their ability to use large datasets to estimate the solution of problems in various areas of knowledge, including geometric processing. This work presents a method that uses deep multiview learning as part of the processing responsible for finding *automatically*, that is, without direct user intervention, correspondence points between 3D shape surfaces represented by triangle meshes. The method is divided into two components: training and correspondence. The former is a multiview training that learns, with the aid of a CNN, to detect interest points in 2D images derived from triangle meshes of the training set. The latter uses the result of the training to infer semantic correspondences with interest points (vertices) in 3D shapes. The discovery of these points does not require new training or human interaction during the correspondence pipeline.

Keywords: *geometric processing*, *3D shapes*, *point correspondence*, *deep multiview learning*.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Motivação e Justificativa

Em processamento geométrico, a determinação de *correspondências* entre formas tridimensionais é um problema importante em diversas aplicações de computação gráfica, em áreas tais como animação, jogos digitais, projeto assistido por computador (CAD) e visualização, entre outras. Genericamente, o problema consiste em, dadas duas ou mais formas geométricas, estabelecer uma relação significativa entre seus elementos, parcial ou totalmente. Dentre as aplicações, pode-se citar reconstrução de superfícies, parametrização cruzada, transferência de informação, detecção de simetria, reconhecimento e busca de formas e modelagem estatística de formas. Exemplos dessas aplicações são ilustradas na Figura 1.1.

A definição precisa do que é uma correspondência significativa depende da aplicação em particular, variando desde o caso menos complicado de como identificar partes das formas que são geometricamente similares, até o problema mais complexo de relacionar elementos que representam as mesmas partes ou exercem a mesma função. Na literatura pode-se encontrar uma diversidade de abordagens orientadas a conteúdo para a solução de problemas de correspondência entre formas, concentrando-se nas similaridades geométricas e estruturais entre as formas envolvidas. Todavia, tais critérios de similaridade podem ser inadequados em cenários em que as partes correspondentes são geometricamente dissimilares. Sob estas circunstâncias, o problema vai além de uma análise puramente geométrica, tal que a determinação de correspondências pode envolver o entendimento semântico tanto da estrutura global quanto das funcionalidades de cada uma das partes das formas. Tal análise semântica usualmente requer a utilização de conhecimento prévio: para encontrar uma correspondência entre as partes que possa ser dissimilar geometricamente, é preciso levar em conta lembranças do reconhecimento de partes similares e assim fazer uso desse conhecimento para estabelecer uma correspondência entre partes desconhecidas. Incorporar esse processo de reconhecimento na correspondência entre formas resulta em uma abordagem orientada a conhecimento. Não obstante, a fim de colocar a questão em termos mais específicos, deve-se considerar não somente os aspectos estruturais e semânticos que caracterizam determinado problema de correspondência, mas também as representações das formas de entrada e da própria correspondência adotadas por distintos métodos de solução do problema [128].

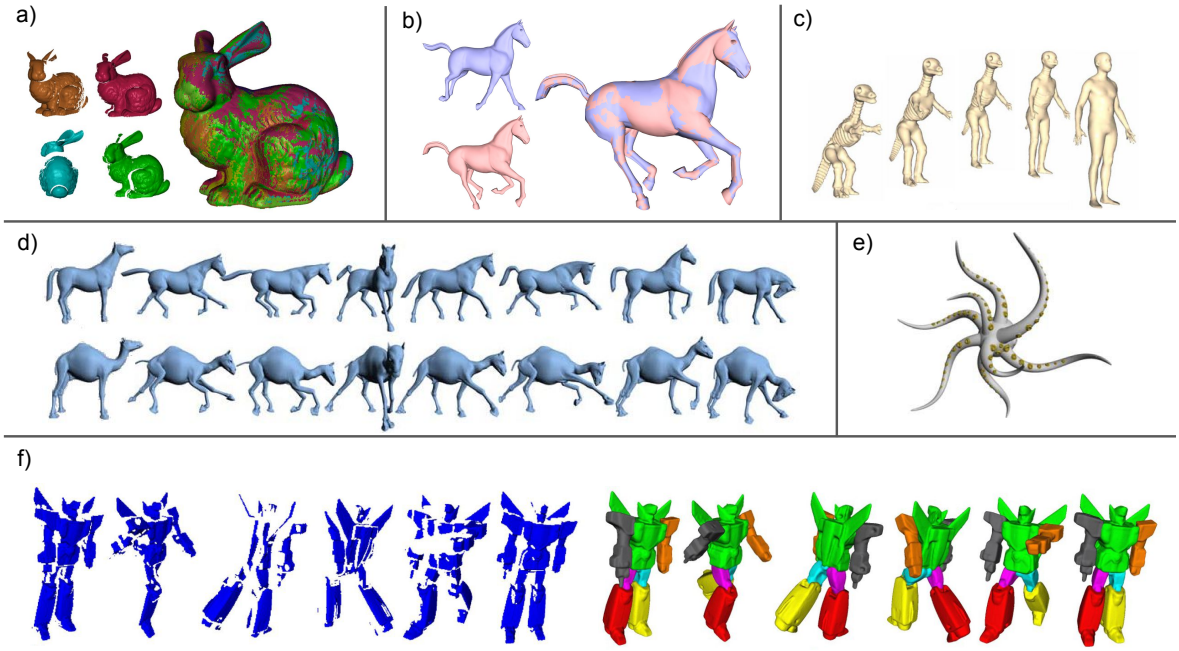

Figura 1.1: Exemplos de aplicações que utilizam métodos de correspondência: (a) um conjunto de digitalizações (à esquerda) é alinhado rigidamente para reconstruir a forma do coelho (à direita); (b) um cavalo em duas diferentes poses (à esquerda) é alinhado não rigidamente (à direita); (c) um esqueleto de dinossauro é transformado em um humano; (d) o movimento definido em um cavalo (acima) é transferido ao camelo (abaixo); (e) uma aplicação de correspondência parcial: as ventosas nos tentáculos do polvo detectadas como similares (em amarelo); e (f) um conjunto de digitalizações de um objeto em movimento (em azul à esquerda) fornece um único modelo reconstruído no qual o movimento é definido (à direita). Imagens retiradas do artigo de Van Kaick et al. [128].

Por vezes independentemente da aplicação, diversos métodos em processamento geométrico requerem que seja estabelecida uma *correspondência inicial* entre pares de *pontos característicos* de duas ou mais formas. Informalmente, um ponto característico pode ser entendido como um que possui uma propriedade, por exemplo, curvatura, que destacadamente o diferencia dos demais pontos em sua vizinhança. Assim, dada uma forma *origem* e uma forma *alvo*, é necessário se determinar, para um conjunto discreto de pontos característicos da forma fonte, quais são os pontos correspondentes da forma alvo. Estes pontos são denominados *pontos de controle* ou *pontos de interesse*. Embora comumente assumido que a correspondência inicial seja parte da entrada de um determinado método de processamento geométrico, o problema é tão difícil como o de descobrir uma correspondência total entre todos os pontos, aqui considerados como sendo os elementos de interesse, das formas fonte e alvo, uma vez que, para tal, deve-se levar em consideração a estrutura global de ambas as formas [148].

No tocante à representação das formas, uma das alternativas mais amplamente utilizadas em computação gráfica para modelar geometricamente a superfície de uma forma é a malha de triângulos. Teoricamente, qualquer superfície pode ser discretamente aproximada por uma malha de triângulos. Além disso, unidades de processamento gráfico (GPUs) são capazes de renderizar malhas de triângulos muito eficientemente. Portanto, neste trabalho são utilizadas malha de triângulos para representar (a superfície de) uma forma. Assim, o problema de correspondência inicial aqui reduz-se a determinar, para um conjunto de *vértices* característicos de uma malha alvo, quais são os pontos correspondentes sobre uma malha origem, como exemplificado na Figura 1.2.

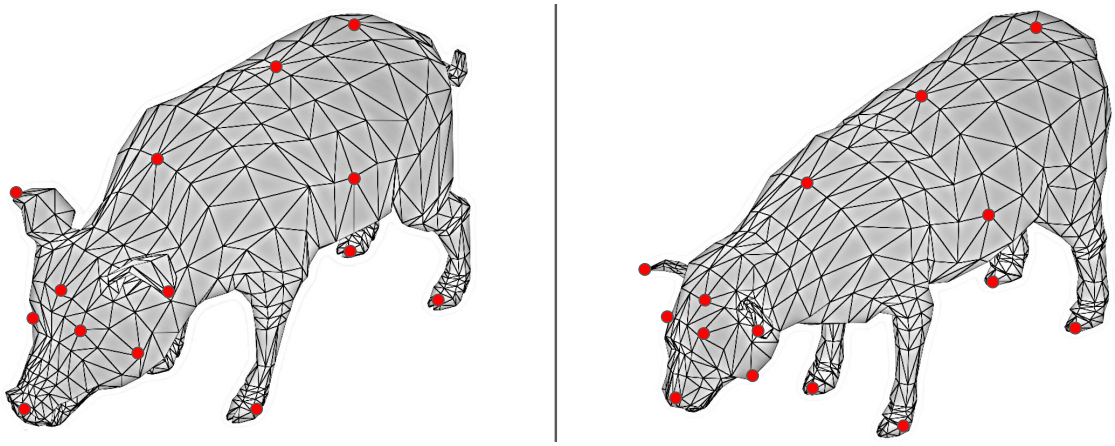

Figura 1.2: Exemplo de duas malhas e possíveis pontos de controle entre elas.

Um exemplo clássico de método que toma como entrada um conjunto inicial de pontos de controle foi proposto por Medalha et al. [90]. Trata-se de uma técnica de *morphing* de *malhas dinâmicas*, i.e., da transformação suave de uma forma animada em outra em um certo intervalo de tempo. Nesse contexto, uma malha dinâmica é uma sequência de malhas de triângulos que representam poses distintas de uma forma ao longo do tempo de *morphing*. Assim, a transformação citada leva em conta não somente a geometria, mas também o movimento de cada uma das formas, conforme ilustrado na Figura 1.3. Contudo, ainda é necessária a intervenção do usuário para definir o conjunto de pontos de controle iniciais, para computar uma aproximação grosseira da geometria da malha fonte na malha alvo a partir de restrições definidas por esses pontos. Mesmo contando com uma ferramenta gráfica para definição dos pontos, ainda é um processo laborioso, de tentativa e erro e do qual a parametrização final depende diretamente: pontos de controle que não são correspondências significativas podem inviabilizar o processo como um todo. Portanto, o principal inconveniente do método é exatamente a especificação manual dos pontos de controle iniciais que guiam o processo de parametrização.

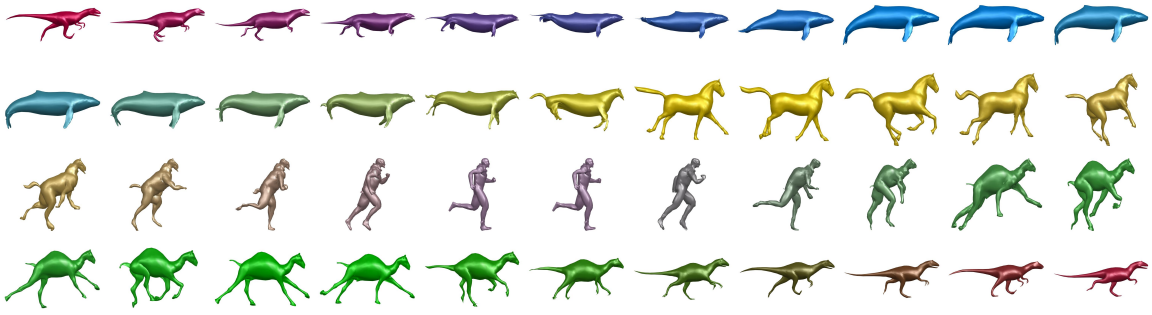

Figura 1.3: Exemplo de *morphing* de malhas dinâmicas: modelo animado de um alossauro que se transforma em outros modelos de outros e volta a ser um alossauro ao longo de um intervalo de tempo (retirado de [90]).

Definir relações semânticas entre duas formas sem a interação do usuário é considerado um problema ainda em aberto. Sem o auxílio do usuário, a performance dos métodos de correspondência automática existentes dependem de um alinhamento inicial que geralmente não toleram grandes variações entre as formas [148].

Em soluções axiomáticas ou orientadas a modelo, em que determinada hipótese geométrica é afirmada e então validada através de algum esquema numérico, uma ou

mais grandezas geométricas são usadas para codificar informações geométricas locais na vizinhança de um ponto da forma (descritores pontuais), tais como a orientação da normal, curvatura, ou propriedades de onda ou calor. Outro tipo de grandeza geométrica são as relações globais entre pares de pontos (descritores emparelhados), as quais incluem distâncias geodésicas ou de difusão, por exemplo. Dado um par de formas, um mapa de correspondência denso entre elas busca minimizar a discrepância entre tais descritores.

Outra linha de pesquisa, de algoritmos orientados a dados, apoiam-se em ferramentas modernas de aprendizagem de máquina, ao invés de axiomaticamente modelar a classe de deformações e as propriedades geométricas das formas envolvidas, as propriedades são deduzidas a partir dos próprios dados. Entre estas propriedades estão generalizações aprendíveis da assinatura de núcleo de calor, assim como modelos que interpretam correspondência como um problema de classificação.

Alguns métodos atuais generalizam redes neurais convolucionais (CNNs) para estruturas não Euclidianas com o objetivo de aprender descritores que obtenham resultados mais precisos durante o processo de aprendizagem.

Nas redes neurais profundas, geralmente a representação da geometria de uma malha dá-se geralmente através dos chamados *descritores de características* — ou ainda *features* ou simplesmente descritores, dependendo do autor. Um descritor é basicamente uma representação numérica compacta da forma de um objeto 3D, no caso desta tese, malha de triângulos. Enquanto descritores globais representam cada forma como um todo através de um único vetor de valores reais, os descritores locais descrevem características de regiões ao redor de pontos de referência de cada forma. Em ambos os casos, os descritores formam um espaço, não necessariamente Euclidiano, em que a métrica de distância entre estes descritores é significativa. Distâncias no espaço de descritores idealmente deveriam retratar as dissimilaridades entre objetos que estes descritores representam. Tais dissimilaridades podem ser geométricas, topológicas, ou semânticas [66]. Nesse texto, a fim de evitar confusões utilizaremos o termo *features* para denotar características representadas por descritores de forma, ou mais especificamente, propriedades geométricas e topológicas das formas 3D.

Um denominador comum nas abordagens orientadas a dados tem sido o regime de treinamento supervisionado [27, 80, 88, 134, 44] — todas contam com exemplos que dependem da *verdade absoluta*, ou *gabarito*, das correspondências entre as formas empregadas. A maior desvantagem deste tipo de configuração de treinamento supervisionado é o fato de que exemplos disponíveis com verdade absoluta das correspondências são raros [47].

Do ponto de vista de modelagem probabilística, o problema de aprendizagem de features pode ser interpretado como uma tentativa de extrair um conjunto de variáveis latentes (ocultas) a partir dos dados. Essas variáveis latentes fornecem outro meio, em uma representação simplificada, de representar os dados e formam um espaço conhecido como espaço latente. Muitas abordagens atuais empregam métodos baseados na decomposição própria (*eigen-decomposition*) do Laplaciano [110, 6, 122, 80, 91] para comparar features de superfícies de formas deformáveis. Porém, tais técnicas são projetadas para pares de formas cujas deformações entre elas são aproximadamente isométricas, i.e., quando envolvem pares de malhas a partir de determinado nível de dissimilaridade, acabam considerando deformações muito genéricas muitas das vezes

não consistentes com a intuição humana de correspondência.

Outrossim, apesar das malhas de triângulos encontrarem-se imersas em um domínio tridimensional, é possível também empregar técnicas multivisão mapeando diversos pontos de vista das malhas para imagens bidimensionais naturalmente adequadas às CNNs. Abordagens multivisão recentes demonstraram vantagens em tarefas de classificação, identificação e rotulação de partes de formas. Uma das vantagens, é a possibilidade de ter à disposição grandes conjuntos de dados invariantes a translação. De fato, trabalhos como os de Zhu et al.[152], Kalogerakis et al. [55] a SimNet presente em [38] obtiveram resultados do estado da arte em tarefas classificação de formas, segmentação e transferência de deformação, respectivamente. No Capítulo 2 é realizada uma revisão bibliográfica destas e das principais possibilidades que servem de inspiração para solução do problema de correspondência entre formas 3D.

Nesta tese, propõe-se um método eficaz para detecção de pontos de correspondência semântica entre malhas de triângulos, a representação mais comum de uma superfície em processamento geométrico. Realiza-se tal tarefa através de 2 etapas distintas: treinamento e correspondência. O treinamento, realizado uma única vez sobre conjuntos de malhas 3D, aprende a detectar features correlacionadas à um gabarito personalizado, empregando a abordagem multivisão. Tal gabarito descreve os pontos semânticos desejados. A partir disso, o pipeline de correspondência é capaz de encontrar pontos de correspondências com esse gabarito, em malhas de triângulo de variadas complexidades ou oriundas de conjuntos de dados diversos, desde que as malhas avaliadas pertençam à classes de formas semelhantes as existentes no treinamento. Toda etapa de correspondência ocorre sem intervenção manual do usuário durante o processo.

No contexto desse trabalho, que emprega aprendizagem de máquina, o termo *classes de formas suportadas* ou *classes de formas semelhantes* refere-se à classes de animais e humanos específicos, cujas características anatômicas são similares ou iguais às características anatômicas das formas englobadas pelo processo de treinamento. Mais especificamente, o método apresentado trabalha com formas pertencentes às classes de humanos, felinos, equinos, camelídeos e terópodes. A composição precisa do conjunto de treinamento é fornecida mais adiante no Capítulo 4.

Uma característica intrínseca dos tipos de animais e humanos pertencentes às classes suportadas, é a simetria bilateral sagital, presente em seres humanos e na maioria dos vertebrados. O plano sagital é um plano imaginário que parte o organismo ao meio dividindo-o em direita e esquerda, formando duas partes simétricas. Lado esquerdo ou direito aqui refere-se ao contexto de orientação espacial, i.e., esquerda é o que está relacionado à, situado ou pertencente ao lado do corpo no qual o coração está majoritariamente situado. Para o problema de correspondência aqui proposto, cujas formas são dotadas de simetria bilateral sagital, lida-se com algumas features ambíguas. Portanto, parte da solução engloba também uma abordagem para classificar features simétricas como à esquerda ou direita da forma.

1.2 Hipótese

Diante do exposto, considerando malhas arbitrárias pertencentes ao conjunto de formas suportadas, a correspondência semântica de pontos de interesse malhas de triângulos depende não apenas da geometria, mas também do contexto e relações semânticas entre suas partes. Apesar dos avanços em trabalhos de aprendizagem com o objetivo de deduzir tais relações semânticas, os mesmos esbarram na necessidade de treinamento sobre uma grande quantidade de malhas que possuam suas respectivas verdades absolutas disponíveis, exigem deformações aproximadamente isométrica entre as malhas, ou requerem intervenção manual do usuário entre os cálculos de correspondência. Contudo, abordagens multivisão atualmente têm obtido sucesso em determinados problemas geométricos como classificação e segmentação de formas. Sendo assim, a hipótese desta tese é que *aprendizagem de máquina pode ser empregada eficazmente em uma ferramenta computacional para detecção de pontos característicos e correspondência de pontos, visto serem essas operações inerentemente semânticas. Correspondência de pontos pode ser obtida a partir de conjuntos de pontos característicos, convenientemente determinados para classes de formas semelhantes, os quais podem ser determinados pelas análise de imagens dessas formas por redes neurais profundas.* Embora existam trabalhos que empreguem aprendizagem de máquina para a solução de problemas geométricos, até onde temos conhecimento, o problema de correspondência de pontos relatado, assim como a utilização de uma abordagem multivisão como parte da solução trata-se de algo inédito.

1.3 Objetivos e Contribuições

O objetivo geral do trabalho é a proposição de um método para a detecção de pontos correspondência entre superfícies representadas por malhas de triângulos, sem intervenção direta do usuário. Os objetivos específicos podem ser definidos como:

- Facilitar a definição de pontos de correspondência iniciais entre pares de formas em técnicas de *morphing* dentro dos conjuntos de formas suportadas pelo método.
- Possibilitar que os pontos semânticos desejados (gabarito) possam ser definidos de acordo com a aplicação.
- Permitir a desambiguação esquerda/direita de pontos de correspondência simétricos em formas dotadas de simetria bilateral sagital.

A principal contribuição é a facilitação ou possibilidade de automatização da tarefa de correlacionar pontos específicos entre a superfície de formas dentro dos conjuntos de classes cobertas pelo método, sem exigência de orientação específica ou de isometria nas deformações entre as formas. Em relação as contribuições específicas, estas podem ser resumidas como a seguir:

- Criação de uma ferramenta gráfica para visualização e edição dos pontos de correspondência, que permite também renderizar imagens de malhas para treinamento de redes neurais em aplicações do tipo multivisão.

- Para ferramentas CAD e motores 3D utilizados em jogos digitais, pode-se aplicar o método para automatizar, ao menos parcialmente, transferência de texturas, estimativa de poses, transferência de pontos de fixação de um modelo de ator para outro qualquer, transferência de nós de animação entre formas 3D, dentre outras tarefas de correlação entre atores.

- Através de pontos de correspondência específicos, é possível determinar um conjunto de coordenadas para automaticamente orientar e alinhar malhas em relação à algum ponto de referência.

- Se adicionadas relações de conectividade entre os pontos de correspondência, o método pode servir como base para classificação automática de formas em classes predeterminadas através da análise das proporções entre as arestas dos pontos de correspondência de cada forma.

1.4 Organização do Texto

O restante do texto é organizado como segue. O Capítulo 2 realiza uma revisão bibliográfica dos problemas de processamento geométrico que envolvem correspondência de pontos entre formas tridimensionais, bem como de métodos que versam sobre o emprego de abordagens baseadas em aprendizagem profunda para solução de problemas similares de processamento geométrico. Os trabalhos relacionados serviram como inspiração e base para a solução do problema proposto. Em seguida, o Capítulo 3 apresenta a metodologia, o processo de implementação assim como a descrição dos problemas encontrados durante o desenvolvimento da solução. No Capítulo 4 relatamos os principais experimentos realizados para avaliar os resultados obtidos. Por fim, no Capítulo 5 como conclusão discute-se os resultados obtidos, problemas enfrentados e as algumas possibilidades de trabalhos futuros.

Capítulo 2

Revisão da Literatura

2.1 Considerações Iniciais

Nesse capítulo são relatados os principais trabalhos que utilizam aprendizado profunda para tentar solucionar problemas de processamento geométrico, destacando-se métodos para classificação, segmentação e correspondência entre formas.

Uma parte da revisão literária também abrangeu algumas técnicas baseada em features, grafos ou similaridade visual. Em muitos trabalhos são abordadas técnicas mais recentes e relevantes relacionadas à transferência de deformação entre formas 3D não rígidas considerando variadas técnicas existentes. A revisão de tais trabalhos serviu como orientação e inspiração para a solução do problema de pontos de correspondência.

A Seção 2.2 inicia com um breve histórico dos trabalhos pioneiros na utilização de redes neurais profundas para analisar problemas cuja estrutura subjacente é não-Euclidiana. Em seguida, são mostrados alguns dos principais avanços na criação e aperfeiçoamento de descritores de features, a estrutura mais frequentemente utilizada para representar malhas tridimensionais como entrada destas redes neurais. Paralelamente, são introduzidos trabalhos classificados como *geometric deep learning* (GDL) Parte das abordagens elencadas na revisão da literatura , mesmo relacionadas à problemas geométricos semelhantes, não fizeram parte da solução final adotada. Tais trabalhos encontram-se disponíveis no Apêndice C, juntamente com o relato dos experimentos que corroboraram nessa decisão. Mesmo que essas técnicas revisadas não pertençam diretamente à solução proposta, servem de norte e inspiração para outros trabalhos que porventura desejem seguir por esse caminho. Mas nesse capítulo optou-se por elencar trabalhos relacionados à aprendizagem profunda multivisão, técnica que é efetivamente utilizada nesse trabalho.

Na Seção 2.3, faz-se uma revisão de trabalhos que empregam redes neurais profundas para classificação de formas tridimensionais, havendo destaque para redes neurais convolucionais (CNNs, *convolutional neural networks*), que obtiveram bons resultados neste tipo de tarefa. Aborda-se também propostas que se utilizam de métodos multivisão, pelo fato de que estes adaptam-se mais facilmente às CNNs e possuírem uma grande quantidade de dados disponíveis para treinamento.

2.2 Descritores de Formas e GDL

Nas últimas décadas, a aplicação de técnicas de inteligência artificial sobre dados tridimensionais tornou-se um campo de estudo próprio, com diferentes terminologias utilizadas através de diversos trabalhos publicados [138, 14, 33, 20, 2, 106, 105, 66, 59]. Em particular, as redes neurais profundas transformaram-se em ferramentas capazes de solucionar ou melhorar soluções de uma gama de problemas de áreas diversas como visão computacional, linguagem natural, comportamento humano, dentre outras. Vale ressaltar também que essas ferramentas foram mais bem sucedidas em dados cuja estrutura subjacente é Euclidiana ou pode ser representada por grades. Dados 3D digitalizados por diferentes tipos de dispositivos podem adotar diferentes formas variando tanto na estrutura como também em suas propriedades. Alguns autores, classificam diferentes representações de dados 3D em duas principais categorias: dados estruturados Euclidianos e dados não-Euclidianos [2], como ilustrado na Figura 2.1.

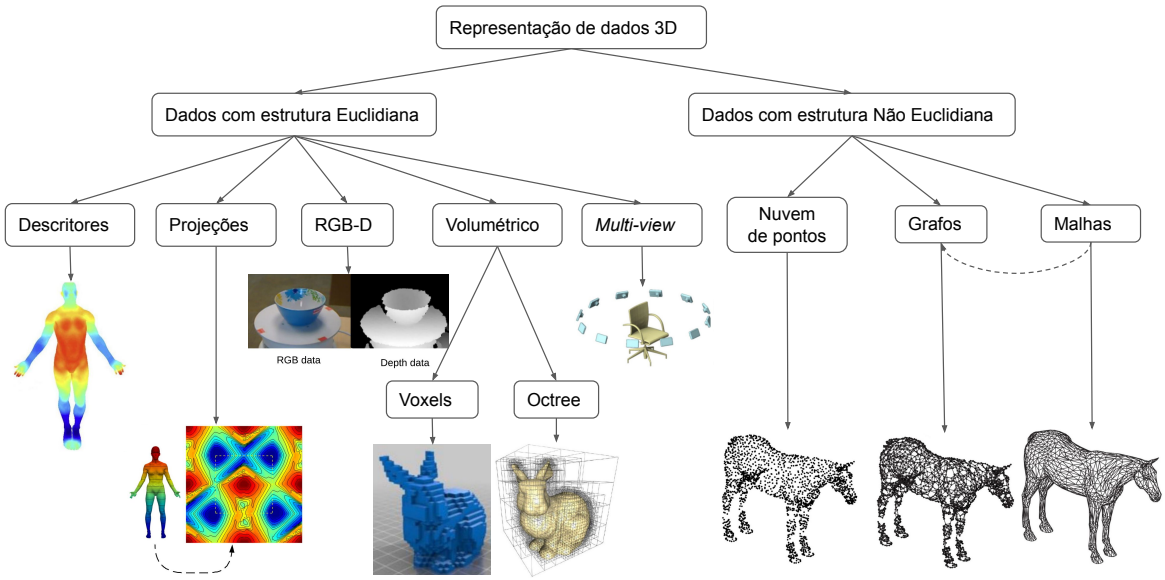

Figura 2.1: Diversas representações de dados tridimensionais (retirado de [2]): representações Euclidianas (descritores [80]; projeções [86]; RGB-D [67]; volumétricas: voxels [151] e octrees [131]; multivisão [120]) e não-Euclidianas (nuvem de pontos, grafos e malhas) [20].

Aos domínios não-Euclidianos pertencem duas estruturas prototípicas importantes em computação e processamento geométrico: grafos e superfícies que são variedades Riemannianas, que apesar de originadas de áreas diferentes da matemática, possuem características em comum. A natureza não Euclidiana destes domínios implica na falta de propriedades familiares como parametrização global, sistema comum de coordenadas, estrutura de espaço vetorial, ou invariância de translação. Portanto, operações como o processo de convolução, bem conhecido em domínios Euclidianos, não pode ser definido da mesma forma em domínios não Euclidianos [20].

Diante da utilização das redes neurais profundas em tarefas que lidam com fala, imagens, ou vídeo, houve crescimento no interesse em aplicar aprendizagem profunda também em dados cujo domínio subjacente é não-Euclidiano. Bronstein et al. [20] classificaram tal abordagem como *Geometric Deep Learning* (Aprendizagem Geométrica Profunda). Essas redes profundas têm auxiliado na solução de tarefas complexas como localização ou classificação de objetos, segmentação semântica, estimativa de mo-

vimento, correspondência entre formas, por exemplo [20]. Assim, existe interesse dos pesquisadores em permitir que estruturas de dados representadas inicialmente como grafos ou malhas de triângulos possam ser analisadas por redes neurais profundas. A representação da entrada de malhas tridimensionais nas redes neurais dá-se geralmente através dos chamados descritores de formas. Um descritor atribui para cada ponto da forma um vetor em um espaço de features (uni ou multidimensional) representando as propriedades geométricas locais e globais do ponto relevantes para determinada tarefa. Esta informação é então utilizada em tarefas de mais alto nível. Quando na escolha de um descritor de features, leva-se em conta quais propriedades da forma o descritor deve capturar e sobre quais transformações da forma eles continuarão invariantes ou, pelo menos, insensíveis á transformação [80]. Exemplos comuns desta família de descritores são assinatura de *kernel* de calor (HKS, *heat kernel signature*) e assinatura de *kernel* de onda (WKS, *wave kernel signature*).

As primeiras investigações envolvendo técnicas de inteligência artificial, mais especificamente, redes neurais agindo sobre modelos cuja estrutura subjacente é não Euclidiana, apareceram na literatura com os trabalhos pioneiros de Gori et al. [42] e Scarselli et al. [111]. Ambos trabalharam com um modelo neural chamada GNN (*graph neural network*), que estende redes neurais recursivas e pode ser utilizado em diversas classes de grafos ou problemas modelados como tal. Até então, aplicações associadas a aprendizagem de máquina lidavam com estes tipos de problemas aplicando algum procedimento de preprocessamento, que transformava o grafo em uma representação mais simples (e.g., vetores ou sequências de números reais). Entretanto, informações valiosas podem ser perdidas durante o processamento e, como consequência, a aplicação pode sofrer com generalização e desempenho ruim [42].

Para representar dados de uma forma 3D em que determinadas propriedades fossem invariantes, Aflalo et al. [1] fazem uso de conceitos similares aos encontrados na área de processamento de sinais clássico.

O conceito de descritor de features tornou-se fundamental em análise de formas, e descritores baseados no operador Laplaciano mostraram-se eficientes, invariantes à isometria e de certa forma lidam bem com uma variedade de transformações. Em um dos trabalhos mais relevantes sobre esse assunto, Litman et al. [80] formularam uma família genérica de descritores espectrais paramétricos (Figura 2.2). Afim de que o descritor seja otimizado visando uma tarefa específica, este deveria levar em conta as estatísticas do corpus das formas as quais ele é aplicado ("sinal") e aquelas da classe de transformações as quais ele fez-se insensível ("ruído"). Embora tais estatísticas sejam complexas para modelar axiomaticamente, é possível aprende-las por exemplos. No trabalho é apresentado um modelo de aprendizado para construção de descritores espectrais otimizados relacionados à métrica de aprendizado de Mahalanobis.

Barra et al. [10] propuseram um método que seleciona features mais significativas das formas a partir de um grande conjunto delas para classificação e identificação de formas 3D (*3D shape retrieval*). Em sua abordagem, a construção do conjunto representativo é considerada uma tarefa de aprendizagem de máquina que utiliza uma técnica de aprendizado supervisionado para identificar os conceitos semânticos alto-nível das classes. No intuito de lidar com uma representação semântica capaz de contemplar características globais e locais, foi adotado um descritor de features que combinasse a estrutura global da forma 3D, codificado em um grafo topológico, mais especificamente, como um grafo de Reeb estendido (ERG, *Extended Reeb graph*) combinado com uma

Figura 2.2: Exemplo de descritores de forma de 16 dimensões baseadas no kernel de calor (linha 1), kernel de onda (linha 2) e kernel treinado (linha 3) propostos por Litman et al. [80]. A figura mostra a distância euclidiana normalizada entre um descritor em um ponto de referência (pulso, barriga ou peito) e descritores calculados nos pontos restantes da mesma forma sintética. A cor azul escuro indica pequena distância enquanto que a cor vermelha indica grande distância. Uma escala em comum do mapa de cores foi utilizada em cada linha para cada descritor, e foi saturada na distância média da forma mais à direita de cada grupo (i.e., pelo menos metade de cada forma é sempre vermelha). As formas de cada grupo são originárias do repositório TOSCA, sua isometria aproximada e uma forma humana escaneada do conjunto de dados SCAPE (esquerda, centro e direita) (retirado de [80]) .

descrição geométrica local, representada pelos índices harmônicos esféricos das partes da forma.

Litman et al. [79] também apresentaram um método de aprendizado supervisionado de descritores de formas para aplicações cujas tarefas envolvessem identificação de formas 3D. A proposta envolvia um novo arcabouço seguindo o paradigma *bag-of-features* (BoF) supervisionado, em que o treinamento discriminativo ocorre já na etapa de construção do dicionário do BoF. BoF é um descritor global de formas montado pela substituição de descritores locais com registros próximos em um dicionário geométrico e então calculando a frequência da aparição destas palavras geométricas.

Descritores de features passaram a possuir um papel crucial em uma gama de aplicações de análise e processamento geométrico, dentre eles correspondência, segmentação, identificação ou reconstrução de formas. Destacaram-se os descritores orientados a dados que desempenharam um papel importante na descoberta de relações geométricas, estruturais ou semânticas entre formas. Métodos orientado a dados, em contrapartida às abordagens tradicionais, reúnem informações de coleções de modelos 3D para melhorar a análise, modelagem e edição das formas (Figura 2.3). Igualmente, são capazes de aprender modelos computacionais que aprendem sobre propriedades e relacionamentos de formas sem apoiar-se em regras codificadas ou instruções explicitamente programadas [138].

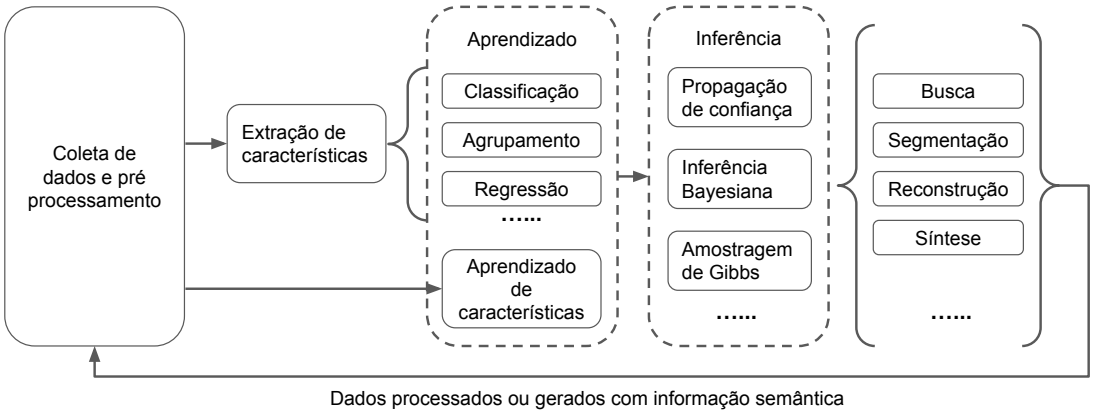

Figura 2.3: Pipeline geral do processamento geométrico baseado em dados ([139]), composto por quatro estágios: coleta de dados e pré processamento, extração de features (ou aprendizagem delas), aprendizagem e inferência. A inferência serve para aplicações que produziriam novas formas ou cenas através de modelagem de reconstrução ou síntese. Estes novos dados, geralmente dotados de rótulos para as formas ou partes delas, pode ser utilizado para melhorar os conjuntos de dados de entrada e aprimorar as tarefas de aprendizado futuras, tornando-se um ciclo.

No campo da visão computacional, as pesquisas ultimamente presenciaram o ressurgimento da "aprendizagem profunda" (*deep learning*), especialmente das técnicas de redes neurais convolucionais, permitindo aprender features específicas contextualizadas a partir de exemplos [87]. A característica principal de uma rede neural convolucional é sua habilidade de aprender abstrações hierárquicas a partir de grandes coleções de dados, exigindo assim pouco conhecimento prévio específico da tarefa desejada. Dentro da comunidade de análise de formas tridimensionais, métodos de aprendizagem profunda eram pouco estudados. Tendo como estudos mais representativos nessa área a aplicação do classificador de florestas para correspondência de formas de Rodolá et al. [103] e o aprendizado de descritores por *bag-of-features* com codificação esparsa, de Litman et al. [79].

Uma das razões para impedimento da adoção das CNNs ou métodos similares em processamento geométrico e análise de formas é de que, ao contrário das imagens bidimensionais, que podem ser modeladas como espaços invariantes ao deslocamento (*shift-invariant spaces*), formas tridimensionais são tipicamente representadas como variedades Riemannianas em que não há invariância de deslocamento; e consequentemente não permite a aplicação da noção do processo de convolução clássico. De fato, invariância à isometria é uma propriedade necessária para a análise robusta de formas 3D, fato fundamentado pela popularidade das assinaturas intrínsecas das formas para análise de formas 3D deformáveis dentro da comunidade de geometria [122]. Nesse sentido, Bruna et al. [21] propuseram uma formulação espectral das CNNs em grafos utilizando um noção de convolução generalizada (não invariante a deslocamento). Embora seu trabalho tenha permitido estender CNNs pra um domínio não Euclidiano, não permitia aplicar o mesmo modelo sobre diferentes domínios, pois os coeficientes da convolução eram expressos em uma base específica dependente do domínio.

Métodos espectrais ganharam popularidade em alguns domínios da computação gráfica e processamento geométrico, sobretudo processamento de formas 3D, computação de descritores de forma, distâncias, e correspondência. Estruturas geométricas espectrais são intrínsecas e deste modo invariantes à deformações isométricas, são eficientemente computadas, e podem ser construídas em diferentes representações. Con-

tudo, nesta construção há a desvantagem de que elas são isotrópicas, i.e., insensível a direção [18]. Assim sendo, CNNs espectrais (SCNN) podem ser utilizadas para aprender sob um domínio não-Euclidiano, mas não através de diferentes domínios. Além disso, filtros definidos no domínio da frequência carecem de uma interpretação geométrica clara e não têm garantia de estarem situados no domínio espacial [18]. Embora a primeira arquitetura de rede neural convolucional intrínseca (CNN Geodésica ou GCNN) [88] tenha produzido bons resultados em diversos *benchmarks* de correspondência e recuperação de formas, há desvantagens, e. g. o processo de mapeamento é limitado a malhas e não há garantias de que o mapeamento é sempre topologicamente significativo [17].

Boscaini et al. [16] trabalharam com a generalização de redes neurais convolucionais com domínios não-Euclidianos para análise de formas deformáveis. O trabalho seguiu a linha baseada em análise de frequências localizadas, isto é, uma generalização da transformada de Fourier em janela para variedades, utilizada para extrair o comportamento local de alguns descritores locais densos. Em suma, a proposta apresentada pelos autores combina as ideias da GCNN (operador espacial) de Masci et al. [88] e SCNN (filtros no domínio da frequência). A principal vantagem deste método é que por tratar-se de uma construção espectral, é quase que diretamente aplicável a qualquer representação da forma (malha, nuvem de pontos, dentre outras), se abastecido por uma discretização apropriada do operador de Laplace-Beltrami. Relacionado aos trabalhos anteriores de Masci et al. [87] (GCNN) e de Boscaini et al. [18] (Descritores de Difusão Anisotrópica), o próprio Boscaini, em conjunto com outros autores [17], apresentaram um método chamado Redes Neurais Convolucionais Anisotrópicas (ACNN) para aprendizagem profunda em domínios não-Euclidianos (Figura 2.4). Este método utiliza kernels de calor anisotrópicos como alternativa no modo de extração de retalhos intrínsecos das variedades.

Figura 2.4: Exemplos de correspondência obtida pelo método ACNN aplicada sobre conjunto de malhas em formas humanas do repositório FAUST. É mostrada uma textura transferida da forma de referência mais à esquerda para outras formas em diferentes poses através da correspondência (retirado de [17]).

Nota-se que nas redes neurais convolucionais mais recentes que utilizam das representações espectrais, a maior parte dos esforços são focados em projetar alguma expansão dos filtros espectrais baseados em grafo Laplaciano [21, 30, 70]. Para Li et al.[72] porém, para o problema de correspondência entre formas, tais métodos ainda são ineficientes ao analisar a informação da estrutura geral das formas devido a informações direcionais ignoradas durante o processo de aprendizagem. Em seu trabalho, o autor propõe uma arquitetura de CNN para variedades. Através da filtragem de

autovalores e autofunções de múltiplos Operadores de Laplace-Beltrami Anisotrópicos (ALBO) [18], operadores gerais de convolução em grafos para uma versão anisotrópica para variedades Riemmanianas são estendidos; cada um deles é determinado por um ângulo de rotação ao redor das normais em planos tangente da forma subjacente.

Em um dos trabalhos do estado da arte atual em relação à correspondência entre formas, Li et al.[71] utiliza polinômios de Chebyshev para representar filtros espectrais onde os coeficientes de expansão são aprendidos sob tarefas de correspondência de formas. No trabalho de Li et al. os operadores de convolução estendidos agregam features locais de sinais através de um conjunto de kernels orientados ao redor de cada ponto, o que permite capturar compreensivamente informação intrínseca do sinal. Os kernels são aprendidos pela aplicação de filtros espectrais, baseados da auto-decomposição (*eigen-decomposition*) de múltiplos Operadores de Laplace-Beltrami Anisotrópicos. A expansão explícita das bases de polinômios de Chebyshev são utilizadas para representar os filtros espectrais, cujos coeficientes de expansão são treináveis.

2.3 Aprendizagem Profunda Multivisão e Classificação de Formas 3D

Com exceção do trabalho de Wu et al. [151], a maioria dos descritores de formas anteriores foram "ajustados manualmente" de acordo com alguma propriedade geométrica particular da forma da superfície ou volume, como por exemplo, representar formas por modelos simbolizados por histogramas ou BoF construídos a partir de normais da superfície, curvaturas, ângulos, distâncias, área dos triângulos ou volumes de tetraedros coletados em pontos de amostragem da superfície, dentre outras características geométricas da forma [120]. Enquanto Wu et al. [151] apresentaram um classificador para formas 3D utilizando uma rede de convicção profunda treinada sobre representações de voxels, Su et al. [120] trataram o problema em um contexto diferente: reconhecer formas 3D a partir de uma coleção de suas visualizações renderizadas em imagens 2D. Em suma, o autor apresentou ideias pra compilar a informação na forma de múltiplas visualizações 2D de um objeto em um descritor de forma compacto com o uso de uma nova arquitetura chamada *Multi-View* CNN (Figura 2.5) ou rede neural convolucional com arquitetura multivisão.

A representação multivisão de objetos 3D mostrou-se a forma mais direta de estender conceitos de redes neurais profundas utilizadas para análise de imagens (i.e., aprendizagem de descritores de imagens de propósito geral) para o caso 3D [66]. A grosso modo, CNNs com esta arquitetura analisam as formas 3D em três passos [66]:

- renderizar visualizações 2D da forma a partir de pontos de vista arbitrários ou cuidadosamente selecionados;
- passar cada visualização 2D através de uma CNN pré-treinada, obter as ativações dos neurônios de uma das camadas, não necessariamente a ultima, como um descritor para a visualização 2D de entrada; e
- reunir os descritores das diferentes visualizações em um único descritor global.

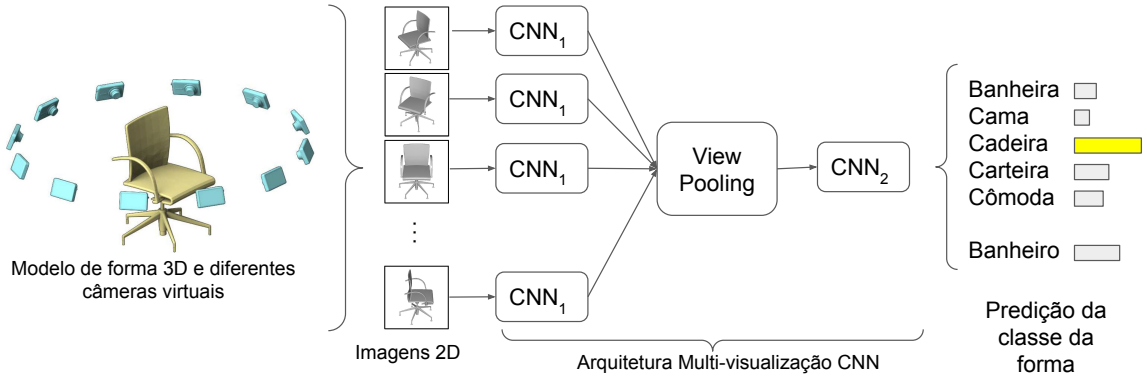

Figura 2.5: Ilustração de uma arquitetura de rede CNN multivisão para reconhecimento de formas 3D. Um conjunto de câmeras são posicionadas ao redor do modelo 3D. Cada câmera captura uma única imagem que será uma das entradas da CNN, a qual extrairá features baseada nas imagens. A saída da CNN multivisão é agregada utilizando um *pooling* de visão (*view pooling*), obtendo um descritor de forma compacto, e então alimentando outra CNN que trata de predizer a pontuação de cada classe possível (retirada de [120]).

Representar modelos 3D como uma coleção de imagens renderizadas de diferentes pontos de vista, cuidadosamente posicionadas em torno dos objetos, é uma ideia direta para comparação de modelos tridimensionais. Isso decorre basicamente de três fatos: no sistema visual humano, objetos 3D são representados por um conjunto 2D de visualizações ao invés da representação 3D plena; em segundo lugar, se dois objetos são similares, eles também parecem similares de todos os ângulos de visualização; por fim, problemas de análise de imagens 2D têm sido investigados a décadas, antes das tecnologias 3D serem bem difundidas [66]. Enquanto computacionalmente pode parecer mais lógico construir classificadores de formas 3D diretamente de modelos 3D, Su et al. [120] apresentam um modelo que constrói classificadores de formas 3D a partir de renderizações 2D destas formas. Em seu trabalho, foi introduzida uma arquitetura CNN padrão treinada para reconhecer visualizações das formas renderizadas de maneira independente umas das outras. Particularmente, embora uma representação 3D em resolução total contenha praticamente toda a informação sobre a forma do objeto, para uma representação baseada em voxels, entrada de uma rede profunda que possa ser treinada com amostras em um tempo razoável, a resolução precisa ser significantemente reduzida. Tomando como exemplo o trabalho de Wu et al. [151], "3D Shapenets" que utiliza uma representação grosseira da forma, uma grade 30x30x30 de voxels binários. Uma única projeção do modelo 3D de mesmo tamanho de entrada corresponde a uma imagem de 164x164 pixeis, ou ligeiramente menor se múltiplas projeções forem usadas. De fato, há uma troca entre aumentar a quantidade de informação da profundidade explícita (modelos 3D) e aumentar a resolução espacial. Com o advento desta arquitetura multivisão CNN Su et al. [120] obtiveram resultados do estado da arte da época em relação a classificação e identificação de objetos 3D (Figura 2.6), inclusive em comparação com aqueles trabalhos que operam diretamente em representações 3D das formas.

Nessa mesma linha de pesquisa, Zhu et al. [152] apresentam um trabalho onde renderiza formas 3D em imagens de profundidade 2D e então aplica um processo de auto-codificação (*auto-encoder*) sobre as imagens 2D, obtendo boa performance em tarefas de identificação de formas 3D. Contudo, sua abordagem projetiva está sujeita a perder informações, visto que as visualizações projetadas são tratadas sem dependência.

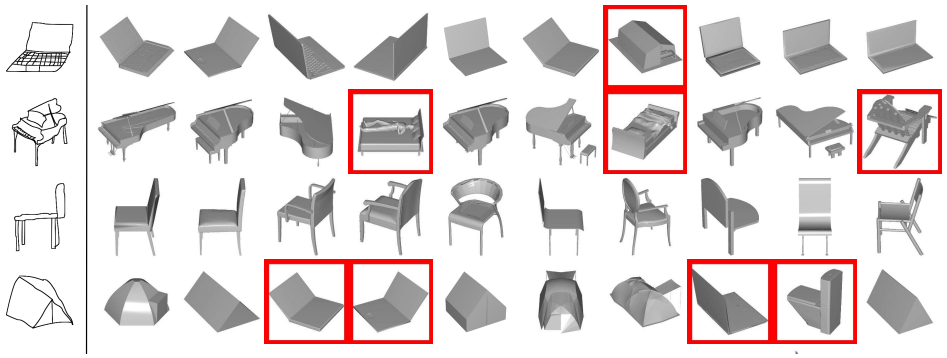

Figura 2.6: Exemplos de identificação de formas 3D através de rascunhos no trabalho de Su et al. [120] (retirado do trabalho dos autores). A coluna mais à esquerda corresponde a pesquisa, as colunas restantes são os 10 melhores resultados obtidos para cada consulta. Os erros estão destacados em vermelho.

À luz dos principais trabalhos da época [120, 152] e apoiando-se no estudo biológico de Welchman et al. [135], cuja pesquisa concluiu que a percepção de formas tridimensionais no córtex visual humano pode ser formado por sugestões de profundidade a partir de múltiplas visualizações, Xie et al. [137] adotaram uma representação de profundidade de imagens multivisão e propôs Aprendizagem de Máquina Profunda Extrema Multivisão (MVD-ELM, *Multi-View Deep Extreme Learning Machine*, Figura 2.7) para tratar a aprendizagem de features projetivas para formas 3D. Porém, o contrário das abordagens multivisão já existentes, o método desses autores garante que os mapas de features aprendidos a partir de diferentes visualizações sejam mutuamente dependentes via pesos compartilhados e em cada camada, suas *desprojeções* juntas formam uma reconstrução 3D válida da forma 3D de entrada através do uso de kernels de convolução normalizados. Desprojeção, introduzido por Miles Reid [101], é uma técnica para descrever transformações bi-racionais em geometria dimensional superior explicitamente em termos de álgebra comutativa. Segundo testes realizados pelos autores, o trabalhou levou a aprendizagem de features 3D mais precisas com resultados encorajadores em diversas aplicações e a propriedade de reconstrução 3D permitia visualização clara das features aprendidas.

Além de características concretas de uma forma 3D, existe também a percepção humana de estilo ou padrão sobre a forma, que pode ser utilizada para orientar, auxiliar projetistas ou até mesmo constituir aplicações com sistemas de recomendação que possam sugerir itens avaliados como similares ou compatíveis com determinada questão. Porém, identificar automaticamente um estilo em uma coleção composta de diferentes categorias e configurações é uma tarefa complexa. Além disso, a similaridade dos objetos não necessariamente corresponde a similaridade dos estilos. Objetos estilisticamente similares podem possuir grande variação em sua forma mais geral [76]. Considerando esse problema de identificação de similaridades de estilo entre formas 3D, Lim et al. [76] propuseram um arcabouço para aprender similaridades de estilo empregando um método similar ao de Su et al. [120], ou seja, uma arquitetura multivisão CNN que utiliza de imagens renderizadas a partir de formas 3D como entrada para redes neurais. Trabalhando com uma ideia similar à de Lun et al. [85], porém em contexto distinto, Lim et al. [76] propõem um método que faz uso da aprendizagem métrica profunda (*deep metric learning*). Em categorização de imagens 2D baseadas em métrica profunda, geralmente uma rede neural profunda aprende um espaço de imersão de menor dimensão das imagens 2D, baseadas em dados rotulados. A distância

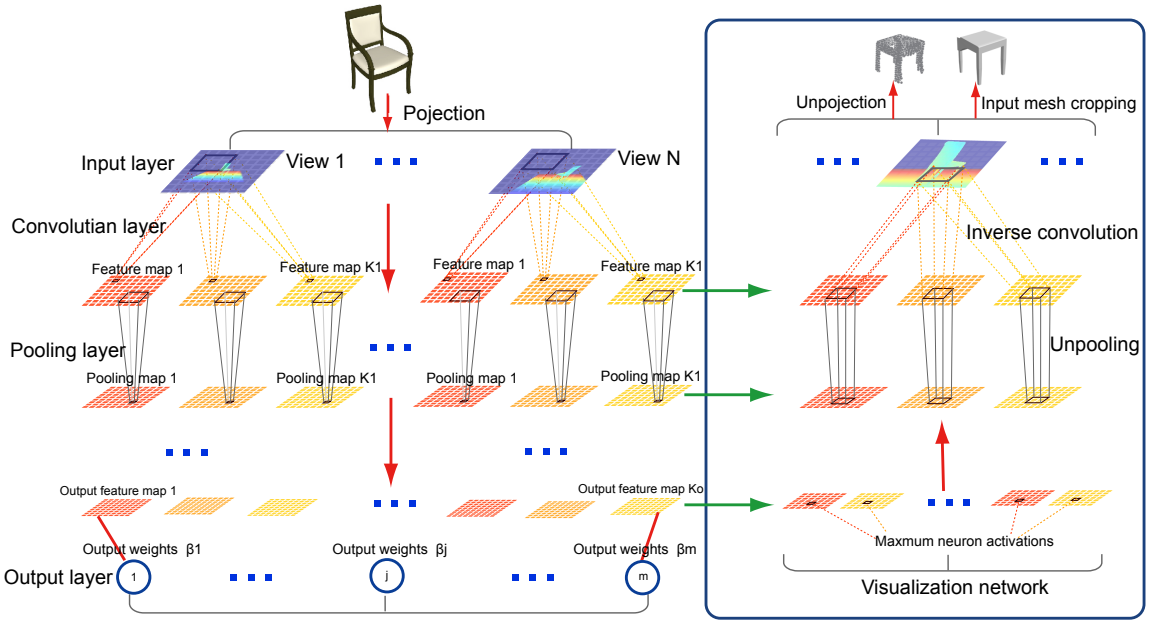

Figura 2.7: A arquitetura da MVD-ELM (esquerda) e a visualização (direita). Na figura à esquerda as imagens de profundidade multivisão para uma determinada forma 3D (cadeira) é entrada aos respectivos canais. Em cada canal, a imagem de profundidade passa por uma cascata de camadas de convolução e *pooling*. Cada camada de convolução produz Kl mapas de features com os kérneis de convolução gerados aleatoriamente. Quando na última camada, os mapas de features de saída são utilizados para calcular os pesos de saída otimizados. À direita, a rede de visualização executa na direção oposta da rede de treinamento, com o inverso das operações de convolução e *pooling* originais, até chegar na camada de entrada onde obtém-se as regiões maximamente ativadas nas imagens de profundidade. Estas regiões são então projetadas nas formas da entrada e utilizadas para produzir as formas em partes características (retirado de [137]).

entre imagens é avaliada neste espaço de imersão [76]. Neste caso, dadas duas formas para as quais existe disponível relativa informação estilística como entrada. (e.g., um objeto C é estilisticamente mais similar ao objeto A do que outro objeto B), uma rede neural é treinada sobre imagens renderizadas destas formas. A saída são dois valores de distância, (B,A) e (B,C), os quais correlacionam com a sua semelhança estilística. Este método possuía a vantagem de não depender de descritores explicitamente pré-calculados, além de evitar o processo de ter que definir e então ponderar a importância relativa de cada descritor de característica.

A natureza estrutural em grade dos dados multivisão, i.e., operam em dados sob uma estrutura Euclidiana, permite que abordagens de redes neurais convolucionais (CNNs), podem ser diretamente aplicadas a eles. Em contrapartida, certamente que representar dados 3D em 2D implica na geração de uma representação mais simples. A tentativa de adaptar esta arquitetura CNN para análise de formas rígidas e não rígidas juntamente com a falta de uma representação unificada para formas levou alguns pesquisadores a buscar a análise de formas deformáveis e não deformáveis via aprendizagem profunda por caminhos distintos [117]. A comunidade de visão computacional enfatizou representações extrínsecas das formas (e.g., Wu et al. [151]), mais adequadas ao aprendizado de formas rígidas, enquanto que a comunidade de geometria buscou adaptar CNNs para variedades não-Euclidianas adotando propriedades intrínsecas das formas (e.g., Masci et al. [87]) para criação de descritores de features. Por conseguinte, Sinha et al. [117] propuseram uma representação que serve para aprendizagem tanto de

objetos rígidos como também não rígidos utilizando descritores intrínsecos ou extrínsecos como entrada de CNNs padrão (Figura 2.8). O tratamento de Sinha et al. converte uma forma 3D em uma "imagem de geometria" (*geometry image*) de maneira que CNNs padrão possam utilizá-las diretamente para aprender. Como o próprio nome sugere, Imagens de Geometria são um tipo particular de parametrização de superfície em que a geometria é re-amostrada em uma grade 2D regular semelhante a uma imagem. O método parametriza a forma 3D sobre um domínio esférico, mapeia as amostras em um octaedro e por fim corta o octaedro ao longo das arestas para ter como saída uma imagem de geometria plana e regular. Tal processo é justificado por por dois fatores: Cortes são definidos a posteriori em relação a parametrização; a simetria esférica permite criar fronteiras de uma imagem de geometria regular sem descontinuidades [117]. Ao contrário de outras abordagens, os pixeis das imagens de geometria podem codificar propriedades da superfície extrínsecas ou intrínsecas conforme adequado à tarefa desejada. Uma CNN padrão poderia portanto automaticamente aprender abstrações discriminativas da forma 3D. Mas há limitações, embora continua, a representação não é sem emendas (*seamless*), o espaço de parametrização, a saber mapas de preservação de área possuem um número muito alto de graus de liberdade e assim sendo podem representar a mesma forma em diversas maneiras arbitrárias na imagem. Por fim, a convolução sobre a imagem de geometria não é invariante a translação [86].

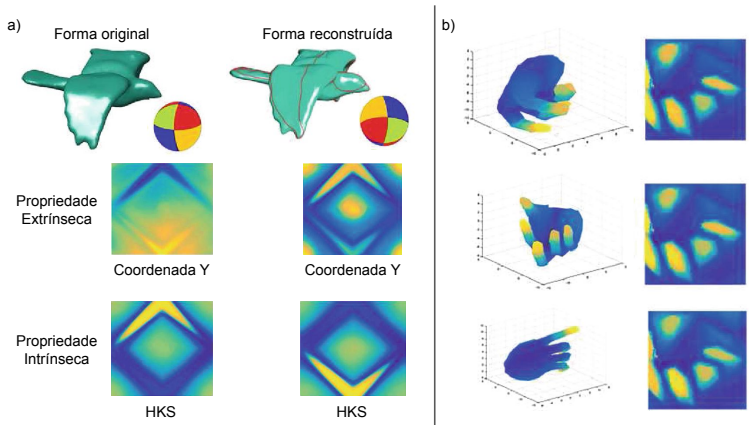

Figura 2.8: Imagens de geometria e diferença entre coordenadas pontuais e de HKS no contexto da análise de formas articuladas (retirado de [117]): (a) Propriedades intrínsecas vs propriedades intrínsecas das formas. No topo a forma original (esquerda) e a forma reconstruída (direita) a partir da imagem de geometria. As imagens de geometria são criadas parametrizando a forma 3D sobre um domínio esférico, amostrando sobre um octaedro e finalmente cortando o octaedro ao longo de suas arestas para obter uma imagem de geometria plana e regular. Na figura, os cortes da aresta são exibidos em vermelho forma reconstruída). A linha central e inferior mostra a codificação da imagem de geometria na coordenada Y e na assinatura do kernel de calor (HKS), respectivamente de duas parametrizações esféricas (esquerda e direita). As duas parametrizações esféricas são rotacionadas simetricamente por 180 graus ao longo do eixo Y. As imagens de geometria para a coordenada Y mostram um eixo assim como um giro da intensidade. As imagens de geometria para HKS exibem apenas um giro no eixo. Isso ocorre porque a HKS é uma assinatura intrínseca da forma enquanto que pontos de coordenadas em uma superfície não são. (b) Exibe descritores intrínsecos (neste caso HKS) são invariantes a articulações da forma.

Parte das arquiteturas de redes profundas adaptadas para atuar sobre dados de formas 3D substituem o vetor de pixeis 2D pelo seu análogo 3D, i.e., uma grade de voxels denso e regular, mantendo suas estrutura Euclidiana, é processado utilizando as operações de convolução e *pooling* [151, 2, 96]. Não obstante, para dados 3D densos, requerimentos de processamento e memória crescem de forma cúbica conforme a

resolução. Consequentemente, as redes existentes na época limitavam-se a resoluções 3D baixas, tipicamente na ordem de 30^3 voxels. Em modelos densos, como o de Wu et al. Wu:2015:3DS, por exemplo, tarefas de classificação de objetos e refinamento de formas são capazes de gerar formas apenas em uma resolução muito grosseira, devido à limitações de memória e processamento. Riegler et al. [102] observaram que frequentemente os dados 3D destas representações em voxels são esparsos, e.g., nuvem de pontos, ou malhas, tendo como consequência desperdício de processamento ao aplicar convoluções 3D nativamente. Na época haviam poucas arquiteturas de rede neural que exploravam explicitamente a esparsidade nos dados. Como estas redes não necessitavam de convoluções densas exaustivas possuíam o potencial de tratar resoluções maiores. Baseando-se nestas observações, o método toma malhas 3D como as de Wu et al. [151], "voxeliza" a entrada em uma resolução de 64^3 e treina uma rede convolucional simples que minimiza o erro de classificação. Os autores mostram o valor máximo das respostas através de todos os mapas de features em diferentes camadas da rede, observando que altas ativações ocorrem somente perto da fronteira dos objetos. Riegler et al. [102] combinam estruturas de dados de *octree* [89] e de grade para permitir CNNs 3D com resolução mais alta. O método limita a CNN 3D ao interior do volume das formas 3D, mas torna-se menos eficiente do que soluções baseadas na representação total de voxels quando a resolução do volume é menor do que 64^3 [131]. Partindo desse ponto, Wang et al. [131] apresentam uma rede neural convolucional para análise de formas 3D baseada em *octree*, intitulada O-CNN. Seu método limita a CNN 3D aos octantes da fronteira da forma 3D e emprega uma nova estrutura de *octree* para treinar e avaliar a O-CNN em GPU (*graphic processing unit* ou unidade de processamento gráfico). Construída sobre a representação das formas 3D em *octree*, o método toma os vetores normais médios de um modelo 3D amostrados nos melhores octantes folha como entrada e realiza operações da CNN 3D nos octantes ocupados pela superfície da forma 3D (Figura 2.9). Ademais, definiu-se uma nova estrutura de dados do tipo *octree* para armazenar a informação do octante e features da CNN em memória de placas gráficas, permitindo executar todo o treino e avaliação O-CNN em GPU.

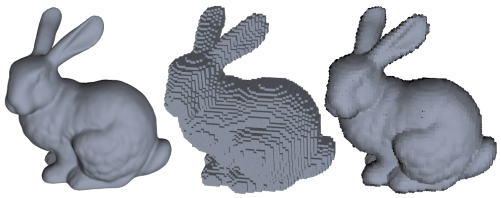

Figura 2.9: Esquerda: a forma 3D original. Centro: a forma 3D voxelizada. Direita: a representação em *octree* como normais amostradas nos melhores octantes folhas (retirado de [131]).

Motivados pela representação de imagens de geometria utilizada por Sinha et al. [117] na aprendizagem de superfícies de formas 3D, Sinha et al. [118] propuseram o que teria sido a primeira abordagem geradora de superfícies de formas 3D utilizando redes neurais profundas. A intenção era gerar nuvens de pontos 3D específicas de determinada categoria, que representasse a superfície de objetos 3D rígidos e não rígidos. A arquitetura da rede neural foi inspirada pelas Redes Residuais Profundas (*Deep Residual Networks*) [48]. Sinha et al. [118] desenvolvem uma rotina para criar imagens de geometria consistentes que representem a superfície da forma de uma categoria de objetos 3D. Em seguida utiliza essa representação consistente para geração da superfície de forma de categoria específica a partir de uma representação paramétrica ou uma

imagem, desenvolvendo novas extensões de redes neurais residuais para a tarefa de geração de imagem de geometria. Em resumo, o método gera superfícies 3D baseando-se em uma representação de imagem de geometria, i.e., uma remodelagem de uma superfície arbitrária em uma grade totalmente regular. Segundo os autores, as imagens de geometria da saída da rede permitem gerar superfícies de forma para imagens ainda não vistas, poses intermediárias de formas e interpolação entre superfícies de formas, como mostrado na Figura 2.10.

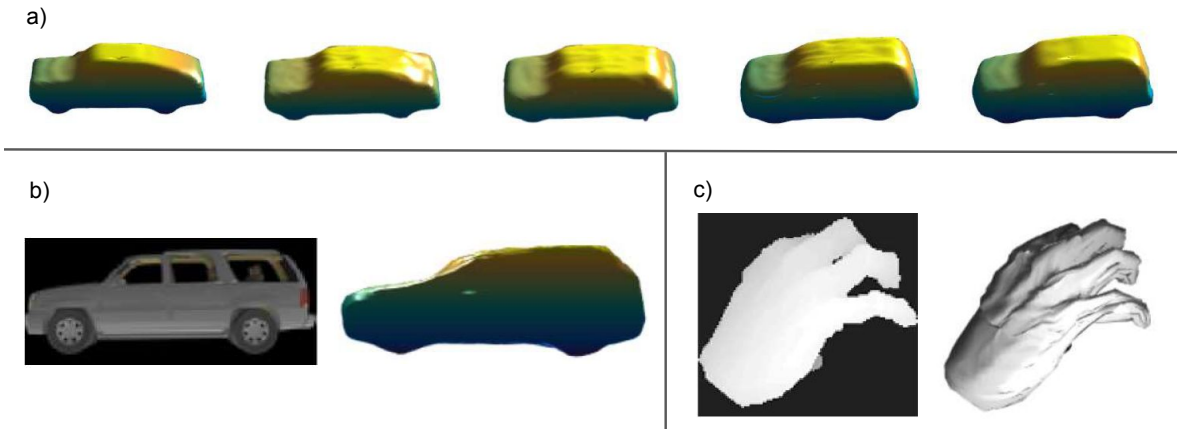

Figura 2.10: Resultados possíveis utilizando a rede proposta por Sinha et al. [118] (figura retirada da mesma fonte): (a) Resultado dos modelos intermediários obtidos pela interpolação da superfície de formas 3D entre a forma original (esquerda) e final (direita). (b) Reconstrução de uma superfície 3D rígida a partir de uma imagem RGB. (c) Reconstrução de uma superfície 3D não rígida a partir de uma imagem de profundidade. As superfícies foram construídas com estimativa implícita do ponto de vista.

2.4 Considerações Finais

A princípio, um ponto comum detectado entre abordagens que identificam formas 3D é a criação de um descritor ou assinatura que capture as propriedades únicas daquela forma, que sejam capazes de distinguir formas pertencentes à classes distintas, e que sejam invariantes a determinadas classes de transformações que cada forma pode ser submetida [124]. Apesar dos avanços relatados na literatura para solução de problemas de correspondência entre formas tridimensionais não-rígidas [19, 110, 122, 60, 6], a tarefa ainda não possui uma solução definitiva, sendo considerado um problema em aberto [103]. É importante observar que a definição precisa do que é uma correspondência significativa depende da aplicação em particular, variando desde o caso menos complicado de como identificar partes das formas que são geometricamente similares, até o problema mais complexo de relacionar elementos que representam as mesmas partes ou possuem a mesma função.

No contexto de correspondência entre formas, classificamos os métodos de interesse em 3 grandes grupos, similar a classificação de identificação de formas sugerida por Tangelder et al.[124]:

- Baseado em features (*feature based*): features referem-se a propriedades geométricas e topológicas das formas 3D. Assim, através da comparação e mensuração das features das formas é possível discriminá-las.

- Baseado em grafos (*graph based*): métodos baseados em grafos tentam extrair um significado geométrico de uma forma 3D utilizando um grafo que mostra como os componentes da forma estão ligados entre si [10, 4, 8].

- Similaridade baseada em visualização (*view based similarity*): baseia-se na ideia de que se dois modelos 3D são similares, então são similares de todos os ângulos de visualização [120], i.e., uma abordagem multivisão.

Abordagens multivisão demostraram algumas vantagens em tarefas de classificação, identificação e rotulação de partes de formas. Sendo a principal delas foi a possibilidade de ter a disposição uma grande quantidade de dados naturalmente invariantes. Destacamos os trabalhos de Zhu et al.[152], Kalogerakis et al.[55] a SimNet presente no trabalho de Gao et al.[38] como passos iniciais na definição do pipeline do trabalho desenvolvido nessa tese.

Zhu et al.[152] projeta formas 3D no espaço 2D e utiliza um auto-codificador para aprender features sobre imagens 2D. A partir do aprendizado, o autor agrega as features aprendidas sobre as imagens para identificar e classificar formas 3D. Kalogerakis et al.[55], por sua vez, tratou o problema de segmentação de formas3D em partes semânticas rotuladas. Utilizando uma abordagem multivisão relacionada a métodos de aprendizagem por segmentação de imagens e formas 3D, o método alcançou precisão do estado da arte, inclusive superior ao trabalho anterior do mesmo autor [56] que tomava como entrada propriedades geométricas das formas. Presente no trabalho de Gao et al. [38], a SimNet é um dos 3 componentes do método de transferência de deformação proposto para formas 3D. A rede SimNet é responsável por calcular a similaridade visual entre duas formas, ambas em seus respectivos espaços latentes. Tal rede é treinada sobre resultados de uma técnica chamada de distância do campo de luz(*light field distance*), inicialmente proposta por Chen et al[22]). Resumidamente, uma forma 3D é projetada em imagens a partir de múltiplos pontos de vista. A similaridade entre dois modelos 3D pode ser medida somando-se a similaridade de todas as imagens correspondentes. Na técnica, o sistema de câmera ao redor de cada modelo é rotacionado até que a maior similaridade geral (correlação cruzada) entre dois modelos a partir de todos ângulo de visualização seja alcançada.

Capítulo 3

O Método

3.1 Considerações Iniciais

O problema de encontrar um mapeamento ou correspondência entre superfícies de duas formas tridimensionais (i.e., superfícies bidimensionais imersas no $\mathbb{R}^3$) é um problema fundamental em computação gráfica. Na literatura encontramos uma diversidade de abordagens orientadas a conteúdo para a solução de problemas de correspondência entre formas, ou seja, concentrando-se nas similaridades geométricas e estruturais entre as formas envolvidas. Todavia, em cenários onde a geometria das malhas apresenta notável dissimilaridade, o problema vai além de uma análise puramente geométrica, tal que a determinação de correspondências pode envolver o entendimento semântico. Apesar dos avanços relatados para solução do problema de correspondência entre formas tridimensionais não-rígidas [19, 110, 122, 60, 6], até onde temos conhecimento a literatura não apresenta uma solução definitiva, e continua sendo um problema em aberto [103].

Embora diversas técnicas estudadas e revisadas no Capítulo 2 e Apêndice C não lidem diretamente do mesmo problema deste trabalho, foi possível identificar, em algumas delas, limitações para nosso propósito, e.g. falta de memória em abordagens volumétricas ou ausência de invariância a translação em superfícies 3D. Abordagens com aprendizagem não supervisionada, como no trabalho de Gao et al. [38], apesar de consideradas como possível inspiração, especialmente por não requerer uma grande quantidade de conjuntos de dados pre-anotados. Todavia, verificamos que a natureza destes trabalhos consistem basicamente em aprender a definir um espaço de deformação n-dimensional para formas origem e destino e em seguida mapear a transformação de um espaço em outro. Tais propostas são eficazes para tarefas relacionadas à transferência de deformações ou animações — exigindo como entrada sequências de formas deformadas —, mas não fornecem informações sobre correspondência entre as formas envolvidas, apenas define espaços de deformação latentes e um mapa de transformação entre eles. Portanto, as abordagens como as que utilizam representações volumétricas [151] ou mapeamento de espaços latentes de deformação [44, 43, 142, 108] mostraram-se também inadequadas como uma solução direta do problema dentro do contexto desejado.

Diante disso, a solução proposta para o problema, descrita com detalhes na Seção 3.2, foi obtida através da utilização de uma abordagem multivisão combinada

com alguns algoritmos de refinamento executados no domínio espacial. A motivação baseou-se em resultados de trabalhos multivisão de formas 3D [22, 75] nos quais é possível identificar pontos comuns na literatura, observados em trabalhos como no de Zhu et al. [152], Kalogerakis et al. [55] assim como na *SimNet* presente no trabalho de Gao et al. [38]. Apesar da inspiração, os trabalhos citados tratam de problemas distintos como classificação de formas, segmentação e comparação visual de deformações entre pares de formas no espaço latente.

Mesmo tratando-se de problemas da mesma natureza, os trabalhos citados não tratam do mesmo problema e nem possuem pipeline geral semelhante ao trabalho aqui proposto. A inspiração deu-se pela utilização da abordagem multivisão para solucionar problemas geométricos de "comparação" sem a obrigatoriedade de um grande dataset anotado para treinamento. Zhu et al. utiliza a comparação possibilidade de comparação entre as imagens para classificar formas. Kalogerakis et al. classifica todas as faces de uma malha em categoria de segmentos genéricos, e ao contrário do nosso trabalho, o pipeline de classificação dá-se exclusivamente pela CNN, além das principais imagens serem imagens de profundidade. A SimNet, por sua vez, trata-se de uma técnica de comparação visual entre formas, transformada em treinamento para fornecer uma solução diferenciável ao problema de transferência de deformações. O método apresentado nessa tese pode ser dividido em duas etapas: treinamento e correspondência. Depois de concluída a etapa de treinamento, o método diferencia-se dos demais principalmente por permitir a detecção de pontos de correspondência ou pontos de interesse entre formas variadas, além de permitir a correspondência entre conjuntos de dados de datasets distintos, sem a intervenção do usuário.

A etapa de treinamento toma como entrada malhas 3D variadas das classes de formas desejadas que cubram características semânticas desejadas nas correspondências, além de um gabarito com os pontos de interesse (features desejadas). A correspondência é aprendida de acordo com esse gabarito. Informa-se através da interface gráfica de uma ferramenta, para uma malha de cada conjunto de malhas de treinamento, os pontos referentes ao gabarito. Automaticamente a ferramenta gera imagens e posição dos pixeis correspondentes as features para malhas de treinamento; estas imagens e posições de pixeis são tomadas como entrada por uma CNN, que é então treinada para para detectar os pixeis candidatos a cada feature. Após essa fase, a etapa de correspondência pode ser executada, sem necessidade de repetir o treinamento.

A etapa de correspondência é totalmente livre da intervenção manual do usuário, i.e., o pipeline segue sem requisitar auxílio do usuário para a detecção das correspondências. Através da mesma ferramenta utilizada na etapa anterior, gera-se automaticamente imagens das malhas desejadas, que em seguida são fornecidas para a CNN treinada realizar detecção das features. De posse da resposta da rede, realiza-se o mapeamento reverso dos pontos encontrados nas imagens, projetando-os de volta à malha 3D como vértices tridimensionais; por fim, refina-se as informações obtidas para definir, através dos pontos descobertos, um único vértice da malha destino para cada feature desejada.

Na Seção 3.2 é apresentada a solução proposta para essa tese. Apresentamos um método composto de uma abordagem multivisão para aprendizagem semi-supervisionada. A técnica é combinada com alguns algoritmos de refinamento no espaço 3D para fornecer uma solução de correspondência entre formas sem a necessidade de interação do usuário após o treinamento.

3.2 Solução Baseada em Multivisão

Ao analisar formas 3D, é possível representá-las por imagens obtidas a partir de diversos pontos de vista, juntamente com CNNs 2D [115], embora computacionalmente possa parecer mais direto construir os classificadores diretamente da representação tridimensional das formas. Construir classificadores de formas 3D a partir de renderizações das mesmas é uma prática comum [120], relevante na literatura atual [118, 137, 120, 96] e que relata resultados eficazes, especialmente em tarefas de correspondência ou segmentação de formas [53]. Embora esta representação não contemple toda a informação geométrica sobre a forma do objeto, permite o treinamento o treinamento permite ponderar resoluções mais significativas de imagem e tempo de execução. Em especial, a natureza estrutural dos dados operando sob uma estrutura Euclidiana, permite aplicá-los diretamente em redes neurais convolucionais. A tentativa de adaptar e utilizar esse tipo de arquitetura CNN para análise de malhas rígidas e não rígidas surgiu em função da falta de uma representação unificada para formas 3D, influenciando alguns pesquisadores a considerar outros caminhos para a análise de formas deformáveis e não deformáveis via aprendizagem profunda [117].

Apesar da representação mais simples, dados bidimensionais não precisam lidar com uma estrutura complexa em um espaço 3D e o número limitado de dados 3D disponíveis para aprendizagem de features [152]. Outrossim, ao contrário de uma convolução diretamente sobre a superfície de uma geometria, os dados são invariantes à translação [86]. Projetar formas 3D no espaço 2D de onde features são aprendidas já provou-se eficaz, como demonstrado por exemplo pelo trabalho de Zhu et al. [152] que reconhece formas tridimensionais agregando features aprendidas sobre imagens bidimensionais renderizadas a partir de uma técnica de multivisão. No contexto de aprendizagem profunda sobre malhas tridimensionais, a abordagem multivisão pode algumas vezes obter resultados tão corretos ou até melhores do que abordagens que utilizam aprendizado profundo geométrico. Mesmo não agindo diretamente sobre a topologia e geometria das superfícies, é um tipo de técnica bem amadurecida na comunidade científica e permite a extração de uma boa quantidade de dados na forma de imagens.

Em relação ao tempo de processamento e consumo de memória, o processo de geração das imagens mostrou-se também menos custoso do que nas propostas que baseadas em geometria. Parte disso deve-se à grande quantidade de informações geométricas geradas em etapas de pré-processamento, e.g., cálculo de dados informando a distância geodésica entre todos os vértices ou matrizes baseadas no Laplaciano. De fato, para a abordagem multivisão sequer há exigência de que as malhas sejam totalmente conexas ou possuam todas as arestas ou vértices e arestas do tipo *manifold*.

3.2.1 O Método

O método proposto permite correlacionar vértices de malhas de triângulos com um gabarito no qual foram rotulados pontos de interesse. Os pontos do gabarito podem ser definidos sob uma política específica dependendo do propósito desejado; podem ser arbitrários, localizarem-se em pontos de junção, em determinadas partes anatômicas, dentre outras possibilidades. Apesar de técnicas de multivisão já terem sido empregadas em outros problemas na área de processamento geométrico, este trata-se de um trabalho inédito. O pipeline do método envolve os seguintes passos: definição do gabarito;

preparação de dados para o treinamento; treinamento da CNN; detecção de features em imagens utilizando a CNN; mapeamento reverso pixeis para vértices; separação de features simétricas; e pós-processamento — composto por remoção de *outliers* e desambiguação de features simétricas em relação ao plano sagital. Tal pipeline e a relação entre eles podem ser sumarizadas pela Figura 3.1, que exibe as etapas de treinamento e correspondência.

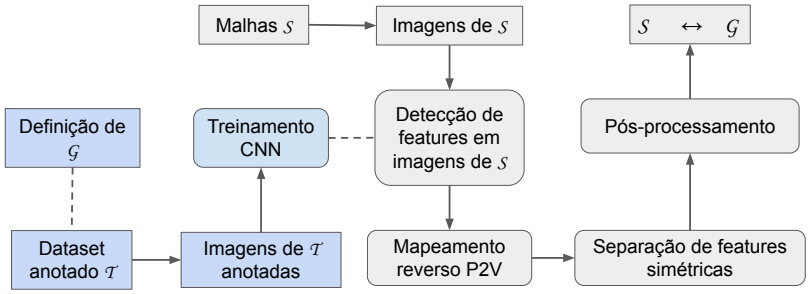

Figura 3.1: Em azul claro é representado a etapa de treinamento: Com o gabarito semântico $\mathcal{G}$, marca-se as malhas referência do dataset $\mathcal{T}$ de onde gera-se imagens em múltiplas visualizações. Por fim, estas imagens são fornecidas para uma CNN treinar a detecção de features. Em cinza, a etapa de correspondência gera imagens das malhas desejadas para correspondência $\mathcal{S}$, detecta as features com a CNN já treinada, realiza o mapeamento reverso pixel/vértice e ajusta os resultados.

O pipeline proposto foi desenvolvido empregando uma CNN cujo propósito é a detecção de regiões de interesse em imagens. Mais detalhes sobre tal CNN são mostrados na Seção 3.2.3, já os experimentos e comparações com outras CNNs para o mesmo propósito são relatados no Capítulo 4. A etapa de treinamento compreende os passos de definição do gabarito até o treinamento da CNN; a partir do passo detecção de features em imagens utilizando CNN, inicia-se a etapa de correspondência. Tais passos são descritos em maiores detalhes a seguir.

Treinamento Define-se um gabarito $\mathcal{G} = \{g_1, g_2, ...g_n\}$ representando índices de n features, $n \geq 3$, cada feature é definida por um ponto de interesse específico sobre as classes de formas cobertas pelos conjuntos de treinamento. Por exemplo, em classes que representem formas humanas, os pontos podem ser pontos nos olhos, pontas dos dedos, centro do peito, e assim por diante. A única restrição em relação a estas features é que 3 destas features são obrigatórias, sendo ou não pontos de interesse da tarefa planejada. Os 3 pontos obrigatórios são utilizados para eliminar ambiguidade entre features com simetria bilateral. As regras de como isso ocorre estão descritas na Subseção 3.2.2 e tais definições são criadas apenas para o gabarito no processo de treinamento.

Seja um conjunto de malhas $\mathcal{S} = \{m_{s_1}, m_{s_2}, ...m_{s_{n_j}}\}$, para o qual deseja-se realizar a correspondência com as features do gabarito $\mathcal{G}$. Toda malha $m_{s_i} \in \mathcal{S}$ possui mesma conectividade, i.e., são sequências de animação de uma forma ou diversas poses de malhas com mesma topologia. O objetivo do método é encontrar um conjunto de n vértices para as malhas do conjunto que correspondam às features definidas pelo gabarito $\mathcal{G}$. Nesse caso, basta encontrar a solução para uma única malha de $\mathcal{S}$, pois as malhas do conjunto possuem mesma conectividade. Não é exigido que as malhas possuam deformações isométricas, sejam orientadas (nem entre elas), ou obedeçam alguma ordem específica ou tampouco existem restrições em relação a qualidade dos triângulos ou superfície. O conjunto $\mathcal{S}$ pode ser composto de apenas uma malha, i.e., o método

funciona também para a correspondência com uma única malha. Embora o pipeline seja o mesmo para uma ou várias malhas, teoricamente um conjunto maior e variado possibilita uma cobertura melhor das poses e consequentemente uma quantidade maior de vértices candidatos a cada feature, melhorando a precisão do método.

Para o treinamento, além do gabarito $\mathcal{G}$ o método requer um conjunto de conjuntos $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, ...\mathcal{T}_t\}$ de malhas de treinamento, em que $\mathcal{T}_i$ é um conjunto de malhas de mesma conectividade, em poses variadas ou sequências de animação ou deformação (Figura 3.2 (a)). $\mathcal{T}$ pode conter apenas um conjunto com um único tipo de malha, entretanto, quanto maior e mais variado o número de conjuntos de treinamento presentes em $\mathcal{T}$, mais features a rede aprenderá a identificar. Não é exigida relação alguma entre os conjuntos de $\mathcal{T}$, nem ordem, orientação, topologia, complexidade geométrica ou quantidade de poses de cada conjunto $\mathcal{T}_i$.

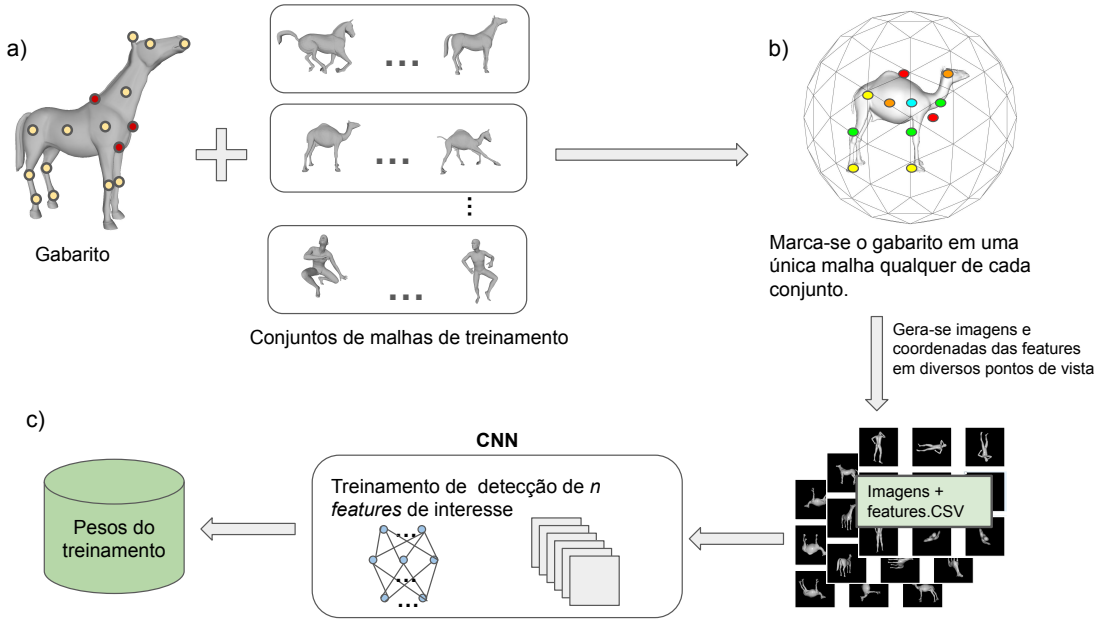

Figura 3.2: A entrada é composta pelo gabarito $\mathcal{G}$ e malhas de treinamento $\mathcal{T}$ (a), em que uma malha de cada conjunto deve ser marcada na ferramenta MIG de acordo com o gabarito (b). (a) mostra um exemplo visual de marcação de pontos, sedo os pontos em vermelho representando features especiais do plano sagital bilateral. A ferramenta renderiza imagens das formas e indica em um arquivo os pixeis correspondentes as features marcadas e visíveis em cada ponto de vista. Tais imagens são tomadas como entrada por uma CNN treinada para detecção de features no domínio 2D.

Inicialmente, com o auxílio de uma ferramenta intitulada MIG (*Mesh Image Generator*), dotada de interface gráfica, marca-se as features definidas no gabarito $\mathcal{G}$ em uma única malha de cada conjunto $\mathcal{T}_i$. Detalhes sobre a ferramenta e suas funcionalidades estão descritos na Subseção 3.2.3. Devido a presença de malhas de mesma topologia em cada conjunto, basta indicar as features em uma única malha de cada conjunto de $\mathcal{T}$ (Figura 3.2 (b).

Através da mesma ferramenta é possível renderizar uma quantidade parametrizável de imagens para cada malha de cada conjunto $\mathcal{T}_i$, em diversos pontos de vista. Conforme explanado na Subseção 3.2.3, as imagens são padronizadas e geradas em tons de cinza com o modelo de iluminação de *Phong*. Também para cada imagem, a ferramenta salva um arquivo do tipo csv contendo posições dos pixeis das features marcadas e visíveis — do ponto de vista que a imagem foi gerada — em coordenadas de (x,y)

de imagem. Tais coordenadas são salvas para todas as imagens geradas para todas as malhas e não apenas das malhas marcadas, pois apesar de malhas de um mesmo conjunto possuírem mesma conectividade, apresentam poses e deformações distintas, tratando-se de imagens distintas umas das outras.

As imagens geradas a partir de $\mathcal{T}$ e arquivos com posição dos pixeis das features alimentam uma CNN pré-treinada de detecção de regiões em imagens 2D. Após completo o treinamento, são salvos os pesos encontrados durante o treinamento e encerra-se a etapa de treinamento (Figura 3.2 (c)).

Correspondência Assim como foi feito para o conjunto $\mathcal{T}$, através da ferramenta MIG, gera-se imagens de diversos pontos de vista também para o conjunto as malhas de $\mathcal{S}$. É ideal utilizar as mesmas configurações de cena para gerar imagens de $\mathcal{T}$ e $\mathcal{S}$, visto que todas as imagens de $\mathcal{T}$ serão tomadas como entrada da CNN para treinamento e as imagens geradas para as malhas de $\mathcal{S}$ passarão pela mesma rede para detecção de features, Figura 3.3.

Ao passar as imagens geradas pelo pipeline de detecção da CNN sobre uma ou mais malhas $m_{s_i} \in \mathcal{S}$, a rede retorna um conjunto de conjuntos de pixeis (x,y) $\mathcal{P} = \{p_1, p_2, ...p_n\}$, em que cada p_k é formado por grupos de pixeis candidatos à feature k definida em $\mathcal{G}$.

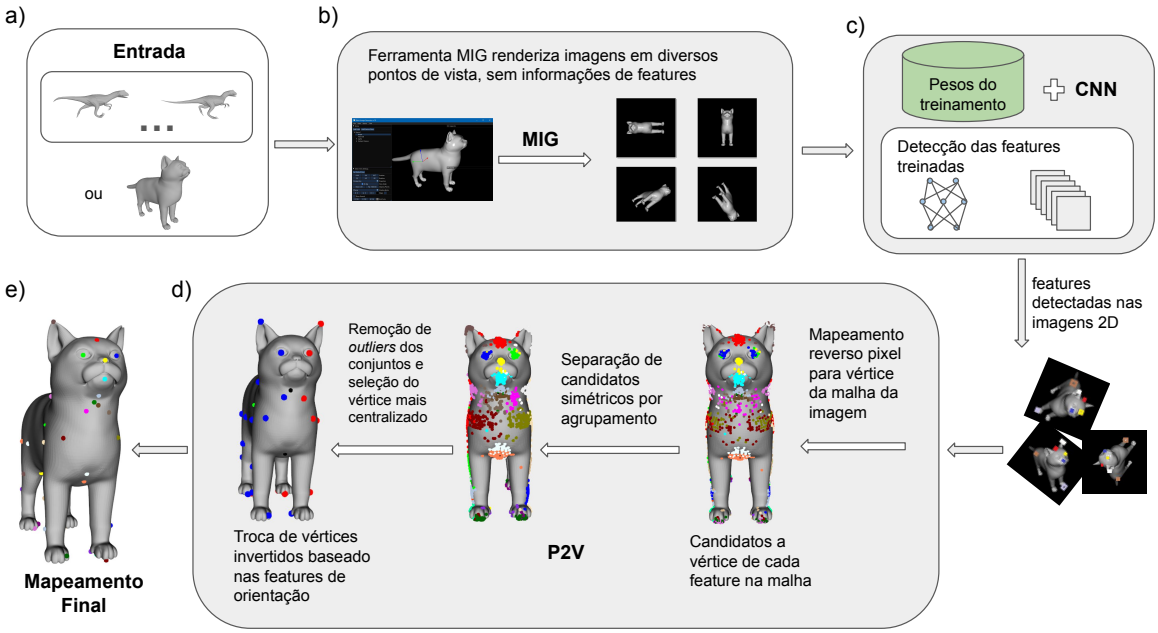

Figura 3.3: (a) A entrada pode ser uma sequência de malhas $\mathcal{S}$, representando uma animação ou deformações variadas de uma forma; ou uma única malha, se desejado. (b) A ferramenta MIG renderiza imagens das malhas automaticamente. (c) As imagens geradas passam pela CNN já treinada na etapa de treinamento e devolvem como saída, para cada imagem, pixeis detectados como features. (d) No módulo P2V, os pixeis são mapeados de volta a superfície da malha de origem. Os grupos de features simétricas são separadas por *clustering* e os vértices em posições atípicas são ignorados por um estimador robusto de covariância. Define-se como vértice representante da feature aquele que se encontra mais ao centro de cada conjunto. Por fim, calcula-se automaticamente o lado dos vértice simétricos de acordo com o gabarito, resultando na correspondência final (e).

Para cada pixel de cada conjunto de $\mathcal{P}$, calcula-se a projeção inversa dos pixeis das imagens de volta para um vértice da superfície da forma 3D fonte da captura da

imagem, i.e., sabendo das configurações de cena e da malha utilizada para renderizar determinado pixel em uma imagem, realiza-se o processo inverso determinando o ponto da superfície da malha que resultou naquele pixel e qual o vértice mais próximo desse ponto, Figura 3.4. A tarefa é realizada por um módulo da ferramenta MIG chamado P2V (*pixel to vertex*), cuja posição geral no pipeline do método pode ser verificada na Figura 3.3 (d). Mais detalhes sobre a tarefa são fornecidos na Seção 3.2.4.

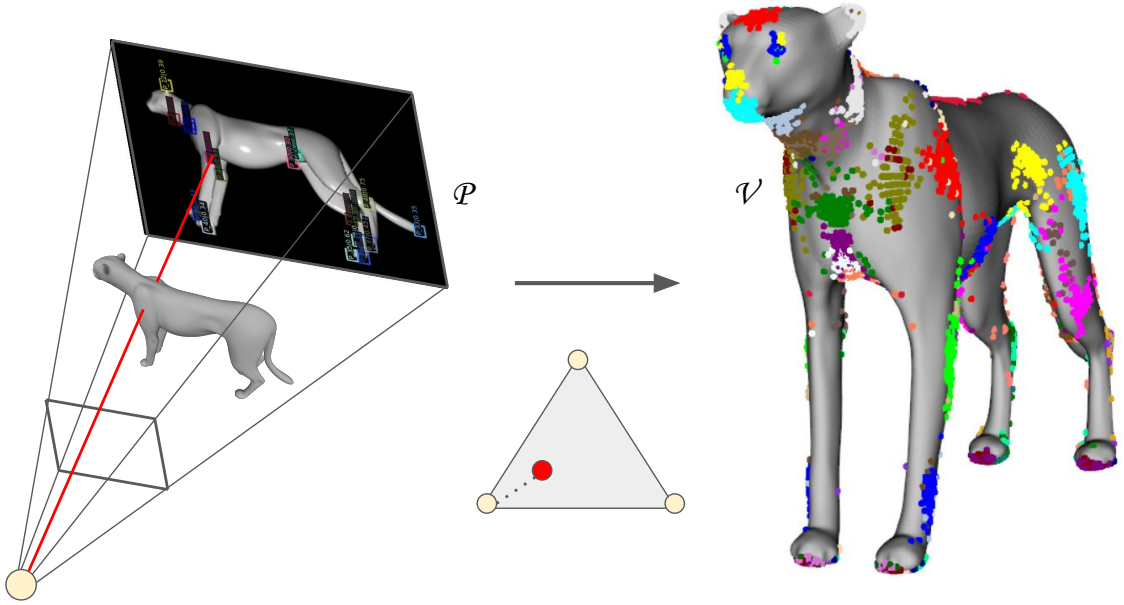

Figura 3.4: Para cada pixel analisado do conjunto $\mathcal{P}$, monta-se a cena original, de onde é possível determinar qual ponto da superfície gerou determinado pixel. O vértice mais próximo do ponto é então um dos candidatos ($\mathcal{V}$) às features definitivas.

O processo de mapeamento inverso resulta em uma lista de conjuntos de vértices $\mathcal{V} = \{v_1, v_2, ...v_n\}$ em que cada conjunto v_k é formado pelos vértices $\{v_{k_1}, v_{k_2}, ...v_{k_{n_k}}\}$, com n_k igual à quantidade de vértices candidatos à feature k. Devido a alguns fatores como a resolução da imagem, precisão da rede, falsos positivos e aproximações numéricas durante o processo de mapeamento reverso, a posição de cada vértice de um conjunto v_k, representada por $\mathbf{v_k}$, é uma posição candidata à feature k aproximada. Contudo, ao partir de resultados positivos de uma CNN cuja detecção seja precisa, a maior parte desses pontos estarão próximos ou ao redor da solução real.

Um dos comportamentos característicos das CNNs empregadas para tentar detectar features, é não identificar separadamente regiões de lados simétricos em imagens que não pertencem à formas idênticas às treinadas. Neste caso, pontos simétricos são aqueles que pertencem ao lado direito e esquerdo do plano sagital da forma analisada, e.g., uma feature que representa um olho esquerdo é identificada como olho esquerdo e também como olho direito; o mesmo acontece para a feature que representa o olho direito.

Para solução do problema adota-se um algoritmo de agrupamento Gaussiano. Esse algoritmo iterativamente aprende a identificar, sem supervisão, os dois conjuntos simétricos que apresentam distribuição aparentemente Gaussiana mais equilibrada a partir de um conjunto de pontos tridimensionais. Com isso, o método define os pares de conjuntos simétricos para cada feature que apresenta um par simétrico. Isso ainda

não classifica se o conjunto refere-se a uma feature localizada à direita ou esquerda do plano sagital. Tal tarefa será efetuada em um passo mais adiante.

A etapa seguinte define, para cada conjunto de vértices v_k, qual é o vértice do conjunto que mais precisamente representa a feature k. Embora a distribuição das posições dos vértices aparentemente seja majoritariamente Gaussiana, existem posições *outliers*, i.e., valores atípicos, como ilustrado na Figura 3.5. A solução não pode resumir-se a média dos pontos de cada conjunto, pois dependendo da posição e quantidade de *outliers*, tal média seria distorcida. A solução proposta é obtida através do uso de um estimador robusto de covariância proposto por Rousseeuw e Driessen [109], que elimina os *outliers* e encontra o ponto central do conjunto.

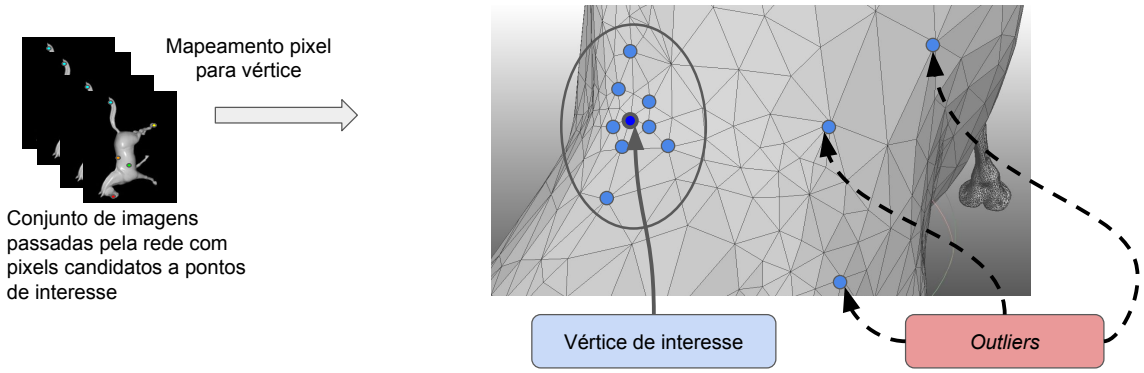

Figura 3.5: É possível perceber um conjunto de vértices de interesse, que na prática formam uma distribuição aparentemente Gaussiana com o vértice mais representativo ao centro. Os valores atípicos (*outliers*) são vértices fora da distribuição correta em uma média muito acima dos demais.

Após execução do algoritmo de covariância robusta, cada conjunto v_k será transformado em apenas um vértice, representando a feature k. Falta ainda corrigir a questão relacionada a simetria bilateral sagital. Sabendo quais são os pares de conjuntos simétricos de $\mathcal{V}$ através do gabarito $\mathcal{G}$, o método baseia-se em 3 features obrigatórias, especificadas neste gabarito para identificar o lado de cada feature simétrica. As posições dos vértices expressas pelas 3 features em questão, **a**, **b** e **c**, formam um plano imaginário que separa simetricamente as formas, e cuja normal aponta sempre à direita (mais detalhes na Subseção 3.2.2) da forma em relação ao plano sagital. Tomando como base o triângulo formado por **a**, **b** e **c**, detecta-se um ponto da malha que é interceptado por um raio partindo do centroide desse triângulo, em direção ao lado direito do plano sagital. Esse ponto interceptado atua como uma âncora de referência, à direita da forma. Realiza-se uma busca geodésica percorrendo a superfície a procura da menor distância de cada um dos dois pontos de cada par simétrico até essa âncora. O elemento do par que estiver em uma distância menor pertence ao lado direito, naturalmente o outro elemento pertence ao lado esquerdo. Por fim, troca-se as features que eventualmente estejam localizadas do lado incorreto, com seu respectivo par, obtendo o conjunto final de correspondência, Figura 3.3 (e).

3.2.2 Entrada

Para etapa de treinamento o método toma como entrada conjuntos, $\mathcal{T}$, de malhas 3D e um gabarito, $\mathcal{G}$, informando a estrutura semântica das n features de interesse. Já

a detecção de correspondência é executada sobre uma única malha ou algum conjunto, $\mathcal{S}$, de malhas com mesma topologia.

Cada conjunto $\mathcal{T}_i$ pertencente ao conjunto de malhas de treinamento $\mathcal{T}$, pode ser formado por sequências de animação ou poses variadas para uma determinada malha. Não assume-se que as poses entre os conjuntos de $\mathcal{T}$ sejam equivalentes ou correspondam a poses específicas, porém quanto mais variadas as poses e classes de formas disponíveis, maior será a capacidade de detecção do método.

Para o gabarito $\mathcal{G}$, composto por n índices de pontos de interesse há duas regras:

1. Indicar os pares de ponto de interesse simétricos lateralmente. Isto é, deve-se informar um conjunto de pares $\mathcal{G}_s = \{(p_1, q_1), (p_2, q_2), ...(p_{n_p}, q_{n_p})\}$ com $n_p \leq n$, em que cada elemento do conjunto são pares de índices de pontos simétricos.

2. Devem ser selecionados, entre os elementos de G, 3 índices de pontos de interesse a,b e c, cujas posições são aqui representadas por $\mathbf{a}$, $\mathbf{b}$ e $\mathbf{c}$, respectivamente. Tais features, servirão mais adiante (Subseção 3.2.4) para solucionar um problema de ambiguidade provocada pela simetria bilateral sagital. Os 3 elementos devem satisfazer as seguintes propriedades:

   (a) Devem estar localizados na superfície de um mesmo segmento não simétrico, e.g., cabeça ou tronco.

   (b) Os pontos $\mathbf{a}$, $\mathbf{b}$ e $\mathbf{c}$ devem definir o plano sagital das formas, i. e., representar features sobre o plano sagital — plano que divide as classes de formas suportadas em lado direito e esquerdo.

   (c) Devem estar distribuídos de forma que $(\mathbf{a} - \mathbf{c}) \times (\mathbf{b} - \mathbf{c}) = \mathbf{N}$ represente a normal do plano sagital, com direção apontando à direita do mesmo plano, assim como exemplificado na Figura 3.6.

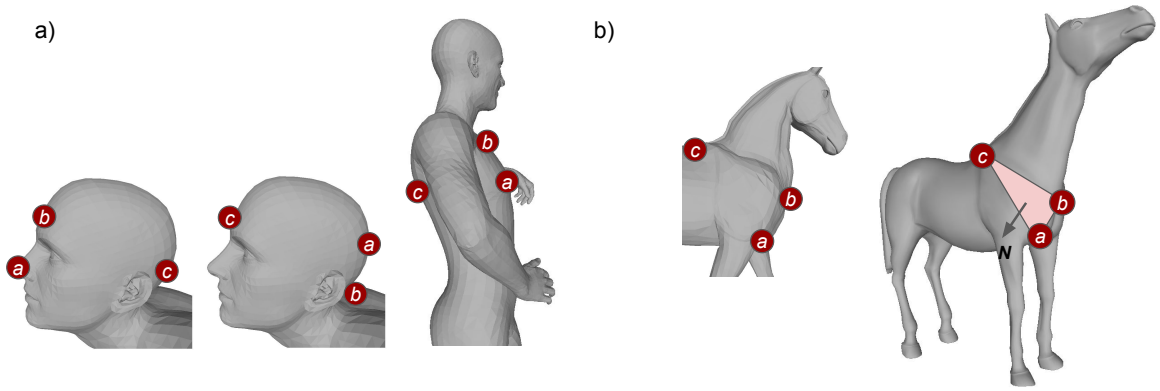

Figura 3.6: Ilustração de algumas possibilidades de definição da features a, b e c. A normal $\mathbf{N}$ do plano sagital, obtida por $(\mathbf{a} - \mathbf{c}) \times (\mathbf{b} - \mathbf{c})$ deve sempre possuir direção apontando à direita do mesmo plano, (b).

É importante ressaltar que as informações exigidas para $\mathcal{G}$ são estruturas para nortear as marcações de pontos de interesse nos treinamentos, e as correspondências serão realizadas em relação aos n pontos de interesse desse gabarito. A disposição e quantidade de pontos depende do propósito desejado e a princípio, não possui restrições, além das já citadas. Nesse trabalho, para experimentos relatados na Subseção 4

definimos um gabarito baseado nos resultados do trabalho de Medalha et al. [90], ilustrado na Figura 3.7, composto por 66 pontos localizados em regiões significativas das formas, como junções de membros, ponta de dedos, orelhas, tornozelos, dentre outros identificados na Tabela 3.1.

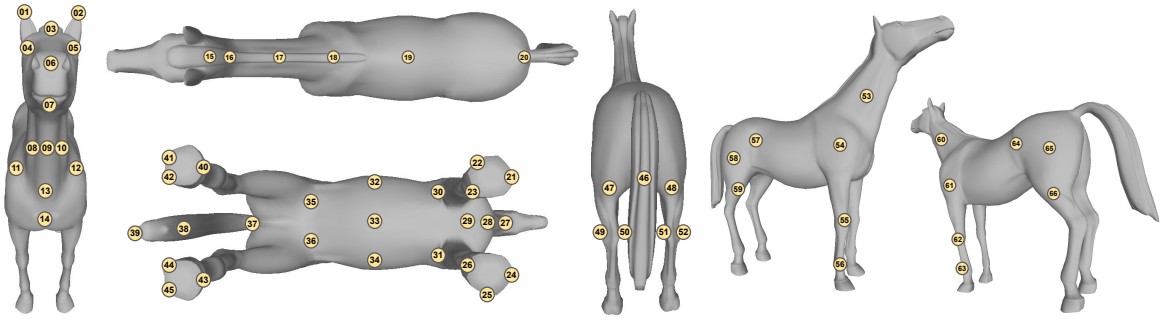

Figura 3.7: Mapeamento sobre malha de um cavalo, o significado semântico de cada índice pode ser verificado na Tabela 3.1

Para cada conjunto de treinamento $\mathcal{T}_i = \{m_{t_1}, m_{t_2}..., m_{t_{n_{t_i}}}\} \in \mathcal{T}$ é preciso relacionar cada elemento g_k do gabarito $\mathcal{G}$ com um vértice das malhas m_{t_i}. Reforçamos que basta correlacionar os elementos para uma única malha de m_{t_i}, pois as malhas do mesmo conjunto possuem mesma conectividade.

A correlação é realizada através da ferramenta MIG, em que uma das suas funções dentro do pipeline é facilitar através de interface gráfica a marcação desses pontos de interesse nas malhas de treinamento. A interface gráfica da ferramenta é ilustrada na Figura 3.8.

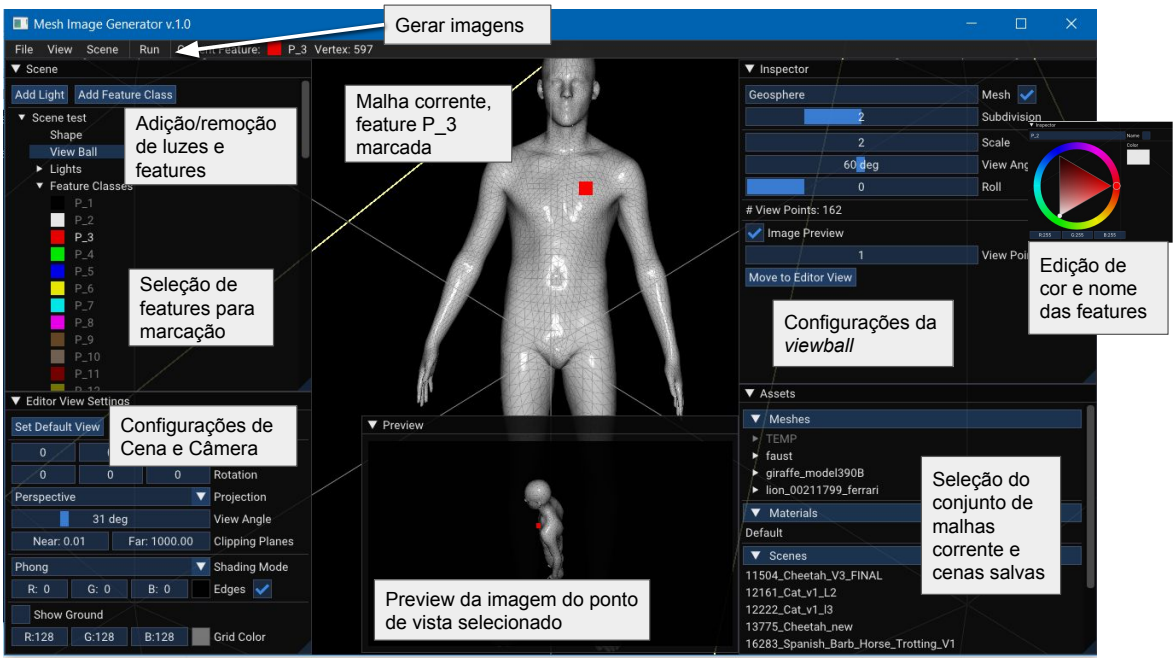

Figura 3.8: Funcionalidades acessíveis através da interface gráfica da MIG.

Ao executar a ferramenta, é possível carregar e modificar uma cena já com um gabarito definido, ou criar e modificar um novo gabarito. Seleciona-se graficamente um

Nr.	Semântica	Nr.	Semântica
01	ponta da orelha direita	34	esquerda abdômen frente
02	ponta da orelha esquerda	35	centro virilha direita
03	centro testa	36	centro virilha esquerda
04	olho direito	37	anus
05	olho esquerdo	38	meio cauda parte frontal
06	ponta do focinho/nariz	39	ponta da cauda
07	frente do queixo	40	ponta dedo mais longo pé direito
08	lateral direita pescoço de frente	41	calcanhar lado direito do pé direito
09	centro do pescoço de frente	42	calcanhar lado esquerdo do pé direito
10	lateral esquerda pescoço de frente	43	ponta dedo mais longo pé esquerdo
11	ombro direito frente	44	calcanhar lado direito do pé esquerdo
12	ombro esquerdo frente	45	calcanhar lado esquerdo do pé esquerdo
13	base pescoço frente	46	meio da cauda parte traseira
14	início osso esterno (limite peito de frente)	47	dobra atrás do joelho esquerdo
15	topo cabeça	48	dobra atrás do joelho direito
16	encaixe cervical na cabeça	49	tornozelo esquerdo à esquerda visto trás
17	centro pescoço costas	50	tornozelo esquerdo à direita visto trás
18	base pescoço costas	51	tornozelo direito à direita visto trás
19	centro das costas	52	tornozelo direito à esquerda visto trás
20	base da cauda ou cóccix	53	meio pescoço lado direito
21	ponta dedo mais longo mão direita	54	ombro direito
22	base da mão direita à direita próximo do punho	55	joelho direito lateral
23	base da mão direita à esquerda próximo do punho	56	punho direito lateral
24	ponta dedo mais longo mão esquerda	57	encaixe quadril lado direito
25	base da mão esquerda à esquerda próximo do punho	58	coxa direita
26	base da mão esquerda à direita próximo do punho	59	joelho direito
27	papo (embaixo queixo)	60	meio pescoço lado esquerdo
28	meio peito	61	ombro esquerdo
29	peito entre membros superiores	62	joelho esquerdo lateral
30	axila direita	63	punho esquerdo lateral
31	axila esquerda	64	encaixe quadril lado esquerdo
32	direita abdômen frente	65	coxa esquerda
33	centro abdômen	66	joelho esquerdo

Tabela 3.1: Descrição dos 66 pontos definidos no gabarito utilizado.

elemento g_k e com um clique relaciona-o com um vértice de uma das malhas de m_{t_i}. É permitido também a qualquer momento alternar a malha atual com outra do mesmo conjunto.

A malha dentro do MIG é normalizada de tal maneira que a maior dimensão de sua caixa delimitadora, do tipo AABB(*axis aligned bounding box*), seja a unidade (Figura 3.9). O centro da caixa delimitadora é posicionado na origem do centro de coordenadas. Em seguida, cria-se uma geoesfera — chamada aqui de *viewball* —, obtida através subdivisão de um dodecaedro, com raio igual a 1, cujo centro coincide com o centro da caixa delimitadora da malha normalizada. Uma câmera virtual é posicionada em cada vértice dessa *viewball*, apontando para o centro da caixa delimitadora, i.e., a direção de projeção vai do vértice da *viewball* para o centro da caixa delimitadora.

Definida a cena e informada a correlação de $\mathcal{G}$ com as malhas de $\mathcal{T}_i$, para cada câmera virtual, renderizam-se κ imagens em θ rotações de câmera em torno da direção de projeção, em que a soma dos ângulos das rotações totaliza 360°. Para um conjunto $\mathcal{T}_i$ são renderizadas $n_{t_i} \times \kappa \times \theta$ imagens. Os valores *default* para κ e θ são 162 e 4, respectivamente. Além disso, um arquivo do tipo csv é salvo informando os rótulos e posições (x,y) dos pixeis correspondentes as features visíveis em cada imagem, Figura 3.11. Repete-se o processo de correlação com gabarito e renderização das imagens e csvs para cada conjunto $\mathcal{T}_i$ de $\mathcal{T}$. Ao final, todas as imagens geradas para $\mathcal{T}$ são for-

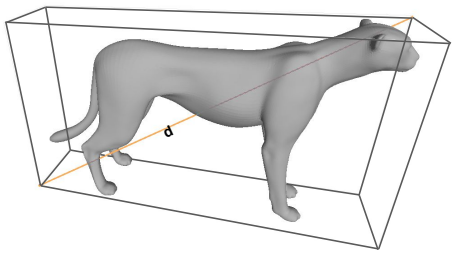

Figura 3.9: Exemplo de uma caixa delimitadora criada sobre uma forma 3D. A maior medida da caixa é a diagonal indicada por d.

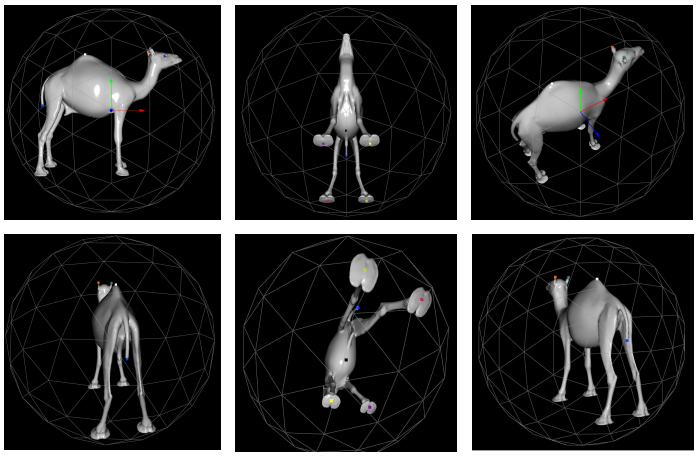

Figura 3.10: Geoesfera em que o centro de cada face da esfera representa um ponto de vista de uma câmera virtual apontado para o centro da mesma esfera. A partir destes pontos de vista, é possível capturar imagens da forma de ângulos perspectivas variadas.

necidas à CNN responsável por aprender a detecção de features. Essa é toda entrada necessária para o treinamento.

Após treinamento, o método está pronto para executar o processo de detectar correspondências para malhas destino $\mathcal{S}$. A entrada para esta etapa é bem mais simples. Utilizando o MIG, basta renderizar o conjunto de imagens para $\mathcal{S}$ — sem marcações e nem informar gabarito — e fornecê-las à CNN já treinada. Fornece-se os resultados dos rótulos e posição dos pixeis preditos pela CNN ao módulo P2V (Subseção 3.2.4) que retorna como saída a correspondência final.

3.2.3 Detecção de Features em Imagens

Para a detecção de features nas imagens, em primeiro lugar procurou-se por uma rede detectora de objetos baseada em CNN já bem avaliada na comunidade acadêmica. Inicialmente avaliamos um protótipo do pipeline utilizando resultados das redes Yolov5 [100, 99] e ATAS [73]. Esta avaliação inicial objetivou apenas a validação inicial da ideia, com uma verificação manual de experimentos utilizando de 5 pontos de interesse marcados em formas humanas. A validação geral do pipeline foi efetuada utilizando-se a rede baseada em CNN Yolov5, que apresentou resultados de detecção suficientes para validação da proposta, além de ser ser dotada de ferramentas que facilitam a visualização de métricas e gráficos sobre seu desempenho. Depois de definido o pipeline e critérios de avaliação do método, outras CNNs foram experimentadas e comparadas, tais avaliações são exibidas mais adiante no Capítulo 4.

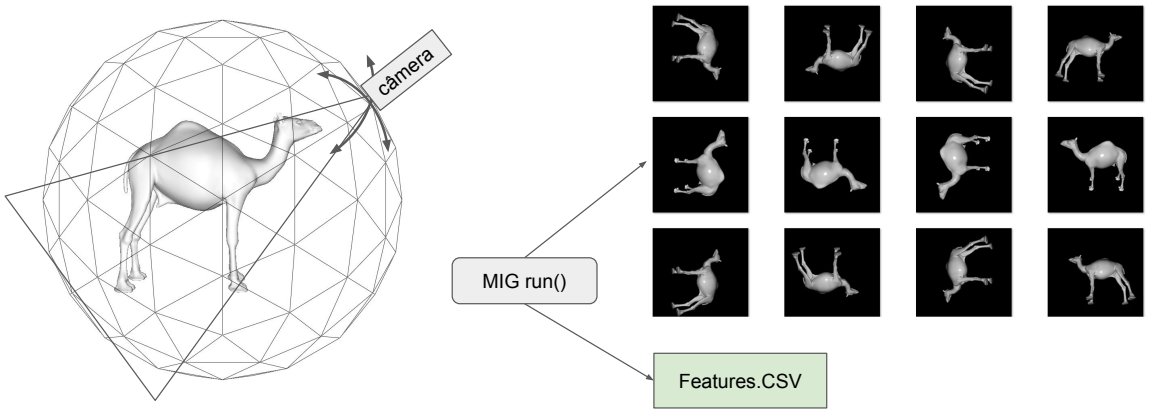

Figura 3.11: A ferramenta MIG automaticamente gera imagens a partir de pontos de vista partindo do centro das faces da *viewball*. A quantidade de faces é parametrizável, i.e., é possível definir a quantidade de pontos de vistam dependendo do propósito desejado para as imagens. Além disso, cada pontos de vista permite rotação ao redor do eixo de projeção da câmera virtual um uma quantidade de graus também parametrizável.

Experimentações iniciais foram executadas sobre um conjunto de treinamento composto por 4 conjuntos de animações de malhas, sendo 3 dos conjuntos (dinossauro, cavalo, camelo) obtidos das malhas utilizadas por Medalha et al. [90] e um conjunto de formas humanas montadas na ferramenta Daz3D [29]. Cada conjunto era composto por 15 poses de formas, sobre as quais foram geradas imagens em 162 pontos de vista em 4 rotações de câmera, totalizando 9720 imagens por conjunto.

Basicamente, há 3 tipos comuns de métricas para análise de detecção de objetos em CNNs: precisão (*precision*), memória (*recall*), taxa de precisão média (*mean average precision*). Em adição, um limite definido como aceitável de interseção de regiões retangulares para uma predição ser considerada positiva verdadeira.

Precisão refere-se a medida da porcentagem correta de predições positivas dentre todas as predições obtidas, já a memória é a medida de predições positivas corretas dentre todos os casos positivos reais. A primeira pode ser descrita por

$$\text{precision} = \frac{t_p}{t_p + f_p}, \tag{3.1}$$

enquanto a segunda é descrita por

$$\text{recall} = \frac{t_p}{t_p + f_n}, \tag{3.2}$$

em que t_p é quantidade de verdadeiros positivos, f_p refere-se à quantidade de falsos positivos, e f_n são falsos negativos (falha em prever um objeto que estava presente). Note que a precisão aumenta conforme diminui a quantidade de predições falso negativas, enquanto que a memória aumenta conforme diminui a quantidade de falso negativos. Para nosso propósito, veremos mais adiante que a precisão possui um peso maior nos resultados finais.

Note que nessa tese o método implementado utiliza um pixel para representar cada feature encontrada, mas as CNNs de detecção detectam regiões retangulares, não pontos. O ponto considerado como pixel candidato, e posteriormente vértice candidato à feature é tomado sempre como centro desses retângulos delimitadores. Ou seja, o

retângulo deve estar centralizado em relação à feature procurada, mas a largura e altura do retângulo não é tão relevante.

Para cada retângulo delimitador, calcula-se a sobreposição entre o retângulo predito b_p e o retângulo verdadeiro (correto) b_t. Essa medida é chamada de IoU (*intersection over union*), em que $\text{IoU} = A(b_p)/A((b_p \cup b_t))$, sendo $A(\cdot)$ a área do polígono em questão. Calcula-se os valores de precisão e memória utilizando IoU em relação à um determinado limite. Por exemplo, para um limite de 0.5, caso $\text{IoU} \geq 0.5$, classifica-se a predição como verdadeiro positivo, caso contrário a predição é classificada como falso positivo. Dito isso, significa também que para uma predição, obtém-se diferentes valores de falsos ou verdadeiros positivos dependendo do limite considerado.

A definição geral de precisão média AP (*average precision*) é encontrar a área formada sob os valores de precision − recall. Já a chamada mAP (*mean average precision*) corresponde a média de AP, i.e.,

$$\text{mAP} = \sum_{i=1}^{n_{a_p}} V(\text{AP}_i)/n_{a_p}. \tag{3.3}$$

em que n_{a_p} é a quantidade de inferências no conjunto e $V(\text{AP}_i)$ é a precisão média para a i-ésima inferência a_p. Todavia, em determinados contextos calcula-se P para cada classe e tira-se a média para obter o mAP, mas em outros considera-se AP e mAP como a mesma medida. Por exemplo, para o conjunto de dados COCO [77], considerado um dos *benchmarks* em testes de detecção, não há diferenciação entre AP e mAP.

Diante das diversas variações na metodologia de cálculo de mAP entre as redes, e a influência de cada arquitetura de rede na precisão e memória, a comparação desses valores por si só não garantiria a escolha da CNN mais adequada a tarefa aqui proposta. A arquitetura de algumas redes pode também influenciar nos resultados, já que para o nosso propósito interesse apenas o centro da região detectada, não importando a área da região; redes como a GFL [74] por exemplo utilizam uma abordagem diferente que trata erros de classificação e IoU de forma unificada.

Na teoria o método deve obter acertos na correspondência sempre acima dos obtidos pela CNN. Isso acontece por causa de alguns fatores comentados a seguir. Os valores verdadeiros positivos são os que de fato mais influenciam no mapeamento sobre as malhas. As métricas calculadas para a CNN utilizam uma média dos acertos, enquanto que na realidade, o método exige apenas que a CNN tenha precisão acima de 50% para garantir mapeamento correto.

A identificação de features não precisa ocorrer em todas as imagens em que ela está presente, contanto que a quantidade de verdadeiros positivos seja maior do que a de falsos positivos. Por exemplo, imagine que uma determinada feature f esteja presente em 11 imagens de um treinamento e que a CNN identifique somente 4 como verdadeiros positivos, 2 como falsos positivos e não identifique nada em 5 delas. De acordo com as Equações (3.1) e (3.2), temos $\text{precision} = 4/(4+2) \approx 0.66$ e $\text{recall} = 4/(4+5) \approx 0.44$, que resulta em uma precisão média $\text{AP} \approx 0.55$. Mas, ao mapear os pixeis para a superfície da malha de origem, teremos 4 vértices corretos e 2 que serão eliminados como *outliers*. Consequentemente, só serão considerados os 4 vértices corretos para calcular o vértice que representa a feature f.

A medida de erro desejada nesse trabalho não é necessariamente aquela fornecida apenas pela CNN, mas sim aquela que melhor indique se a correspondência semântica

está acontecendo. A validação é um aspecto importante do problema, já que é necessária para permitir comparar efetivamente resultados de diferentes métodos. A forma mais comum de validação nesse caso é a inspeção visual de resultados [128]. Exibir etapas de *morphing* entre formas é também uma maneira similar de avaliar a qualidade visual das correspondências, visto que espera-se uma transição suave de uma forma à outra para uma boa correspondência [148]. Tais procedimentos podem permitir uma comparação qualitativa dos resultados. Contudo, como as comparações visuais podem ser subjetivas ou laboriosas, procedimentos mais objetivos ou quantitativos também são importantes para permitir a comparação de resultados.

Em função do exposto, uma estratégia foi definida para tentar normalizar as comparações e tentar medir a efetividade do método. Após executar o pipeline de treinamento, executa-se os seguintes passos:

1. Seleciona-se r malhas de classes formando um conjunto $\mathcal{T}_e = \{t_{e_1}, t_{e_2}, ...t_{e_r}\}$ em que t_{e_u} é a u-ésima malha selecionada para estimar o erro. As malhas em $\mathcal{T}_e$ não fazem parte dos conjuntos de treinamento.

2. Com o auxilio da ferramenta MIG, marca-se as features esperadas $\mathcal{X}_u = \{x_{u_1}, x_{u_2}, ...x_{u_n}\}$ em cada malha t_{e_u} de $\mathcal{T}_e$, de acordo com o gabarito $\mathcal{G}$.

3. Executa-se o pipeline de correspondência, obtendo vértices preditos $\mathcal{Y}_u = \{y_{u_1}, y_{u_2}, ...y_{u_n}\}$. Agora é possível comparar os vértices obtidos $\mathcal{Y}_u$ com os vértices marcados $\mathcal{X}_u$.

São considerados dois tipos de erro para comparar resultados discretamente: quantitativo e qualitativo. Quantitativamente, mede-se quantas features foram preditas corretamente e quantas não foram preditas. Qualitativamente, medimos para as features encontradas, qual a distância geodésica entre as features preditas e marcadas. O erro quantitativo de cada malha t_{e_u} é dado por:

$$\omega_u = \sum_{i=1}^{u_n} \text{val}(x_{u_i}, y_{u_i})/u_n, \tag{3.4}$$

com

$$\text{val}(x_{u_i}, y_{u_i}) = \begin{cases} 1 \text{ se } D(x_{u_i}, y_{u_i}) > \varepsilon, \\ 0 \text{ se } D(x_{u_i}, y_{u_i}) \leq \varepsilon, \end{cases} \tag{3.5}$$

em que $D(x_{u_i}, y_{u_i})$ é a porcentagem da distância geodésica entre os vértices x_{u_i} e y_{u_i}, relativa a maior distância entre dois vértices da malha t_{e_u}. ε é a porcentagem limite aceitável de distância geodésica entre x_{u_i} e y_{u_i} para que a medida seja considerada correta. Essa porcentagem limite é parametrizável. O erro médio quantitativo total é portanto:

$$\Omega = \sum_{u=1}^{r} \omega_u/r. \tag{3.6}$$

Já o erro qualitativo para cada malha t_{e_u}, é a média da soma das porcentagens das distâncias geodésicas entre os vértices de $\mathcal{X}_u$ e $\mathcal{Y}_u$ relativa também a maior distância entre vértices da malha t_{e_u}, e pode ser expresso como:

$$\lambda_u = \sum_{i=1}^{u_n} D(x_{u_i}, y_{u_i})/u_n. \tag{3.7}$$

O erro qualitativo total é portanto:

$$\Lambda = \sum_{u=1}^{r} \lambda_u / r. \tag{3.8}$$

Na Figura 3.12 é possível verificar visualmente o erro quantitativo. A implementação disponibiliza este tipo de visualização facilitar a identificação de possíveis features não mapeados ou mapeados incorretamente dentro do ε exigido. Note ainda que, na Figura 3.12 (b), exibe-se os resultados de uma sequência de poses (poderia ser uma animação também) de 3 malhas de mesma topologia. Neste caso, como trata-se de mesma topologia entre as 3, é possível remover os erros identificados pela medida qualitativa, resultando em uma correspondência considerada 100% correta.

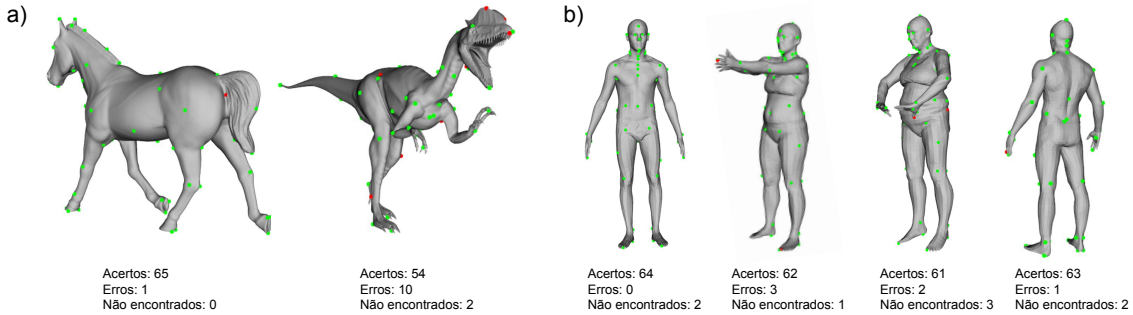

Figura 3.12: A implementação disponibiliza uma ferramenta que possibilita a visualização das correspondências consideradas corretas ou incorretas de acordo com o ε informado. Para $\varepsilon = 5\%$, a posição dos vértices corretos estão em verde; as incorretas, vermelhas. Como ilustrado na forma mais à esquerda (cavalo), pode acontecer de não ser possível determinar com precisão uma determinada feature se a posição correta não está visível. A feature marcada em vermelho deveria posicionar-se embaixo da cauda, mas a região não existe porque a cauda funde-se com as nádegas do cavalo. Já em (b), são 4 malhas com mesma topologia e consequentemente, mesmo mapeamento. Nesse caso, features deslocadas ou não encontradas em uma malha podem ser compensadas por outra.

O programa retorna para cada malha a quantidade de features consideradas acertos,quantidade de features consideradas erros, quantidade de features não encontradas e erro geodésico médio. Deixamos claro que o objetivo principal da métrica é comparar o desempenho entre redes, parâmetros e configurações discretamente. Mais adiante, no Capítulo 4, realiza-se uma comparação entre algumas CNNs. É importante dizer que não faz parte do escopo desse trabalho avaliar e comparar a vasta diversidade de arquiteturas e possíveis configurações de CNNs de detecção de regiões em imagens. Tal comparação, requer a definição de uma metodologia específica e uma série de avaliações empíricas considerando parâmetros e hiper-parâmetros das redes. Não obstante, as redes que mais se destacam na tarefa discutida — considerando mAP —, apresentam resultados similares ao considerarmos a tarefa de detecção de classes em imagens, como discutido mais adiante no Capítulo 4.

3.2.4 P2V

Como pode ser observado na Figura 3.3 (d), módulo P2V faz parte da etapa de correspondência e resolve tarefas como mapeamento reverso pixel para vértice, identificação de elementos de conjuntos simétricos, remoção de *outliers* e refinamentos finais do método.

Mapeamento reverso pixeis para vértices No processo de correspondência, após gerar imagens do conjunto $\mathcal{S}$, o próximo passo é transformar os resultados extraídos das imagens pela CNN em vértices correspondentes ao gabarito $\mathcal{G}$. É importante ter em mente que a precisão nos resultados de correspondência é influenciada pela precisão dos resultados obtidos da CNN, que por sua vez é influenciada pela variação de conjuntos de treinamento.

A quantidade de features identificadas em cada imagem 2D varia, pois dependendo do ponto de vista determinadas features não serão visíveis daquela posição. Além disso, a CNN pode não ter sido capaz de identificar a feature (falso negativo) naquela imagem. Por isso, é importante que seja gerada uma quantidade razoável de imagens na ferramenta MIG em diversos pontos de vista para garantir a presença de uma feature em mais de uma imagem.

Combinando os resultados do processamento de todas as imagens, tem-se a lista $\mathcal{P} = \{p_1, p_2, ...p_n\}$ de conjuntos de pixeis (x,y) em que cada conjunto de pixeis p_k representa pixeis candidatos à feature k.

Para cada pixel pertencente aos grupos de $\mathcal{P}$, carrega-se a cena com configurações originais que geraram a imagem a qual o pixel foi gerado. Executam-se então os seguintes passos:

(i) Sabendo das configurações da câmera, traça-se um raio de pixel partindo da posição da câmera virtual em direção a do pixel em questão.

(ii) Calcula-se o ponto de interseção do raio de pixel com a as faces da superfície da malha carregada. No caso de mais de um ponto de interseção, considera-se o ponto mais próximo da origem do raio.

(iii) A partir do ponto encontrado na superfície da malha em uma determinada face, considera-se como vértice projetado o vértice daquela face que encontra-se mais próximo do ponto.

O processo de reversão transforma a lista $\mathcal{P}$ na lista de conjuntos de vértices $\mathcal{V} = \{v_1, v_2, ...v_n\}$ em que cada conjunto v_k é formado por vértices candidatos à feature k.

Os membros de v_k são vértices candidatos à feature k, mas podem ser vértices cujas posições são distintas. Isso acontece por fatores como, por exemplo, problemas de aproximação numérica devido a resolução das imagens, variação nas posições das features detectadas pela CNN ou até mesmo detecções falso positivo. Todavia, é esperado que grande maioria dos vértices estejam mais próximos da solução real, com pequenas aproximações e um pequeno número de *outliers*. De fato, verificando os dados provenientes da CNN, o vértice mais adequado encontra-se no centro de uma distribuição dos vértices do conjunto, excluindo-se os *outliers*.

Não obstante, ao lidarmos com a projeção de um pixel de uma imagem com resolução limitada de volta para a superfície 3D da forma, ocorrem aproximações numéricas. Para algumas poucas imagens, alguns pontos de interesse encontrados pela rede podem apresentar resultados falso positivos. Ao avaliarmos os resultados obtidos, detectamos dois problemas evidentes: se o pixel que será projetado de volta à superfície da malha estiver posicionado muito próximo da fronteira (silhueta) da malha na imagem, a aproximação pode indicar que ele não toca a superfície, Figura 3.13 (a).

Também no caso de um pixel muito próximo da fronteira da malha na imagem ou casos de valores falso positivos, um ponto de interesse pode ser erroneamente mapeado distante da média dos outros pontos encontrados, tornando-se um ponto em posição atípica em relação a média (*outlier*).

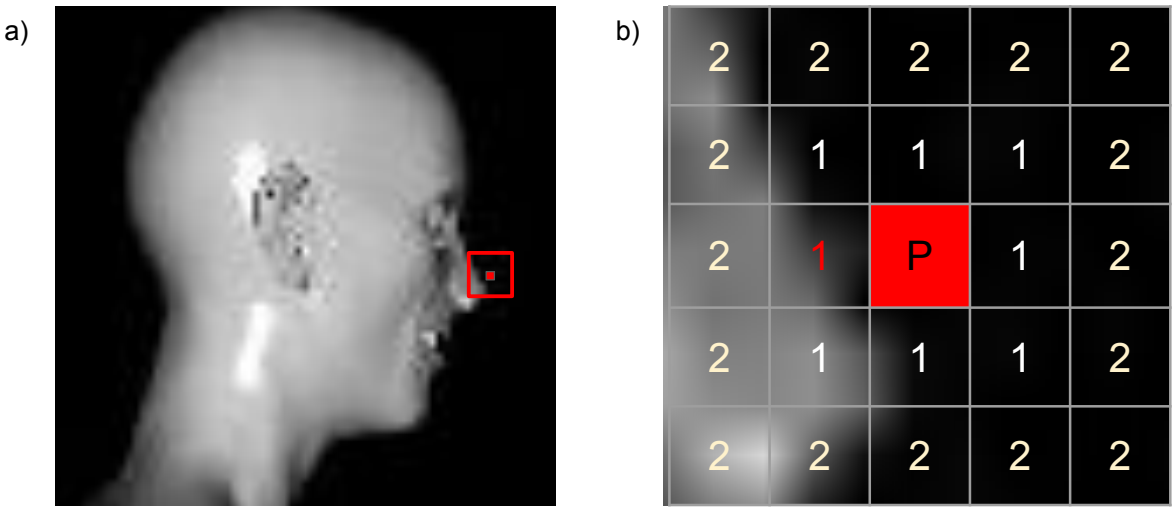

Figura 3.13: Nesta imagem retornada pela CNN, do ponto de vista ilustrado, ao calcular o mapeamento reverso pixel para malha, o raio partindo em direção à P não intercepta a superfície da malha. Dispara-se então raios em direção aos pixeis ao redor de P, em cada nível de vizinhança incrementalmente, até que a superfície seja interceptada.

Para o primeiro problema, a solução adotada é bastante simples: traça-se o raio de pixel partindo do pixel em direção à posição da câmera, caso a superfície da malha seja interceptada obtém-se o ponto desejado; caso não haja interseção — teoricamente deveria existir, pois a CNN identificou o pixel como um ponto de interesse da malha em imagens com apenas uma forma — repete-se a operação de disparar um raio em direção à câmera, mas a partir dos pixeis vizinhos do pixel origem. Repete-se o processo incrementando o nível de vizinhança ao redor do pixel de origem enquanto a superfície não for encontrada, Figura 3.13 (b).

O segundo problema, resume-se a ignorar os vértices em posição de *outliers* antes de calcular qual o vértice mais ao centro de cada conjunto v_k. O vértice mais central da distribuição de pontos poderia ser encontrado por estimadores de covariância empírica e estimadores de covariância reduzida, mas estes são muito sensíveis à presença de *outliers* nos dados. Para identificação dos *outliers*, optamos por um estimador de covariância robusta para calcular a covariância dos dados válidos, mais especificamente, a tarefa é solucionada através do método de MCD (*minimum covariance determinant*) como relatado mais detalhadamente na Subseção 3.2.6.

3.2.5 Features Simétricas

Como já exposto no Capítulo 1, formas de humanos e animais em geral representam seres biologicamente dotados da característica de simetria bilateral sagital. Ou seja, é possível traçar um plano imaginário, chamado plano sagital, que divide o corpo em duas metades iguais, ou praticamente iguais, formando duas partes simétricas.

Durante o treinamento, a CNN aprende a detectar regiões nas imagens que corres-

pondem às features desejadas. Todavia, no caso geral, ao detectar um ponto simétrico, a rede nem sempre identifica corretamente se essa região pertence ao lado direito ou esquerdo das formas. É possível identificar quais são os índices de pares de features simétricas por um conjunto $\mathcal{G}_s \in \mathcal{G}$, composto pelos índices de pares de features simétricas, dado obrigatório já definido no início do pipeline.

Isso quer dizer que, após o mapeamento reverso dos pixeis para vértices, cada par de features simétricas $(p_i, q_i) \in \mathcal{G}_s$ está representada em $\mathcal{V}$ por dois conjuntos de vértices para features simétricas v_{p_i} e v_{q_i}, em que um dos deveria pertencer à esquerda e outro à direita. Mas, ocorre que v_{p_i} possui vértices tanto do lado esquerdo como do lado direita da forma; o mesmo acontece com v_{q_i}. É possível notar esse comportamento dos conjuntos na Figura 3.14.

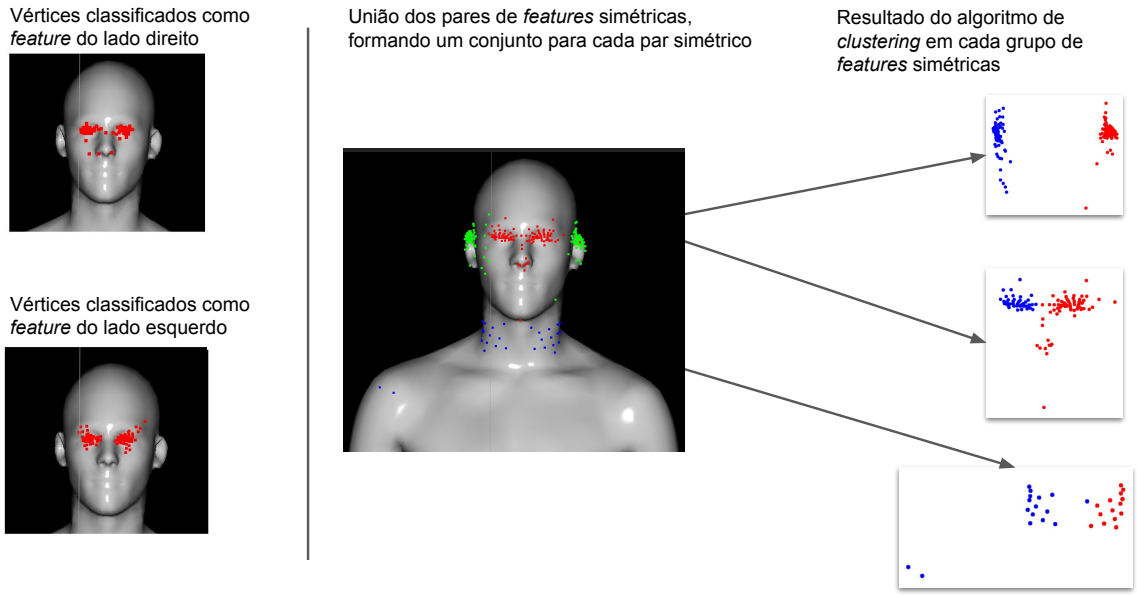

Figura 3.14: Um par de features dotadas de simetria retornam da CNN com uma classificação única, porém é possível notar uma distribuição aparentemente gaussiana ao redor de cada uma das features. É possível solucionar o problema separando os pontos em 2 conjuntos distintos através de algoritmos de *clustering*, técnica comum na área de análise de dados.

Como solução empregada para desambiguação destes vértices, primeiro une-se os conjuntos v_{p_i} e v_{q_i}. Em seguida, separa-se $v_{p_i} \cup v_{q_i}$ em 2 grupos $v_{g_a} = \{v_{a_1}, v_{a_2}, ...v_{a_{n_p}}\}$ e $v_{g_b} = \{v_{b_1}, v_{b_2}, ...v_{b_{n_s}}\}$, $n_s + n_p = n$, através de um algoritmo de agrupamento (Figura 3.15). O grupo que representa o lado direito será incluído em v_{p_i} e o grupo que representa o lado esquerdo em v_{q_i}.

Há três técnicas amplamente conhecidas podem ser utilizadas para formar os agrupamentos: *k-means*, modelo de mistura Gaussiana (GMM ou *Gaussian mixture model*) e agrupamento espectral (*spectral clustering*). A técnica mais adequada depende diretamente das características dos dados. A opção definida para os conjuntos de dados deste trabalho foi o modelo de mistura Gaussiana, que trata-se de um modelo probabilístico que parte do princípio de que todos os dados dos pontos originaram-se de uma mistura de um número finito de distribuições Gaussianas com parâmetros desconhecidos.

A solução da tarefa de agrupamento dá-se por um algoritmo de maximização de expectativas para ajustar os conjuntos. No nosso caso, desejamos dividir os dados

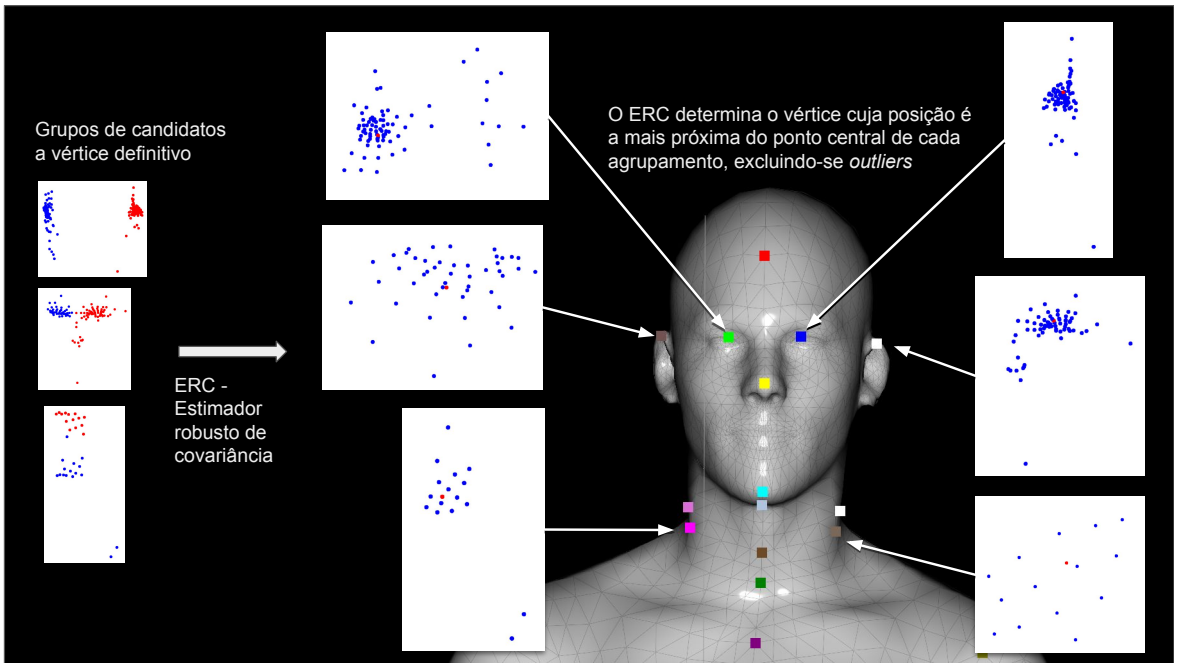

Figura 3.15: O estimador robusto de covariância toma como entrada um conjunto de posições de vértices e iterativamente aprende a calcular quais posições são atípicas e possuem distância até o centro do conjunto muito discrepante do restante do conjunto. Ignorando estes valores atípicos, o vértice mais adequado ao mapeamento é aquele mais próximo da posição central do conjunto. Obs.: nas imagens, os conjuntos de pontos representados são tridimensionais e foram rotacionados para que fosse possível observar o ponto central mais facilmente.

em 2 conjuntos, que posteriormente serão classificados em rótulos esquerda ou direita. Como os dados iniciais não indicam o rótulo do conjunto ao qual pertencem, a principal dificuldade vem do fato de que desconhece-se de quais pontos vieram de quais componentes. O algoritmo de maximização de expectativas, contorna o problema através de um processo iterativo. Em primeiro lugar, o algoritmo supõe componentes aleatórios — normalmente distribuídos ao redor da origem — e calcula para cada ponto a probabilidade dele ser gerado por cada componente do modelo. Em seguida, o algoritmo ajusta os parâmetros para tentar maximizar a vizinhança dos dados. A repetição do processo garante convergir para um ótimo local [112].

A opção por pelo algoritmo de mistura Gaussiana foi tomada considerando a distribuição das posições observadas dos vértices nos grupos $v_{s_{p_i}}$ e $v_{s_{q_i}}$, ilustradas na Figura 3.16. O algoritmo permite ainda informar a política de inicialização de valores e um tipo de covariância que será considerado. Para inicialização dos valores optamos pelo algoritmo *k-means* e matrizes de precisão (inverso das matrizes de covariância). Uma matriz de covariância é simétrica e positiva semi-definida, então a mistura Gaussiana pode ser equivalentemente parametrizada por matrizes de precisão. Em relação ao tipo de covariância, optamos pelo parâmetro de covariância esférica, em que cada componente possui sua própria variância.

Por tratar-se de um algoritmo iterativo e não supervisionado, um ponto importante é a inicialização dos valores. Para features bem identificadas pela CNN, a inicialização dos agrupamentos pelo algoritmo *k-means* retorna melhores resultados. Isso ocorre provavelmente porque o *k-means* pode agrupar os dados em *clusters* que sejam mais separáveis ou distintos uns dos outros, o que pode facilitar a identificação de padrões e a classificação dos dados. Ao agrupar os dados em *clusters*, o algoritmo

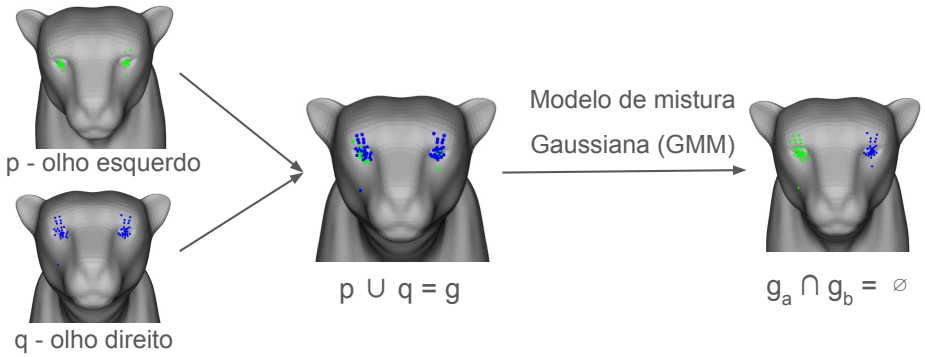

Figura 3.16: A figura mostra conjuntos p e q retornados como vértices candidatos ao centro dos olhos da forma. É evidente a confusão entre pontos a esquerda e direita de cada feature; o método então une os dois conjuntos e utiliza o GMM para separar o agrupamento em novos agrupamentos distintos esquerda/direita.)

também ajudar a reduzir a dimensionalidade dos dados. Mas em situações atípicas, em que a CNN pode ter maior dificuldade em identificar determinadas features simétricas — uma feature embaixo dos cascos dianteiros de um cavalo, por exemplo, é muito parecida visualmente nos 4 cascos —, dependendo da posição destas features, os valores falsos positivos podem formar pequenos grupos de *outliers*. Como essa etapa é realizada antes da etapa de remoção dos *outliers*, dependendo da inicialização pode acontecer do algoritmo formar grupos indesejados, como por exemplo no caso ilustrado na Figura 3.17. Mesmo não sendo um caso comum, a situação pode causar instabilidade no resultado entre várias execuções. Para eliminar tal instabilidade, repete-se a inicialização da mistura Gaussiana 30 vezes e toma-se como resposta correta a de maior pontuação. Esse número de repetições é parametrizável e 30 foi uma quantidade de repetições superdimensionada em relação aos resultados obtidos.

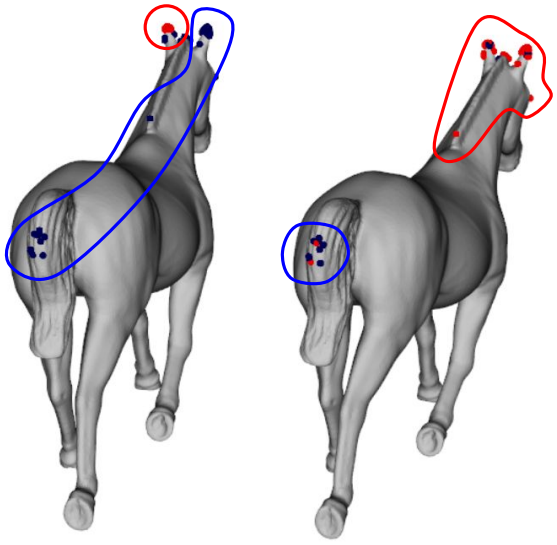

Figura 3.17: Dependendo a inicialização, em casos bem específicos, o algoritmo de agrupamento pode selecionar grupos não desejados. A figura mostra agrupamentos de pontos de features definidas na ponta das orelhas do cavalo. À esquerda da figura destaca-se o agrupamento desejado; os pontos incorretos apontados na cauda serão removidos na etapa de remoção de *outliers*. À direita, o agrupamento não permitirá o reconhecimento de um dos pontos das orelhas e resultará em um ponto na orelha e outro na cauda. Porém, considerando que a detecção de feature retornará mais acertos do que falso positivos, o agrupamento correto possuirá melhor pontuação.

Separados os vértices de $v_{p_i} \cup v_{q_i}$ em 2 grupos $v_{g_a} = \{v_{a_1}, v_{a_2}, ...v_{a_{n_s}}\}$ e $v_{g_b} = \{v_{b_1}, v_{b_2}, ...v_{b_{n_s}}\}$, resta ainda decidir a qual dos lados do plano sagital cada agrupamento v_{a_i} e v_{b_i},$i < n_s$, pertencem. Mas antes disso, passe-se pela etapa de remoção de *outliers* e seleção de apenas um vértice por feature. Com apenas um vértice por feature, ai então o método decidirá à qual lado do plano sagital cada feature simétrica pertence, Subseção 3.2.6.

3.2.6 Pós-processamento

Para determinar os *outliers* em v_k e encontrar o vértice k que melhor representa g_k, utilizamos o algoritmo MCD proposto por Rousseeuw e Driessen [109]. MCD trata-se de um estimador de dados multivariados e dispersos cujo objetivo é encontrar $\mathbf{h_o}$ observações (de um total de $\mathbf{h_t}$) cuja matriz de covariância possui o menor determinante. O estimador pode ser aplicado em dados com distribuição aparentemente Gaussiana mas também se comporta bem em outros dados com distribuição simétrica.

O objetivo do algoritmo MCD é encontrar $\mathbf{h_o}$ obervações extraída de um total de $\mathbf{h_t}$, cuja matriz de covariância clássica possui o menor determinante — que implica menor distância média dos pontos em relação ao ponto central. A estimativa MCD é então a média destes $\mathbf{h_o}$ pontos, e a estimativa de espalhamento é sua matriz de covariância. A etapa chave do algoritmo baseia-se no fato de que, iniciando de qualquer aproximação do MCD, é possível calcular outra aproximação com um determinante ainda menor. Nesse caso, $\mathbf{h_t}$ é composto pelas posições dos vértices presentes em v_k. Após calcular o estimador de determinante de covariância mínima, pode-se fornecer pesos às observações de acordo com suas distâncias de *Mahalanobis*, resultando em uma estimativa reponderada da matriz de covariância do conjunto de dados.

Antes da aplicação do MCD, verificamos, para todos os vértices candidatos à feature k de cada grupo $v_k = \{v_{k_1}, v_{k_2}, ...v_{k_{n_k}}\}$, qual v_{k_w} repete-se o maior número de vezes. Caso o vértice v_{k_w}, que mais se repete, apareça em mais de 1/3 das n_k amostras de v_k, v_{kw} já é selecionado como vértice definitivo correspondente a g_k, desconsiderando por ora a simetria bilateral. Isso acontece porque, além de ser um vértice que aparece em mais de 30% das amostras, a partir de elementos com presença de aproximadamente 25% em $\mathbf{h_t}$, as distâncias (determinante das matrizes de covariância) entre as $\mathbf{h_o}$ selecionadas pelo MCD resultam em 0. Caso o vértice que mais se repita não alcance 1/3 de n_k, aplica-se o algoritmo MCD. A figura 3.18 exemplifica a aplicação do algoritmos sobre conjuntos de pontos com diversos *outliers*, permitindo a obtenção do centroide da distribuição dos pontos mais adequados.

Após a execução do MCD, obtém-se um ponto 3D ao centro da distribuição de vértices v_k, excluindo-se os *outliers*, mas ainda é preciso localizar o vértice mais próximo desse ponto, correspondente a g_k. Se o retalho da superfície onde localizam-se os vértices v_k possuir uma curvatura muito irregular, pode ser que a distância Euclidiana do ponto ao centro da distribuição encontrada até o vértice mais próximo encontre um vértice que não encontra-se no centro do retalho. A solução implementada encontra o vértice mais adequado, calculando em qual deles a soma das distâncias até os outros ao redor é a menor possível. Cria-se uma matriz simétrica $M_{n_k \times n_k}$ de distâncias entre cada vértice de v_k para todos os outros. O vértice g_k será aquele representado pela linha cuja soma das colunas seja menor do que dos outros. Ou seja, o índice do melhor

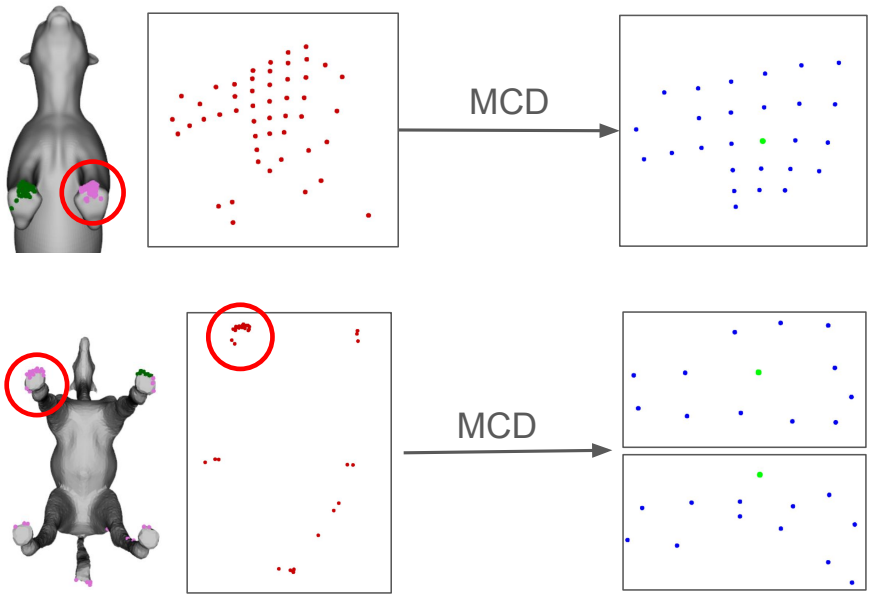

Figura 3.18: Exemplo da aplicação do MCD em resultados de agrupamento de vértices em features de pona do casco de 2 modelos. Repare que o modelo mais abaixo possui *outliers* bem espalhados em relação a região mais densa (e correta) de pontos. Mesmo assim o algoritmo mostrou-se eficiente em remover os pontos atípicos e encontrar o centroide da distribuição.

vértice em v_k é dado por $\arg\min(l_w)$, $l_w = \sum_{j=1}^{n_k} d_{i,j}$ em que $d_{i,j}$ é a distância do vértice da linha i até o vértice da coluna j.

Mesmo após definidos os vértices correspondentes a cada g_k, para features simétricas falta ainda conferir e decidir quais os lados corretos em relação ao plano sagital. O processo inicia-se encontrando por uma âncora $\mathbf{s_r}$ do lado direito da forma, i.e., um ponto à direita do plano sagital, cujo lado direito é dado pela direção da normal $\mathbf{N}$, definida obtida através dos pontos $\mathbf{a}$, $\mathbf{b}$ e $\mathbf{c}$ definidos na Subseção 3.2.2. Tal âncora assumirá o papel de ponto de referência para encontrar, entre os pares de pontos simétricos, qual está mais próximo do lado direito da forma.

O triângulo formado pelas posições $\mathbf{a}$, $\mathbf{b}$ e $\mathbf{c}$ dos 3 pontos de interesse a, b e c indicados em $\mathcal{G}$, possui normal $\mathbf{N}$, centroide $\mathbf{c_t}$ e pertence ao plano sagital. Calcula-se a interseção da malha com o vetor partindo de $\mathbf{c_t}$ na direção N. A âncora $\mathbf{s_r}$ é definida como a posição do vértice mais próximo do primeiro ponto de interseção encontrado na superfície da forma, estando sempre posicionada à direita da forma, Figura 3.19.

Em seguida, para cada par de features simétricas v_{a_i} e v_{b_i}, calculam-se as respectivas distâncias geodésicas mínimas de suas posições até a âncora $\mathbf{s_r}$. A distância é calculada pelo algoritmo de *Dijkstra*, procurando pelo caminho de custo mínimo pela malha como se ela fosse um grafo em que os vértices são nós e os pesos das arestas é o comprimento das arestas. O vértice que possuir menor distância até $\mathbf{s_r}$ pertence ao lado direito, enquanto o vértice com a menor distância pertence ao lado esquerdo (Figura 3.20).

A efetividade da rotina é garantida por trata-se de uma busca por vértices simétricos em dois lados simétricos e com conectividade dos elementos da malha semelhantes. Isto é, considerando a malha como sendo um grafo $\text{Graf}_m = \mathcal{D}, \mathcal{E}, \mathcal{C}$ cujo peso das arestas é igual ao comprimento das arestas da malha, e em que $\mathcal{E}$ corresponde aos nós esquerdos, $\mathcal{D}$ corresponde aos nós do lado direito e $\mathcal{C}$ os vértices não simétricos que

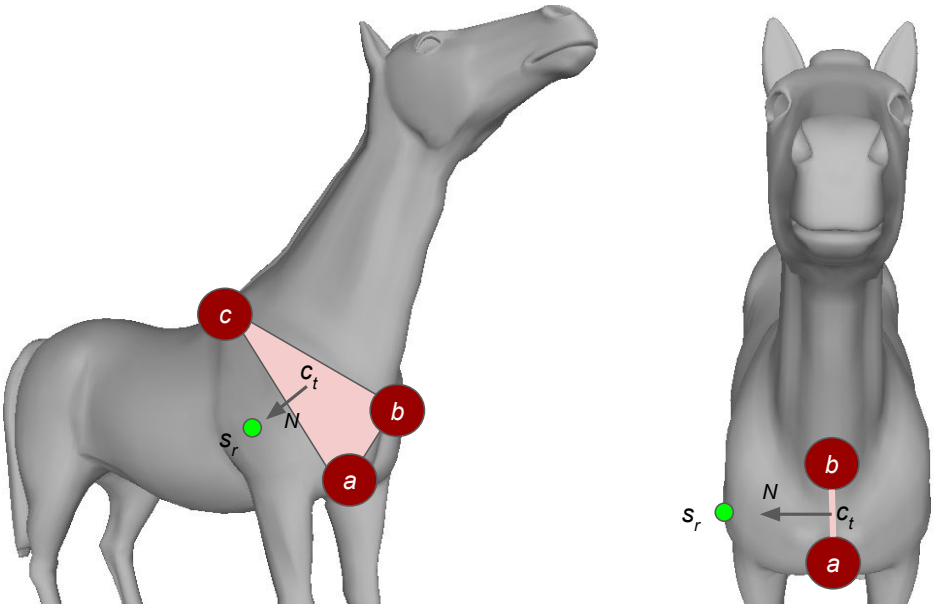

Figura 3.19: Ilustração do triângulo formado pelos pontos (**a**, **b**, **c**) cuja normal **N** e sempre aponta para o lado direito do plano sagital da forma. Partindo do centroide $\mathbf{c_t}$ do triângulo e seguindo na direção da normal **N** encontra-se um ponto na superfície da forma. O vértice mais próximo desse ponto é definido como a âncora $\mathbf{s_r}$.

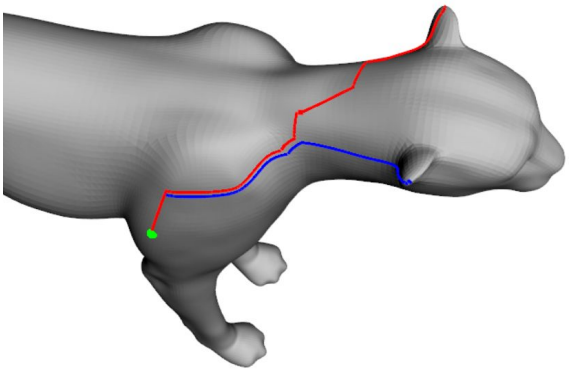

Figura 3.20: A figura mostra o traçado encontrado pelo algoritmo de Dijkstra do ponto âncora $\mathbf{s_r}$ até os pontos simétricos de um determinada feature. O ponto pertencente ao lado direito sempre resultará no menor traçado.

conectam $\mathcal{D}$ até $\mathcal{E}$. Considere o ponto de partida em $\mathcal{D}$ como a âncora $\mathbf{s_{r_d}} \in \mathcal{D}$ e seu ponto simétrico $\mathbf{s_{r_e}} \in \mathcal{E}$. A menor distância de $\mathbf{s_{r_d}}$ até um nó qualquer de $\mathcal{D}$ é d_{d_v}, a menor distância de $\mathbf{s_{r_e}}$ até seu nó simétrico em $\mathcal{E}$ é d_{e_v}, e d_{e_d} a menor distância para ir de um nó de $\mathcal{D}$ para $\mathcal{E}$. Como $\mathcal{D}$ e $\mathcal{E}$ são subgrafos simétricos, $d_{d_v} \approx d_{e_v}$ e além disso, partindo de $\mathbf{s_{r_d}}$ até algum nó de $\mathcal{D}$, temos a distância d_{d_v}, enquanto que até o nó simétrico em $\mathcal{E}$, a distância será $> (d_{d_v} + d_{e_d})$. Assim, a âncora sempre estará mais próxima do par do lado direito do que do seu simétrico do lado esquerdo.

Caso v_{a_i} esteja à direita, consequentemente v_{b_i} deverá estar à esquerda, basta portanto somente conferir cada g_k e trocá-lo com o par simétrico, caso estejam em posições trocadas. Ao fim, teremos definido todos os k vértices correspondentes a $\mathcal{G}$.

Capítulo 4

Experimentos e Resultados

4.1 Considerações Iniciais

Ao contrário dos experimentos das abordagens baseadas nos domínios espectral e espacial, na abordagem multivisão foi possível montar conjuntos de dados com classes mais heterogêneas e variadas, pois para a renderização das imagens não há restrições em relação a topologia das malhas. Como já citado no Capítulo 3, o desenvolvimento do pipeline, assim como os experimentos iniciais forma criados empregando a rede Yolov5. Durante essa fase, foram observados os comportamentos dos dados a respeito de resultados do mapeamento pixel para vértice, espalhamento dos vértices que representam uma determinada feature, simetria entre determinadas features, dentre outros comportamentos relatados nas seções a seguir. O resultado de algumas estratégias utilizadas para solucionar ou mitigar problemas encontrados, assim como o resultado de correspondências encontradas pelo método são mostradas nesse capítulo.

O capítulo encontra-se organizado da seguinte forma: a Seção 4.2 detalha o conjunto de dados utilizado para treinamento da CNN, ressaltando suas características mais relevantes. Na Seção 4.3, além do desempenho da rede Yolov5 nos experimentos, mostra-se um comparativo entre algumas das principais redes baseadas em CNNs presentes na ferramenta *MMDetection* na tarefa de correspondência aqui definida. A Seção 4.4 mostra alguns resultados práticos a aplicação do método sobe diversas malhas. Mostra-se inclusive a aplicação de uma técnica de *morphing* tomando como entrada pontos de correspondência aprendidos pelo método. Finalmente, a Seção 4.5 realiza um breve comentário sobre os resultados obtidos.

4.2 Definição dos Dados

Selecionamos, dentro das malhas disponíveis, 6 classes de formas 3D para treinamento: malhas de formas humanas criadas em poses variadas com o software Daz3D [29] (15 poses), malhas de um cavalo (15 poses), malhas de um camelo (15 poses)e malhas de um terópode (15 poses) utilizados em [90], malhas de um gato (10 poses) e malhas de uma leoa (10 poses) disponibilizadas por [121], Figura 4.1. Não é obrigatório que poses entre as classes sejam equivalentes ou correspondam a poses específicas de outro conjunto de treino.

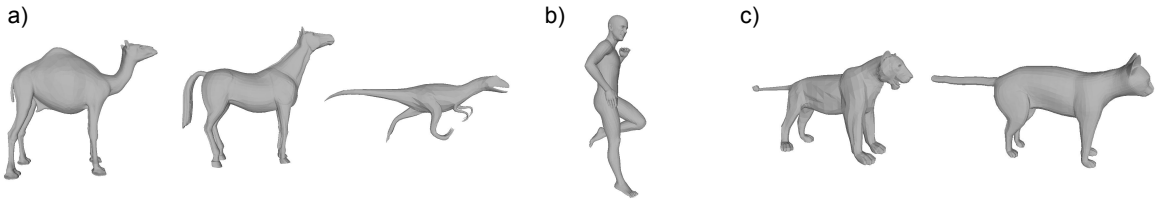

Figura 4.1: Malhas utilizadas no treinamento dos experimentos: dataset criado no softwareDaz3D [29], 3 animais do dataset utilizado no trabalho de Medalha et al. [90] e 2 animais do dataset utilizado por Sumner et al. [121]

Uma única malha de cada conjunto foi marcada na ferramenta MIG de acordo com o gabarito ilustrado na Figura 3.7, composto de 66 pontos semânticos de interesse, definidos conforme os resultados obtidos pelo trabalho de Medalha et al. [90].

Empregando a ferramenta, foram renderizadas imagens para uma geoesfera composta por 162 vértices, i.e., 162 pontos de vista, em 4 rotações ao redor do eixo de projeção para um dos pontos de vista, totalizando 51840 imagens. Todas as imagens foram renderizadas em resolução 512x512, com raio da geoesfera igual a dois, ângulo de vista da câmera virtual de 60° e retângulo delimitador da região de 16x16 pixeis. As imagens são renderizadas empregando o modelo de iluminação de *Phong* e com fundo na cor preta. São utilizadas 2 luzes posicionadas acima, à esquerda e à direita da câmera virtual, com o material da malha com cor de reflexão difusa e ambiente igual a branco.

Para comparação de resultados entre diferentes parâmetros de execução, definimos um conjunto de avaliação composto por malhas com dados anotados seguindo o processo descrito na Subseção 3.2.3. Esse conjunto rotulado é composto de $r = 13$ malhas, cobrindo as principais características semânticas representadas nas classes de formas suportadas pelo treinamento. São cobertas classes de equino, camelídeo, felino, humano e terópode, sendo:

- 1 malha de equino retirada de [127]. Essa malha possui geometria diferente da forma de cavalo utilizada no treinamento, fornecendo um exemplo de uma classe treinada, mas de complexidade geométrica distinta.
- 1 malha de camelídeo de [90] e 1 malha de girafa retirada de [25]. O camelo possui mesma topologia do modelo do treinamento, mas em uma pose (deformação) diferente. Teoricamente, essa malha sempre deve possuir alta taxa de acerto. A girafa apresenta algumas características semelhantes ao camelo, mas um nível de detalhamento mais baixo, além de ser um tipo de animal que não existe no treinamento.
- 5 malhas de felinos — 3 grandes felinos, 2 pequenos — retirado de [127]. Modelo um pouco similar às classes de gato e leoa existente no treinamento, compartilhando algumas características semânticas comuns, mas com geometria diferente.
- 4 malhas de formas humanas retirada de [15]. Modelos humanos, mas com nível de detalhamento e biotipo distintos dos humanos presentes no treinamento.
- 1 malha de terópode retirada de [127]. Malha bem diferente do conjunto de treinamento, a região da cabeça não identifica-se com nenhum dos modelos treinados.

4.3 Comparações e Redes Detectoras de Objetos baseadas em CNNs

A CNN principal usada para geração das imagens dos experimentos relatados foi a Yolo5v [100, 99], em sua última versão. O treinamento dessa rede específica ocorreu pelo ambiente virtual *Google Colab Pro+, GPU A100-SXM4-40GB, Intel(R) Xeon(R) CPU @ 2.00GHz 8-core, 53 GB RAM*, levando aproximadamente 15 horas, para 250 épocas, *lr* igual a 0.01, tamanho do lote = 32, 66 classes e o restante com os parâmetros *default*. A taxa de confiança (*score*) utilizada na CNN para se considerar uma detecção foi de 0.25.

Foram avaliadas taxas de confiabilidade no intervalo de 0.5 até 0.15. Na tabela, estão listados os dados de 0.30 até 0.15. Nota-se o seguinte comportamento geral: quanto maior a taxa de confiabilidade — até um certo limite —, maior a chance que aumentar a qualidade dos pontos; em contrapartida, maior a probabilidade de não encontrar determinados pontos.

No nosso caso, a taxa de confiança limite é por volta de 0.25, definida com base nos experimentos de *correspondência* informados na Tabela A.5, sobre o conjunto de avaliação com 13 malhas. Note que a malha *giraffe_model390B* não encontrou pontos essenciais do gabarito e sequer pôde ser avaliada.

N. Confiança	**Acertos**	**Erros**	**Não Ident.**	**Ω**	**Λ**
0.30	56,61	3,07	1,23	4,77	1,64
0,25	61,38	3,69	0,92	5,67	1,84
0.20	61,15	4,15	0,69	6,36	2,00
0.15	60,61	4,92	0,46	7,49	2,17

Tabela 4.1: Avaliação dos acertos para malhas do conjunto rotulado para níveis de confiança de 0.30 a 0.15 nas detecções em imagens, para limite de erro $\varepsilon = 0.05$. Os dois últimos valores de Média Geral são Ω (3.6) e Λ (3.8), respectivamente.

O caminho inverso também ocorre, ao diminuir a taxa de confiabilidade, aumentamos a quantidade de pontos encontrados, mas a qualidade geral decresce. Não coincidentemente, a taxa de confiabilidade que produz melhores resultados gerais de detecção na CNN é ao redor de 0.252, Figura 4.2.

Os parâmetros de configuração foram adotados para obtenção de uma solução generalista, i.e., para malhas variadas dentro das classes de formas suportadas, e não apenas para uma única classe de forma como somente formas de cavalos ou de humanos. Ao considerar a correspondência para um caso específico de malha, é possível que a alteração de parâmetros melhore os resultados para o caso específico em questão. Para problemas em que a presença de pontos é mais importante do que a precisão, pode-se diminuir levemente a taxa de confiança; para outros em que a precisão é mais importante a taxa de confiança pode ser levemente incrementada. Nos experimentos com o dataset de avaliação, a média de vértices candidatos para cada feature ficou em aproximadamente 205, enquanto que o menor e o maior número de vértices candidatos para uma determinada feature em uma malha foi 1 e 403, respectivamente.

Em relação as métricas de erro de detecção de regiões na CNN, como esperado e já relatado na Subseção 3.2.3, para o conjunto gerado de imagens de formas do conjunto

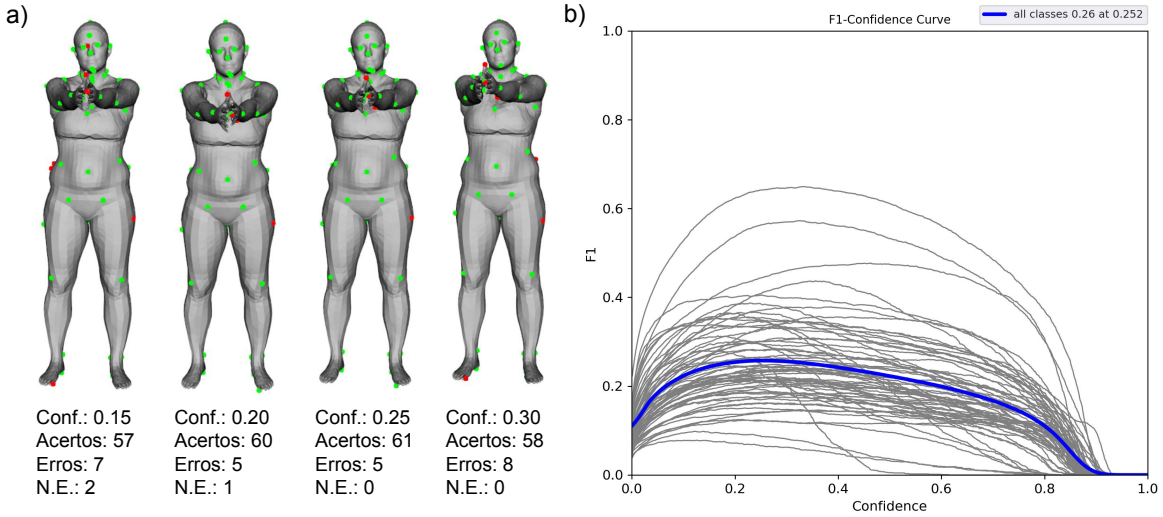

Figura 4.2: Em (a), são mostradas imagens de uma malha do grupo rotulado com erros (dentro da margem especificada) marcados em vermelho e acertos em verde, com as correspondências retiradas de variados níveis de confiança na etapa de detecção na CNN. É possível visualizar a diferença de acertos entre os variados níveis de confiança. Em (b) encontra-se o gráfico F1 $\times$ confidence da CNN. A medida Fα é média harmônica ponderada da precisão e memória de um classificador, tomando $\alpha = 1$. Isto é, no gráfico tanto precisão como memória possuem a mesma importância.

de avaliação e olhando somente para valores dos erros obtidos pela validação, tem-se a impressão (incorreta) de que os dados não serão satisfatórios.

Em um primeiro momento, para o conjunto gerado de imagens de formas do conjunto de avaliação, os valores dos erros obtidos pela validação pode levar à impressão (incorreta) de que os resultados da detecção não seriam suficientes para o pipeline de correspondência. Impressão essa já desmentida na Subseção 3.2.3, que mostra que poucas detecções na rede já são suficientes para o pipeline de correspondência. Nos experimentos realizados com a Yolov5 com o dataset anotado de validação, obteve-se: precisão igual a 0.278; memória igual a 0.248; e mAP50 de 0.191. Os resultados individuais obtidos por feature podem ser verificados na Tabela A.5, no Apêndice A. Os resultados obtidos de precisão, memória e mAP são listados na Figura 4.3. Observando a matriz de confusão dos dados avaliados, Figura 4.4, percebe-se claramente que parte dos erros dá-se por causa de features simétricas.

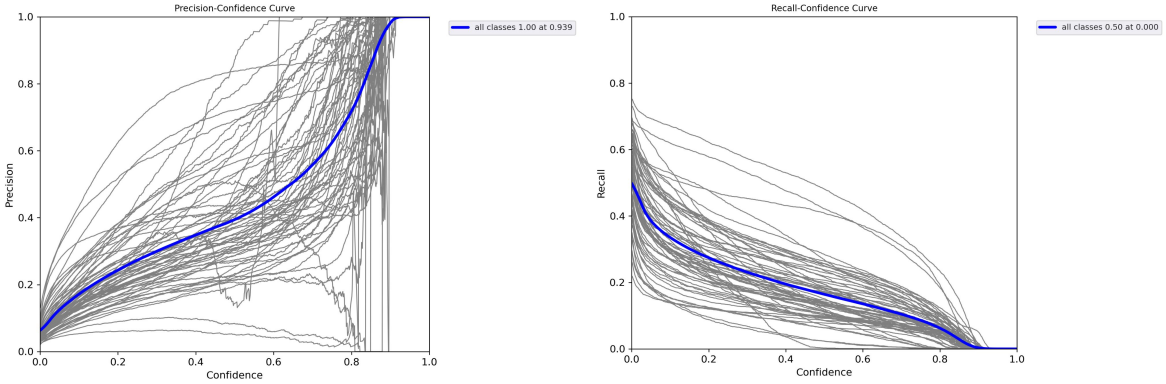

Figura 4.3: Nos gráficos de confiança e de memória, nota-se a grande variação nos resultados para algumas das features no teste de validação dos conjuntos rotulados. É possível notar também que o nível de confiança mais equilibrado ao considerarmos as duas medidas é por volta de 0.25.

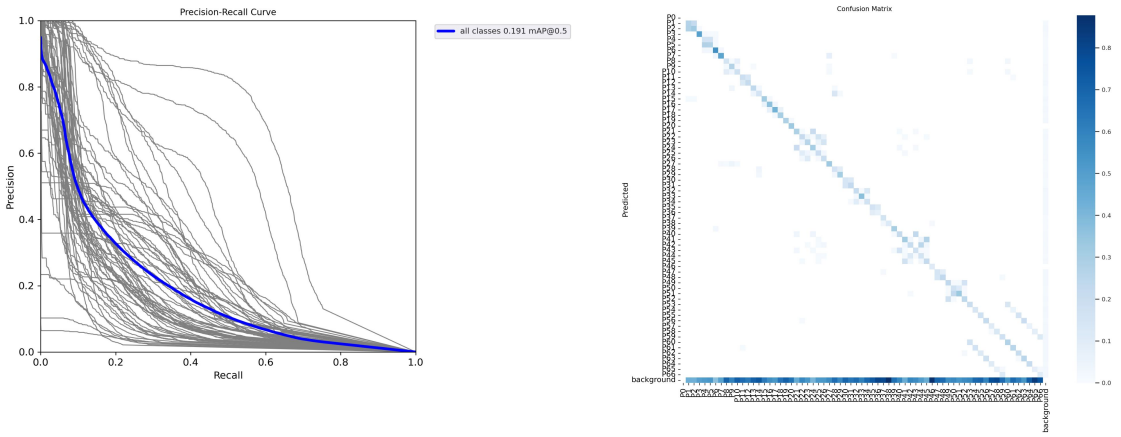

Figura 4.4: À esquerda o gráfico mostra a relação entre precisão e memória da detecção no conjunto de validação rotulado na CNN. À direita localiza-se a matriz de confusão $feature \times feature$. Pode-se notar em alguns pontos da diagonal que determinados pares de features (i,j) possuem classificação parecida com pares (j,i) por ocasião das features com simetria.

Durante a fase final de implementação, organizamos testes de desempenho com redes presentes no *MMDetection* com o intuito de encontrar a CNN com melhor precisão e efetividade diante dos dados característicos desse trabalho. *MMDetection* trata-se de uma ferramenta de detecção de objetos em código aberto em *Pytorch*, da qual fazem parte modelos de rede do estado da arte em detecção de objetos em imagens. Foram avaliadas as redes *ResNeSt* [150], *Deformable Convolutional Networks* (DCN) [28], *Generalized Focal Loss* (GFL) [74] e VFNet [149]. Embora durante a fase inicial de validação do método a rede Yolov5 foi utilizada para os experimentos, paralelamente o mesmo treinamento foi adaptado para outras CNNs. O objetivo de testar o desempenho de outras CNNs não é, de forma alguma, encontrar a melhor configuração ou realizar uma análise detalhada de cada CNN, mas sim encontrar a CNN cuja arquitetura melhor se encaixa no método aqui proposto, considerando a metologia definida.

A comparação entre desempenho de CNNs sob critérios como caixa delimitadora, precisão, memória e nível de confiança requer uma análise mais detalhada, com metodologia própria e não é o propósito desse trabalho. Outrossim, antes de possuir um pipeline concreto e o conjunto de 13 malhas rotuladas, a comparação só poderia ser realizada baseando-se nas métricas das próprias redes ou comparações visuais. Por essa razão, a comparação entre diferentes CNNs foi realizada já nas etapas finais do trabalho. Mesmo assim, comparativamente, a rede Yolov5 mostrou-se mais adequada, no geral, como pode-se verificar na Tabela 4.2 composta pelos valores médios de acertos e erros seguindo as métricas definidas no Capítulo 3. Isso não quer dizer absolutamente

CNN	**Acertos**	**Erros**	**Não Ident.**	**Ω**	**Λ**	**mAP**	**mAP50**
Yolov5	60.75	3.75	1.5	5.83	1.88	0.89	0.58
GFL	50.75	15.25	0	23.10	5.52	0.69	0.90
DCNet	54.25	11.75	0	17.80	4.69	0.49	0.75
ResNeSt	52.75	13.125	0.125	19.95	5.69	0.58	0.85
VfNet	49.12	16.87	0	25.56	6.96	0.73	0.91

Tabela 4.2: Avaliação dos acertos para malhas do conjunto rotulado nas detecções em imagem, para limite de erro $\varepsilon = 0.05$. Os dois últimos valores de Média Geral são Ω (3.6) e Λ (3.8), respectivamente.

que a rede Yolov5 sempre será a mais adequada a problemas de natureza similar aos

apresentados aqui. Os testes foram executados com as configurações padrão da ferramenta *MMDetection*, com dados no padrão *Coco dataset* [77]. É possível que, diante de ajuste dos parâmetros, os resultados se modifiquem. Os treinamentos foram realizados em uma máquina *AMD Ryzen Threadripper 1900X 8-Core*, com placa de vídeo *NVidia GeForce RTX 3080 Ti, 64GB RAM e SO Debian GNU/Linux 11*. O tempo aproximado médio de treinamento de cada rede nestas configurações foi de aproximadamente 22.8 horas.

A comparação permitiu verificar ainda algumas características desejáveis às CNNs candidatas ao método. Algumas redes identificaram, para o conjunto de testes rotulados, todas as features. Porém, a qualidade da precisão não ficou entre as maiores, Figura 4.5. O retângulo delimitador das regiões utilizado nos testes foi de 16x16.

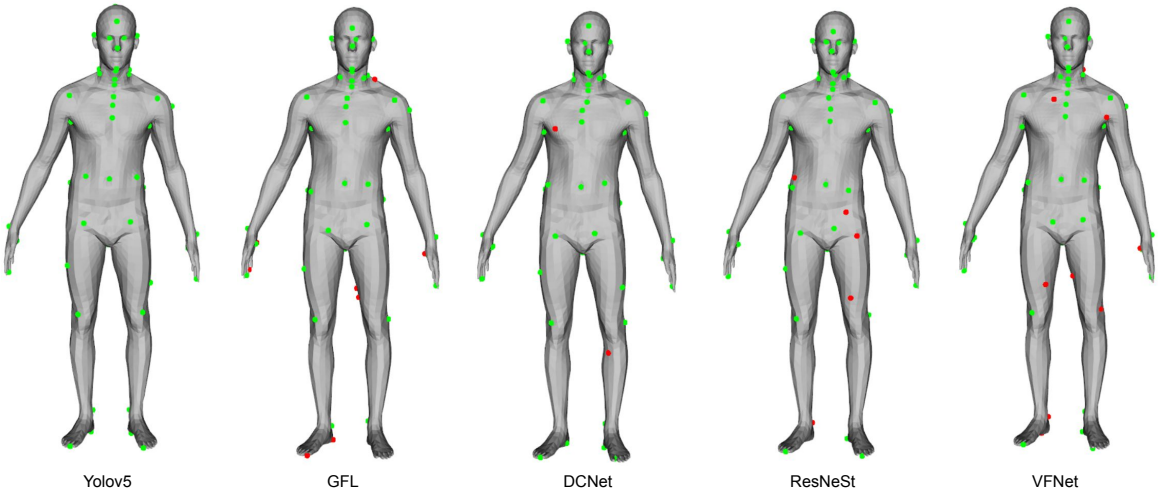

Figura 4.5: Comparação dos resultados obtidos por 5 CNNs distintas sobre um modelo humano do dataset FAUST. Em verde, os pontos considerados corretos pela métrica Λ (3.8), em vermelhos os considerados incorretos, para um $\varepsilon = 0.05$

Como utilizamos o centro do retângulo como ponto candidato, redes que apresentam uma centralização menos precisa do retângulo apresentam pixeis espalhados de forma mais irregular. Devido ao fato de que o método parte do princípio de que pontos para uma determinada feature estarão espalhados ao redor do valor verdadeiro, isso influência nos resultados, principalmente na etapa de clusterização, Figura 4.6.

A Figura 4.7 (a) exibe a captura de alguns resultados provenientes de mapeamentos reversos pixel para vértice, logo antes do processo de clusterização. É possível notar nos exemplos a distribuição Gaussiana das features. Também é possível perceber a ambiguidade provocada pela simetria bilateral sagital das formas, simetria essa identificada na figura pela coloração aplicada a cada feature. Já os resultados obtidos após clusterização, que separa as features simétricas em pares através do algoritmo de mistura Gaussiana, foram capturados e estão exibidos na Figura 4.7 (b). A separação é realizada para as features marcadas como simétricas no gabarito. Teoricamente, é possível tentar realizar a clusterização sem identificar as simetrias, no entanto features muito próximas ou falta de precisão da CNN levaria a erros de identificação.

Por isso, decidiu-se que tal definição faz parte do gabarito. Em algumas features da mesma figura ainda é possível notar que mesmo após a separação, ainda restam *outliers*.

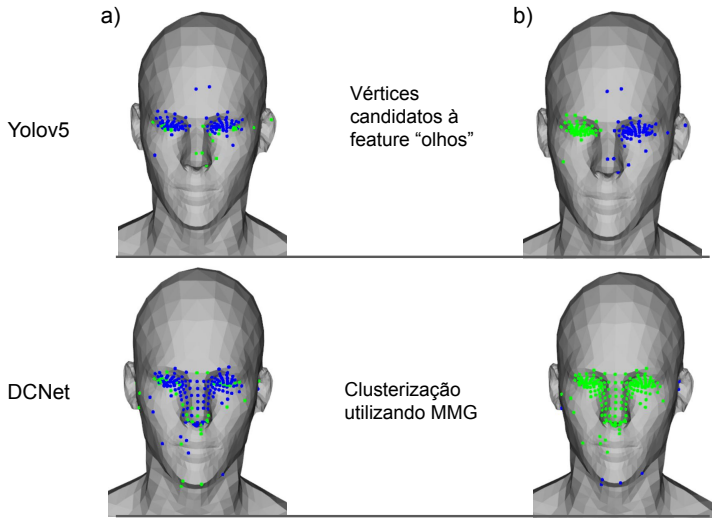

Figura 4.6: Comparação dos vértices candidatos à duas features “olhos”, obtidos a partir dos resultados da Yolov5 (linha superior) e DCNet (linha inferior). Na coluna (b), os resultados da separação em dois conjuntos através do algoritmo de GMM. Nota-se que há muitos pontos dispersos e os conjuntos selecionados não são corretos no caso da DCNet.

Para o algoritmo de mistura Gaussiana, a etapa de inicialização é repetida diversas vezes — o valor mínimo entre 30 o número de pontos candidatos à feature — e o melhor resultado é tomado como correto. Na prática, o valor é superdimensionado, mas a quantidade utilizada apresentou estabilidade nos resultados de todos os testes. A instabilidade provocada pela inicialização não é comum, e ocorre somente em casos em que os pontos candidatos estão mais esparsos do que deveriam, em especial devido a introdução de alguns valores falso positivos. Em casos não atípicos, a instabilidade não acontece.

Após a etapa de pós-processamento, segue-se para a identificação do lado do plano sagital para os vértices simétricos. Para depuração, implementou-se um tipo de visualização que exibe os lados calculados de cada feature simétrica, permitindo identificar imediatamente features em posições incorretas. Exemplos dos resultados alcançados durante o desenvolvimento são mostrados na Figura 4.8. Na figura, exibe-se apenas as features simétricas. Para os vértices identificados com a semântica correta, o lado a qual a feature pertence só estará incorreto quando o ponto de interesse correspondente não foi identificado pela CNN ou a detecção não foi precisa o suficiente para evitar a seleção de um vértice falso positivo.

Para o conjunto de avaliação rotulado, calculamos médias dos resultados qualitativos (Equações (3.7) e (3.8)) e quantitativos (Equações (3.4) e (3.6)) exibindo-os mostrados na Tabela 4.1. Os resultados individuais para cada uma das 13 malhas do conjunto rotulado podem ser analisadas nas Tabelas A.1, A.2, A.3 e A.4, do Apêndice A A. A métrica serve para comparação entre diferentes CNNs e configurações.

4.4 Exemplos

Para efeito de comparação e correlação dos valores discretos com a percepção da correspondência, a Figura 4.9 exibe pontos de correspondência encontrados para 4 malhas de do conjunto de validação (equino, camelídeo, felino e humano) utilizando

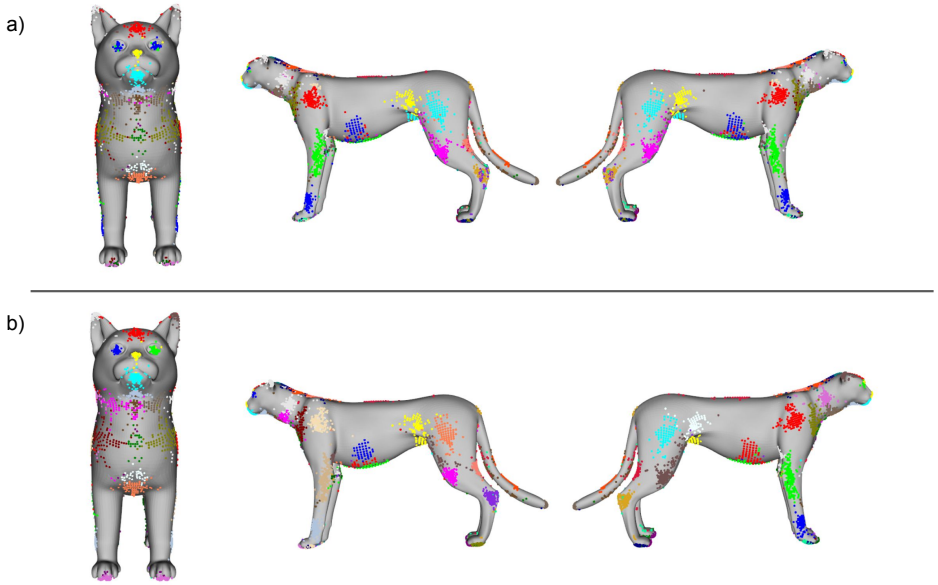

Figura 4.7: Em (a) estão plotados pontos das inferências retornadas da CNN após o mapeamento inverso pixel para vértice, sem alterações. Cores diferentes servem para melhor identificar as nuvens de pontos de cada feature, cores diferentes indicam grupos diferentes. Em (b), os mesmos pontos, mas separados pelo processo de clusterização do método.

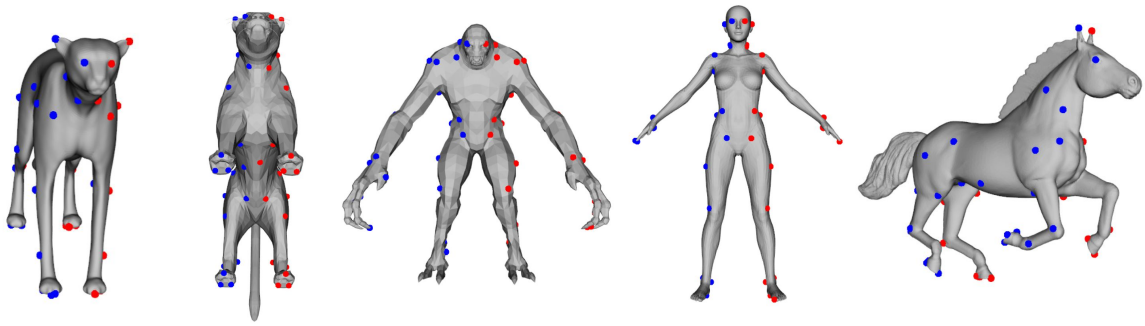

Figura 4.8: Após a identificação de 1 vértice por feature, separa-se as features simétricas em conjuntos esquerda e direita. Em vermelho, features classificadas como esquerda e em azul, direita. Os mapeamentos marcados incorretamente são fruto da classificação incorreta da feature, não da escolha incorreta do lado.

o nível de confiança 0.25; a informação é complementada pela Figura 4.2 que marca graficamente em vermelho os erros encontrados, de acordo com diferentes níveis de confiança.

Como um dos objetivos do trabalho era prover um auxílio para métodos de *morphing*, sem a necessidade de marcar os pontos de correspondência manualmente, coletamos algumas das correspondências obtidas pelo método e as fornecemos como entrada para o método implementado por Medalha et al. [90]. Outros dos resultados obtidos são exibidos na Figura 4.10. Mais resultados do processo de *morphing* sobre outras malhas podem ser verificados no Apêndice B, nas Figuras B.1 B.2 eB.3.

Por fim, para demonstrar visualmente a capacidade de generalização do método, executamos a etapa de correspondência também sobre um pequeno grupo de malhas disponíveis, algumas provenientes de conjuntos dados conhecidos como FAUST [15], SCAPE [5], PSB [25] e outras malhas de animais de uso livre arbitrariamente coletados do site Free3D [127]. Nessas malhas, a avaliação dos resultados foi visual, já que não

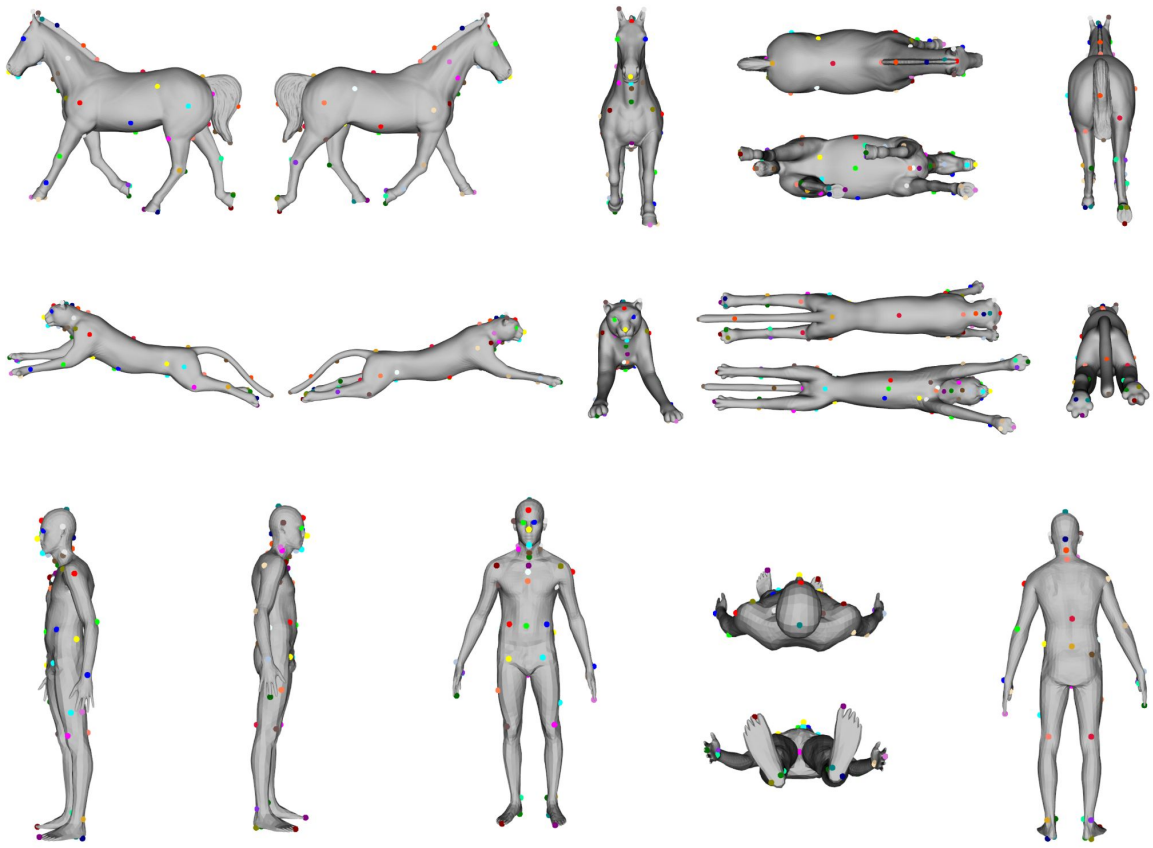

Figura 4.9: Parte dos resultados obtidos para o conjunto de avaliação, mostrando 3 formas diferentes. O restante dos resultados podem ser verificadas nas figuras do Apêndice B.

existe gabarito para as malhas dessa etapa. Uma amostra dos resultados da última etapa após detecção dos vértices *outliers* e seleção do vértice definitivo estão ilustradas nas Figuras B.4, B.5 e B.6, inclusas no Apêndice B.

4.5 Considerações Finais

Neste capítulo, mostramos alguns dos experimentos e resultados obtidos durante o desenvolvimento do método. Comparamos os resultados entre algumas CNNs, tanto da detecção de regiões como também dos resultados da correspondência obtidos para um conjunto de validação, montado com malhas rotuladas. Porém, deixamos claro que não faz parte do escopo desse trabalho a análise detalhada das CNNs comparadas. Diante dos resultados mostrados, nota-se que, naturalmente quanto maior a precisão e memória da CNN, melhor para o método. Contudo, percebe-se que deve ser levado em conta também a distribuição dos valores falso positivos, influência de acertos considerados corretos por causa do limite da IoU e taxa de confiança.

Por sua vez, a rede utilizada (Yolov5), seguindo as métricas (3.6) e (3.8), permitiu ao método apresentar resultados finais com taxas de acerto quantitativo e qualitativo de 94.3% e 98.16%, respectivamente, tomando o conjunto de validação rotulado com 13 malhas como entrada.

É interessante notar que, a mAP das features na CNN, considerandos-se taxa de

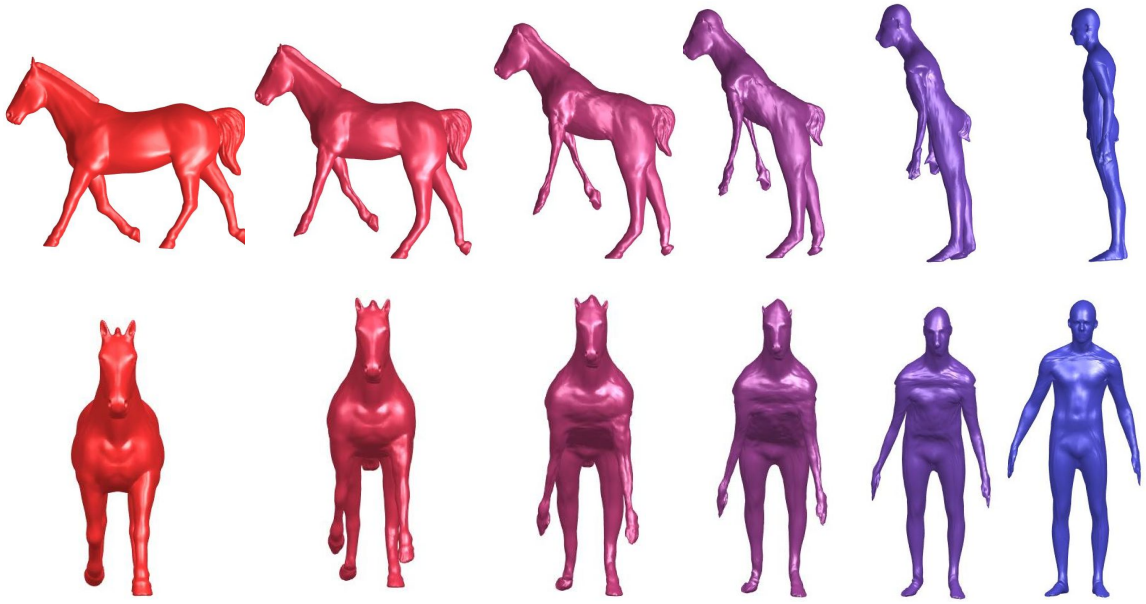

Figura 4.10: Exemplos do resultado do *morphing* proposto por Medalha et al [90]. Os resultados das malhas obtidas em comparação com as malhas destino demonstram que os pontos semânticos automaticamente encontrados pela nossa etapa de correspondência são de fato adequados. No exemplo, a transformação da malha de um cavalo retirada de [127] e de um modelo humano do dataset FAUST [15].

confiança de 0.25 para imagens do mesmo dataset de validação, foi de menos de 0.20. Com as imagens de correspondências exibidas no Apêndice B B, mais precisamente as Figuras B.4, B.5 e B.6, reforçamos que as correspondências são consistentes em relação à semântica. Como demonstração final, exibimos o resultado de um processo de *morphing* sobre algumas das correspondências encontradas, demonstrando uma das utilidades do método.

Capítulo 5

Conclusão

5.1 Considerações Finais

Nesta tese, apresentamos um método baseado em aprendizagem multivisão profunda, que possibilita a detecção de pontos de interesse semânticos em malhas de triângulos variadas dentro de uma determinada classe de formas semelhantes.

Durante a busca inicial por soluções, a prototipação de abordagens axiomáticas ou baseadas em representações espectrais não obtiveram os resultados esperados. A principal dificuldade decorreu da disponibilidade limitada de dados rotulados, em comparação com abordagens bidimensionais. Porém, através da adoção de um método que emprega abordagem multivisão, alcançou-se os objetivos. Por tratar-se de uma técnica cuja etapa de treinamento dá-se sobre imagens 2D, evitou-se um problema comum encontrado ao processarmos malhas de triângulos: a indisponibilidade de um conjuntos de dados grande o suficiente para treinamento.

A correspondência dá-se em relação a um gabarito semântico definido conforme o propósito desejado. Consequentemente, o método possibilita obter automaticamente, sem intervenção do usuário, correspondências entre malhas de datasets de origens variadas, contanto que pertençam à classe de formas suportada pelo treinamento.

O objetivo principal, i.e., propor um método para a detecção de pontos correspondência entre superfícies representadas por malhas de triângulos, sem intervenção direta do usuário, foi plenamente alcançado. Durante as validações efetuadas no Capítulo 4, utilizando a métrica de medição de erros definida na Subseção 3.2.3, os resultados finais apresentaram taxas de acerto quantitativo e qualitativo de 94.3% e 98.16%, respectivamente, para 66 features. No caso de malhas representando uma sequência de animação ou para um conjunto de malhas de mesma topologia em poses diversas, a taxa de acerto pode ser ainda maior, pois o resultado final pode ser a compilação de todas as malhas com mesma topologia.

Como exemplificado na Subseção 3.2.3, o método requer apenas que a CNN utilizada acerte a predição da feature desejada em poucas imagens, uma única predição dentro do conjunto de imagens geradas para uma malha é suficiente, desde que a predição seja precisa. A limitação encontrada pelo método é que ele só pode detectar pontos de interesse que sejam visíveis de algum ponto de vista.

Após o mapeamento inverso pixeis para vértices, trabalhar sobre uma malha de

triângulos permitiu solucionar alguns problemas gerados pela etapa de detecção da CNN. O primeiro deles é a geração de features simétricas devido a simetria bilateral sagital, não classificadas corretamente em esquerda/direita pela pela CNN. A solução do problema de simetria bilateral pode ser aplicada em outros trabalhos que precisem realizar a mesma tarefa, e tratava-se de um objetivo específico do trabalho. O segundo problema trata-se da eliminação de *outliers*, valores atípicos que podem distorcer o resultado final.

Encontrar correspondências semânticas automaticamente pode auxiliar na solução de uma série de problemas geométricos. Em ferramentas CAD ou motores 3D utilizados em jogos digitais, pode-se aplicar o método para automatizar, ao menos parcialmente, transferência de texturas, estimativa de poses, transferência de pontos de fixação de um modelo de ator para outro qualquer, transferência de nós de animação entre formas 3D, dentre outras tarefas de correlação entre atores. Outra possibilidade é, através de pontos de correspondência específicos, determinar um conjunto de coordenadas para automaticamente orientar e alinhar malhas em relação à algum ponto de referência. Se adicionadas relações de conectividade entre os pontos de interesse do gabarito, o método pode servir como base para classificação automática de formas em classes predeterminadas através da análise das proporções entre as arestas dos pontos de correspondência. De fato, o método permite que gabaritos — pontos semânticos desejados — sejam definidos conforme a tarefa requerida, um dos objetivo específicos do trabalho.

Além da contribuição principal, destaca-se também a implementação de uma ferramenta gráfica para visualização e edição dos pontos de correspondência, que permite renderizar imagens de malhas para treinamento de redes neurais em aplicações do tipo multivisão.

Como parte da demonstração dos resultados obtidos, informamos os pontos de correspondência obtidos pelo método como correspondência inicial entre pares de malhas, tomadas como entrada pelo método de *morphing* proposto por Medalha et al.[90]. O processo retornou os resultados esperados, criando com sucesso *morphing* entre as formas. Portanto, o objetivo específico de facilitar a definição de pontos de correspondência iniciais entre pares de formas em técnicas de *morphing* também foi alcançado. Considerando todas as obervações realizadas, conclui-se que todos os objetivos, principal e específicos, foram plenamente alcançados. Finalmente, diante dos resultados, verifica-se que a hipótese definida no Capítulo 1 — *aprendizagem de máquina pode ser empregada eficazmente em uma ferramenta computacional para detecção de pontos característicos e correspondência de pontos, visto serem essas operações inerentemente semânticas. Correspondência de pontos pode ser obtida a partir de conjuntos de pontos característicos, convenientemente determinados para classes de formas semelhantes, os quais podem ser determinados pelas análise de imagens dessas formas por redes neurais profundas.*— foi confirmada.

5.2 Trabalhos Futuros

Futuramente, é válida uma análise aprofundada em relação a comparação entre CNNs, variando configurações e parâmetros de treinamento. Além disso, mesmo que nos experimentos realizados, 162 pontos de vista tenham se mostrado suficientes para

abranger todas as características necessárias para obtenção dos resultados apresentados, uma análise da relação entre precisão e diversas configurações e número de imagens geradas por malha seria importante.

O método apresentado, limitou-se à classe de formas suportada — equinos, camelídeos, humanos, terópodes e felinos. Trabalhos futuros podem estendê-lo para novas classes de animais, aumentando a variedade e melhorando a generalização do método. Outra extensão possível trata-se da questão relacionada à simetria, é possível estender o método para tratar outros eixos de simetria além da bilateral.

Apêndice A

Tabelas Complementares

Conf.	**Malha**	**Acertos**	**Erros**	**N.E.**	$\omega_{\mathbf{u}}$	$\lambda_{\mathbf{u}}$
0.3	11504_Cheetah_V3L	62	4	0	6.06	2.50
0.3	12161_Cat_v1_L2	59	7	0	10.60	3.25
0.3	12222_Cat_v1_l3	63	3	0	4.54	1.33
0.3	13775_Cheetah_new	65	1	0	1.51	0.77
0.3	16283_SBH_Trotting	65	1	0	1.51	0.43
0.3	Camel9	65	1	0	1.51	0.27
0.3	dilophosaurus	54	6	6	10	1.74
0.3	giraffe_model390B	NC	NC	5	NC	NC
0.3	lion_00211799_ferrari	60	6	0	9.09	1.98
0.3	tr_reg_000	62	1	3	1.58	1.15
0.3	tr_reg_016	57	7	2	10.93	3.97
0.3	tr_reg_081	61	2	3	3.17	2.58
0.3	tr_reg_090	63	1	2	1.56	1.36
Média Geral		**Ac.**	**Err.**	**N.E.**	**Ω**	**Λ**
		56.61	3.07	1.23	4.77	1.64

Tabela A.1: Avaliação dos acertos para malhas do conjunto rotulado para nível de confiança de 0.3, $\varepsilon = 0.05$. Os dois últimos valores de Média Geral são Ω (3.6) e Λ (3.8), respectivamente

Conf.	**Malha**	**Acertos**	**Erros**	**N.E.**	$\omega_{\mathbf{u}}$	$\lambda_{\mathbf{u}}$
0.25	11504_Cheetah_V3	61	5	0	7.57	2.64
0.25	12161_Cat_v1_L2	59	7	0	10.60	3.23
0.25	12222_Cat_v1_l3	63	3	0	4.54	1.33
0.25	13775_Cheetah_new	65	1	0	1.51	0.78
0.25	16283_SBH_Trotting	65	1	0	1.51	0.47
0.25	Camel9	65	1	0	1.51	0.26
0.25	dilophosaurus	56	8	2	12.5	2.10
0.25	giraffe_model390B	58	6	2	9.37	2.37
0.25	lion_00211799_ferrari	60	6	0	9.09	1.98
0.25	tr_reg_000	63	1	2	1.56	1.20
0.25	tr_reg_016	60	5	1	7.69	3.66
0.25	tr_reg_081	61	2	3	3.17	1.85
0.25	tr_reg_090	62	2	2	3.12	2.08
Média Geral		**Ac.**	**Err.**	**N.E.**	**Ω**	**Λ**
		61.38	3.69	0.92	5.67	1.84

Tabela A.2: Avaliação dos acertos para malhas do conjunto rotulado para nível de confiança de 0.25, $\varepsilon = 0.05$. Os dois últimos valores de Média Geral são Ω (3.6) e Λ (3.8), respectivamente

Conf.	**Malha**	**Acertos**	**Erros**	**N.E.**	$\omega_{\mathbf{u}}$	$\lambda_{\mathbf{u}}$
0.20	11504_Cheetah_V3	60	6	0	9.09	3.41
0.20	12161_Cat_v1_L2	60	6	0	9.09	3.18
0.20	12222_Cat_v1_l3	63	3	0	4.54	1.34
0.20	13775_Cheetah_new	65	1	0	1.51	0.74
0.20	16283_SBH_Trotting	62	4	0	6.06	1.67
0.20	Camel9	65	1	0	1.51	0.26
0.20	dilophosaurus	58	7	1	10.76	1.92
0.20	giraffe_model390B	55	9	2	14.0625	2.61
0.20	lion_00211799_ferrari	60	6	0	9.09	1.95
0.20	tr_reg_000	63	1	2	1.56	1.26
0.20	tr_reg_016	61	5	0	7.57	3.60
0.20	tr_reg_081	61	3	2	4.68	2.02
0.20	tr_reg_090	62	2	2	3.12	2.09
Média Geral		**Ac.**	**Err.**	**N.E.**	**Ω**	**Λ**
		61.15	4.15	0.69	6.36	2.00

Tabela A.3: Avaliação dos acertos para malhas do conjunto rotulado para nível de confiança de 0.20, $\varepsilon = 0.05$. Os dois últimos valores de Média Geral são Ω (3.6) e Λ (3.8), respectivamente

Conf.	**Malha**	**Acertos**	**Erros**	**N.E.**	$\omega_{\mathbf{u}}$	$\lambda_{\mathbf{u}}$
0.15	11504_Cheetah_V3	59	7	0	10.60	3.61
0.15	12161_Cat_v1_L2	59	7	0	10.60	3.18
0.15	12222_Cat_v1_l3	62	4	0	6.06	1.86
0.15	13775_Cheetah_new	64	2	0	3.03	1.01
0.15	16283_SBH_Trotting	62	4	0	6.06	1.69
0.15	Camel9	65	1	0	1.51	0.26
0.15	dilophosaurus	58	7	1	10.76	1.82
0.15	giraffe_model390B	56	10	0	15.15	2.75
0.15	lion_00211799_ferrari	60	6	0	9.09	1.94
0.15	tr_reg_000	63	1	2	1.56	1.19
0.15	tr_reg_016	58	8	0	12.12	4.61
0.15	tr_reg_081	61	3	2	4.68	2.02
0.15	tr_reg_090	61	4	1	6.15	2.27
Média Geral		**Ac.**	**Err.**	**N.E.**	**Ω**	**Λ**
		60.61	4.92	0.46	7.49	2.17

Tabela A.4: Avaliação dos acertos para malhas do conjunto rotulado para nível de confiança de 0.15, $\varepsilon = 0.05$. Os dois últimos valores de Média Geral são Ω (3.6) e Λ (3.8), respectivamente

Classe	P	R	mAP50	Classe	P	R	mAP50
P1	0.351	0.355	0.258	**P34**	0.173	0.179	0.132
P2	0.314	0.388	0.252	**P35**	0.157	0.163	0.0644
P3	0.434	0.46	0.395	**P36**	0.0607	0.0677	0.0243
P4	0.377	0.236	0.228	**P37**	0.191	0.154	0.105
P5	0.398	0.334	0.234	**P38**	0.186	0.112	0.102
P6	0.7	0.594	0.606	**P39**	0.385	0.292	0.232
P7	0.578	0.551	0.497	**P40**	0.198	0.246	0.136
P8	0.28	0.167	0.163	**P41**	0.308	0.35	0.243
P9	0.439	0.286	0.285	**P42**	0.219	0.218	0.148
P10	0.2	0.185	0.157	**P43**	0.216	0.195	0.144
P11	0.22	0.187	0.124	**P44**	0.266	0.187	0.166
P12	0.21	0.214	0.135	**P45**	0.302	0.261	0.219
P13	0.36	0.245	0.191	**P46**	0.467	0.114	0.134
P14	0.322	0.199	0.171	**P47**	0.166	0.158	0.0789
P15	0.291	0.311	0.205	**P48**	0.203	0.218	0.126
P16	0.386	0.359	0.339	**P49**	0.222	0.245	0.141
P17	0.393	0.433	0.38	**P50**	0.238	0.192	0.148
P18	0.389	0.297	0.267	**P51**	0.277	0.333	0.223
P19	0.27	0.26	0.208	**P52**	0.216	0.236	0.137
P20	0.567	0.313	0.346	**P53**	0.265	0.227	0.191
P21	0.278	0.311	0.202	**P54**	0.158	0.142	0.103
P22	0.223	0.251	0.155	**P55**	0.183	0.19	0.108
P23	0.275	0.29	0.215	**P56**	0.258	0.238	0.17
P24	0.264	0.348	0.223	**P57**	0.14	0.109	0.0981
P25	0.245	0.219	0.159	**P58**	0.152	0.131	0.103
P26	0.288	0.215	0.169	**P59**	0.168	0.182	0.107
P27	0.438	0.342	0.334	**P60**	0.249	0.285	0.212
P28	0.299	0.174	0.214	**P61**	0.236	0.249	0.171
P29	0.367	0.326	0.246	**P62**	0.218	0.222	0.151
P30	0.249	0.151	0.134	**P63**	0.25	0.273	0.183
P31	0.252	0.22	0.134	**P64**	0.234	0.194	0.156
P32	0.183	0.215	0.155	**P65**	0.0965	0.0781	0.0275
P33	0.315	0.351	0.305	**P66**	0.135	0.112	0.0434
Geral					**0.278**	**0.248**	**0.191**

Tabela A.5: Avaliação dos acertos da CNN Yolov5 para o conjunto rotulado com 13 formas e 66 classes de *features*. P = precisão, R= memória e mAP50 é média de acertos da caixa retangular com precisão de pelo menos 50%

Apêndice B

Imagens Complementares

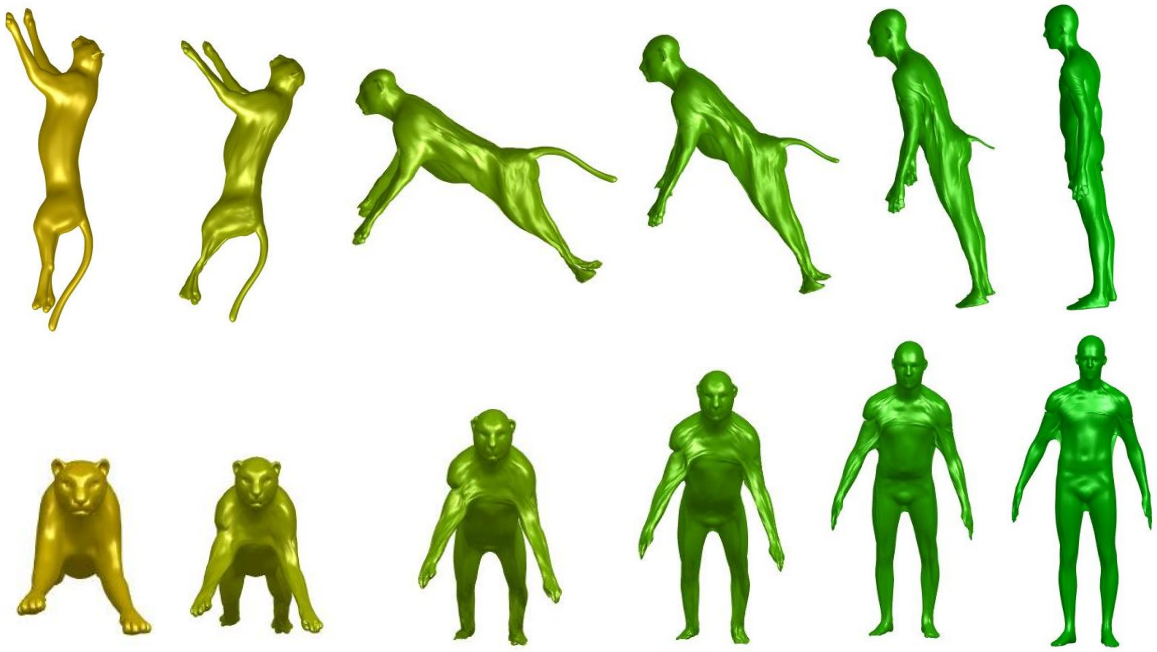

Figura B.1: Exemplos do resultado do *morphing* proposto por Medalha et al [90]. No exemplo, a transformação da malha de um felino retirada de [127] e de um modelo humano do dataset FAUST [15].

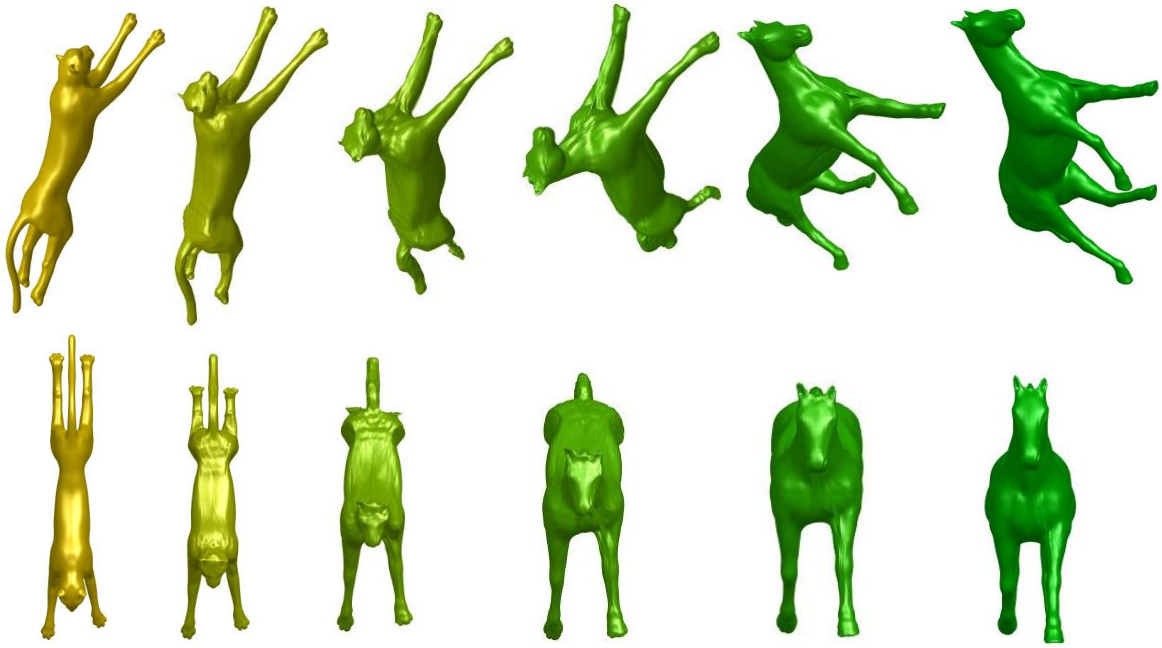

Figura B.2: Exemplos do resultado do *morphing* proposto por Medalha et al [90]. No exemplo, a transformação da malha de um cavalo retirada de [127] e de um felino de mesma origem.

Figura B.3: Exemplos do resultado do *morphing* proposto por Medalha et al [90]. No exemplo, a transformação da malha de um felino em outro de espécie diferente, ambas malhas retiradas de [127].

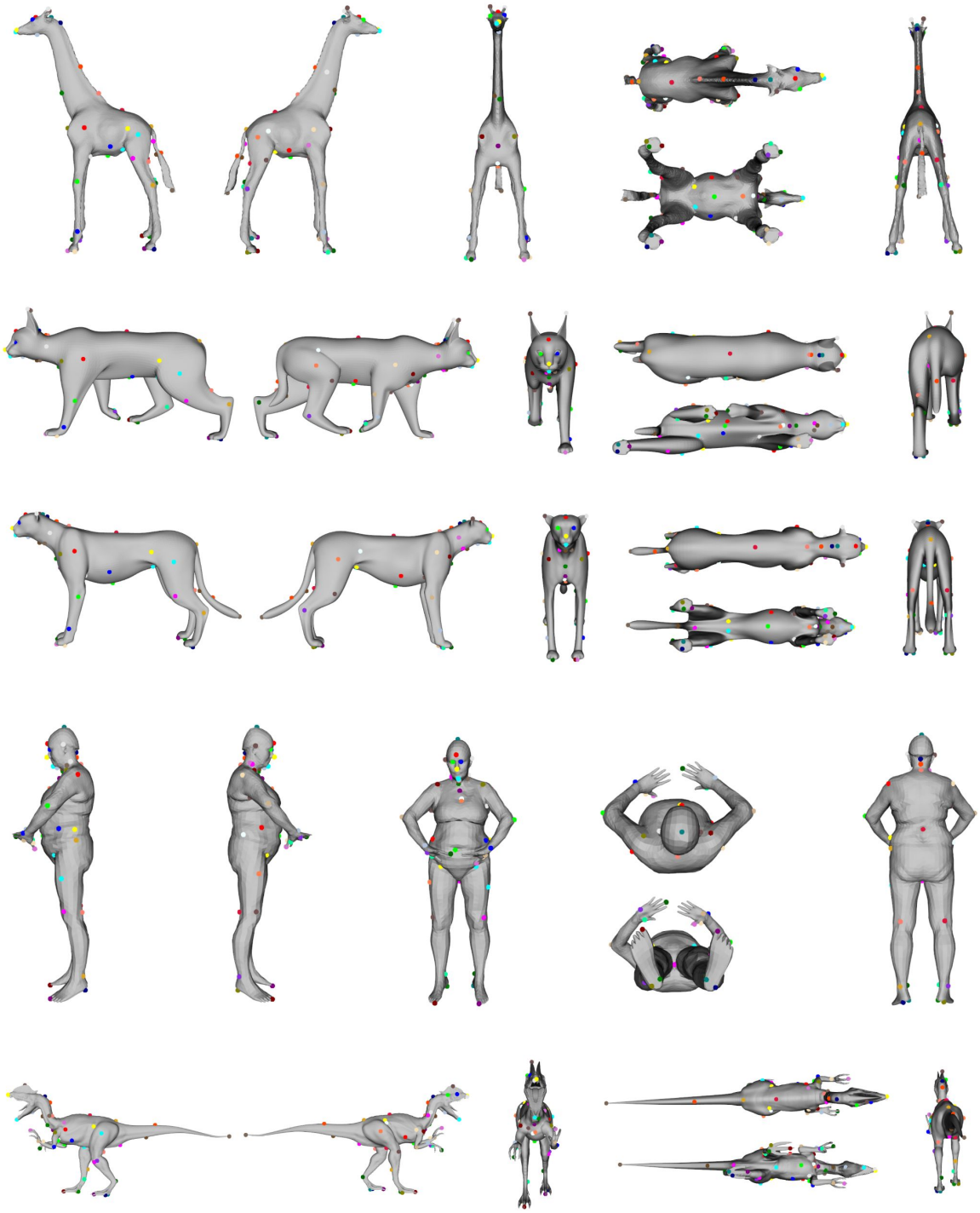

Figura B.4: Resultado dos experimentos de correspondência, com taxa de confiança 0.25, na CNN Yolov5, sobre algumas malhas do conjunto de validação. Percebe-se boa generalização na classe de felinos. Até mesmo uma girafa, forma não vista no treinamento, possui a maioria dos pontos de correspondência corretos. O terópode (última linha) tem cabeça totalmente diferente de todos os modelos treinados, por isso possui poucas correspondências corretas na região.

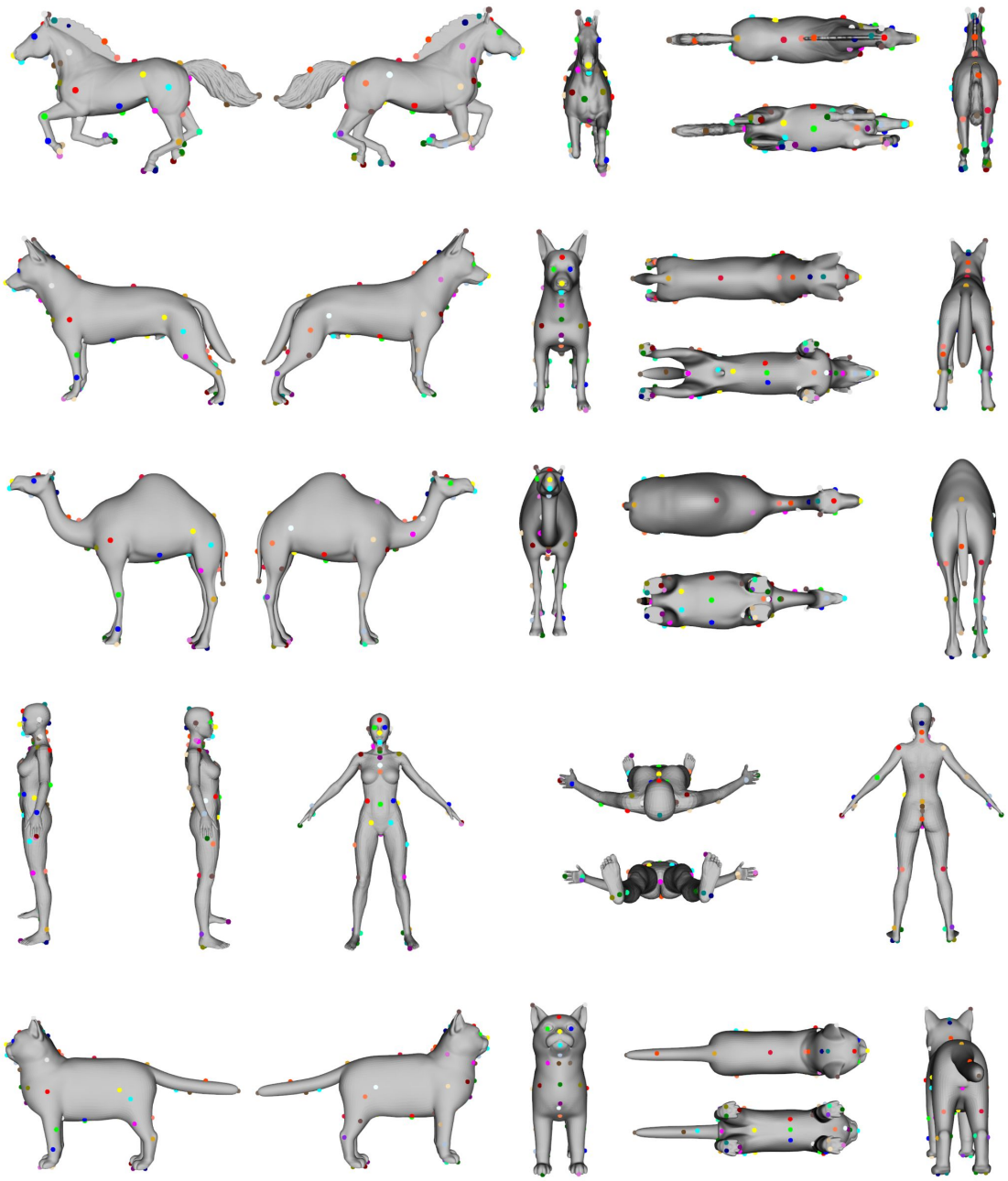

Figura B.5: Resultado dos experimentos de correspondência, com taxa de confiança 0.25, na CNN Yolov5. Comparando-se as correspndências visualmente com o gabarito, nota-se uma correspondência com taxa de acertos quase total. Em média, na avaliação visual, percebe-se 1 a 3 pontos incorretos ou não encontrados (de 66) por malha.

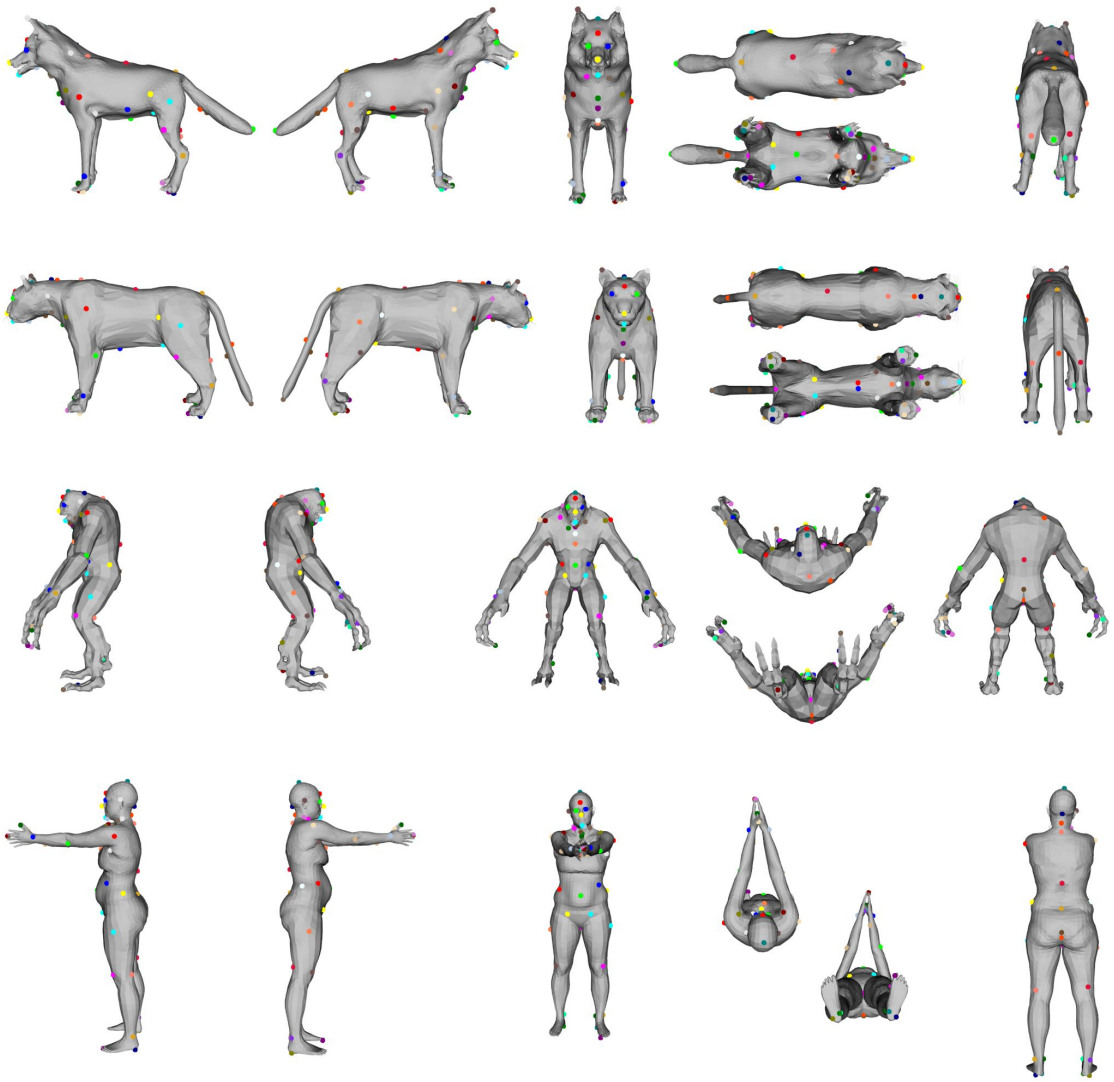

Figura B.6: Resultado dos experimentos de correspondência, com taxa de confiança 0.25, na CNN Yolov5. Na terceira linha, perceba que no monstro, mesmo com a classe de forma não fazendo parte do treinamento e possuindo características diferentes dos humanos, o método ainda encontra algumas correspondências, demonstrando boa capacidade de generalização. Na última linha, um modelo do dataset FAUST. Mesmo com algumas features não encontradas, ao compilarmos os resultados da malha com os resultados de outras malhas de mesma conectividade dentro do mesmo dataset, obtém-se todas as feature para os modelos.

Apêndice C

Revisão Literária e Experimentos Suplementares

C.1 Considerações Iniciais

Nesse capítulo são relatados os principais trabalhos que utilizaram aprendizado profunda na solução de problemas de processamento geométrico semelhantes ao tratado nessa tese, mas que não forma utilizados na solução final. Destacando-se métodos para classificação, segmentação e transferência de deformação entre formas. Tal revisão serve como norte inicial para trabalhos que eventualmente pretendam seguir a linha de pesquisa aqui relatada, auxiliando a identificar algumas dificuldades e possibilidades já encontradas.

Na Seção C.2, são retratados os principais avanços na solução de problemas de segmentação de formas 3D e de correspondência entre elas, apresentando resultados recentes e os principais desafios. Na Seção C.3, são revisados os trabalhos recentes relacionados à problemas de transferência de deformação ou transferência de pose entre pares de formas tridimensionais não rígidas. Ainda, é considerada uma abordagem geral do problema, não se limitando apenas a soluções obtidas via redes neurais profundas.

As primeiras tentativas de desenvolvimento de uma solução para o problema proposto no Capítulo 1, consistiram em estudar métodos que trabalhassem diretamente com pares de formas para que pudéssemos empregar diretamente a correspondência entre 2 formas. Tais métodos, mais próximos dos trabalhos de processamento geométrico que utilizam diretamente a malha, grafos ou relações de adjacência como estrutura subjacente utilizada para alimentar redes neurais. Porém, para correspondência entre pares de formas cujas deformações não são aproximadamente isométricas, tais métodos mostraram-se inadequados para o problema de correspondência aqui tratado, como relatado na Seção C.4 .

A Seção C.4 relata o processo percorrido na busca por soluções através da investigação das principais abordagens espectrais e espaciais relacionadas à problemas de processamento geométricos. Descrevemos a evolução das abordagens avaliadas desde a utilização de propriedades geométricas e espaciais até abordagens que utilizam discretização do operador de Laplace-Beltrami na tentativa de solucionar parte do problema. Apesar das abordagens relatadas nesta seção não serem parte direta da solução encontrada, isso não significa que não sejam alternativas válidas, mas sim que existem

dificuldades e limitações. Para trabalhos que porventura desejem utilizar as mesmas estratégias como ponto de partida, os relatos servem de norte e ponto de partida. Tais trabalhos podem aproveitar-se da experiência e sabendo de alguns problemas enfrentados desde o princípio, procurar alternativas para contorná-los.

C.2 Segmentação e Correspondência por Aprendizagem Profunda

Para efetivamente entender uma malha de triângulos que representa uma forma 3D, um dos pontos chave é identificar a qual parte provável do objeto cada triângulo pertence. Este processo, chamado de rotulação de malha, tem por objetivo deduzir as características inerentes à malha e pode ser empregado em várias aplicações tais como edição, modelagem e deformação de malhas [46], resultando também em avanços em tarefas de segmentação e rotulação das formas tridimensionais, área que tornou-se um objeto de pesquisa desafiador. A tarefa de segmentação, apesar de fundamental, é desafiadora por diversos motivos, dentre os principais pode-se citar: a variedade e ambiguidade das partes das formas que devem possuir o mesmo rótulo semântico; o fato de que detectar fronteiras entre as partes com precisão pode envolver indicações extremamente sutis; features globais e locais devem ser analisadas conjuntamente; e a análise deve ser robusta a ruídos e subamostragem (*undersampling*) [55]. Xie et al. [136] trabalharam para melhorar a performance de treinamento de abordagens do tipo supervisionadas, mantendo a precisão para grandes conjuntos e oferecendo cenários de aprendizagem online. O autor utilizou o conceito de *Extreme Learning Machine* (ELM) [52] para treinar uma rede neural como um classificador. O classificador treinado da ELM é utilizado para deduzir rótulos para todas as faces da forma (malha). Baseado nisso, é realizada uma otimização da segmentação.

Um exemplo didático de um pipeline de segmentação de formas 3D, neste caso com aprendizado supervisionado, é ilustrado na Figura C.1. As formas de entrada são dotadas de informações das partes rotuladas das formas. Um descritor geométrico é extraído para cada ponto nos exemplos de treinamento, e os pontos são mapeados para um espaço de features em comum. A etapa de aprendizagem utiliza um algoritmo de classificação que separa o espaço da entrada não-linearmente em um conjunto de regiões correspondentes a rótulos das partes das formas. Dada uma forma teste (ainda não conhecida pela rede), um modelo probabilístico é utilizado para deduzir rótulos das partes para cada ponto na forma baseando-se em seu descritor geométrico no espaço de features.

Seguindo uma abordagem próxima a de Litman et al. [80], Rodolá et al. [103] propuseram um método para correspondência de formas ajustado à alguma classe específica de formas e deformações (Figura C.2). Enquanto os primeiros baseiam-se em uma abordagem da área de processamento de sinais para formular uma família genérica de descritores de formas deformáveis, os segundos tratam o problema da correspondência de formas como um problema de classificação em que as amostras de entrada são pontos na superfície da forma e a classe de saída é um elemento de um conjunto de rótulos canônicos, que poderá coincidir com a superfície de uma das formas 3D do conjunto de treinamento. Em um cenário onde uma determinada classe é representada por um pequeno conjunto de formas exemplo, o método proposto aprende um

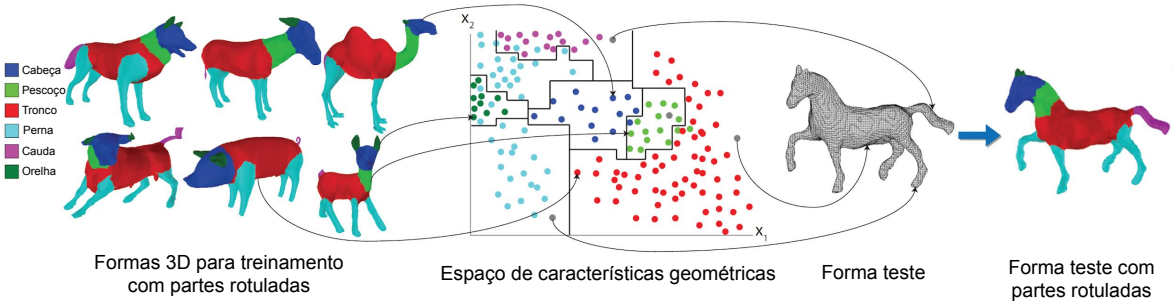

Figura C.1: *Pipeline* de um algoritmo de segmentação supervisionado (retirado de [139]). Dado um conjunto de formas com partes rotuladas, os pontos de cada forma são mapeados para um espaço de features comum baseado em seus descritores geométricos locais. Um classificador é aprendido para separar o espaço de features em regiões correspondentes a cada rótulo das partes. Dada uma forma de teste, seus pontos são mapeados para o mesmo espaço geométrico. Os rótulos então são deduzidos para todos os pontos baseados no classificador aprendido e no modelo da estrutura probabilística subjacente.

descritor de forma capturando a variabilidade das deformações dessa classe. Uma das contribuições do artigo consistiu em um novo classificador de floresta aleatório (*random forest classifier*), que podia atacar o problema de classificação de forma eficiente, partindo de um descritor de forma parametrizável e genérico. O classificador proposto explorava aleatoriamente e espaço de parametrização dos descritores e encontrava as features mais discriminativas que devidamente recuperam o mapa de transformações, caracterizando a categoria da forma em questão. Vale ressaltar que o descritor de forma utilizado foi o kernel de assinatura de onda (WKS), que embora conhecido por ser invariante à transformações isométricas, pode ser explorado pelo classificador de floresta para casar formas que passaram por deformações maiores que isométricas. Em um sentido mais amplo, a saída do classificador de floresta aleatório pode ser visto como um novo descritor por si só, moldado pelas formas e deformações presentes no conjunto de treinamento. Neste sentido, o método proposto pode inclusive ser utilizado como um complemento a outros descritores de forma existentes.

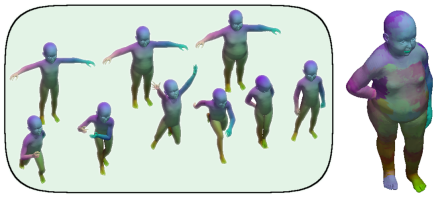

Figura C.2: Exemplo de correspondência densa entre formas utilizando florestas aleatórias (*random forests*) sob deformações não isométricas. Formas na área sombreada são um subconjunto do treinamento. A floresta é treinada com kérneis de descritores de onda e composta de 80K classes de treinamento com 19 amostras por classe. Na figura, as correspondências estão indicadas pela cor (retirado de [103]).

Enquanto Litman et al. [80] e Rodolá et al. [103] utilizaram um conjunto de exemplos para aprender os descritores mais informativos aplicados diretamente no contexto da correspondência de formas, não muito tempo depois Corman et al. [27] apresentaram um novo método para calcular correspondências entre pares de formas não rígidas. Seu método mostrou uma abordagem fundamentalmente distinta, em que ao invés de tentar identificar descritores que consigam diferenciar pontos distintos, tenta encontrar um grupo de descritores ótimos que possam ser utilizados em conjunto para produzir

todo o mapa sobre as formas 3D. Deste modo, a consistência é incorporada diretamente no estágio de aprendizagem, evitando problemas ao tentar obter correspondências consistentes durante o pós-processamento. Para tal, o autor utilizou a representação de mapas funcionais [94], que permitiu formular o problema de aprendizagem de maneira puramente intrínseca, sem depender de parametrização consistente ou posição espacial dos vértices das formas, enquanto permitia controlar diretamente a influência e performance dos descritores na qualidade final do mapa funcional. As formas são representadas por malhas de triângulos e todas funções são expressas como vetores na base dos autovetores do operador de Laplace-Beltrami. O operador é calculado antecipadamente em cada forma e o objetivo é gerar um mapa funcional definido exclusivamente. A Figura C.3 mostra um exemplo das funções mais estáveis aprendidas dos mapas de treinamento.

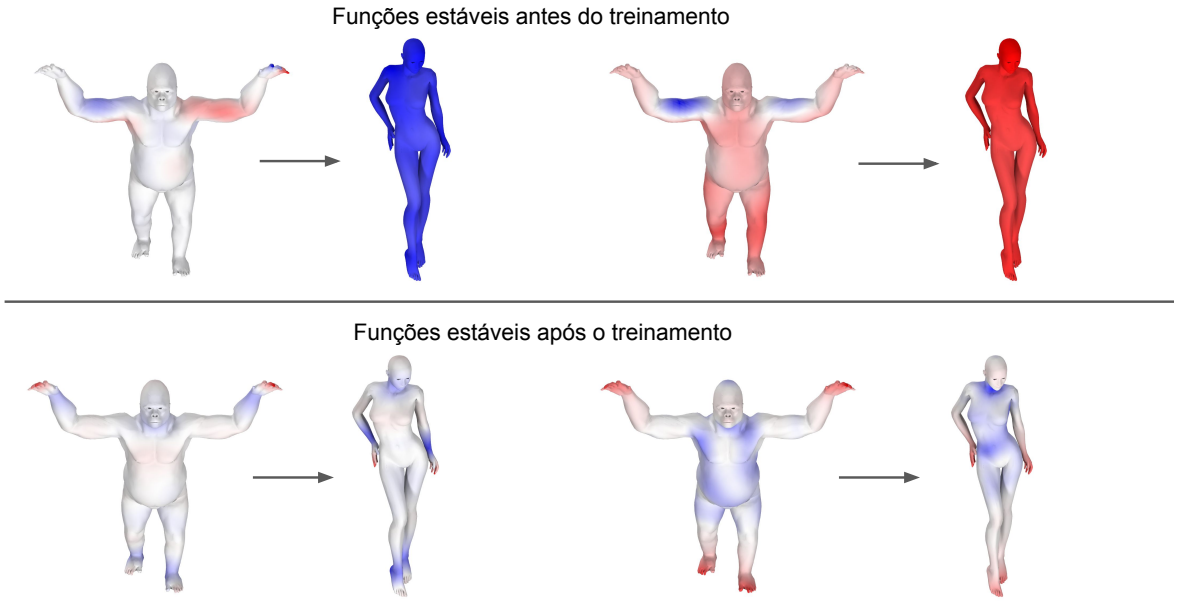

Figura C.3: Cada linha mostra a visualização dos dois primeiros componentes da base extraída da forma de referência (gorila) e então mapeada para uma forma ainda não conhecida (mulher). Na primeira linha, cada uma das funções foi mapeada para a forma não conhecida utilizando seu mapa das verdades absolutas convertido para um mapa funcional. Note que as funções são transferidas de maneira indesejada, devido a incompatibilidade das bases de Laplace-Beltrami e o ruído nos mapas de entrada. Na segunda linha, as funções de sonda foram ponderadas utilizando o método em [27]. As funções estáveis indicam a cabeça, as mãos e os pés como as áreas mais estáveis (retirado de [27]).

Masci et al. [87, 88] propuseram uma extensão do paradigma CNN para domínios não Euclidianos, a qual chamou de *Shapenet*. A base do trabalho é construída sob um sistema local de coordenadas polares geodésicas para extração de "retalhos" (*patches*) das formas, estes retalhos passam então por uma cascata de filtros e operadores lineares e não-lineares. Os coeficientes dos filtros e pesos das combinações lineares são variáveis de otimização aprendidas visando minimizar o custo de uma função específica. A arquitetura foi batizada de rede neural convolucional geodésica ou GCNN (Figura C.4). *Shapenet* foi capaz de aprender features invariantes de descritores de formas, além disso, abordagens anteriores como *kernel* de assinaturas de calor e onda, descritores espectrais ótimos, e contextos de formas intrínsecas poderiam ser obtidos como configurações específicas da *Shapenet*. Vale observar porém que este tipo de representação tem carência de informações do contexto global [86].

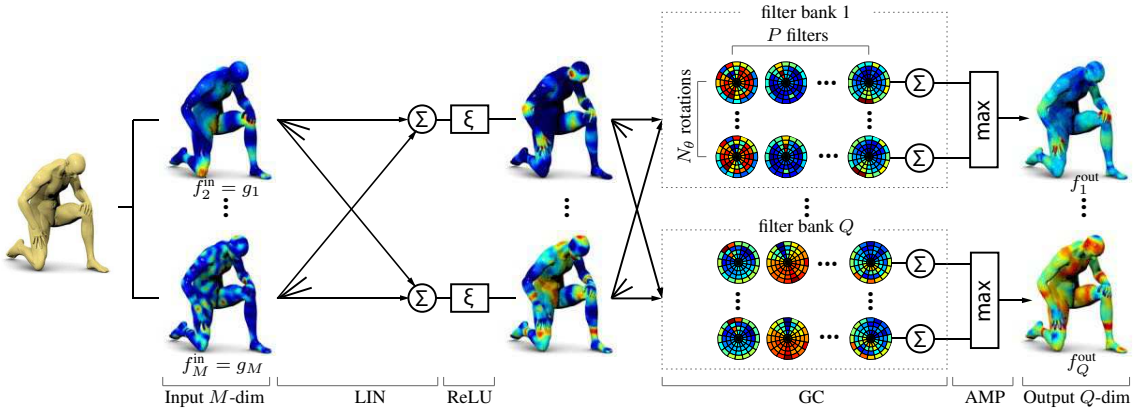

Figura C.4: (
Arquitetura GCNN contendo uma camada convolucional aplicada à vetores da geometria M com 150 dimensões (camada de entrada), produzindo um descritor de features Q de 16 dimensões (camada de saída). A saída de cada camada é entrada para a próxima. Na figura é possível identificar a camada LIN (*linear*) que ajusta as dimensões da entrada e saída através de uma combinação linear, a camada GC (*geodesic convolution*) que substitui a camada convolucional utilizada em CNNs Euclidianas clássicas. Devido a ambiguidade da coordenada angular, o resultado da convolução geodésica é calculado para diversas rotações (N_θ) dos filtros. Por fim, a camada AMP (*angular max-pooling*), utilizada em conjunto com a camada GC, calcula o *max-pooling* sobre as rotações dos filtros.

Segundo seu criador, *Shapenet* foi a primeira extensão do paradigma CNN generalizável para domínios não-Euclidianos, i.e., podia ser treinada com um conjunto específico de formas e aplicado em outra [87], guardadas as devidas limitações. Generalizada por Masci et al. [88] como Redes Neurais Convolucionais Geodésicas para variedades Riemannianas, *Shapenet* não deve ser confundida com o trabalho de Wu et al. [151], da mesma época. Nele, os autores propõem representar uma forma geométrica tridimensional como uma distribuição de probabilidade de variáveis binárias em uma grade de voxels 3D, usando uma Rede Convolucional de Convicção Profunda (*Deep Belief Networks*) [50] para reconhecimento de categoria ou identificação de formas 3D baseadas em mapas de profundidade 2.5D. A rede era capaz de aprender a distribuição de formas 3D entre diversas categorias e poses arbitrárias de dados CAD (*computer aided design*), descobrindo partes de representações hierárquicas da composição. Porém, por tratar-se de uma arquitetura CNN padrão aplicada sobre representações de visualizações 2D de formas volumétricas, é inadequado para formas deformáveis devido a falta de invariância de deslocamento nas variedades Riemannianas [88].

Apesar do progresso na análise de formas com estruturas similares, e.g., formas humanas, e modos similares de deformação, i.e., deformações isométricas ou quase-isométricas, lidar com coleções de formas de famílias diversificadas estruturalmente e geometricamente (móveis, veículos ou máquinas, por exemplo) ainda é um problema em aberto [54]. Baseando-se em trabalhos anteriores na área de Redes de Convicção Profunda (*Deep Belief Networks*), em especial pelos trabalhos de Wu et al. [151] e Xie et al. [136], Huang et al. [54] desenvolveram um modelo generativo profundo para superfícies de formas 3D (Figura C.5) com algumas diferenças em relação as formulações anteriores. O modelo proposto analisa e sintetiza coleções de formas 3D aprendendo,

desde o início, representações da variabilidade de superfícies. Possui dois componentes principais: um modelo de deformação probabilístico e um modelo generativo de superfície. O modelo de deformação probabilístico estima correspondências de pontos difusos e segmentação de partes das formas em conjunto. Ao contrário de trabalhos anteriores, que contam com formas pré-existentes ou primitivos geométricos simples para *templates*, este aprende a geometria do *template* e deformações a partir da coleção de entrada (Figura C.6). Por sua vez, a saída do modelo de deformação fornece a entrada para o modelo generativo de superfície, cujo propósito é aprender as relações geométricas e estruturais das partes e posições correspondentes dos pontos da superfície. A ideia principal do modelo generativo segue o conceito de aprendizagem profunda, aprender as relações dos dados da superfície hierarquicamente: aprende arranjos geométricos de pontos e partes individuais através de uma primeira camada de variáveis latentes. Esta arquitetura hierárquica do trabalho de Huang et al. [54] foi pensada e alinhada de acordo a composição natural das formas, considerando que essas formas usualmente possuem uma estrutura bem definida, que por sua vez é definida por partes, e que estas partes são compostas de retalhos e arranjos de pontos com determinada regularidade [54].

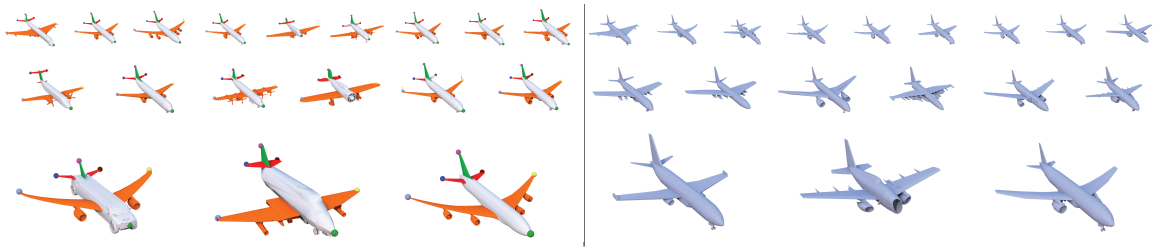

Figura C.5: Parte do resultado do modelo de Huang et al. [54] (figura retirada da mesma fonte): dada uma coleção de formas 3D, um modelo probabilístico é treinado para executar a análise e síntese de junções das formas. À esquerda a figura mostra partes semânticas e pontos correspondentes nas formas deduzidos pelo modelo. À direita figura exibe novas formas sintetizadas pelo modelo.

Tratando-se do problema de rotulação de malhas 3D com o uso do redes neurais convolucionais profundas, Xie et al. [136] propuseram um método para segmentação e rotulação empregando um conjunto de descritores de features e ELM para treinar um classificador da rede neural para rotulação de malhas. Contudo, a simples combinação das características geométricas (e.g., modelo linear e rede neural superficial) é geralmente insuficiente para ser generalizado para todo tipo de malha [46]. Guo et al. [46] propõem então uma nova abordagem para rotulação de malhas utilizando redes neurais convolucionais profundas. Nessa abordagem, treina-se uma CNN de forma supervisionada utilizando um grande *pool* de features geométricas clássicas. Durante o processo de treinamento, estas features baixo-nível são não-linearmente combinadas e comprimidas para gerar uma representação efetiva e compacta para cada triângulo da malha. A partir daí, baseando-se nas CNNs treinadas e nas representações das malhas, um vetor de rótulos é inicializado para cada triângulo, com o propósito de indicar suas probabilidades de pertencer às várias partes do objeto. Ao final, um algoritmo de rotulagem de malhas baseado em grafos é adotado para otimizar os rótulos dos triângulos considerando as consistências do rótulo. Os resultados obtidos mostraram-se robustos para diversas malhas 3D e *benchmarks* públicos.

O problema de segmentação de formas em partes significativas assume um papel importante na análise e compreensão de formas 3D, e é natural que métodos orientados

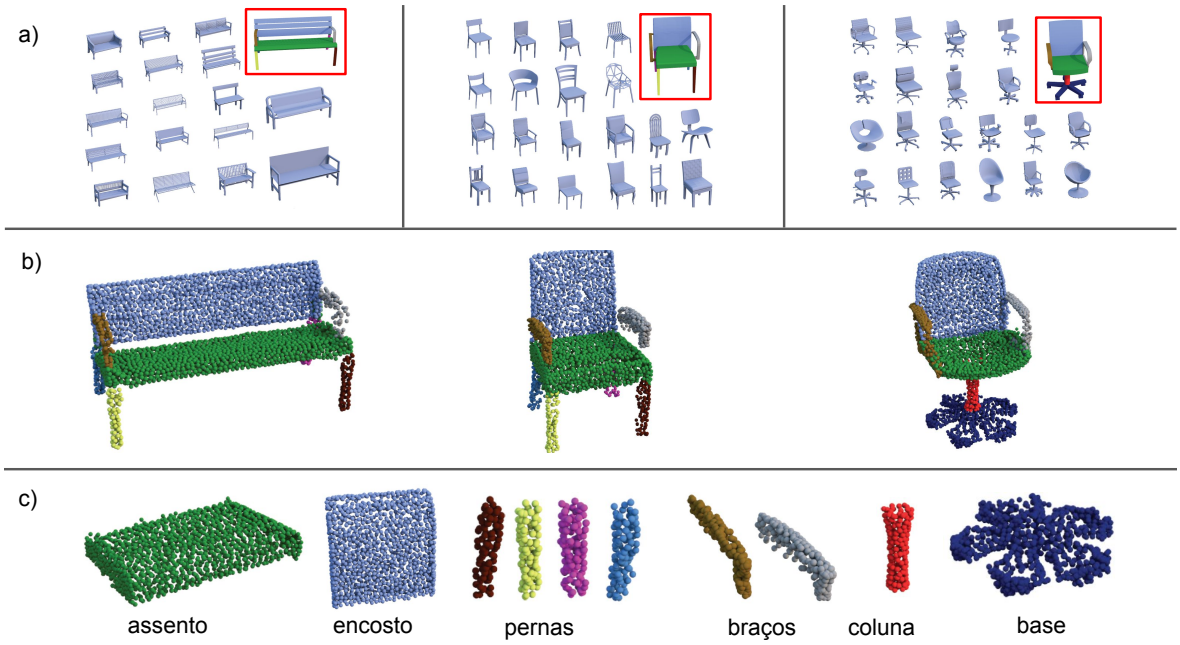

Figura C.6: Aprendizagem de modelos de partes para uma coleção de cadeiras (retirado de [54]). (a) o método agrupa a coleção de entrada em grupos contendo formas estruturalmente semelhantes (e.g., bancos, cadeiras com quatro pernas, cadeiras de escritório). É exigido como entrada uma segmentação rotulada para pelo menos uma forma por grupo (caixa em vermelho). (b) Um modelo (*template*) é aprendido para cada parte semântica por grupo. (c) Dadas as partes dos modelos aprendidos por grupo específico, o método aprende modelos de nível superior para partes semânticas que são comuns entre os diferentes grupos (e.g., assentos, encostos, pernas e braços em cadeiras. Os templates das partes com nível superior permite ao métdo estabelecer correspondências entre formas que pertencem à grupos estruturalmente diferentes, mas que compartilham partes sob o mesmo rótulo.

a dados fossem aplicados com o intuito de melhorar a performance, visto que a segmentação de formas 3D pode na realidade ser considerada um processo de agrupamento de faces no espaço das features (*feature space*). Tal fato já havia sido demonstrado por Kalogerakis et al. [56], cuja proposta consistia em realizar a rotulação e segmentação simultaneamente, e que a performance da tarefa de segmentação poderia ser melhorada por técnicas envolvendo aprendizado e baseada em dados das formas 3D, assim como no trabalho de Xie et al. [136], em que uma prática comum era construir uma assinatura da forma extraindo algum tipo de propriedade geométrica e aplicá-la na segmentação da forma com a utilização de alguma técnica de decomposição. Não obstante, métodos desta natureza geralmente necessitavam de um grande conjunto de resultados de segmentação rotulados manualmente, limitando sua utilização. Shu et al. [116] foram os primeiros a introduzir aprendizagem profunda para segmentação de formas 3D. Vale notar que o método aprende diretamente a partir de formas 3D não rotuladas e sem rotulação manual. O algoritmo proposto consiste em três partes principais:

- pré-decomposição de cada forma em retalhos primitivos para gerar super-segmentação e calcular diversas assinaturas como features baixo-nível das formas;
- aprender features alto-nível, de forma não supervisionada, baseadas em aprendizagem profunda a partir de features baixo-nível; e
- cada resultado da segmentação ou co-segmentação pode ser obtido pelo agrupamento de retalhos no espaço das features alto-nível.

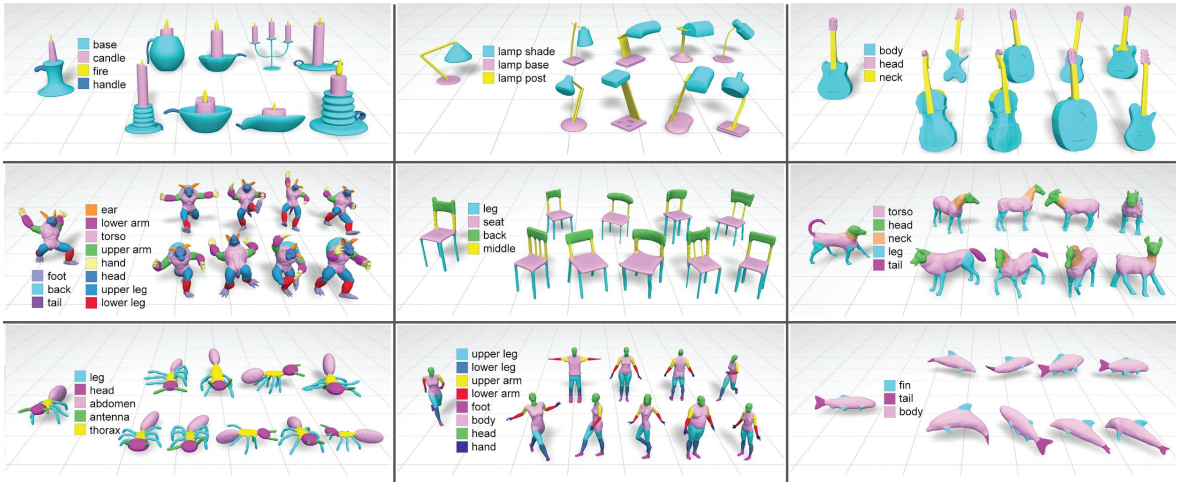

Figura C.7: Resultados da rotulação de malhas de Guo et al, candelabro, lâmpada, violão, tatu, cadeira, quadrúpede, formiga, humano e peixe (retirado de [46]).

Embora tenha obtido bons resultados (Figura C.8), o algoritmo possui alguns problemas. Não é possível distinguir features desejadas e não desejadas na etapa de aprendizagem, mesmo que haja features geométricas suficientes, não há como garantir que os resultados da segmentação serão definitivamente melhores. Além disso, o tempo gasto com o treinamento era longo e a compatibilidade entre algumas bibliotecas não era ideal [116].

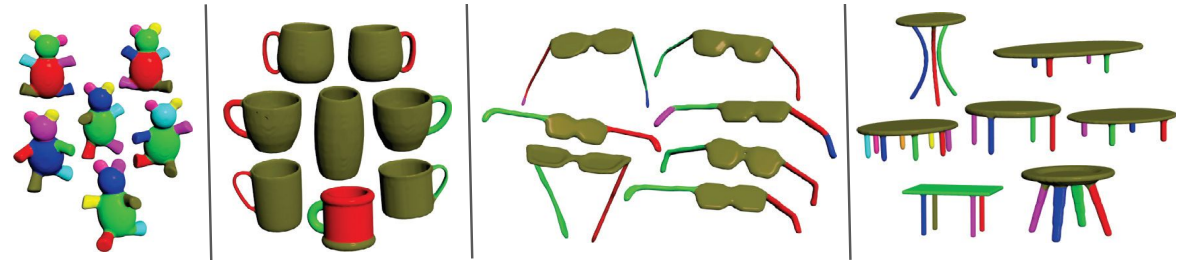

Figura C.8: Resultados representativos de segmentação produzidos pela abordagem de Shu et al. [116] sobre o conjunto de dados de *benchmark* de segmentação Princeton (retirado de [116]).

Inspirados por trabalhos de aprendizagem profunda, mas que tratavam de problemas de correspondências entre imagens 2D [40, 147, 140, 84], Wei et al. [134] introduziram o primeiro arcabouço de correspondências densas e precisas entre formas humanas 3D não-rígidas com diferentes vestimentas ou com dados de entrada apenas parciais, e.g., forma 3D capturada por escaneamento em que há falhas na malha obtida. O autor utiliza uma CNN profunda para treinar um descritor de features em pixeis de mapas de profundidade das formas 3D, mas ao invés de treinar a rede para solucionar o problema de correspondência de formas imediatamente, a rede é treinada para solucionar um problema de classificação de regiões do corpo (segmentadas em retalhos), modificado para melhorar a suavidade perto das regiões de fronteira dos descritores aprendidos. De acordo com Wei et al. [134], esta abordagem garante que pontos próximos do corpo humano estarão próximos no espaço de features, e vice-versa, renderizando os descritores de features apropriados para o cálculo da correspondência densa entre as digitalizações. Como principais contribuições destacaram-se o fato do arcabouço ser o primeiro a encontrar correspondências densas entre formas humanas com roupas e dados de entrada parciais e a arquitetura da CNN profunda a qual aprende uma incorporação

suave usando uma técnica de multissegmentação em formas humanas.

Seguindo trabalhos de segmentação de formas 3D [56, 46] e com uma arquitetura de rede neural similar a rede de segmentação completamente convolucional de Long et al. [114], Yi et al. [143] realizam um estudo sobre o problema de anotação semântica em formas de modelos 3D representados como grafos. Em comparação com imagens 2D (as quais basicamente são grades bidimensionais), grafos representando superfícies de formas 3D são estruturas de dados irregulares e não-isomórficas. Para permitir a predição de funções de vértices nestes grafos de formas 3D, recorre-se ao método CNN espectral que permite o compartilhamento de pesos da rede através de kérneis de parametrização no domínio espectral coberto pelos autovetores do Laplaciano do grafo. A rede, chamada de *SyncSpecCNN* ataca dois desafios principais: como compartilhar coeficientes e direcionar análises em multi-escala em diferentes partes do grafo para uma única forma, e como compartilhar informação através de diferentes formas relacionadas que podem ser representadas por grafos bem distintos. Para alcançar estes objetivos, é introduzida uma parametrização espectral de kérneis convolucionais expandidos e uma rede transformadora espectral, obtendo bons resultados em tarefas como segmentação de partes de formas e predição de pontos chave (Figura C.9).

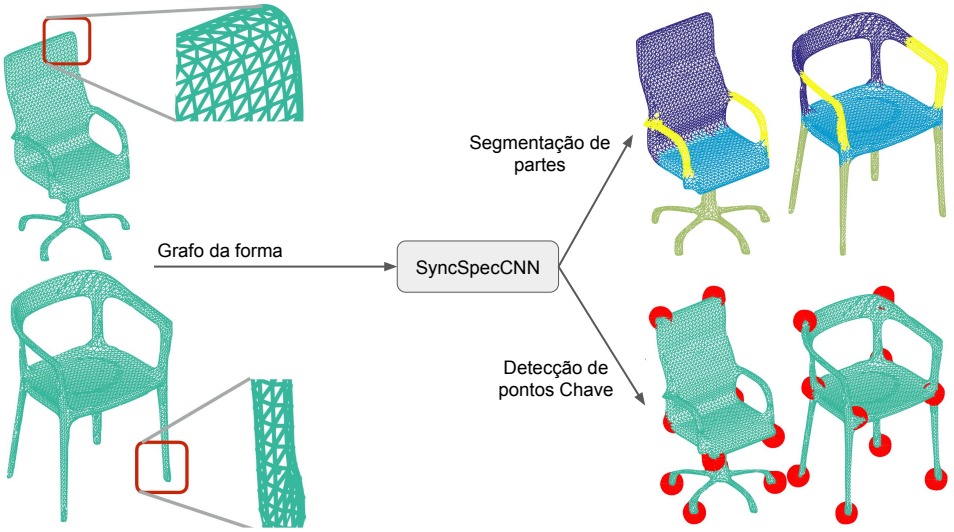

Figura C.9: SyncSpecCNN toma como entrada um grafo da forma com funções de vértices (i.e., função de coordenadas espaciais) e prediz um rótulo por vértice. A figura mostra como exemplo segmentação de partes 3D e predição de pontos chave como saídas (retirado de [143]).

Na filosofia comum de métodos que atuam no domínio espacial como [88, 16], Monti et al. [91] apresentam um trabalho em que formula operações semelhantes a convolução como correspondência de modelos com "retalhos" locais intrínsecos em grafos ou variedades. A abordagem foi apelidada de *MoNet*. A principal novidade é a maneira pela qual o retalho é extraído: utiliza-se uma construção paramétrica ao contrário das abordagens anteriores que utilizavam retalhos fixos, e.g., em coordenadas geodésicas ou coordenadas de difusão. Em particular, é apresentado que operadores de retalhos podem ser construídos como uma função de um grafo local ou pseudo-coordenadas de variedades, além de um estudo de funções representadas como uma mistura de kérneis Gaussianos. Tal construção permite formular propostas anteriores como CNN geodésica (GCNN) [88] e CNN anisotrópica (ACNN) [17] em variedades [91]. A Figura C.10 mostra funções de ponderação do operador de retalhos, que são fixas nas arquiteturas

GCNN e ACNN, e parte dos parâmetros aprendíveis na MoNet.

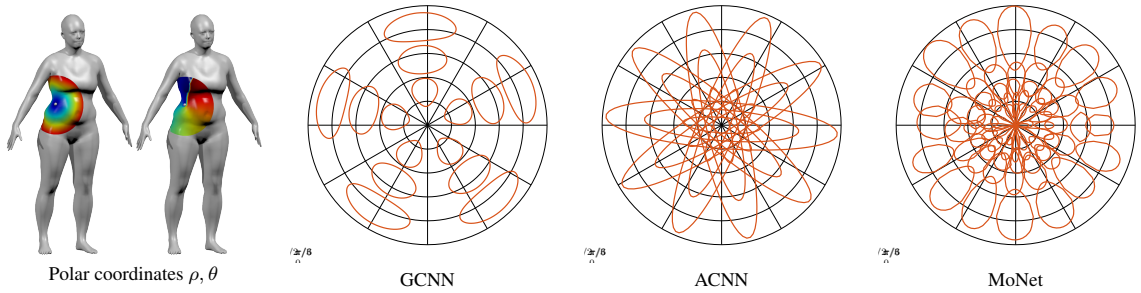

Figura C.10: Funções de ponderação do operador de retalhos da MoNet. À esquerda (formas): coordenadas polares intrínsecas ρ, θ na malha ao redor de um ponto marcado em branco. Nas 3 imagens mais à direita, as funções de ponderação do operador de retalhos $w_i(\rho, \theta)$ utilizado em diferentes generalizações de convolução na variedade (ajustadas manualmente na GCNN [88] e ACNN [17] e aprendidas na MoNet [91]). Os kérneis estão normalizados (retirado de [91]).

Em tarefas de correspondência de formas, há uma variedade de métodos recentes ou clássicos cuja revisão da literatura detalhada está fora do escopo deste trabalho. Contudo, destaca-se uma família de métodos baseados na noção de mapas funcionais, introduzidos por Ovsjanikov et al. [94], modelando correspondências como operadores lineares entre os espaços de funções em variedades. Projetando um modelo de predição no espaço dos mapas funcionais que agem como operadores lineares fornecendo uma representação compacta da correspondência entre formas deformáveis, Litany et al. [78] introduzem uma mudança de paradigma nas tarefas de aprendizado com mapas funcionais. Ao invés de modelar o problema como um problema de rotulação, em que cada ponto da forma alvo recebe um rótulo identificando um ponto em determinado domínio de referência; a correspondência é construída a posteriori compondo as predições de rótulos de duas formas 3D de entrada. Os descritores de forma são descobertos de maneira similar a abordagem de Corman et al. [27], em que a combinação de pesos para um conjunto de descritores de entrada é aprendido de maneira supervisionada e sua construção baseia-se no framework de mapas funcionais [94]. Porém, enquanto o critério de otimização de Corman et al. [27] é definido em termos da divergência com a precisão de um mapa funcional da classificação do conjunto de treinamento no domínio espectral, Litany et al. [78] têm por objetivo recuperar um mapa ótimo no domínio espacial. O processo de aprendizado é modelado via rede profunda residual (*deep residual network*) chamada *FMNet* (Figura C.11) que toma campos de descritores densos definidos em duas formas 3D como entrada, e produz um mapa suave (*soft map*) entre os dois objetos dados (como ilustrado na Figura C.12).

Kalogerakis et al. [55] novamente trazem à tona o problema de segmentação de formas 3D em partes semânticas rotuladas, mas desta vez, utilizando uma abordagem multivisão relacionada a métodos de aprendizagem por segmentação de imagens e formas 3D. Dada uma malha crua de polígonos 3D como entrada, o método gera um conjunto de imagens a partir de múltiplas visualizações que são automaticamente selecionadas para uma cobertura ótima da superfície. Estas imagens alimentam uma rede *feed-forward* que produz como saída mapas de confiança por partes da forma, através de camadas de processamento de imagem (pré-treinadas em grandes conjuntos de dados de imagem). Os mapas de confiança são unidos e projetados na representação do espaço da superfície da forma através de uma camada de projeção. Por fim, a arquitetura incorpora uma camada de Campo Aleatório Condicional (CRF, *Conditional Random*

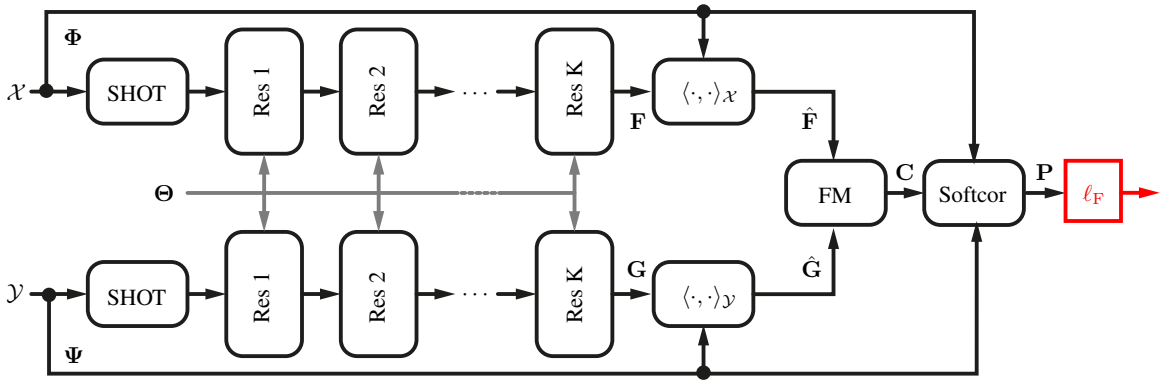

Figura C.11: Arquitetura FMNet (retirada de [78]). Descritores pontuais de um par de formas tomados como entrada passam por uma sequência idêntica de operações (com pesos compartilhados), resultando em descritores refinados $\mathbf{F}$, $\mathbf{G}$. Estes, por sua vez, são projetados nos autovetores do Laplaciano Φ, Ψ para produzir as representações espectrais $\hat{\mathbf{F}}$, $\hat{\mathbf{G}}$. As camadas do mapa funcional (FM) e da correspondência suave (SoftCor), não são paramétricas e são utilizadas para montar a perda (erro) geometricamente estruturada ℓ_F.

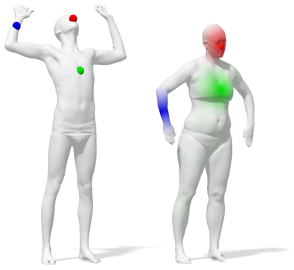

Figura C.12: Dadas malhas origem e destino como entrada, a rede produz como saída uma matriz de correspondência suave cujas colunas podem ser interpretadas como a distribuição da probabilidade sobre a forma destino (retirado de [78]).

Field) baseado em superfície, que promove rotulação consistente de toda superfície. Toda a rede, inclusive o CRF, é treinada de maneira ponto-a-ponto para alcançar melhor performance. A principal contribuição deste trabalho de Kalogerakis et al. [55] é a introdução de uma arquitetura profunda para raciocínio composicional baseado em partes em representações de formas 3D sem a utilização estágios de processamento com geometria construída à mão ou descritores ajustados manualmente [55].

Um dos principais desafios na aplicação de CNN em superfícies curvas é que não há generalização clara do processo de convolução. Em particular, existe a necessidade de duas propriedades consideradas fundamentais na operação de convolução para o sucesso da arquitetura da CNN: localidade e invariância de translação [86]. Embora seja possível parametrizar a superfície localmente em um retalho geodésico em torno de um ponto [87, 88] ou criar imagens de geometria para parametrizar globalmente uma superfície esférica em uma imagem 2D [117], definir uma translação local invariante para o operador de convolução em superfícies curvas não é tarefa trivial. Diante dos obstáculos topológicos e geométricos desta tarefa, Maron et al. [86] notaram que o único tipo de superfície para a qual uma convolução invariante a translação era bem definida era o toro. Baseado nesta observação, dentre outras, o método proposto aplica aprendizagem profunda para superfícies de formas esféricas utilizando uma parametrização sem emendas para um toro planar em que o operador de convolução está bem determinado (Figura C.13). Como resultado, um arcabouço padrão de aprendizagem profunda podia ser prontamente adaptado para aprendizagem semântica de proprie-

dades alto-nível da forma. O método foi demonstrado para dois tipos de aplicações, segmentação semântica e detecção automática de pontos de referência em superfícies anatômicas. A Figura C.14 mostra a aplicação da técnica de segmentação de formas humanas em outras classes de formas diversas. Embora o conjunto tenha sido treinado apenas com dados de formas humanas padrão, a rede produz resultados plausíveis.

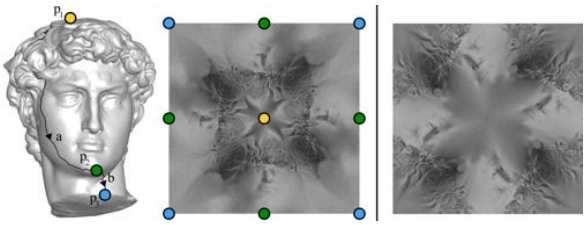

Figura C.13: Calculando a estrutura de um toro plano (centro) em 4 capas (4 exatas cópias da superfície) de uma superfície do tipo esférica (esquerda), definidas por 3 pontos informados (amarelo, verde e azul). A imagens mais à direita mostra o toro plano resultante de uma escolha diferente dos 3 pontos (retirado de [86]).

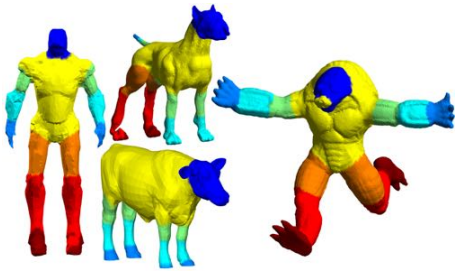

Figura C.14: Resultados do algoritmo de Maron et al. [86] treinado exclusivamente na segmentação semântica do corpo humano aplicada à superfícies de outras classes. O método mesmo assim produz resultados plausíveis, indicando certo grau de generalização (retirado de [86]).

Na subárea de síntese e formas e modelos de formas generativas, assim como de modelos regenerativos profundos, Nash et al. [92] apresentam um trabalho cujo objetivo erá tomar uma coleção de objetos de entrada segmentados com correspondência densa de pontos e aprender um modelo generativo de formas 3D capaz de sintetizar novos modelos de exemplo, completar objetos incompletos, e embutir objetos 3D em um espaço latente de menor dimensão. Seu trabalho assemelha-se ao de Huang et al. [54], mas ao invés de utilizar uma *Beta Shape Machine* (BSM) que trata-se de um modelo probabilístico não direcionado, os modelos são direcionados. O cerne do método é um modelo generativo que descreve a distribuição de probabilidade sobre os pontos da superfície da forma, normais da superfície, e partes existentes em grandes coleções de objetos 3D. As relações entre estas variáveis para objetos 3D é complexa devido às relações hierárquicas das partes, relações de simetria, assim como suavidade local e outras restrições estruturais [92]. O modelo é uma variante do auto-codificador variacional (VAE, *variational auto-encoder*): um modelo capaz de capturar distribuições complexas em cima de dados de alta dimensão [61]. O VAE consiste em uma rede codificadora que mapeia dados para um código latente de menor dimensão, e um decodificador que mapeia o código latente para uma reconstrução dos dados. Desta forma, o VAE é obrigado a tornar o código oculto altamente informativo em relação aos dados associados. No trabalho de Nash et al. [92], o VAE trabalha com uma arquitetura hierárquica na qual as camadas mais altas capturam relações estruturais e

globais dos objetos, e as camadas inferiores capturam a variabilidade nas partes dos objetos. Todavia, o método tem limitações, sendo a mais significante delas a necessidade de conjunto de dados de entrada conter segmentações de malha consistentes e também as correspondências densas. Uma questão para ser levada em conta é que a noção de correspondência de pontos um-a-um para objetos em diversos conjuntos de dados como por exemplo, diversos modelos de cadeiras, é mal fundada [92].

Moldado sob a influência de diversas abordagens relevantes em técnicas de aprendizagem profunda para dados tridimensionais [142, 151, 117, 118, 86, 98, 97], Groueix et al. [44] propõem redes de deformação de formas (*shape deformation networks*), uma solução tudo em um para o problema de correspondência entre formas baseando-se em *templates*. Em primeiro lugar são processadas nuvens de pontos representando as formas de entrada utilizando uma arquitetura similar a de Qi et al. [98]. Em seguida, de maneira similar a Sinha et al. [118], uma representação da superfície é aprendida. A rede de deformação de formas é treinada como parte de uma arquitetura codificador-decodificador, que prontamente aprende uma rede codificadora que toma uma forma 3D alvo como entrada e gera uma representação global de features, e uma rede decodificadora de deformação de formas que toma como entrada a característica global e deforma o template transformado-o na forma alvo. No momento dos testes, o alinhamento da forma com o *template* é otimizado localmente pela distância de Chamfer entre o alvo e a forma gerada sobre a representação da característica global que é passada como entrada para a rede de deformação de formas (Figura C.15) .

Em comparação com a maioria das técnicas de aprendizagem profunda para dados 3D, Groueix et al. [44] não codificam as correspondências explicitamente na saída de uma rede convolucional, mas implicitamente as aprende otimizando parâmetros da rede geradora. Ainda, em contraste com métodos de correspondências de formas baseados em *template*, como o trabalho de Zuffi et al. [153], o método não necessita de um *template* deformável ajustado a mão; os parâmetros de deformação e graus de liberdade são aprendidos implicitamente pelo codificador.

Considerando a segmentação de formas 3D como suporte para muitas outras tecnologias de processamento 3D, Wang et al. [130] projetam uma nova arquitetura de rede totalmente convolucional para formas, intitulada de *Shape Fully Convolutional Networks* (SFCN). Nessa arquitetura SFCN, formas 3D são representadas por grafos, em que novas operações de *pooling* e convolução em grafos são similares às operações de *pooling* e convolução utilizadas em imagens. Além disso, baseada na ideia de segmentação de imagens de uma arquitetura original chamada FCN [114], foi implementada uma nova operação de geração, que funciona como uma ponte para facilitar a execução de convolução e operações de *pooling* diretamente nas formas 3D. Entretanto, há limitações, e.g., as formas envolvidas devem ser malhas de uma variedade, para determinar mais facilmente a conectividade entre triângulos. A arquitetura SFCN não possui função de seleção de features, e para obter melhor compartilhamento de parâmetros, a arquitetura pode necessitar que todas as malhas do conjunto de treinamento possuam a mesma granularidade.

O problema de segmentação de formas tornou-se base e pré-requisito para explorar características inerentes a forma. Por consequência, a co-segmentação de formas 3D, combinada com coleções de modelos, co-segmenta múltiplas partes de formas 3D dentro da mesma categoria (objetos semanticamente semelhantes), também atrai atenção expressiva, e.g., como no trabalho de Yin et al. [146]. Apesar de existirem similari-

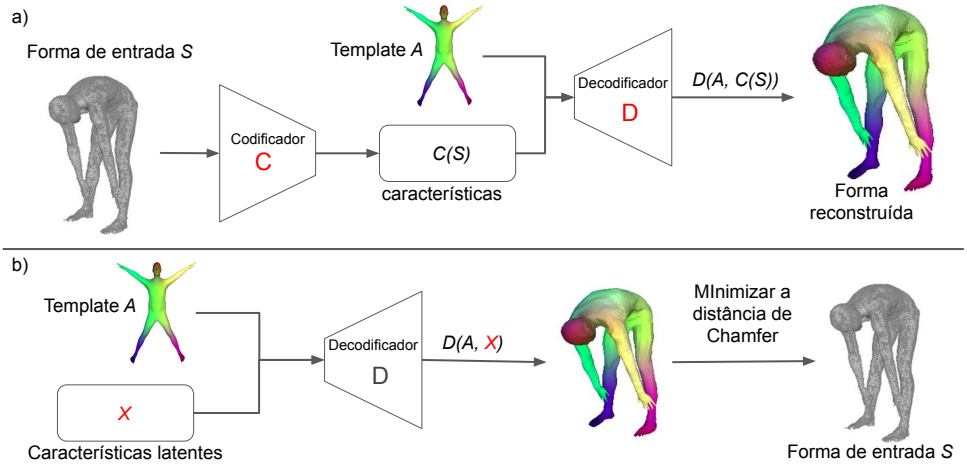

Figura C.15: Visão geral do método de Groeix et al. [44]. (a) Um passo *feed-forward* do auto-codificador codifica a nuvem de pontos S da entrada para código latente $C(S)$ e reconstrói S utilizando $C(S)$ para deformar o template A. (b) A reconstrução $D(A,C(S))$ é refinada executando um passo de regressão sobre a variável latente x, minimizando a distância de *Chamfer* entre $D(A,x)$ e S. Por fim, dadas duas nuvens de pontos S_r e S_t, para relacionar um ponto q_r de S_r com um ponto q_t de S_t, procura-se o vizinho mais próximo p_r de q_r em $D(A,x_r)$, que por definição está em correspondência com p_t; e procura-se pelo vizinho mais próximo q_t de p_t em S_t. A cor vermelha indica elementos que estão sendo otimizados.

dades e correspondências diversas entre algumas formas 3D de uma mesma categoria, co-segmentar efetivamente as partes da forma 3D ainda apresenta desafios. Cada forma 3D individualmente possui sua pose e aparência, e as partes cuja geometria encontra features similares pode ter rótulos semânticos distintos. Neste sentido, Yin et al. [146] propuseram um método efetivo de co-segmentação, combinando features individuais da forma e consistência global. A proposta visava tratar problemas como já encontrados em outros trabalhos como o de Huang et al. [52], que mantinha e propagava features individuais das formas, passando similaridade e consistência por pares. Pelo fato da consistência global ser perdida, uma grande variação na forma 3D poderia levar a resultados incorretos [146]. Dada uma categoria de forma 3D, Yin et al. utiliza cada forma com um dicionário para representar de maneira esparsa toda a categoria daquela forma. É aplicado um algoritmo de cortes normalizados para dividir cada forma em retalhos primitivos reduzindo a complexidade computacional. Em seguida, são extraídas as features do retalho, as quais incluem função do diâmetro da forma, distância geodésica média, contexto da forma e distância da superfície medial, da mesma maneira que no trabalho de Guo et al. [46]. Para descobrir a expressão consistente de cada classe e a estrutura global comum, são introduzidas restrições de baixo nível (*lowrank constraints*). Por fim, são utilizadas representações de erros para ponderar os coeficientes e obter os valores mais seguros. Outrossim, através de um método simples de agrupamento e suavização do processo, chega-se aos resultados finais da co-segmentação.

Ainda em relação a segmentação de formas 3D, George et al. [39] observaram que havia poucos estudos comparativos das soluções existentes além de duas outras coisas em comum. A primeira dela é que técnicas baseadas em features frequentemente são lentas ou sensíveis ao redimensionamento de features. A segunda é que as técnicas apresentadas frequentemente sofrem de problemas de reprodutibilidade. O autor fornece publicamente implementações de diversas técnicas de aprendizagem profundo, a rigor, redes neurais, redes auto-codificadoras e CNNs, cujas arquiteturas possuem ao menos

duas camadas. O estudo também propôs uma nova forma de calcular o fator de conformidade (CF, *conformal factor*), uma nova técnica de CNN, e um estudo compreensivo de diversas técnicas de aprendizagem profunda para comparação de seus patamares. Adicionalmente, foi exibido um estudo abrangente e comparativo de algumas técnicas de aprendizagem profunda para segmentação de malhas. Conclui-se que arquiteturas mais simples ainda são capazes de executar razoavelmente bem utilizando o mesmo conjunto de features geométricas quando comparadas a modelos CNN mais complexos, além de que o tempo de treino é significante menor. Outra contribuição do trabalho de George et al. [39] é a proposta de uma nova CNN para segmentação de malhas. Utiliza-se dados 1D, filtros e uma arquitetura *multi-branch* para treinamento separado de features multi-escala. Ao invés de projetar features geométricas 3D em imagens 2D e utilizar filtros para ajustá-las ao pipeline CNN baseado em imagens, são utilizados dados 1D e filtros que possibilitam mitigar inferências desnecessárias de features não relacionadas. Isso também evita o problema de ajuste de parâmetros para remodelagem e re-amostragem de features, obtendo resultados considerados interessantes (Figura C.16).

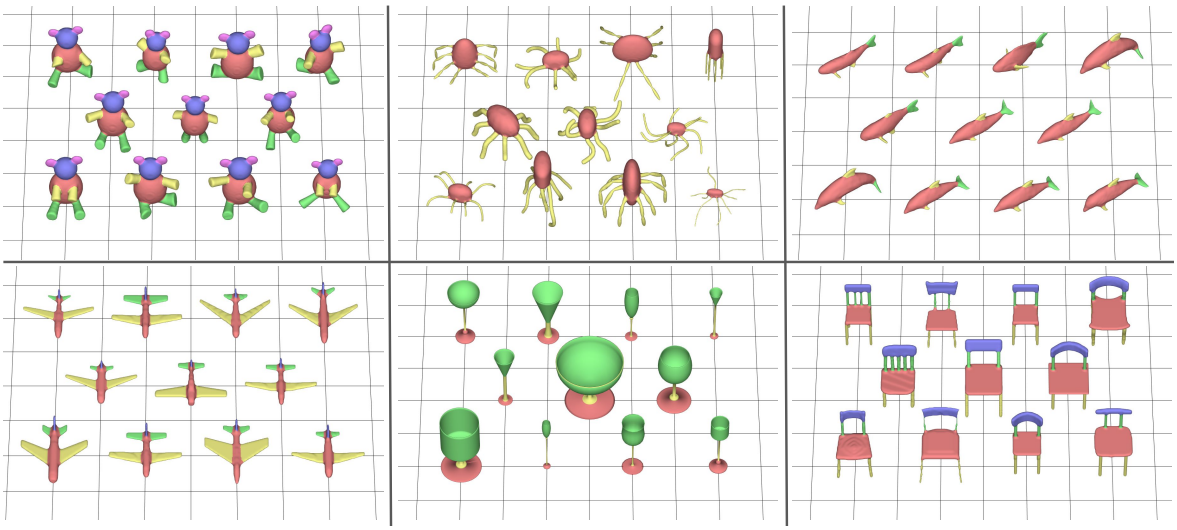

Figura C.16: Visualização de alguns dos resultados da técnica de CNN 1D nos conjuntos de dados PSB e Coseg, com precisão acima de 95% (retirado de [39]).

Os trabalhos elencados até o momento apresentam técnicas de aprendizagem profunda supervisionada ou semi-supervisionada. Todavia, em muitos casos a forma de treinamento supervisionada é proibitiva em termos da quantidade de dados manualmente classificados necessários. Não obstante, apesar das técnicas de aprendizado terem alcançado um bom desempenho, sua utilidade é limitada pela exigência da presença de mapas de dados com a verdade absoluta disponível entre um número suficiente de exemplos de treino. Isto torna difícil aplicar tais abordagens à novas classes de formas em que estes dados com a verdade absoluta não está disponível [107].

Halimi et al. [47] apresentaram a primeira tentativa de criar um arcabouço de aprendizagem totalmente não supervisionado para solucionar o problema fundamental da correspondências entre formas não rígidas. Os autores mostram um esquema de aprendizado para correspondência entre formas 3D densas baseado em um critério puramente geométrico. A abordagem sugerida transita entre soluções baseadas em modelos e baseados em dados, aprendendo descritores pontualmente que resultam em correspondências minimizando a distorção da distância geodésica entre os pares. Para

o modelo, as deformações naturais, tais como mudanças na forma, aproximadamente preserva a estrutura métrica da superfície, produzindo um critério natural para guiar o processo de aprendizado relativo a predições que minimizam as distorções. Nesta base, a exigência por dados rotulados é substituída por um critério puramente geométrico. O modelo de aprendizado resultante é agnóstico a classe, e capaz de levantar qualquer tipo de dados da deformação geométrica para a fase de treinamento. Mas, apesar da minimização da distorção da distância geodésica ter encontrado satisfatória generalização em uma variedade de *benchmarks*, variação na escala global e mudanças topológicas podem exigir uma adaptação adequada. Ciente deste problema, Roufosse et al. [108] também apresentaram um método de aprendizagem profunda, intitulado *SURFMNet* (Figura C.17), que calcula correspondências sem os dados de verdade absoluta. O processo é similar ao apresentado por Halimi et al. [47], mas opera puramente no domínio espectral para melhorar a eficiência. Ambos os método obtêm bons resultados para formas humanas. Tal caraterística advém do fato de que articulação de poses humanas podem ser modeladas como isometrias aproximadas, ou seja, a correspondência latente introduz pouca distorção métrica.

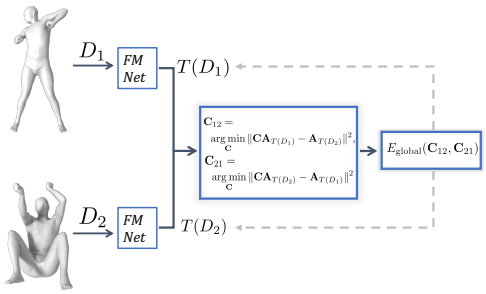

Figura C.17: Visão geral da abrodagem SURFMNet (retirado de [108]): dado um par de formas e seus respctivos destcritores D1, D2, esses são otimizados em uma transformação não linear T utilizando a arquitetura FMNet [78] de modo que os descritores transformados levem à mapas funcionais que melhor satisfaçam às restrições estruturais.

Utilizando formas codificadas como nuvem de pontos, Groeix et al. [43] tratam do problema de correspondência entre superfícies treinando um modelo que toma como entrada um par de formas — origem e destino — e visa descobrir a deformação da origem no destino. A arquitetura do modelo pode ser separada em duas partes, como uma rede de dois estágios:

- Uma rede de predição de parâmetros cuja saída são parâmetros de transformação das formas origem e destino; e
- uma rede de deformação que transforma a forma origem na destino utilizando os parâmetros definidos pela primeira rede.

Tanto a forma origem como a destino são codificadas de uma nuvem de pontos (assim como na *PointNet* [98]) para um vetor de features no espaço latente de onde uma rede do tipo MLP (*multi layer perceptron*) prediz os parâmetros de transformação. Esses parâmetros são entrada de outra rede, para deformar a origem no destino, empilhando camadas de transformação e camadas totalmente conectadas. O treinamento das redes deve ser realizado para uma categoria específica de objetos, já que, segundo o autor, diferentes categorias de formas podem ter diferentes topologias. Porém, apesar de apresentar bons resultados, o método não é competitivo quando existem muitas amostras

de dados para treinamento, pois soluciona a deformação contra cada uma das formas segmentadas fornecidas.

C.3 Transferência de Deformação e Pose entre Formas

Superfícies complexas que variam de acordo com o tempo surgem em diversas aplicações como filmes, jogos digitais ou aplicações científicas. Quer seja produzidas a mão, como em *key-framing*, via simulação de processos físicos como em pele ou tecidos, ou obtidas por captura de movimentos, superfícies que variam com o tempo são geralmente representadas por polígonos [62]. Em muitos casos, pode ser vantajoso utilizar a conectividade de uma única malha estática ao modelar deformações na superfície ao invés de usar uma malha a parte para cada passo de tempo. No entanto, adotar conectividade fixa gera um inconveniente óbvio, a necessidade de muito mais polígonos do que o mínimo necessário em qualquer quadro de animação. Tal fato pode ser agravado em superfícies que passam por deformações não rígidas e extremas, como em técnicas de *morphing* e outras formas de animação não esqueletal. Tais deformações necessitam de malhas densas, já que a conectividade estática deve ser capaz de representar com precisão todas as possíveis deformações da superfície [62].

Nesse caso, mecanismos de controle alto nível, baseados em aproximação geométrica, são desejáveis para realizar a edição, processamento de movimento e da forma [126]. As estruturas subjacentes devem capturar o espaço global em que a forma animada encontra-se definida e manter-se grosseiro o suficiente para servir como representação intermediária para edição da sequência. Uma série de abordagens tem sido propostas para reconstruir estruturas de controle com base em alguma incorporação espacial 3D baseada em forma, e.g., esqueleto de animação (*animation skeleton*) ou gaiolas de deformação (*deformation cages*), exemplificados na Figura C.18. Animação esqueletal provê naturalmente uma disposição para modelar objetos articulados enquanto que as gaiolas de deformação mostram-se mais adequadas a modelagem de evolução do volume e seu mapeamento adapta-se melhor a geometrias não-tubulares [126].

Figura C.18: Exemplo simples mostrando: (a) uma malha de referência. (b) Uma possível representação de sua estrutura esqueletal. (c) Uma possível representação de gaiolas (neste caso, em conjunto com a estrutura esqueletal). Figuras retiradas do trabalho de Chen et al. [24].

Demonstrando a utilidade destes mecanismos de controle, até a publicação do trabalho de Chen et al. [24] não havia na literatura métodos publicados para a combinação de geometrias e movimentos de duas sequências de malhas diferentes. Chen et al. adotaram então uma representação baseada em gaiola dirigida por armação esqueletal [23] como uma estrutura de controle alto nível para codificar sequências de malhas. A gaiola, agente utilizado para incorporar a geometria, é responsável por re-

produzir a geometria e acelerar o processamento geométrico subsequente. Para lidar uniformemente com diversas estratégias de *morphing*, o *framework* proposto por Chen et al. possui quatro etapas (Figura C.19): representação da sequência de malhas, parametrização cruzada, combinação de movimento e interpolação dinâmica da forma.

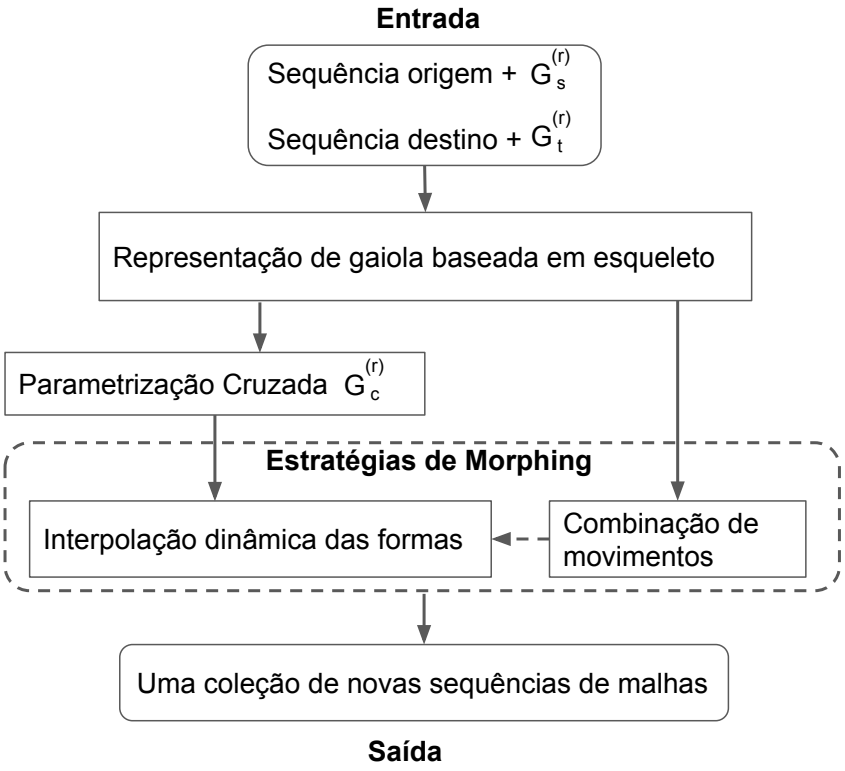

Figura C.19: Fluxograma do framework unificado proposto por Chen et al.[24]. A notação G indica geometria, os sobrescritos r e d referem-se à pose de referência e pose deformada da malha, e os subscritos s, t e c indicam a malha origem, malha destino e malha compatível, respectivamente.

Outra alternativa comum é deformar a superfície das formas diretamente, sem a utilização de estruturas de esqueleto ou ângulos de junção, transferindo a deformação do movimento de uma forma de origem para uma nova forma destino. De fato, no campo de análise estatística de formas, a forma e pose de um corpo geralmente são estudadas como elementos distintos [4], assumindo que mudanças na forma e na pose são independentes.

Algum tempo depois do trabalho de Chen et al. [24], Medalha et al.[90] apresentaram um método alternativo mais simples para *morphing* de malhas dinâmicas totalmente baseado em malha e sem exigir a construção de armação de esqueletos, segmentação de malha ou o uso de qualquer estrutura de controle adicional. Ademais, não é necessário que as duas malhas de entrada compartilhem o mesmo número de vértices ou triângulos ou possuam a mesma conectividade. O pipeline proposto por Medalha et al. é composto resumidamente por dois estágios: parametrização cruzada baseada em *template* e interpolação dinâmica de malhas. Durante a etapa de parametrização cruzada é utilizada uma variante do algoritmo mínimos quadrados (*least-square* - LS) em malhas[119] para fornecer uma aproximação grosseira da geometria da malha origem na malha destino. No método de Medalha et al. os possíveis candidatos a pontos de controle da malha LS são detectados utilizando uma abordagem baseada na assinatura

de *kernel* de calor (HKS)[122]. Em seguida, é realizado um processo iterativo de ajuste fino que adiciona novas restrições ao processamento de malha LS. A parametrização cruzada é executada apenas uma vez para quaisquer dois quadros para estabelecer uma correspondência completa entre vértices das malhas origem e destino. Por fim, utiliza-se tal correspondência no estágio de interpolação dinâmica de malhas para produzir os resultados do *morphing*. A eficácia do método foi apresentada em diversos resultados alcançados pelo trabalho. O principal inconveniente do método proposto por Medalha et al. [90], assim como nos métodos anteriores, é a indicação manual de alguns pares de correspondências de vértices entre as malhas origem e destino para guiar o processo de parametrização.

Poses de uma determinada forma são projetadas para especificamente para a forma em questão, e modelar a pose independentemente da forma ainda é um problema em aberto. Pontos de contatos e mais comumente distância entre partes do corpo têm sido bem exploradas mas outros tipos de restrições podem também ser importantes, e.g., orientação de coordenadas entre as partes do corpo [11]. Redirecionar a animação de uma forma para outra pode assumir classificações e abordagens variadas, dependendo do objetivo, representação suportada, nível de detalhamento e escopo do problema. Resumidamente, as abordagens mais comuns (ou combinação delas) são:

- Redirecionamento de movimento: técnica que lida com o problema de adaptar um movimento animado de uma forma para outra, geralmente explorando e reutilizando dados de captura de movimentos [41, 65].
- Deformação de superfície: métodos de deformação que manipulam diretamente a superfície da forma envolvida.
- Deformação do espaço de incorporação: Um espaço de deformação mapeia um domínio origem para um domínio alvo dentro do espaço Euclidiano ao mesmo tempo que preserva restrições predefinidas.
- Transformação de deformação: método que reutiliza sequências de animações animações existentes de uma forma 3D origem para produzir novas animações. Dada uma forma de origem de referência, uma forma destino de referência e uma forma origem deformada, o objetivo é produzir uma nova forma destino deformada que simula a deformação da forma origem deformada.

Dentre os diversos métodos de deformação de malhas, a deformação no subespaço esqueletal é uma técnica popular e bastante utilizada em aplicações em tempo real. No subespaço de deformação esqueletal, o animador especifica um conjunto de ossos representados como matrizes de transformação afim $\mathbf{M}_k$, que são utilizadas para controle da deformação. Além disso, cada ponto (vértice) da superfície da malha possui um conjunto associado de pesos $\mathbf{w}_k$, para cada osso. Via de regra, estes pesos são translacionalmente invariantes e formam uma partição da unidade (i.e., $\sum_k \mathbf{w}_k = 1$). Dado um conjunto de transformações de ossos, a posição deformada $\hat{\mathbf{v}}$ de um vértice $\mathbf{v}$ da pose canônica (*rest-pose*) é:

$$\hat{\mathbf{v}} = \sum_k \mathbf{w}_k \mathbf{M}_k \mathbf{v} = 1,$$

em que $\mathbf{v}$ é representado na forma homogênea [68].

Tanto na extração como na incorporação do esqueleto deve-se informar quais partes da superfície estarão relacionadas com cada osso. Tal processo é conhecido como *skinning*. Quase todo sistema de deformação de malhas ou deformação do volume pode ser adaptado para uma deformação baseada em esqueleto [8]. Dentre as técnicas utilizadas para tal propósito é possível deformar o subespaço do esqueleto, também conhecido como *linear blend skinning* (LBS) ou *matrix palette skinning*. A LBS é uma técnica de deformação de malhas implementada em quase todo motor 3D moderno, frequentemente utilizada por objetos virtuais movidos por animação esqueletal [58]. Geralmente, trabalhos sobre *skinning* aprimoraram o LBS ao deduzir as articulações das formas a partir de múltiplos exemplos de malhas [64, 132]. Tal prática obtém bons resultados, mas são inadequadas a problemas em que não existe uma sequência de malhas de animação, mas apenas uma única malha disponível para análise.

Outra opção é inferir a articulação utilizando o esqueleto dado como um codificador dos possíveis nós de deformação, não apenas como uma estrutura de animação. Essa abordagem é útil quando não há uma sequência de malhas de animação e pode ser vista em trabalhos como o de Baran et al. [8] ou em programas comerciais como o Maya, o qual atribui pesos baseado na proximidade do vértice ao osso.

Não obstante, em um arcabouço ou aplicação convencional de animação 3D, o usuário deve definir os ossos na personagem manualmente, um processo conhecido como *rigging*. Isso exige colocar as junções dos esqueletos dentro do personagem e especificar quais partes da superfície estão relacionadas com qual osso, ou seja, como os movimentos da forma deformarão sua superfície. O processo é tedioso e faz do processo de animação de personagens mais difícil. A partir disso, diversos trabalhos visaram automatizar o processo de extrair a estrutura esqueletal dos modelos [125, 81, 57, 129]. Avanços na modelagem, deformação, e *rigging* possibilitam a criação de uma forma em determinada pose com relativa simplicidade, porém criar animações com malhas ainda é uma tarefa trabalhosa e que demanda tempo [9].

Recentemente, Liu et al. [83] apresentaram uma nova abordagem de transferência de animação sensível ao contexto, baseando-se em mapeamento harmônico e utilizando a ideia de um grafo de contexto para modelar interações locais entre superfícies da forma origem, a serem preservadas na forma destino, para garantir a fidelidade da pose (figura C.20). Apesar de ser baseado em esqueletos, o trabalho não necessita de uma nova etapa de *rigging* já que a forma manipula as superfícies diretamente, automatizando o processo. Porém, há algumas limitações importantes, as formas de referência (tomadas como base) das formas de origem e destino precisam estar na mesma pose, e possuírem estruturas de esqueleto semelhantes. Além disso, é assumido que há claras correspondências semânticas entre as superfícies dos domínios 3D.

Dados dois conjuntos de formas deformáveis (conjunto origem e conjunto destino) e uma forma origem inédita deformada, a transferência de deformação consiste em produzir deformação realística da forma destino, correspondendo visualmente à forma origem com deformação inédita. Sumner e Popovic [121] propuseram uma alternativa para solução do problema de transferência de animação. Dada uma correspondência entre duas malhas, a técnica proposta copia as deformações dos triângulos da primeira malha para os triângulos da segunda (Figura C.21). Assume-se que a correspondência é literal e portanto partes correspondentes das malhas movem-se de maneira geometricamente idêntica. Apesar da transferência de deformação funcionar bem para formas similares e ser capaz de transferir detalhes sutis de movimento, geralmente é desejável

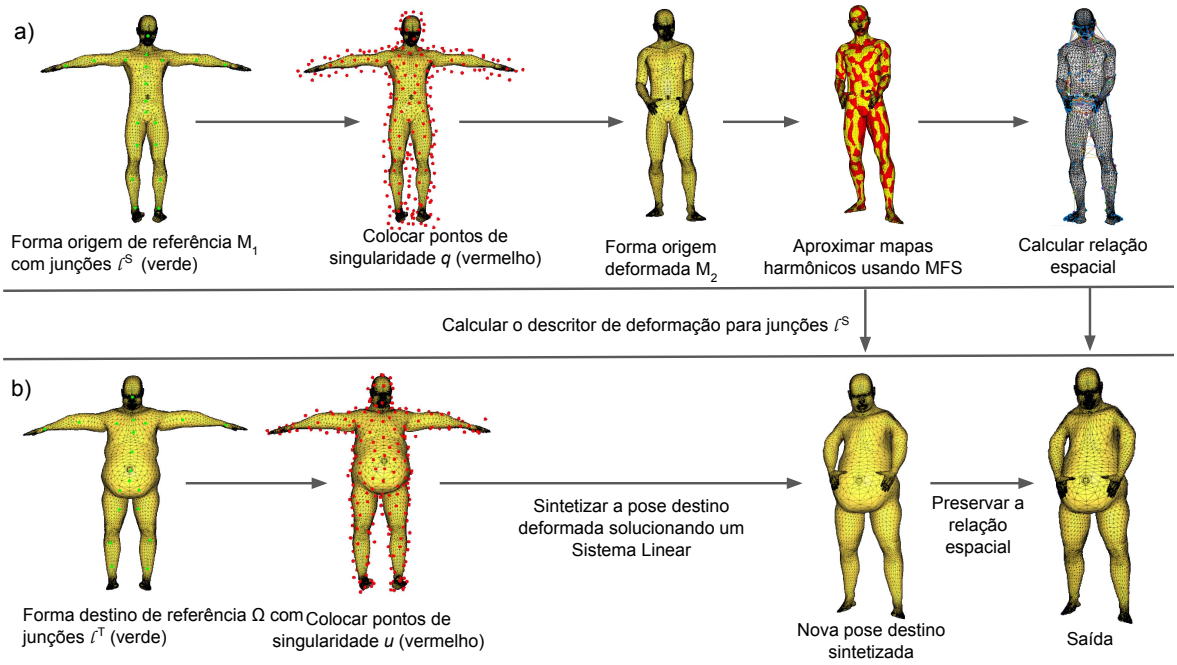

Figura C.20: Visão geral da sintetização de novas poses utilizando mapeamento harmônico (figuras retiradas de [83]). Durante o processo de análise de deformação (a), é calculado o descritor de deformação da pose de referência origem. Durante o processo de síntese de deformação b), são calculados os coeficientes de aproximação do mapeamento harmônico assumindo que a pose de referência destino realiza deformação similar à pose de referência origem. O movimento sintetizado também é restrito (por um arcabouço Laplaciano de deformações) para ter relações espaciais mais similares possíveis àquelas da pose origem deformada.

correspondência semântica [9].

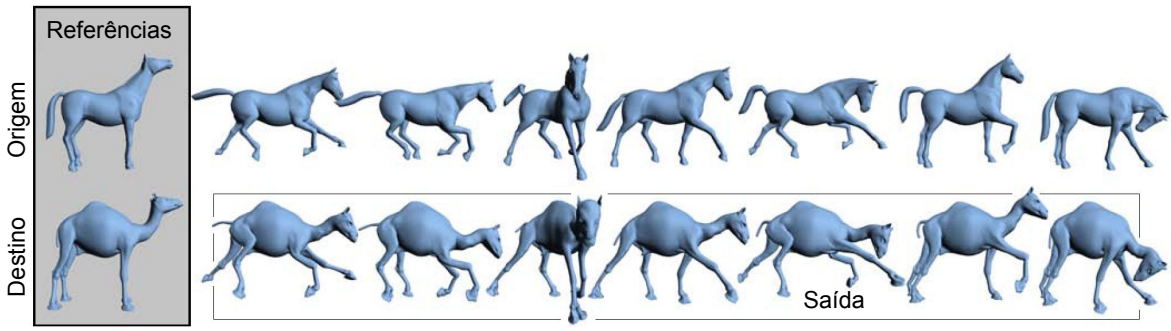

Figura C.21: A transferência de deformação copia as deformações apresentadas por uma malha origem para uma malha destino diferente. Neste exemplo, as deformações da malha do cavalo de referência são transferidas para a malha do camelo de referência, gerando sete novas poses para o camelo. Tanto mudanças esqueletais grosseiras como também deformações de pele sutis são reproduzidas (retirado de [121]).

Todavia, o trabalho de Sumner e Popovic [121] depende de um conjunto de correspondências ponto-a-ponto entre as formas origem e destino (Figura C.22). Em geral, não há métodos automáticos e confiáveis para realizar tal tarefa: portanto os métodos atuais exigem que o usuário especifique um número suficiente de pontos de correspondência iniciais para que seja possível deduzir as correspondências restantes. O processo geralmente baseia-se em tentativa e erro para garantir que os pontos especificados fornecem restrições suficientes [38]. Outro problema característico deste mesmo trabalho é que as deformações de features dos modelos destino podem se misturar com aquelas

do modelo origem quando estes modelos forem diferentes em sua estrutura anatômica.

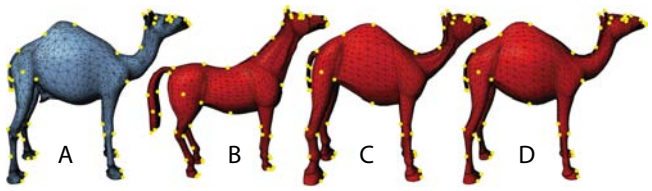

Figura C.22: O algoritmo de correspondência deforma a malha origem na malha destino, controlada pelos pontos de marcação selecionados pelo usuário (mostrados em amarelo). (A) Malha Destino. (B) Malha Origem. (C) Malha origem após a primeira etapa de deformação em que o termo do ponto válido mais próximo é ignorado. (D) Malha final deformada utilizando todos os três termos da função objetiva (retirado de [121]).

Chu et al. [26] estendem o trabalho de Sumner e Popovic [121], adicionando uma nova abordagem baseada em exemplos para mapear as features de deformação dos modelos destino através do uso de alguns exemplos. O autor transfere as poses e detalhes de geometria dos modelos de origem primeiro para o modelo destino e só então ajusta os detalhes de geometria dos modelos alvo transformados através de alguns exemplos alvo especificados pelo usuário.

Para diminuir o esforço do usuário em relacionar pares de pontos de controle entre as malhas origem e destino, Yang et al. [141] desenvolveram um método que automaticamente seleciona um conjunto destes pontos chave na forma origem, porém ainda é necessário que o usuário selecione alguns pontos de controle manualmente na forma destino, correspondentes a pontos predefinidos automaticamente na forma origem.

Se, ao invés de uma sequência de malhas de superfície, os movimentos forem representados primariamente por animação esqueletal, o usuário pode construir um modelo de *skinning* (processo de vincular a malha 3D à configuração de junções criadas) para a malha desejada e utilizar algum método de redirecionamento para esqueleto [31, 51]. Todavia, é importante reconhecer que tal solução é mais complexa e impõe restrições muitas vezes não desejadas aos tipos de transferências que podem ser realizadas.

Sejam animações representadas por uma sequência de malhas indicando a deformação das superfícies ou por uma sequência de posições e orientações de uma estrutura de esqueleto, seria mais produtivo se fosse possível transferir animações - já capturadas por um sistema de captura de movimentos ou mesmo animada manualmente por artistas - para outras formas com geometria ou topologia diferentes dos animados originalmente. Por exemplo, uma vez capturada a sequência de movimentos de uma forma realizando determinado movimento, seria interessante reutilizar esta mesma captura em malhas de triângulos de outras formas sem que fosse necessário ajustar tudo manualmente ou realizar nova captura. Nesse sentido, Baran et al. [8] apresentam um método para animar personagens automaticamente baseando-se em uma animação já capturada. Seu método parte de uma animação de esqueleto já definida e tem como objetivo adaptar o esqueleto à uma malha estática que representa uma nova forma, vinculando também o esqueleto à superfície, permitindo que os dados da animação original animem a nova forma (Figura C.23). Essa abordagem pode ser formulada como um problema de otimização em que deseja-se calcular as posições das junções de tal

modo que o esqueleto resultante ajuste-se dentro da forma da melhor maneira possível, mantendo-se semelhante ao esqueleto de origem. O sistema de animação foi batizado de *Pinocchio* e algumas demonstrações foram criadas com personagens humanoides (Figura C.24). Não obstante, o protótipo não considera o tipo de material, animando materiais teoricamente rígidos ou roupas da mesma maneira, com característica emborrachada. Além disso, há problemas em áreas da superfície mais complexas tais como quadris e regiões do pescoço.

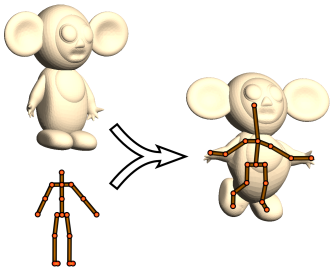

Figura C.23: Neste exemplo, uma malha de triângulos é animada ao incorporar um esqueleto (animação pré-existente) dentro do mesmo, possibilitando aplicar um movimento de caminhar em uma forma que inicialmente era estática (retirado de [8]).

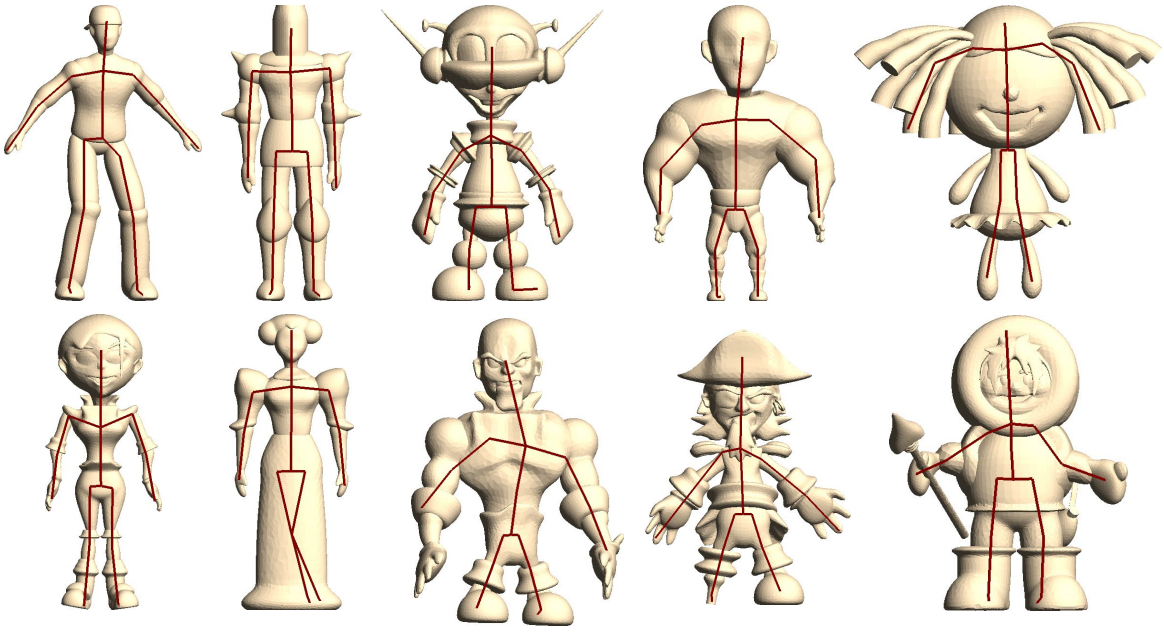

Figura C.24: Alguns resultados de teste para incorporação de esqueleto do sistema de animação *Pinocchio* (retirado de [8]).

Ben-Chen et al. [13] propuseram um novo método de transferência de deformação, que pode ser aplicado à diversas representações de formas — malhas de tetraedros, sopa de polígonos e malhas com múltiplos componentes. A principal característica da abordagem adotada é a deformação do espaço na qual a forma está incorporada. A deformação origem é aproximada por um mapa harmônico utilizando um conjunto de funções harmônicas base. A partir daí, dado um conjunto esparso de pontos de correspondência entre as formas origem e destino selecionados pelo usuário, é gerado uma deformação da forma alvo a qual possui propriedades diferenciais similares às da deformação de origem (Figura C.25). A transformação de deformação é realizada com o auxílio de gaiolas que cercam as formas que serão transferidas. Entretanto,

a construção das gaiolas requer um esforço considerável, além do que, tais métodos podem deformar erroneamente regiões espacialmente adjacentes caso se ajustem na mesma gaiola.

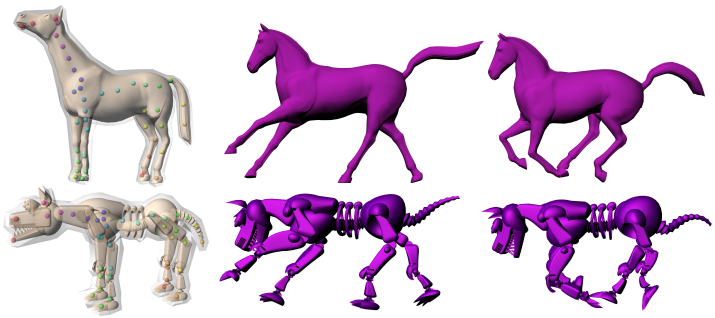

Figura C.25: Transferência de poses de um cavalo galopando para um cão robô, utilizando 40 pontos de correspondência marcados. As poses de referência dentre de suas gaiolas, e as marcações de correspondência são mostradas na coluna mais à esquerda (retirado de [13]).

Na mesma época, Baran et al. [9] propuseram uma abordagem de transferência semântica de deformação baseando-se apenas na malha da superfície, sem a utilização de um esqueleto, transferindo uma deformação de malha já existente de uma forma para outra. O trabalho tenta preservar features semânticas do movimento ao invés de sua deformação literal. A transferência de deformação realiza essa tarefa com um espaço de forma que possibilita interpolação e projeção através da álgebra linear padrão. Dados diversos pares de malhas exemplo, a transferência semântica de deformação deduz uma correspondência entre os espaços de forma dos duas formas. Isso possibilita transferência automática de novas poses e animações. Este método não necessita que o usuário forneça correspondências ponto-a-ponto entre as formas origem e destino, mas exige como entrada pares combinados de malhas origem e destino, assumindo que cada modelo no conjunto origem está semanticamente relacionado à forma correspondente no conjunto destino (Figura C.26). Na prática, caso os conjuntos origem e destino sejam construídos de maneira independente, tal característica provavelmente não será satisfeita [38].

Não obstante, a solução do problema de relacionar malhas origem e destino, depende da natureza das malhas envolvidas e pode tornar-se um problema complexo. Mesmo trabalhos recentes possuem restrições. Azencot el al. [7] propôs a correspondência de formas consistente via otimização acoplada, mas com a limitação de que o método assume que serão fornecidas as correspondências iniciais densas entre as formas envolvidas. Além disso, grande parte dos trabalhos, como no também recente trabalho de Eisenberger et al. [32], requerem que os pares de malhas, apesar de não-rígidas, sejam quase isométricas entre as diferentes poses. Já para abordagens baseadas em dados naturalmente a necessidade de uma base de dados consistente torna-se o primeiro obstáculo.

De fato, o problema de mapeamento entre superfícies tem sido alvo frequente de estudos, sejam eles baseados na geometria [128] ou em abordagens baseadas em conjuntos de dados [139]. Dentre as abordagens baseadas em dados é importante ressaltar também aquelas que surgiram apoiando-se em aprendizagem de máquina, empregando florestas aleatórias [103], redes neurais baseadas em grafos [78] ou redes residuais totalmente conectadas [108]. Boa parte das técnicas do estado da arte dos métodos de transferência de deformação exigem como entrada correspondências ponto

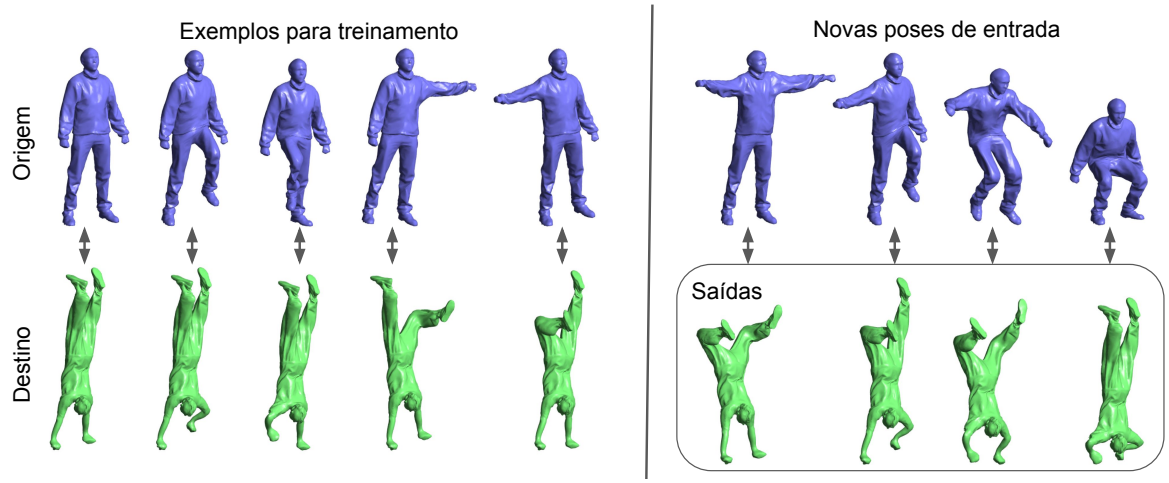

Figura C.26: A transferência de deformação semântica aprende uma correspondência entre poses de duas formas das malhas exemplos e sintetiza novas poses da forma destino a partir das poses da forma origem. Neste exemplo, dadas 5 poses correspondentes de duas formas (esquerda), o sistema cria novas poses da forma destino (baixo, direita) a partir de quatro novas poses da forma origem (cima, direita) (retirado de [9]).

a ponto entre as formas origem e destino, ou pares de formas deformadas origem e destino com deformações correspondentes relacionadas. Na maioria dos casos, tais correspondências não estão disponíveis e não podem ser estabelecidas por um algoritmo automaticamente.

Sendo a deformação de malhas 3D uma representação flexível para representar sequências de animações 3D assim como coleções de objetos da mesma categoria, permitindo diversas formas com deformações não lineares de grande escala, Tan et al. [123] apresentam um dos primeiros métodos de aprendizagem profunda para deformação de formas não rígidas. O autor estuda como analisar a deformação de formas 3D utilizando estas redes neurais profundas. É proposto um novo processo arcabouço chamado auto-codificadores variacionais de malhas (*mesh variational autoencoders*) para explorar o espaço latente probabilístico de superfícies 3D (Figura C.27). O foco principal era produzir um modelo generativo capaz de analisar coleções de modelos e sintetizar novas formas. Para alcançar esse objetivo foi utilizada uma representação de superfícies chamada de RIMD (*Rotation Invariant Mesh Difference*) [36] para representar as deformações, em conjunto com um modelo auto-codificador variacional. Apesar da técnica obter bons resultados em comparação com as técnicas mais atuais da época, o modelo foi pensado para processar somente malhas homogêneas (de mesma topologia [36].

Considerando a maioria dos casos em que não há disponível como entrada as correspondências ou pontos de controle entre formas origem e destino, Gao et al. [38] estudam tornar possível a deformação destas formas, mesmo com esta restrição. A abordagem explora a capacidade de aprendizado das redes neurais profundas para aprender como as formas são deformadas naturalmente a partir de um determinado conjunto, fornece uma métrica diferenciável para medir a similaridade visual a constrói um mapeamento confiável entre os espaços latentes com consistência do ciclo. Segundo os autores, o trabalho foi inspirado pela forma com que seres humanos realizam tal tarefa, observando as forma deformadas para aprender suas features, considerando as similaridades entre as formas origem e destino e refletindo como as formas alvo deveriam ser deformadas para assemelharem-se às formas origem.

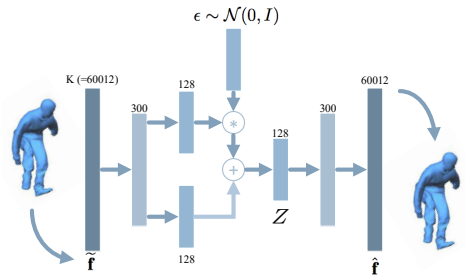

Figura C.27: Pipeline do auto-codificador variacional para malhas (retirado de [123]). A malha da figura possui 2502 vértices e 7500 arestas. Cada **dR** (matriz de diferença de rotação [36]) de uma aresta direcionada possui 3 elementos, e **S** (matriz de escala e cisalhamento [36]) de cada vértice possui 6 elementos, então a dimensão total das features é K = 2502 x 6 x 7500 x 3 x 2 = 60012. Na figura, ε é uma variável aleatória com distribuição Gaussiana com média 0 e variância unitária. O símbolo $\tilde{\mathbf{f}}$ representa as features preprocessadas (neste caso, formato RIMD), Z é o conjunto de features latentes e $\hat{\mathbf{f}}$ é a saída do auto-codificador.

Dado um conjunto de formas origem $\mathcal{S}$ e um conjunto de formas destino $\mathcal{T}$, não relacionados entre si, assim como uma forma origem deformada **s**, o objetivo é produzir uma forma destino deformada **t** que possua deformação visualmente semelhante à deformação de **s**. Ao contrário dos métodos anteriores, não é mandatório ter como entrada correspondências ponto-a-ponto entre as formas origem e destino, nem pares de formas relacionadas. Ao invés disso, os conjuntos das formas origem e destino devem conter diferentes deformações, contanto que em ambos conjuntos as deformações sejam suficientes para cobrir os espaços de deformação relevantes. Isso permite utilizar dois conjuntos de formas deformadas independentes.

Para lidar com grandes deformações, Gao et al representa cada forma deformada utilizando uma representação de deformação de forma chamada ACAP (*as consistent as possible*) [37]. A representação baseia-se em gradientes de deformação, amplamente utilizados em modelagem geométrica. É importante lembrar que, individualmente dentro de seus próprios conjuntos (formas origem e destino), as formas devem possuir mesma conectividade.

Resumidamente, a rede proposta por Gao et al. [38] é composta por 3 componentes: VAE convolucional para codificar as formas no espaço latente, SimNet para calcular a similaridade visual entre duas formas (dos conjuntos $\mathcal{S}$ e $\mathcal{T}$), ambas em seus respectivos espaços latentes, e CycleGAN para transferência de deformação.

É essencial também garantir a similaridade visual entre as formas origem e destino, mas métricas de similaridade bidimensionais não podem ser generalizadas e aplicadas diretamente em domínios de formas 3D. Diante disso, o autor emprega uma técnica chamada de distância do campo de luz (LFD, *light field distance*), proposta inicialmente por Chen et al. [22], para medir a similaridade visual. Nessa abordagem, uma forma 3D é projetada em múltiplas visualizações e features são calculadas baseadas nas imagens projetadas por essas visualizações. Tomando como exemplo a Figura C.28, em que (a) e (c) são dois diferentes modelos de avião com rotações inconsistentes:

- Para o avião da Figura C.28(a), são colocadas câmeras de um campo de luz em uma esfera ao redor do modelo. A Figura C.28(b) mostra onde câmeras são colocadas nos pontos de interseção da esfera.

- As câmeras deste campo de luz do avião (a), podem ser posicionadas, nas mesmas

posições para o modelo de avião da Figura C.28(c), ilustrado em (d).

- Ao somar as similaridades de todos os pares de imagens correspondentes na Figura C.28(b) e (d), a similaridade geral entre os dois modelos 3D é obtida.
- Em seguida, o sistema de câmeras da Figura C.28(d) pode ser rotacionado para uma orientação diferente, assim como ilustrado na da Figura C.28(e), o que leva a outros valores de similaridade entre os dois modelos.
- Após estimar os valores de similaridade, a orientação correspondente correta pode ser encontrada, na qual os dois modelos possuem a maior similaridade de todos os ângulos de visualização, Figura C.28(f). A similaridade entre dois modelos é definida somando-se a similaridade a partir de todas as imagens correspondentes entre a Figura C.28(b) e (f).

Entretanto, calcular as similaridades para todas as possíveis rotações seria impraticável. Daí, as posições das câmeras de um campo de luz são distribuídas uniformemente em vértices de um dodecaedro regular, de forma que as posições reduzidas das visualizações são utilizadas para aproximação.

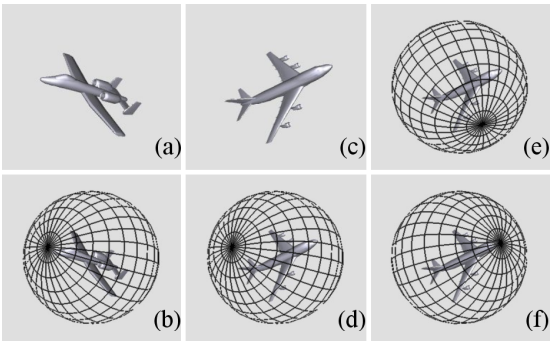

Figura C.28: A ideia geral da medição de similaridade entre 2 modelos 3D utilizando LFD (retirado de [22]).

Chen et al. [22] posicionam câmeras dos campos de luz em 20 vértices de dodecaedro regular, i.e., 20 diferentes visualizações de um modelo, distribuídos uniformemente. Todavia, 20 visualizações representam um modelo 3D apenas de forma grosseira. Diante disso, todas as luzes são desligadas, de modo que as imagens renderizadas correspondam apenas às silhuetas dos modelos, o que melhora a eficiência e robustez da métrica em imagens. Ademais, é utilizada uma projeção ortogonal para acelerar o processo de busca e reduzir a quantidade de features. Por consequência, 10 silhuetas diferentes são produzidas para cada modelo 3D, já que as silhuetas projetadas de dois vértices opostos em um dodecaedro são idênticas. A Figura C.29 mostra um exemplo típico de 10 silhuetas de um modelo 3D.

Um exemplo é ilustrado na Figura C.29: A comparação é realizada entre dois modelos de malhas (vaca e porco) em orientações arbitrárias (a). Primeiro, 20 imagens são renderizadas a partir dos vértices do dodecaedro de ambos os modelos. São comparadas todas as imagens correspondentes aos mesmos ângulos de visualização (b), como, a ordem de 1 até 5 entre os modelos porco e vaca. São extraídos então os valores de similaridade sob esta rotação do sistema de câmeras. Em seguida, a ordem de 1 até 5 é mapeada de maneira diferente como em (d), e são extraídos outros valores de

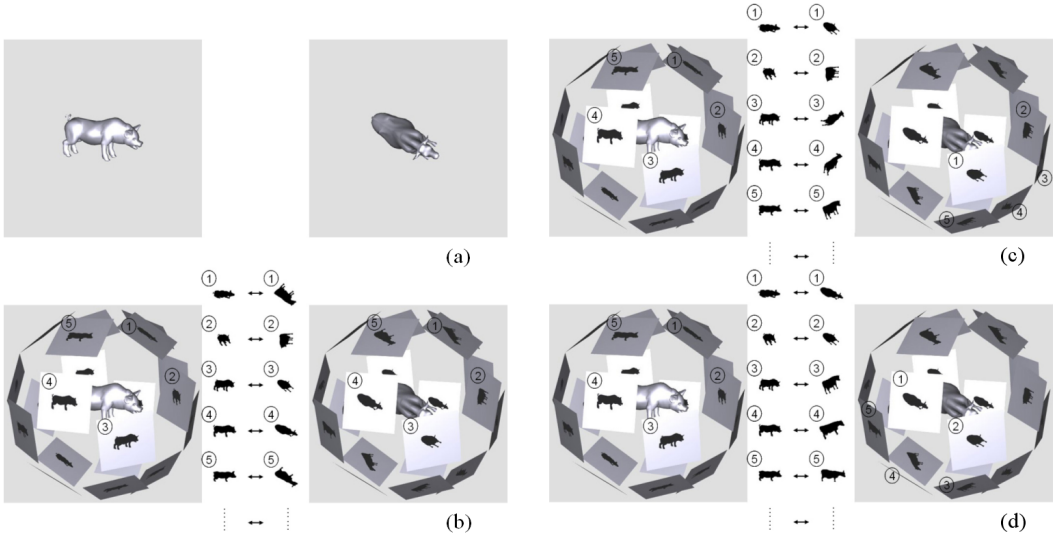

Figura C.29: Comparando descritores de campos de luz entre dois modelos 3D

similaridade. Ao repetir o processo, encontra-se uma rotação de posições de câmera com a melhor similaridade (correlação sendo a maior), como mostrado em (d). Portanto, a similaridade entre os dois modelos é o somatório das similaridades entre todas as imagens correspondentes. O restante do processo pode ser encontrado no trabalho de Chen at al. [22], mas não será detalhado por não fazer parte do foco principal deste trabalho, estando mais relacionado à area de processamento de imagens.

A principal desvantagem do método é que a medida atual de similaridade visual funciona bem quando a deformação é visualmente similar, i.e., pode não funcionar bem para formas semanticamente similares mas visualmente muito diferentes. Dentre outras desvantagens pode-se citar o fato de haver etapas pré-processamento pesadas para cada par de forma origem e destino e do fato da técnica não considerar contatos e colisões entre as superfícies [11].

Na mesma linha de pensamento que o trabalho de Gao et al. [35], considerando que para a maioria de tarefas de modelagem, especialmente aquelas envolvendo modelos tridimensionais, não é comum encontrar dados com pares com correspondências preexistentes, Yin et al. [145] apresentam uma rede neural profunda, chamada LOGAN (*latent overcomplete GAN*), que aprende transformações de formas de propósito geral a partir de domínios não pareados (não relacionados par-a-par). Comparando especificamente com GAO et al. [35], cujo trabalho corresponde a transformação da forma preservando a pose, em que conjuntos de malhas origem e destino representam diferentes poses com mesma conectividade dentro do seu respectivo conjunto, LOGAN é projetada para ser uma rede de transformação de propósito geral para formas representadas como nuvem de pontos onde é permitida uma variação geométrica e topológica maior entre as malhas dos conjuntos de origem e destino. A rede proposta é treinada com dois conjuntos de formas representados como nuvem de pontos, sem pareamento ou pontos de correspondência entre as formas. A arquitetura da rede é composta por um auto-codificador que codifica as formas para um espaço latente, resultando em uma representação chamada pelo autor de representação sobre-completa. O tradutor é baseado em uma rede do tipo adversária generativa (GAN, *generative adversarial network*), que opera no espaço latente, em que um cálculo do erro adversário impõe a tradução entre domínios enquanto que um cálculo de erro de preservação de features garante

que as características desejadas da forma sejam preservadas para uma transformação natural. Depois de treinada, a rede toma como entrada uma forma no formato de nuvem de pontos em um domínio e a traduz para outro. A rede é treinada sobre dois conjuntos de formas, cada forma representada utilizando uma nuvem de pontos como na PointNet++ de Qi et al. [97]. Um vez treinada, a rede toma uma forma de um dos domínios e transforma-a para outro domínio.

Sem correspondências entre as formas, um dos desafios é como normalizar as formas apropriadamente, as relacionando. Ao invés de trabalhar com as formas diretamente, opta-se por é traduzi-las para um espaço latente comum compartilhado pelos domínios origem e destino. O espaço latente é obtido por uma rede auto-codificadora treinada antes de transformar as formas (Figura C.30(a)).

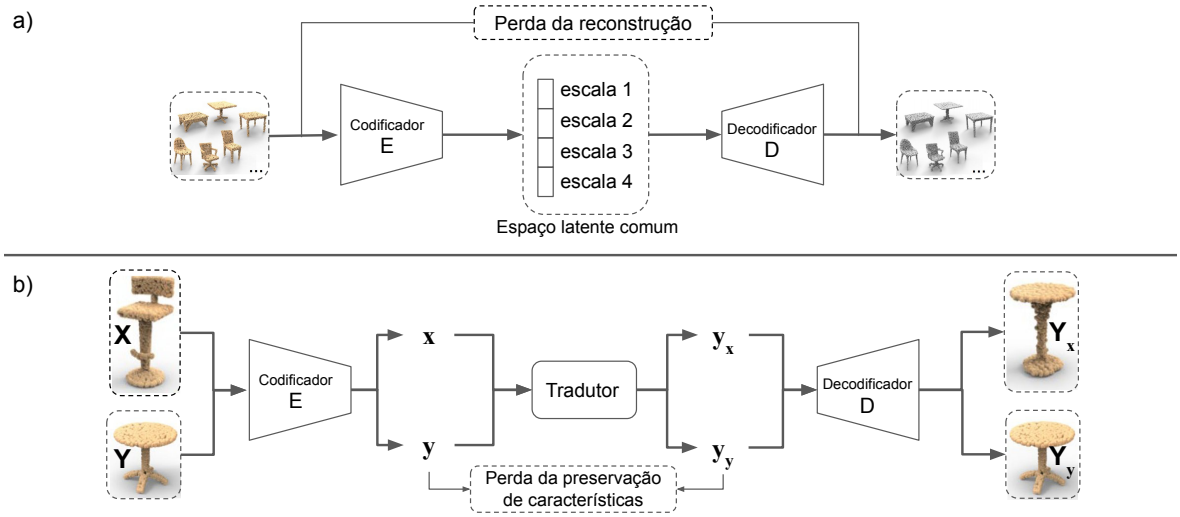

Figura C.30: Visão geral da arquitetura da rede LOGAN, composta de um auto-codificador (a) para codificar as formas de dois domínios de entrada para um espaço latente comum sobre-completo, e um tradutor baseado em GAN (b) projetado com uma perda adversária e uma perda para reforçar a preservação de features.

Ao transformar formas de camelos para cavalos, por exemplo, uma transformação apropriada não deve ser de um camelo específico para um cavalo qualquer, mas sim para um cavalo que claramente seja "derivado" daquele camelo em particular. Ainda, algumas features do camelo que também são comuns aos cavalos devem ser preservadas durante a transformação enquanto que outras podem ser alteradas. Trata-se de um problema chave: quais features devem ser preservadas/alteradas é desconhecido - depende dos domínios das formas e neste caso a rede deve aprender sem supervisão. Para tentar solucionar este problema, a arquitetura de Yin et al. [145] apresentam duas partes principais:

- O auto-codificador codifica as features da forma em múltiplas escalas, o que é comum em redes neurais convolucionais (CNN). Entretanto, ao contrário de métodos convencionais que agregam as features multi-escala (e.g., na PointNet++ [97]), as features multi-escala são concatenadas para produzir um código latente que é "sobre-completo". Especificamente, a forma de entrada pode ser reconstruída utilizando apenas partes do código correspondente a diferentes escalas de features. A intuição é que ao realizar as transformações da forma no espaço latente formado por tais códigos sobre-completos, em que os códigos das

features multi-escala estão separados, facilitaria um desembaraçamento implícito das features preservadas e alteradas.

- Em um tradutor de formas de cadeira para formas de mesa, por exemplo, o treinamento não é somente para transformar um código de cadeira em um código de mesa, mas também para transformar um código de mesa no mesmo código de mesa (Figura C.30(b)). A perda dessa transformação, chamada de perda de preservação de features, ajudaria o tradutor a preservar features da mesa (no caso da entrada ser um código de cadeira) durante a transformação cadeira para mesa.

Não obstante, a tradução da forma entre domínios não é adequada para todos os pares de domínios, e.g., entre uma cadeira e um avião ou um corpo humano e uma face. Parte-se de uma suposição implícita que as formas de ambos domínios deveriam compartilhar de alguns pontos em comum. Em geral, estes pontos em comum podem ser latentes [145]. Além disso, devido à natureza da representação das nuvens de pontos, as formas geradas não são necessariamente claras e compactas. Os pontos podem estar espalhados ao redor dos locais desejados, especialmente em partes estreitas dos modelos. Por fim, a realização de traduções em um espaço comum e a mensuração do erro em relação à preservação das features uma a uma implica também na suposição de que há um alinhamento da escala entre as formas de entrada, i.e., as features comuns a serem preservadas devem estar na mesma escala.

Considerando contatos e inter-colisões entre as superfícies de uma forma 3D ao deformá-la, Basset et al. [11] investigam se a transferência de forma ao invés da transferência de pose preserva melhor o significado contextual original da pose de origem. O autor propõe um método de otimização que deforma a forma origem com pose utilizando três principais funções de energia: similaridade em relação à forma destino, preservação do volume do corpo, e gerenciamento de colisões (preservando contatos existentes e prevenindo penetração entre partes da superfície). Mas, apesar do método não utilizar esqueleto, ainda é necessário uma segmentação das partes do corpo da forma, realizada manualmente. Ainda, o método parte da hipótese de que as deformações de poses humanas são aproximadamente isométricas.

Ao comparar técnicas de *morphing* com deformação de malhas dinâmicas, há alguns pontos em comum. Técnicas de transferência de deformação precisam otimizar simultaneamente dois objetivos concorrentes. Um deles é o alinhamento entre as formas origem e destino, i.e., definindo correspondências entre as posições e componentes das formas ao deformar uma forma para uma pose diferente. O segundo objetivo é alcançar métricas de qualidade, como minimização da distorção e preservação de features geométricas locais. De certa forma, estes dois objetivos são contraditórios, já que alinhar perfeitamente a forma origem à uma forma destino impede a preservação original de detalhes da geometria da origem. Considerando tal problema, Yifan et al. [144] propõem uma arquitetura de aprendizagem para deformação de formas preservando detalhes originais da forma origem. O objetivo é deformar uma forma origem de modo que corresponda à estrutura geral de uma forma alvo, ao mesmo tempo preservando detalhes da superfície da origem, como exemplificado na Figura C.31. O método estende uma técnica de deformação tradicional baseada em gaiolas, em que a forma origem é envolvida por uma malha de controle grosseira (gaiola), e as translações determinadas nos vértices da gaiola podem ser interpoladas à qualquer ponto da malha origem por

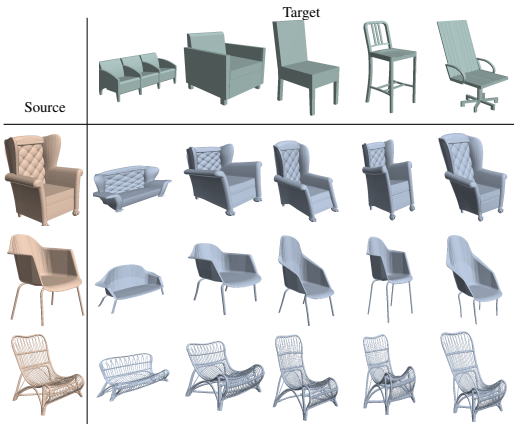

Figura C.31: Sintetização de variações das formas origem (*source*, em marrom), deformando-as para corresponder aos destinos (*target*, em verde) (retirado de [144].

funções ponderadas especiais. A arquitetura faz uso de duas redes neurais, uma para predizer a gaiola e outra para deduzir a deformação da gaiola (Figura C.32). O cálculo dos pesos da gaiola e a deformação baseada na gaiola são implementadas como camadas diferenciáveis da rede.

Em determinadas técnicas baseadas em aprendizado de features de formas 3D (e.g., [43, 133]), as deformações previstas corrompe certas features e exibe distorções, principalmente em áreas com estruturas finas, muito detalhadas ou com discrepâncias grosseiras entre a forma origem e destino. Estes artefatos surgem devido limitações inerentes às redes neurais, atualmente, para captar, preservar, e gerar altas frequências. Um obstáculo adicional é o fato do conjunto de deformações que preservam features é intratável e varia de forma para forma, dificultando o aprendizado. Yifan et al. [144] propõem uma técnica clássica de processamento geométrico chamada de deformação baseada em gaiola (*cage-based deformation*), ou simplesmente *CBD*. Ao invés de definir a deformação unicamente sobre uma superfície $\mathcal{S}$, a deformações espaciais abrangem todo o ambiente do espaço no qual a forma $\mathcal{S}$ está incorporada. Uma CBD controla essa deformação através de uma malha de triângulos grosseira, chamada de gaiola $\mathcal{C}$, que geralmente inclui $\mathcal{S}$. Dada a gaiola, qual quer ponto no espaço ambiente $\mathbf{p} \in \mathbb{R}^3$ é codificado via coordenadas baricêntricas generalizadas, como uma média ponderada dos vértices $\mathbf{v}_j$ da gaiola: $\mathbf{p} = \sum \Phi_j^C(\mathbf{p})\mathbf{v}_j$ em que as funções de ponderação $\{\Phi_j^C\}$ dependem da posição relativa de $\mathbf{p}$ (em relação ao sistema de coordenadas global) aos vértices $\{\mathbf{v}_j\}$. A deformação de qualquer ponto no espaço ambiente é obtida deslocando os vértices da gaiola e interpolando suas novas posições $\mathbf{v}_j^{'}$ com os pesos pre-calculados:

$$\mathbf{p}^{'} = \sum_{0 \leq j < |\mathcal{V}_c|} \Phi_j^C(\mathbf{p})\mathbf{v}_j^{'}. \tag{C.1}$$

Na CBD, para obter as funções de ponderação $\{\Phi_j^C\}$ podem ser adotadas diversas formulações como interpolação, precisão linear ou minimização da suavidade e distorção por exemplo. Yifan et al. [144] optaram por utilizar coordenadas do valor médio (MVC) pela preservação de features e propriedades de interpolação, assim como diferenciabilidade em relação às coordenadas das gaiolas deformadas e da forma origem, permitindo seu uso como uma camada diferenciável da rede neural. Uma das limitações do método é que ele depende muito da predição da gaiola da forma origem, a deformação deduzida pode não ser ótima se houver interseção da gaiola com a forma ou a gaiola não encapsular de forma justa a região a ser deformada.

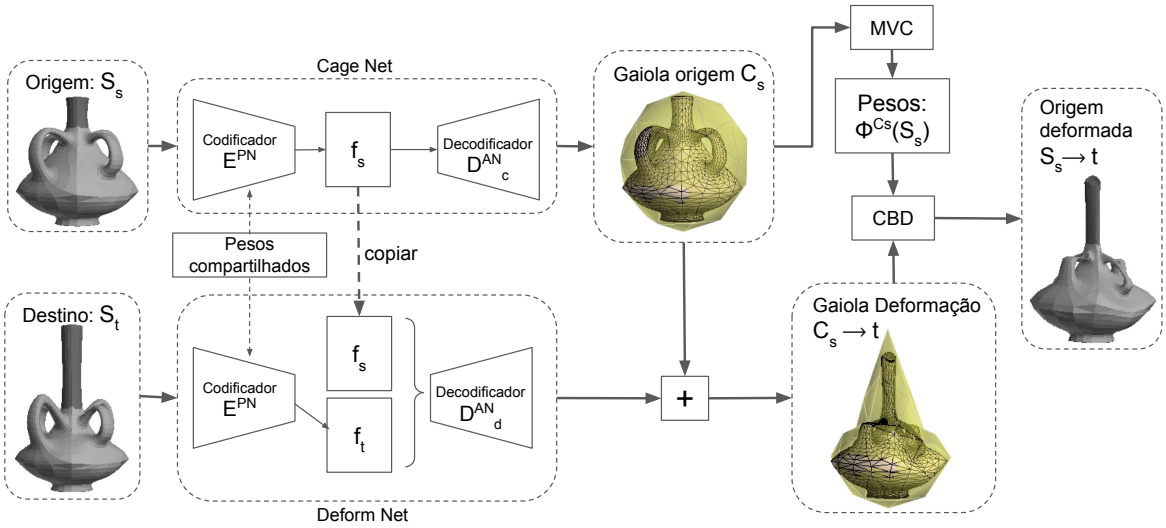

Figura C.32: Visão geral da abordagem de aprendizado por gaiolas [144]. Uma forma origem S_s e uma destino S_t são codificadas pela mesma rede codificadora de pontos E^{PN} em códigos latentes f_s e f_t, respectivamente. O código da forma origem é decodificado por um decodificador do tipo AtlasNet [45] D_c^{AN} para uma gaiola C_s no módulo de predição de gaiola (*Cage Net*). Os códigos das formas origem e destino também são concatenados e decodificados por um decodificador D_d^{AN} para gerar o deslocamento da deformação da gaiola no módulo de predição de deformação (*Deform Net*). O deslocamento é adicionado à C_s para calcular a deformação da gaiola $C_s \rightarrow t$ que deformará a origem no destino. Dada uma gaiola e uma forma origem, a camada MVC calcula o valor médio das coordenadas $\Phi^{C_s}(S_s)$. Tais coordenadas e a deformação da gaiola $C_s \rightarrow t$ são utilizadas pela camada de deformação baseada em gaiola (*CBD*) para produzir uma gaiola origem deformada $S_s \rightarrow t$.

Na mesma linha de problemas, tentando deduzir deformações 3D, Wang et al. [133] propõem uma rede de ponta-a-ponta chamada de 3DN, transforma o modelo de origem sendo influenciado por um modelo alvo desejado. Diferente da maioria dos trabalhos, o modelo alvo pode ser uma imagem 2D, malha 3D, ou uma nuvem de pontos. Dada uma malha qualquer e um modelo alvo, a rede estima o vetores de deslocamento dos vértices (deslocamentos 3D) para deformar o modelo origem ao mesmo tempo que mantém a conectividade de sua malha.

Em seu arcabouço, Wang et al. [133] adotam uma representação intermediária das malhas origem e destino estruturada como nuvens de pontos. Especificamente, é amostrado um conjunto de pontos da malha origem deformada e do modelo alvo e calculada a perda entre os conjuntos de pontos resultantes. Essa abordagem tem por objetivo minimizar os erros no cálculo entre similaridade entre as malhas, que podem ser de densidades variadas entre os diferentes modelos. Ao manter a topologia da malha origem inalterada e preservar propriedades como por exemplo simetrias, o autor foi capaz de gerar deformações de malhas plausíveis.

C.4 Investigação de Soluções nos Domínios Espectral e Espacial

C.4.1 Features Espectrais

Dos trabalhos que obtiveram os melhores resultados em relação a correspondência entre formas utilizando representações baseadas nos Operadores de Laplace-Beltrami, tomamos como base das implementações a proposta de Li et al. [71], que obteve resultados do estado da arte em *benchmarks* de correspondência entre formas como base das avaliações.

Realizamos experimentos da efetividade das técnicas propostas quando aplicadas sobre formas de conjuntos de dados de bases distintas. Assim, como grande parte dos trabalhos anteriores ao de Li et al., a representação espectral utiliza kerneis baseados no Laplaciano das formas, mais especificamente, Operadores de Laplace-Beltrami anisotrópicos e filtros espectrais representados por polinômios de Chebyshev agregam features locais dos sinais através de convoluções.

A opção pelo trabalho de Li et al. foi devido aos excepcionais resultados relatados nos *benchmarks*. Nele, os datasets são formados de diversas animações de um conjunto de formas, i.e., várias deformações de um conjunto de formas da mesma categoria. Tal fato foi confirmado nos experimentos realizados com datasets informados pelos respectivos autores. O segundo passo consistiu em verificar a eficácia do modelo em conjuntos de dados com características diferentes dos selecionados. Em um primeiro momento, utilizando o aplicativo Daz3d [29] criamos uma base de dados de 100 formas humanas masculinas em poses arbitrárias e biotipos diversos. Todas as formas criadas possuem o mesmo número de vértices e arestas, variando apenas na geometria. Ao aplicar o algoritmo de Li et al. [71] sobre esta base, obteve-se resultados pouco acima de 98% de acerto, confirmando a efetividade do método para este tipo de base de dados. O conjunto de testes foi formado por 10 modelos inéditos de mesma natureza. Em um segundo passo, com a mesma ferramenta, foram criadas 15 formas humanas femininas em poses variadas, com parâmetros de complexidade baseadas no mesmo modelo humano.

Com a rede já treinada com modelos de formas humanas masculinas, o objetivo foi apurar o comportamento da rede treinada utilizando modelos de formas femininas sintetizados pela mesma metodologia do conjunto original de treinamento. Todavia, topologicamente os modelos femininos não possuem exatamente o mesmo número de vértices e arestas do que os modelos humanos de treinamento. Portanto, não é possível utilizar a mesma metodologia empregada para medir conjuntos com mesma topologia, em que o mapeamento dava-se na proporção 1:1 e a porcentagem de acerto era calculada de acordo com a quantidade de pares de vértices com correta correspondência. Diante disso, a seguinte estratégia foi adotada para comparação dos resultados do mapeamento entre estas malhas com deformações não isométricas:

- O modelo da rede foi treinado sobre um conjunto composto de 100 modelos de formas humanas masculinas variadas, mas com mesma topologia, em poses distintas. A rede foi treinada por 100 épocas e os hiper-parâmetros adotados forma os mesmos do trabalho de Li et al. [71]. Para modelos de n vértices, a rede fornece como saída n correspondências.

- Toma-se qualquer uma das malhas do treinamento como modelo de referência (todas possuirão o mesmo mapeamento e o mesmo número de vértices). Utilizando o modelo RGB em que cada componente da cor, com valores variando de 0 a 255, à esse modelo de referência são atribuídas cores aos vértices, seguindo um dos eixos de coordenadas globais da orientação do modelo — a escolha do eixo é parametrizável. O resultado final é um modelo tonalizado com uma faixa n de cores RGB variando ao longo de um dos eixos indicados.

- Em seguida, o modelo alvo, composto por m vértices — $m \leq n$ — passa pela avaliação da rede. A saída são correspondências em que cada um dos n vértices dos modelos treinados são mapeados à algum dos m vértices do modelo alvo.

- As cores RGB definidas para o modelo de referência do treinamento são passadas para o modelo alvo, i.e., à cada um dos n vértices dos modelos de treinamento foi atribuída uma cor e após avaliação do modelo alvo também foi definida uma correspondência. As cores dos m vértices do modelo alvo serão as cores dos vértices correspondentes nos modelos de treinamento.

- Um modelo de referência do conjunto de treinamento e o modelo alvo são renderizados lado a lado em uma interface interativa em que é possível rotacionar, ampliar ou reduzir a imagem gerada. Para que o mapeamento seja considerado correto a cor da região desejada no modelo de referência deve ser a mesma cor do modelo alvo C.33.

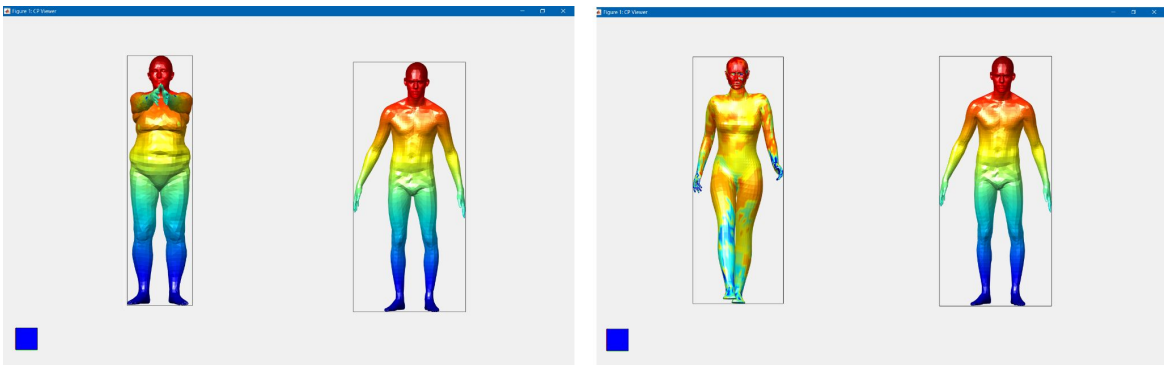

Figura C.33: Resultado da correspondência de pares de formas avaliada visualmente entre modelos de topologias distintas. A figura mais à esquerda mostra a avaliação de duas formas com deformações aproximadamente isométricas, de um mesmo conjuntos de dados (FAUST [15]). A figura mais à direita mostra a comparação entre 2 formas de conjuntos de dados distintos (SCAPE [5] x FAUST).

Como esperado, encontramos limitações quando um dos modelos alvo não pertence ao mesmo grupo de treinamento, mesmo possuindo o mesma conectividade dos modelos treinados. Para modelos da mesma categoria, mas de conjuntos de dados distintos, o método não se mostra eficaz C.34. Apesar de captar a natureza das superfícies dos modelos pertencentes ao conjunto de treinamento, a abordagem por si só não foi capaz de encontrar correspondências entre formas parecidas de uma mesma classe. Naturalmente, não é adequada para formas de classes distintas, como as de modelos utilizadas em *morphing*, ou mesmo formas da mesma categoria mas com complexidades diferentes das utilizadas pelo conjunto de treinamento. É importante notar que o trabalho de Li et al. [71] não tinha como objetivo a comparação entre formas arbitrárias,

mas sim a correspondência entre formas aproximadamente isométricas. Todavia, é importante relatar a incapacidade do método lidar com a tarefa desejada neste trabalho.

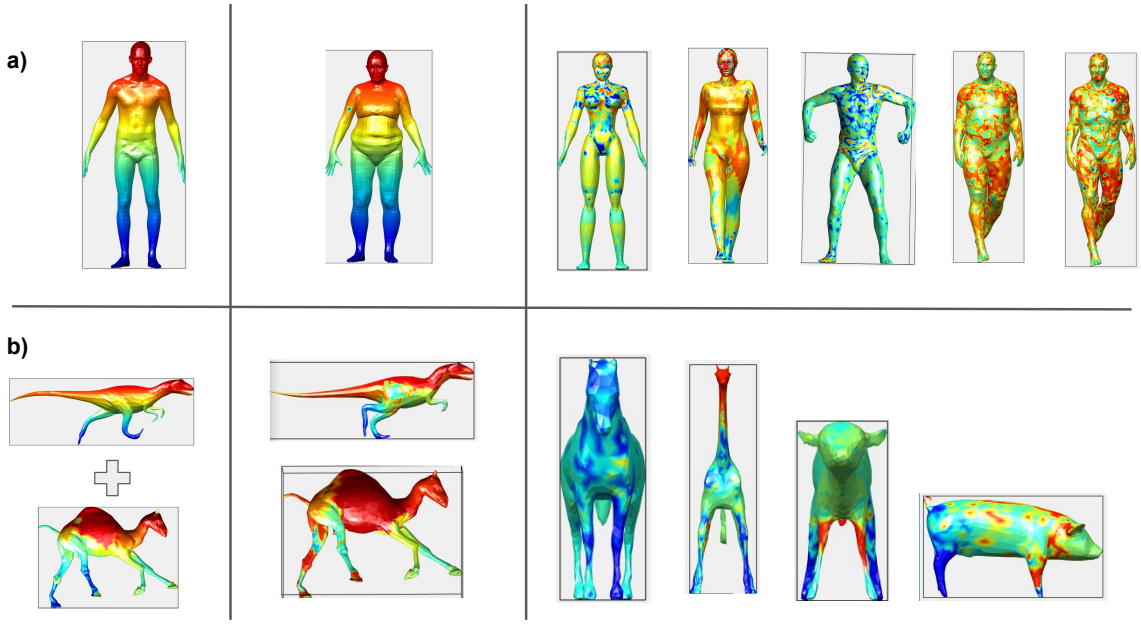

Figura C.34: A figura ilustra resultados obtidos pela ACSCNN treinada durante 100 épocas. A primeira coluna ilustra uma figura dos conjuntos de dados isométricos utilizados pelo treinamento onde as cores indicarão correspondência entre o modelo de treinamento e os modelos testados, cores iguais indicam correspondências. A coluna central ilustra visualmente o resultado das correspondências com malhas inéditas, mas aproximadamente isométricas; neste caso, os acertos superaram 98%. A coluna da direita ilustra resultados das correspondências com malhas em deformações não isométricas.

C.4.2 Features Espaciais

Ao comparar duas formas variadas, seja para classificação, segmentação semântica ou correspondência, de modo geral utilizar somente informações das features locais ao redor de pontos de uma superfície não se mostram eficientes. Como alternativa, alguns trabalhos [113, 124, 93] propuseram extrair, em diversas escalas, propriedades geométricas relacionadas ao volume, ou que sejam capazes de representar relações hierárquicas entre os retalhos de da superfície das formas [56]. Tais features podem ser utilizadas para tarefas de processamento geométrico, mas isoladamente não foram capazes de efetuar correspondências entre formas cujas deformações não são isométricas.

Sabendo disso, paralelamente a investigação das técnicas que propõem representações espectrais, também verificamos até onde seria possível efetuar a correspondência entre formas através de features extraídas analiticamente, i.e., através de uma solução axiomática, mesmo sabendo que isso isoladamente é insuficiente para solução do nosso problema. A grosso modo, investigamos se ao segmentar formas seria viável descobrir correspondências das partes segmentadas para obter critério adicional de comparação e alinhamento e comparação das formas completas.

O primeiro desafio ao lidar com segmentação foi encontrar ou definir as bases de dados adequadas. Objetivando um estudo de caso baseando-se em formas que também poderiam ser utilizadas no processo de *morphing* de animais, encontramos a base de

dados com malhas tridimensionais de animais com classificação de segmentos PSB ou *Princeton Segmentation Benchmark* [25]. Tal conjunto provê dados para análise quantitativa de como as pessoas decompõem objetos em partes. Permite também a avaliação de algoritmos de segmentação automática de malhas. A definição dos segmentos foi criada manualmente por um grupo de oitenta pessoas, que segmentaram as malhas em partes avaliadas como funcionais. Embora possua 19 categorias, para o propósito pretendido optou-se pela utilização de classes de animais e humanoides.

Mesmo assim, o dataset não pôde ser integralmente utilizado. Composto por poucos modelos de cada categoria (20), a classificação dos segmentos era ambígua para nosso propósito, e.g., patas traseiras e dianteiras de ambos os lados eram simplesmente classificadas como patas, sem maior especificidade. Ainda, diversas formas de mesma classe possuem uma quantidade distinta de segmentos e alguns segmentos não eram desconectado uns dos outros.

Minimizamos parte dos problemas implementando soluções simples: identificar entre os modelos, aqueles sem separação entre membros, separação das faces pertencentes a cada membro, classificação de membros entre direita/esquerda, trás/frente, identificação de modelos com quantidade de segmentos diferentes e normalização dos tipos de segmentos. Por exemplo, classes em que segmentos de um chifre ou orelha de um animal era classificado separadamente da cabeça foram conectados; patas foram classificadas como traseira, dianteira, esquerda e direita. Mesmo assim a quantidade de exemplos era de 16 malhas por categoria. Inicialmente, selecionamos 4 categorias: pássaros, humanos, urso de pelúcia (*teddy*) e quadrúpedes. Os testes relatados a seguir foram executados majoritariamente sobre a categoria de quadrúpedes.

Tomando como base o trabalho de Kalogerakis et al. [56], calculamos um conjunto de reduzido de propriedades geométricas da superfície das formas por segmento. Para cada vértice de cada forma, são geradas features relacionadas à curvatura; features estas já conhecidas em trabalhos de correspondência parcial e formas 3D (e.g., variação média da superfície, curvaturas principais, média curvatura, curvatura gaussiana, diferença de curvatura [34]), features extraídas por PCA (e.g., covariância local dos centros das faces), média e mediana SDF(*shape diameter function*) [113], distância da superfície medial [82] e distância geodésica média [49] entre todas as faces. Estas features descrevem discretamente alguns dados relevantes da curvatura, volume e formato das formas. As features dependentes da extensão da superfície (curvatura e PCA) foram calculadas em 5 escalas diferentes, selecionando retalhos cúbicos em 5 raios geodésicos diferentes ao redor de cada face, sendo os raios geodésicos de 5%, 10%, 20%, 30%, 50%. Por fim, cada conjunto de features foi normalizado por *standardization*, i.e., normalizados pela transformação das features subtraindo o valor da média e dividindo pelo desvio padrão.

Considerando que cada característica é definida por face, em diversas escalas e combinações, é notável a grande quantidade de dados a serem processados para malhas compostas por uma grande quantidade de triângulos. Aplicamos então algumas técnicas simples para reduzir a quantidade de dados e verificar, se o conjunto reduzido ao menos pode indicar uma correlação entre segmentos. A partir daí, em caso positivo, um conjunto de dados maior poderia ser considerado como parte da entrada de uma rede neural para solucionar um problema maior. Devido a variedade de classes de formas, decidimos por não utilizar média ou mediana de valores de um conjunto de triângulos. Ao invés disso, selecionamos um subconjunto de triângulos distribuídos ao

longo de cada segmento (Figura C.35), mais especificamente, pontos nas extremidades e no meio de cada segmento, semelhantes a pontos de um esqueleto [129], considerando 3 pontos por segmento. Os pontos são definidos comparando a distância geodésica média de cada triângulo (face) do segmento em comparação com as distâncias na superfície completa; define-se faces posicionadas mais nas extremidades e uma face posicionada mais ao meio, entre estas extremidades. Os pontos utilizados nos experimentos são os centroides das faces selecionadas.

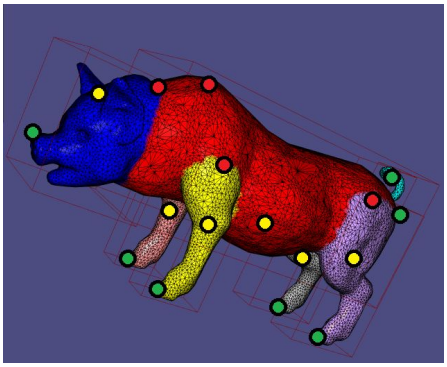

Figura C.35: Ilustração da seleção de 3 faces representativas para cada segmento de uma forma. Círculos verdes marcam um triângulo candidato a ponto da extremidade fechada do segmento; círculos amarelos mostram um triângulo candidato a ponto central do segmento; e pontos vermelhos denotam um possível candidato a ponto da extremidade aberta do segmento.

Estes poucos pontos, mesmo representando uma aproximação grosseira e limitada da geometria de cada segmento, permitiram uma investigação inicial, na qual empregamos o *algoritmo húngaro*. Resumidamente, trata-se de um algoritmo combinatorial simples de otimização que soluciona um problema de atribuição em tempo polinomial, neste caso, para corresponder segmentos únicos de uma forma A com segmentos únicos de uma forma B, através de um grafo bipartido totalmente conectado de segmentos de A e B, em que o peso de cada aresta é calculado pela distância Euclidiana entre as features dos segmentos C.36. Quanto menor a diferença entre as distâncias, maior a probabilidade de corresponder ao mesmo segmento. Baseando-se nesses valores, o algoritmo soluciona um problema de otimização em que cada segmento de uma partição do grafo deve ligar-se a outro segmento da outra partição, com custo geral mínimo.

Durante os experimentos, foi possível perceber que o algoritmo húngaro, aplicado sobre todas as combinações de pares de formas de uma determinada classe, identifica com maior precisão correspondências de alguns segmentos com geometria mais similar entre classes de formas semelhantes, como cabeças e troncos. No caso de membros, há identificação parcial, mas a simplificada do algoritmo ou a falta de dados mais refinados não permitiu identificar a à qual lado do plano sagital o segmento pertence C.37. A porcentagem de acerto obtida nas 4 categorias testadas é exibida na TabelaC.1.

Quadrúpedes	47%
Pássaros	65%
Humanos	51%
Teddy	66%

Tabela C.1: Resultado algoritmo Húngaro 16 features por classe de segmento

No intuito de apenas a influência das features na identificação dos membros, adicionamos uma heurística ao algoritmo, dividindo-o em 2 etapas:

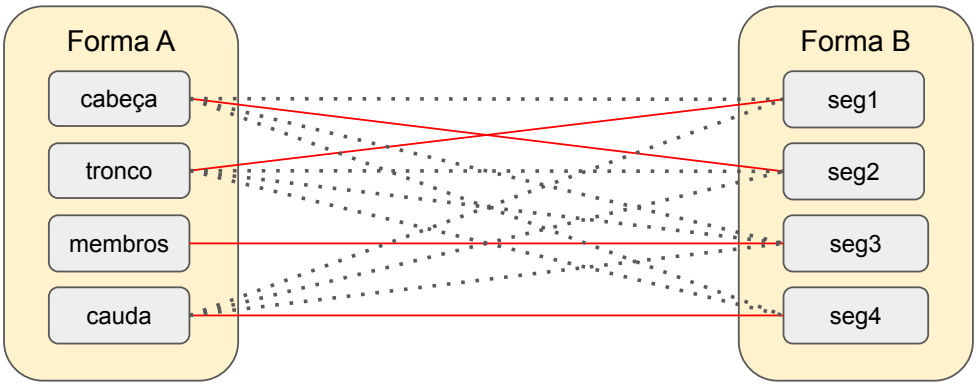

Figura C.36: Ilustração da correspondência desejada pelo algoritmo húngaro para segmentos de 2 formas. A Forma A (esquerda) possui os segmentos definidos cabeça,tronco,membros e cauda, enquanto a Forma B representa uma forma segmentada sem classificação das classes de segmentos. O algoritmo cria um grafo bipartido onde cada segmento da parte A é ligado à todos segmentos da parte B. O peso das arestas é definido pela distância euclidiana entre os valores das features de cada segmento. Cada elemento de uma partição deve ligar-se à algum elemento da outra partição obedecendo a combinação de custo geral mínimo.

1. Executar o algoritmo húngaro sobre a todos os segmentos de um par de formas, fixar o resultado da classificação de cabeça e tronco, ignorando o restante. Naturalmente, considera-se que todas as formas tratadas possuam pelo menos cabeça e tronco. Definidos os segmentos que representam o tronco e a cabeça, adiciona-se as seguintes features à já existentes: ângulo entre cada triângulo do segmento e a cabeça, ângulo entre triângulo e o tronco, distância Euclidiana entre triângulo e cabeça, e distância Euclidiana entre triângulo e tronco.

2. Executar o algoritmo húngaro novamente, sem previamente identificar cabeça e tronco, mas com novas features.

Neste teste a porcentagem de acertos melhorou em 2 das 4 categorias (Tabela C.2). Durante a depuração, ao renderizar os vetores resultantes das novas features assim como o centro da forma e dos segmentos identificados como tronco e cabeça, foi possível observar que nas 2 categorias em que o algoritmo não foi mais eficiente o mesmo não foi capaz de definir, somente com os dados informados, a orientação correta destas categorias, fato já esperado, dada a diferença anatômica entre as 4 diferentes classes que levaram a ambiguidades na comparação C.38.

Quadrúpedes	47%	63%
Pássaros	65%	60%
Humanos	51%	52%
Teddy	66%	86%

Tabela C.2: Resultado algoritmo Húngaro considerando heurística sobre troncos e cabeças

Outrossim, na heurística utilizada foi adotado um viés (definir dois segmento específicos primeiro: tronco e cabeça). Na tentativa de demonstrar que tais heurísticas poderiam ser aprendidas sem a introdução desse viés ou indicação direta, decidimos por passar os dados das features por uma rede neural profunda.

A rede neural escolhida como base para a abordagem foi a PointNet [98], caracterizada no Capítulo 2. PointNet é uma rede conhecida pela classificação e segmentação

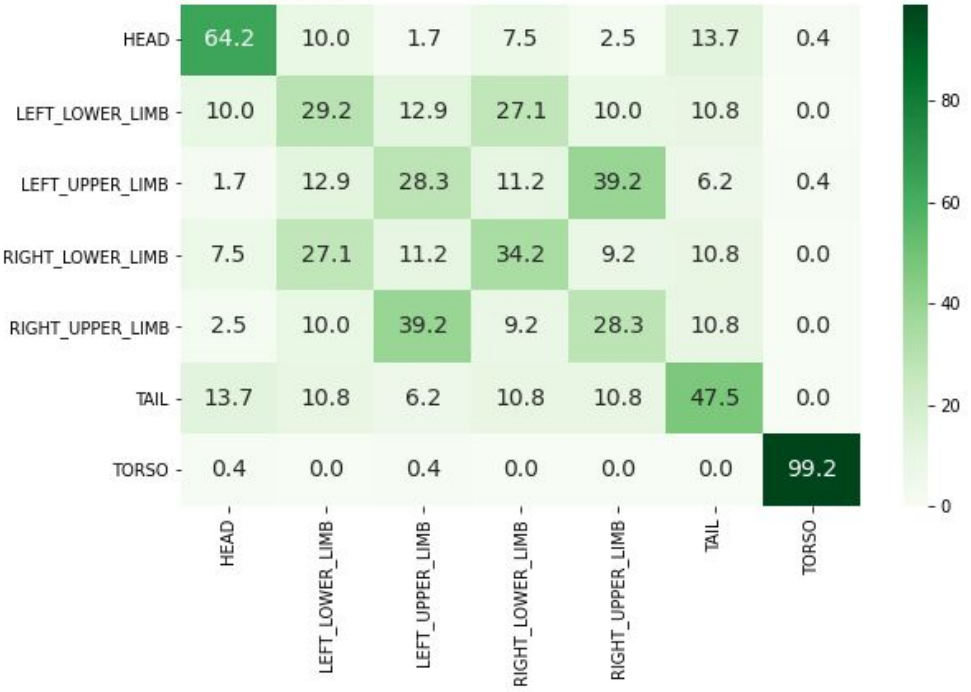

Figura C.37: Matriz de Confusão: Quadrúpedes 16 features para cada segmento, índice geral de acerto de 47.3%. É possível identificar uma dificuldade em identificar o lado dos membros em relação a esquerda/direita, enquanto que troncos e cabeças são identificados com maior precisão. Na tabela, o par linha/coluna contém números que representam a porcentagem total de segmentos de uma classe (linha) que foram relacionados ao outra classe (coluna), portanto trata-se de uma matriz simétrica.

de modelos no formato de nuvem de pontos. O propósito inicial da rede no contexto da validação desejada foi classificar e segmentos. A classificação do segmento implicaria na correspondências com outros segmentos de mesma classificação. Ainda, para o problema principal de correspondência entre formas, a obtenção da classificação dos segmentos atuaria como uma subdivisão da tarefa, onde cada segmento só poderia corresponder a outro do mesmo tipo, restringindo a possibilidade de correspondência para uma região específica da malha.

Diferentemente da PointNet original, que classificava categorias de formas, o objetivo aqui foi tentar classificar segmentos de diversas formas, inclusive pertencente à classes distintas, e.g., segmentos de cabeça, membros, cauda e tronco de malhas de classes de animais quadrúpedes variados. Todos os conjuntos de dados utilizados até o momento passaram por adaptações para tornarem-se adequados à rede e problema específico tratado. As malhas foram normalizadas em escala e posicionadas no centro do sistema de coordenadas. Ao contrário do algoritmo húngaro, que operava sobre dados invariantes em relação a posição ou rotação das malhas, a PointNet opera sobre a posição de pontos tridimensionais que definem a geometria da forma. Além disso, a topologia das malhas originais é formadas por triângulos e faces, obrigando-as a passar por uma etapa de transformação das malhas de triângulos em nuvem de pontos. Para isso, implementamos uma rotina que recebe como entrada uma malha de triângulos e gera como saída uma nuvem de pontos, com p pontos gerados por uma função de aleatoriedade ponderada pela área de cada face, resultando em uma representação em nuvem de pontos com pontos distribuídos com uma distribuição aproximadamente uniforme sobre a superfície da forma. Foram efetuados testes com nuvens de pontos com

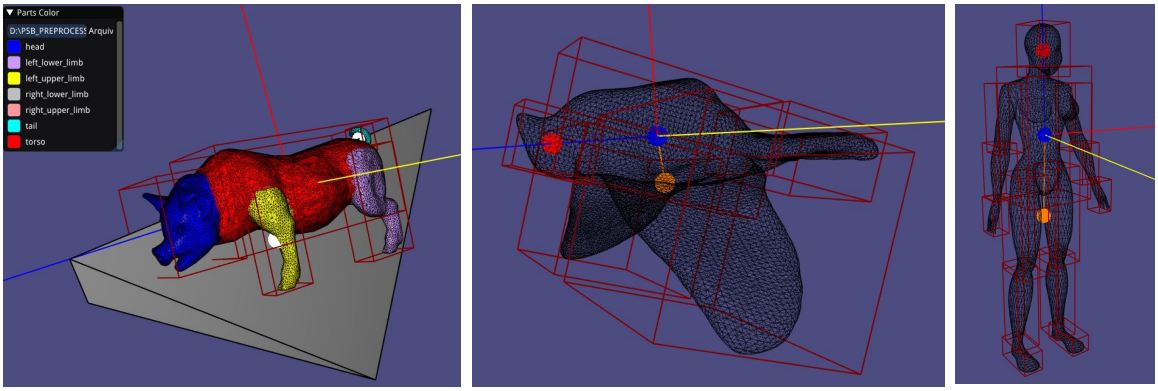

Figura C.38: Considerando heurística aplicada, foi possível detectar alguns problemas: nas imagens, a linha azul de cada uma representa um vetor que passa pelo centro do tronco e pela cabeça detectada, servindo como referência para orientar verticalmente a malha. O vetor laranja passa pelo centro do corpo e centro do conjunto de membros detectados, com propósito de identificar orientação dos membros em relação ao corpo. O vetor amarelo é o produto vetorial dos vetores vermelho e laranja, i.e., a normal do plano formado pelos vetores vermelho e laranja e deveria identificar uma direção lateral da forma. Por fim, o vetor vermelho é o produto vetorial dos vetores azul e amarelo, identificando orientação vertical da forma. Observa-se que a heurística dos vetores funciona no quadrúpede (esquerda da imagem), mas falha nas outras duas. Na imagem central, o algoritmo geometricamente identifica, incorretamente, o pássaro como uma forma cujas asas são os membros e posiciona de cabeça para baixo. Já no caso de formas humanas (à direita), devido as posições e orientações dos membros em relação ao corpo, o algoritmo não foi capaz de detectar onde é a frente ou lateral da forma.

2000, 4000 e 8000 pontos, sem diferença relevante entre os resultados.

As nuvens de pontos foram geradas para classes de quadrúpedes de variadas classes de formas: conjunto composto por 20 formas com 5 segmentos por forma. Todo ponto foi relacionado à um dos segmentos, de acordo com a forma original. Ao passar os segmentos pela rede, para classificá-los, obtivemos uma taxa de acerto de aproximadamente 49%. O resultado relativamente baixo de acertos de certa forma esperado, pois a maioria dos segmentos correspondentes possuem geometrias bem distintas e muitas vezes ambíguas (membros, por exemplo); além disso, rede utiliza apenas a posição espacial dos pontos para aprender cada classificação. Além disso, o projeto original da rede era a classificação geral das formas, não a classificação de segmentos bem diferentes.

Em seguida, efetuamos algumas alterações para incluir as features utilizadas no algoritmo húngaro também na rede. Por tratar-se de uma grande quantidade de propriedades geométricas, optamos por tomar como entrada da rede as faces selecionadas anteriormente (Figura C.35), que remetem à um esqueleto da forma, mas desta vez para um conjunto de dados de entrada composto por todas essas chaves da forma, sem separação por segmento, totalizando 3 pontos por segmentos. A expectativa foi que a rede tomaria como entrada estas faces, mais especificamente, a posição do centro das faces, e retornaria como saída a qual classe de segmento cada ponto tem maior probabilidade de pertencer (classificação de cada ponto).

As novas features foram introduzidas como nova entrada, em um etapa intermediária da rede, após transformações afins da posição dos pontos na rede original, como ilustrado na Figura C.39. As posições espaciais continuam sendo entrada no início do processo, mas as novas features não representam posições no espaço e portanto não faz sentido passá-las por etapas de transformação afim da PointNet. As novas features passam por uma camada totalmente conectada, em seguida são imersas em um

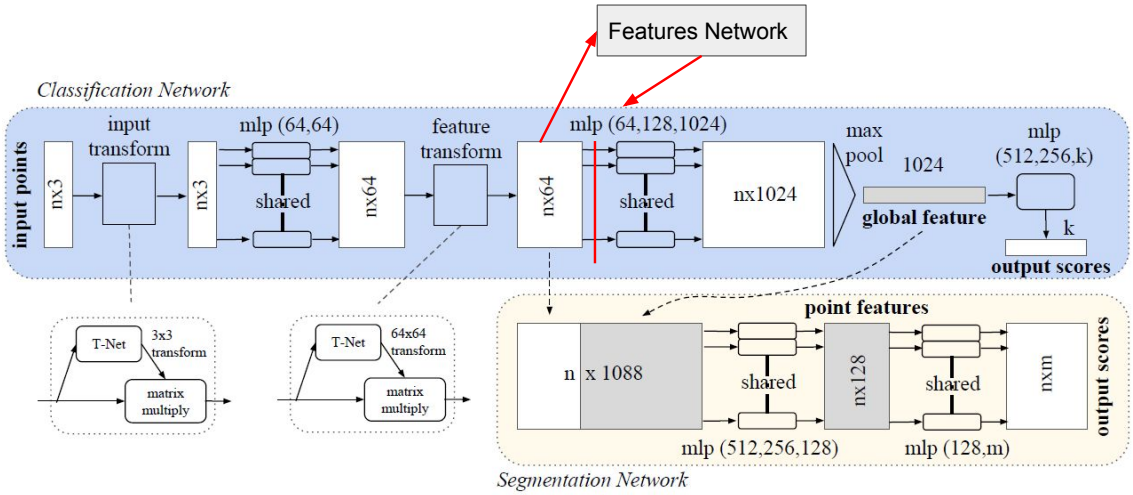

Figura C.39: A imagem mostra a arquitetura original da PointNet [98] indicando o local da introdução de uma subrede que trata as novas features, identificada na imagem como *Features Network*. Fonte da figura original: [98].

espaço de 64 dimensões (Figura C.40). O resultado é concatenado com 64 dimensões resultantes do aprendizado das posições das faces. As 128 features passam por outra camada totalmente conectada e então são imersas em 64 dimensões novamente, a partir daí, estas 64 features seguem pelo fluxo original da PointNet. A dimensão da imersão das novas features foi definida empiricamente; já os outros hiper-parâmetros são os mesmos do trabalho original de Qi el al. [98]. A precisão foi computada como a porcentagem de pontos que corresponderam à classe correta após o treinamento. O conjunto de testes foi definido como 20% das amostras do conjunto total. Os resultados obtidos, assim como a comparação com os experimentos anteriores são mostrados na Tabela C.3. É possível observar que houve melhora na quantidade de acertos na classe de quadrúpedes. Também foram realizados os mesmos testes com as formas humanas, mas analisando os erros obtidos na classificação das mesmas, nota-se que as poses humanas e as features fornecidas pelo conjunto de formas utilizado não possuem variação suficiente para que a rede fosse capaz de aprender significativamente a classificação dos membros.

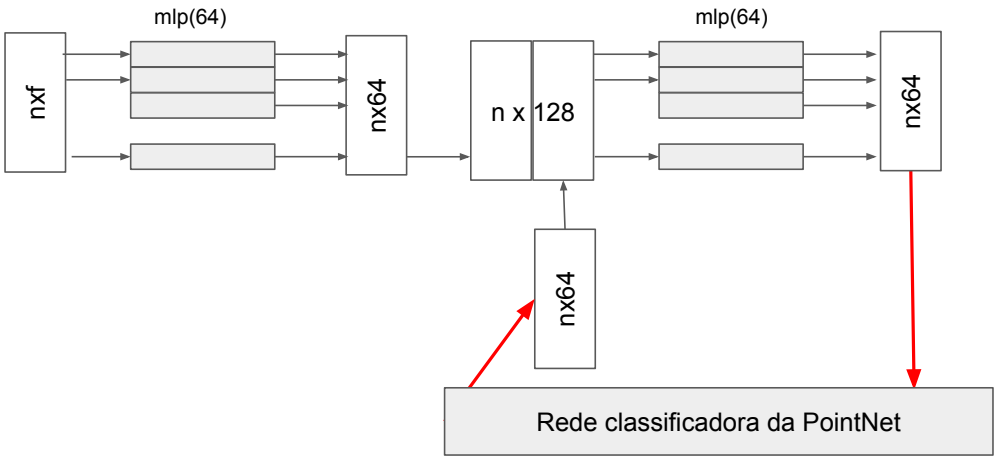

Figura C.40: Detalhe da rede identificada na Figura C.39 como *Features Network*, em que n é o número de vértices (ou pontos) da malha 3D e f equivale a quantidade de features novas, neste trabalho são 22.

Quadrúpedes	47%	63%	65%	48,51%	53,21%

Tabela C.3: Comparação experimentos algoritmo húngaro e PointNet modificada

Diante dos resultados obtidos, nota-se que as features extraídas das formas 3D axiomaticamente podem contribuir para a correspondência entre partes de formas 3D, ao serem parte do processo de aprendizado. Em contrapartida, os conjuntos de dados utilizados nos experimentos apresentaram limitações que afetam diretamente os resultados obtidos e também a utilização das mesmas ideias em redes mais robustas. As principais limitações percebidas foram:

- A quantidade de formas utilizadas nos testes foi relativamente pequena (entre 16 a 20 formas por categoria).
- Presença de malhas com inconsistências geométricas ou topológicas, podendo causar inconsistências ou impedir a geração de determinadas features.
- A separação semântica de segmentos do conjunto de dados utilizado (PSB) foi realizada manualmente por usuários. Mesmo visualmente, é possível notar imprecisão e irregularidade na fronteira de separação entre as partes. Apesar de existir na literatura atual diversos métodos de segmentação [139, 54, 136, 56, 116, 55], na maioria dos trabalhos ainda existe um componente subjetivo na definição semântica dos dados de entrada.

Outrossim, bases de dados compostas por malhas tridimensionais de animais ou formas orgânicas de livre acesso mostraram-se escassas na internet, com exceção de datasets composto exclusivamente de formas humanas. Tratando-se de base de dados de mesma natureza e com a segmentação de partes validada, a quantidade de modelos é ainda menor. Ponderando as dificuldades encontradas diante da adaptação de novas bases de dados, que neste caso envolveria todo um novo trabalho de pesquisa relacionado a efetividade e execução de segmentações, optamos por utilizar como parte da entrada das redes em desenvolvimento os principais dados geométricos das formas (SDF, distância geodésica, curvatura principal, dentre outros) utilizados como entrada para solução de problemas de segmentação [56, 113], possibilitando a utilização destes dados pelas redes, mas sem segmentar a forma previamente.

C.4.3 Investigação da Combinação de Domínios Espectrais e Espaciais

Adotamos uma combinação da filtragem espacial e espectral (Figura C.41) como objetivo de avaliar se a combinação dos dois tipos de features podem indicar parte da solução do problema de correspondência aqui discutido. Uma parte da rede proposta é responsável por realizar convoluções em features extraídas do domínio espacial e gera uma determinada pontuação considerando apenas as características posicionais. Outra parte da rede utiliza filtros de *Chebyshev* para convolução sobre features obtidas em domínios espectrais baseadas no operador de Laplace-Beltrami anisotrópico, considerando para pontuação características intrínsecas extraídas das superfícies em suas representações espectrais. Ambos tipos de features são combinados para tentar obter um ranking de correspondência entre vértices das formas origem e destino.

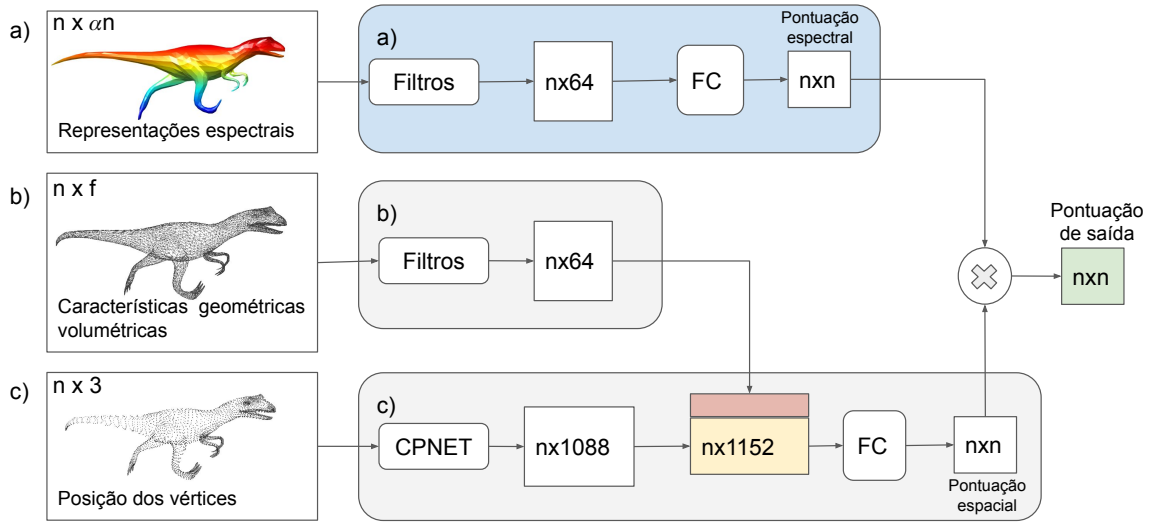

Figura C.41: A rede proposta toma como entrada três conjuntos de features pre-calculadas para cada malha de entrada. Na imagem, n é o número de vértices de cada malha, f é o número de features do domínio espacial, em nossos experimentos foram 421. (a) Utiliza ALBO como kernels de difusão anisotrópica em α ângulos de rotação em relação a curvatura principal da superfície, resultando em n pontuações para cada vértice. (b) Utiliza features extraídas de diversas propriedades geométricas, aprendendo 64 features para cada vértice. Estas features serão utilizadas em c) para compor a pontuação espacial. (c) A posição espacial dos vértices é utilizada como entrada para uma versão modificada da PointNet [98], que combinada com os resultados de (b) resulta em n pontuações para cada vértice. As pontuações espaciais e espectrais são combinadas para gerar o ranking total de correspondência entre vértices, de onde os vértices selecionados podem ser obtidos por uma função *softmax*

Para o treinamento estão disponíveis algumas sequências de animações de malhas de triângulos com número de vértices bem distinto entre os elementos do conjunto. Tais malhas são as mesmas utilizadas pelo trabalho de Medalha et al. [90]. Como resultado do trabalho citado, estão disponíveis correspondências densas entre alguns pares destas formas, que por sua vez, representam formas de classes bem distintas. A disponibilidade destas correspondências permite a realização de um treinamento supervisionado ou semi-supervisionado com o objetivo de aprender, mesmo que aproximadamente, novas correspondências sem uma indicação inicial dos pontos pelo usuário.

A correspondência absoluta, i.e. gabarito, é definida em relação à uma forma de referência ou origem. Cada vértice da forma destino possui um correspondente, geralmente não exclusivo, na forma origem, i.e., a forma destino pode ter uma quantidade de vértices maior ou menor que a forma origem. Tomando como exemplo uma forma origem O com n_vo vértices e uma forma destino D com n_vd vértices, a rede neural fornecerá como saída um vetor de tamanho n_vd em que cada índice corresponde à um vértice da forma D e o conteúdo de cada índice indica o índice do vértice correspondente na forma O.

Filtragem Espectral

Entre as abordagens espectrais que solucionam problemas de correspondência, o trabalho de Li et al. [71] apresentou resultados do estado da arte em diversos *benchmarks* de correspondência de formas, todavia, como já relatado, a abordagem funciona bem para conjuntos de formas pertencentes ao mesmo dataset, composto por formas

cujas deformações são isométricas ou aproximadamente isométricas. Para correspondências semânticas, entre conjuntos de dados distintos ou entre categorias distintas (e.g. dois quadrúpedes de espécies bem distintas) não apresenta bons resultados. De fato, entre estas classes de formas, as features locais das superfícies ao redor de cada ponto podem ser bem variadas. Além disso, o trabalho encontra uma correspondência rígida entre formas — cada vértice da forma origem é mapeado à um único vértice da forma destino — condição geralmente não satisfeita por malhas utilizadas nos trabalhos de *morphing*, i.e., malhas origem e destino com topologia distintas.

Dados os filtros desejados, a convolução em variedades podem ser aplicadas através de multiplicação no domínio espectral. Operadores tradicionais de convolução em variedades baseados na auto-decomposição do Operador de Laplace-Beltrami são homogeneamente dispersos pelas superfícies, insensíveis a informações intrínsecas de direção na forma. Embora não diretamente aplicável ao problema tratado nesse trabalho, a proposta de Li et al. [71] identifica features locais das superfícies das formas utilizando ALBO. ALBO é definida ao considerar mudanças na velocidade de difusão ao longo das curvaturas principais da superfície da forma [3]. Li et al. estende a informação ao considerar múltiplos ângulos anisotrópicos tomando a direção da curvatura principal como referência, mantendo o ALBO intrínseco a forma. As informações adicionais podem desempenhar um importante papel em aplicações que necessitam de uma descrição geométrica de alta qualidade [71].

Não obstante, no experimento a seguir, a informação espectral utilizando o ALBO é empregada. Tais informações utilizadas como entrada da parte da rede responsável pelo aprendizado da pontuação espectral, são calculadas em uma etapa de pré-processamento, quase de forma idêntica ao trabalho de Li et al. [71].

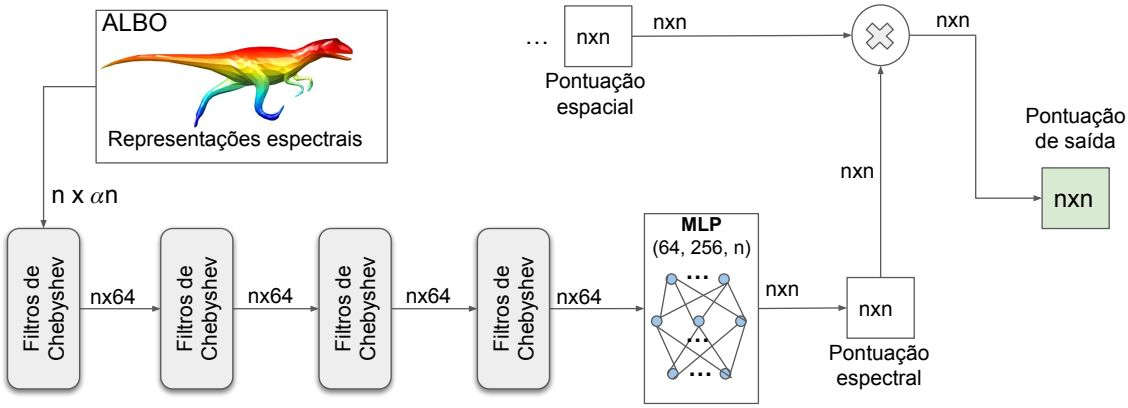

Figura C.42: A representação do ALBO da forma em α ângulos de rotação alimenta a rede que aplica filtros de *Chebyshev* para extrair features relevantes ao problema. Ao final, a pontuação obtida por essa rede é multiplicada pela informação espacial obtida por outra parte da rede, resultando na pontuação geral, utilizada para aprender a correspondência.

Filtragem Espacial

Diversos trabalhos [113, 124, 93] propuseram extrair, em diversas escalas, features com propriedades relacionadas ao volume ou que sejam capazes de representar relações hierárquicas entre os retalhos de da superfície das formas [56]. Porém, tais

features podem ser utilizadas para tarefas de classificação, comparação ou segmentação de formas, mas isoladamente não são capazes de realizar correspondências entre formas cujas deformações são não isométricas.

Algumas dessas features são calculadas e utilizadas como parte da entrada de da rede experimental adotada. O objetivo é utilizar parte da rede para calcular um ranking de probabilidades de cada ponto pertencer à uma determinada região da forma baseando-se nas features representando propriedades volumétricas e contexto dos retalhos da forma. Este ranking faz o papel de um operador de convolução aplicado sobre os resultados de outra parte da rede responsável por aprender correspondência entre pontos baseando-se no LBAO (Operador de Laplace-Beltrami Anisotrópico).

Para cada vértice de uma malha de triângulos, são calculados 421 features. Antes de calculá-las, a escala da malha é normalizada de acordo com o percentil de 30% das distâncias geodésicas entre todos pares de vértices como realizado em [56]. Alguns destes cálculos requerem que a computação seja realizada sobre as faces ao invés dos vértices, neste caso os valores destas features do vértice é calculado como sendo a média dos valores das faces incidentes no vértice ponderada pela área destas faces. O cálculo das distâncias geodésicas entre todos os vértices também é computado e necessário tanto para a obtenção de algumas das features apresentadas [49], e no caso da forma de referência, é necessário para interpretar o erro aproximado como uma distância geodésica ponderada em relação aos dados da correspondência absoluta da forma.

São extraídas features, em várias escalas — selecionando retalhos da malha compostos por conjuntos de faces dentro de um limite determinado da escala, limite este determinado em relação ao raio geodésico relativo a mediana de todos os pares de distâncias geodésicas — da curvatura de superfície por combinações das curvaturas principais encontradas para cada retalho. Valores extraídos da Análise de Componentes Principais (PCA) da forma local, diâmetro de forma [113], distâncias de pontos da superfície medial [82] e contexto de formas [12]

Tais features, apesar de geralmente utilizadas para classificação ou segmentação em treinamentos supervisionados, aqui são utilizadas para realizar uma "classificação" não de segmentos, mas em vértices correspondentes, i.e., para uma malha de n vértices, parte da rede produzirá um ranking de probabilidades para n regiões para as quais os vértices podem ser mapeados. O objetivo não é conseguir aprender o mapeamento total, mas sim definir probabilidades de cada um dos vértices pertencer a cada uma das regiões; o que resultado será uma contribuição estatística para o restante da rede, especialmente para os resultados obtidos pelo aprendizado utilizando LBAO.

Vale ressaltar que mesmo com a adoção destas features, ao tratarmos de superfícies de diversas formas arbitrárias, com diferentes topologia e geometria, algumas das features encontradas serão redundantes entre regiões das superfícies. De fato, nesta proposta a rede realiza filtros sobre tais features (Figura C.43), mas só define a pontuação espacial após a adição de novas features espaciais baseadas na rede projetada por Qi et al. [98]. A combinação de ambas ocorre para melhorar a classificação das regiões dos pontos das formas, como ilustrado na Figura C.44.

O trabalho original de Qi et al. utiliza uma arquitetura simples trabalhando sobre nuvens de pontos, i.e., usa coordenadas 3D de cada ponto, em que nos estágios iniciais cada ponto é processado independentemente e identicamente. A rede original aprende um conjunto de critérios de otimização que seleciona pontos informativos e

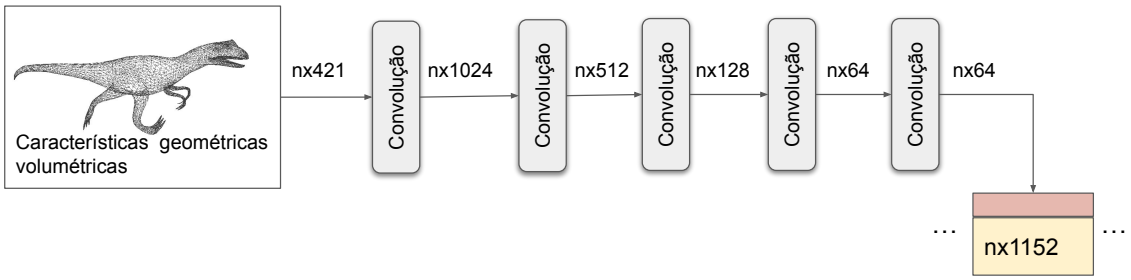

Figura C.43: 421 características geométricas passam por filtros resultando em 64 features que posteriormente são concatenadas com outras 1088.

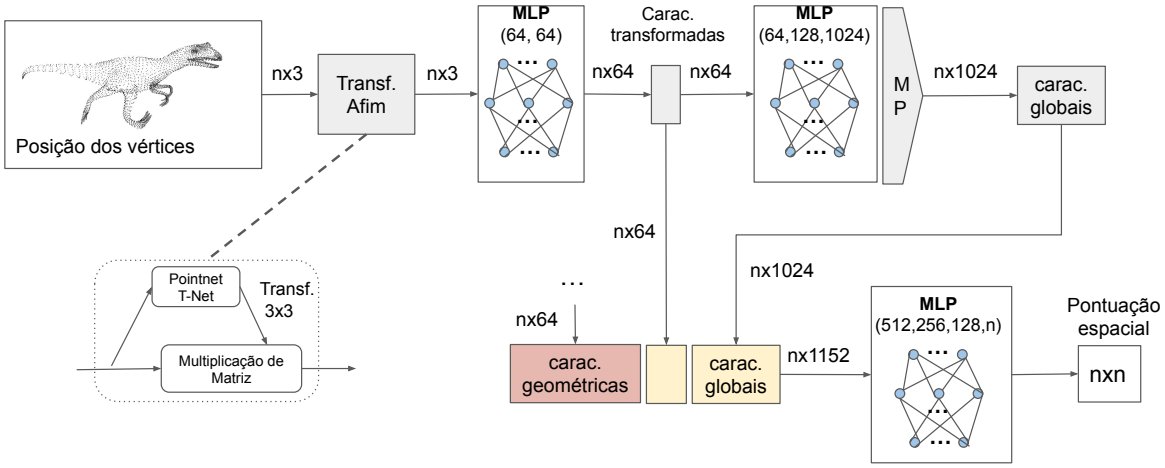

Figura C.44: Modificação da rede PointNet de Qi et al. [98]. Ao final a rede recebe 64 features providas por outras convoluções e aprende n pontuações espaciais para cada vértice.

codifica o motivo pelo qual o ponto foi selecionado. Em os valores ótimos aprendidos são agregados em um descritor global para a forma completa, o que pode ser utilizada para classificação de formas ou como parte da rotulação de pontos. Intuitivamente a rede aprende a resumir um conjunto de pontos de entrada em um conjunto de pontos chave, que grosseiramente correspondem ao um "esqueleto" de objetos. Para o propósito deste experimento, toma-se como entrada as posições dos vértices das formas. Não há necessidade de relacionar os pontos na entrada, já que a rede aplica transformações afins sobre estes pontos independentemente. Mas, assim como projetado no aprendizado com as features das formas, ao invés de classificar, a implementação tenta aprender uma correspondência, i.e., classificar cada ponto da forma de referência para um ponto da forma alvo.

Dentro do pipeline da rede, obtém-se 1088 features por vértice que representando os pontos chave codificados. Estas features são concatenadas com 64 features obtidas pela estrutura da rede responsável por processar as features extraídas axiomaticamente, resultando em 1152 features por vértice. Por fim, estas features atravessam uma MLP para obter os *ranking* das possibilidades de mapeamentos entre vértices de entrada e vértices de referência. O resultado será a contribuição estatística para o restante da rede, contribuindo com os resultados obtidos pelo aprendizado utilizando LBAO.

Regularização Parte da solução proposta neste trabalho substitui a correspondência direta usada no trabalho de Li et al. [71] por uma solução semelhante à adotada por Litany et al. [78]. Litany et al. criaram um *framework* para aprendizagem de corres-

pondência entre formas 3D deformáveis. Não obstante, ao invés de modelar o problema como uma tarefa de rotulação, em que cada ponto de uma malha de origem é relacionada a um ponto de uma outra malha de referência; a correspondência é construída a posteriori, a partir das probabilidades de correspondência entre as duas formas. A predição é realizada em um espaço de mapas funcionais — i.e. imerge as representações das duas formas em um espaço latente e descobre operadores lineares que fornecem uma representação compacta da correspondência entre esses espaços. Para cálculo do erro aproximado, o critério de correspondência utilizado é geometricamente significativo e foi originalmente introduzido por [63] para avaliação de mapas aproximados.

C.5 Considerações Finais

Baseando-se nos trabalhos mais atuais, fica claro que deve-se considerar não somente os aspectos estruturais e semânticos que caracterizam determinado problema de correspondência, mas também as representações das formas de entrada e da própria correspondência. Percebeu-se claramente a utilização de redes neurais profundas para solucionar tarefas relacionadas a malhas tridimensionais, com variadas terminologias [139, 14, 33, 2, 106, 105, 66, 59, 20].

Diversos métodos encontrados, sejam baseados em distorção métrica [19, 104, 95] ou baseados na decomposição própria (*eigen-decomposition*) do Laplaciano [110, 6, 122, 80, 91] captam principalmente deformações aproximadamente isométricas, e acabam considerando deformações muito genéricas muitas das vezes não consistentes com a intuição humana de correspondência. Isso ocorre em tarefas que envolvem modelos de formas de classes distintas ou que não são isométricas, como por exemplo no problema de detectar pontos de correspondência iniciais para morphing [69, 90].

Todavia, esses métodos baseados na decomposição própria do Laplaciano, e.g., HKS ou WKS, capturam bem informações relacionadas a deformações (deslocamentos) e relações de distância entre um ponto e seus vizinhos. Isto é, servem bem para tarefas que envolvem correlações entre malhas cujas deformações são isométricas ou aproximadamente isométricas, transferência de deformações ou correlação entre pares específicos de formas rígidas, mas não servem diretamente para solução do problema aqui tratado.

Durante o decorrer do processo de experimentos, realizamos testes empíricos com uma série de variações do modelo, combinando as features obtidas de domínios espaciais e espectrais baseando-se em configurações de outros trabalhos já citados aqui. Contudo, a diferença entre os resultados obtidos não foi significativa. Nenhuma das alternativas mostradas foi capaz de generalizar corretamente os pontos de interesse desejados. Observamos ainda o seguinte comportamento: ao diminuir a quantidade de camadas a rede montada aproxima-se mais da solução de uma malha específica, mas comporta-se arbitrariamente em relação à outras. Ao aumentar a quantidade de camadas, a rede aparenta entrar em *overfitting* e encontrar soluções nas dimensões extras que não correspondem aos pontos desejados e em alguns casos encontrando a mesma solução para features próximas mas distintas. Como pode ser notado pelas imagens da Figura C.45, tanto configurações que aprendem uma única feature como(a) como também configurações que aprendem probabilidades sobre várias features (b) não apresentaram os resultados esperados.

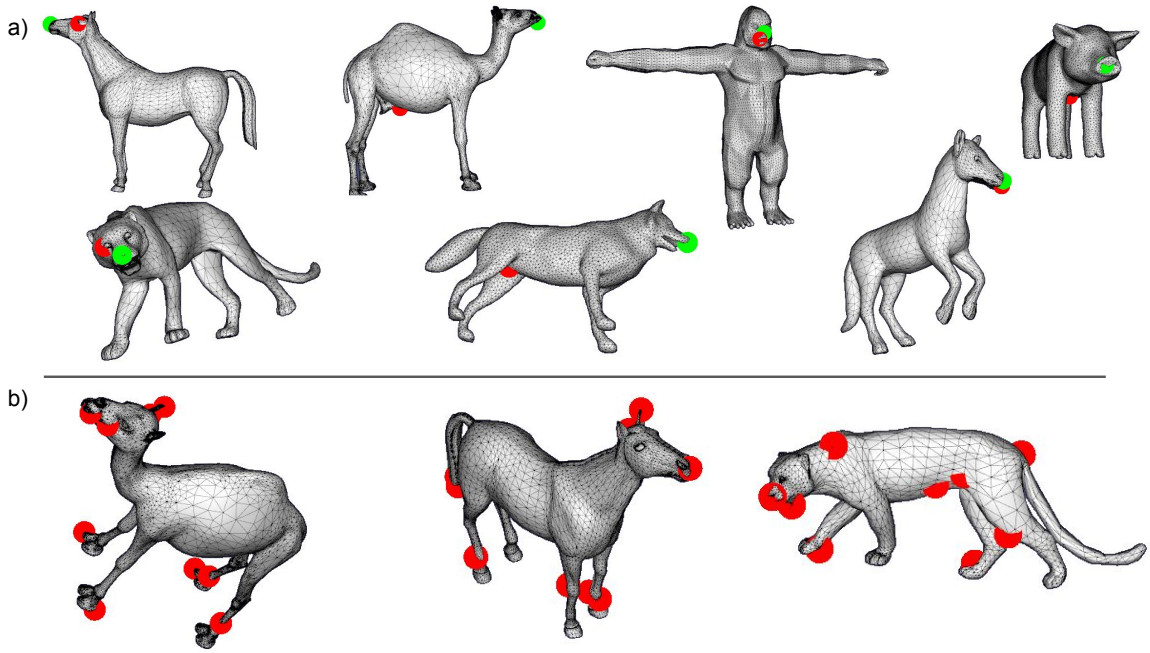

Figura C.45: Em verde, estão marcados os pontos de interesse verdadeiros desejados. Em vermelho os pontos marcados pela rede. Em (a) são exibidos os resultados obtidos por uma configuração da rede que tenta aprender apenas uma feature (correta em verde). Em (b) são exibidos resultados obtidos por uma configuração da rede que tenta aprender 10 features de uma só vez. As posições corretas deveriam ser patas(4), olhos(2), nariz, orelhas(2), peito e costas. Perceber-se claramente a inconsistência entre as marcações.

Destaca-se também a dificuldade em encontrar conjuntos de dados consistentes disponíveis, além do fato de que a performance e consumo de memória da rede aumenta em função da quantidade de triângulos das malhas. Todavia, temos que deixar claro que as abordagens são eficazes para seus respectivos problemas, apesar da dificuldade enfrentada em reproduzir alguns deles com bases de dados diferentes das originais. Abordagens espectrais comportam-se bem quando o problema refere-se, de alguma maneira, a comparação entre deformações e padrões da superfície. Portanto, são naturalmente adequadas para comparações de deformações aproximadamente isométricas ou isométricas, entre duas formas rígidas (sem deformações), ou ainda para definir espaços latentes de deformações. Diante disso, concluímos que para nosso propósito, mesmo tendo sucesso em segmentar partes teríamos ainda um grande desafio em calcular ou aprender a correspondência entre partes semânticas de classes arbitrárias.

As dificuldades, principalmente de encontrar datasets disponíveis e adequados cada abordagem, motivaram a busca por outras alternativas, como métodos multivisão. Contudo, trabalhos que porventura desejem empregar estratégias parecidas com esta como ponto de partida, podem aproveitar a experiência e tomar conhecimento dos problemas a serem enfrentados já no início da pesquisa.

Referências Bibliográficas

[1] Aflalo, Y., Bronstein, A. M., Bronstein, M. M., and Kimmel, R. Deformable shape retrieval by learning diffusion kernels. In *Proceedings of the Third International Conference on Scale Space and Variational Methods in Computer Vision* (Berlin, Heidelberg, 2012), SSVM'11, Springer-Verlag, pp. 689–700.

[2] Ahmed, E., Saint, A., Shabayek, A. E. R., Cherenkova, K., Das, R., Gusev, G., Aouada, D., and Ottersten, B. E. Deep learning advances on different 3d data representations: A survey. *CoRR abs/1808.01462* (2018).

[3] Andreux M., Rodola E., A. M., and D., C. Anisotropic laplace-beltrami operators for shape analysis. In *Computer Vision - ECCV 2014 Workshops* (Cham, 2015), R. C. Agapito L., Bronstein M., Ed., Springer International Publishing.

[4] Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. Scape: Shape completion and animation of people. *ACM Trans. Graph. 24*, 3 (July 2005), 408–416.

[5] Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. Scape: Shape completion and animation of people. *ACM Trans. Graph. 24*, 3 (jul 2005), 408–416.

[6] Aubry, M., Schlickewei, U., and Cremers, D. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proceedings of the IEEE International Conference on Computer Vision* (11 2011), pp. 1626–1633.

[7] Azencot, O., Dubrovina, A., and Guibas, L. Consistent shape matching via coupled optimization. *Computer Graphics Forum 38*, 5 (2019), 13–25.

[8] Baran, I., and Popoviundefined, J. Automatic rigging and animation of 3d characters. *ACM Trans. Graph. 26*, 3 (July 2007), 72–es.

[9] Baran, I., Vlasic, D., Grinspun, E., and Popoviundefined, J. Semantic deformation transfer. *ACM Trans. Graph. 28*, 3 (July 2009).

[10] Barra, V., and Biasotti, S. Learning kernels on extended reeb graphs for 3d shape classification and retrieval. In *Proceedings of the Sixth Eurographics Workshop on 3D Object Retrieval* (Aire-la-Ville, Switzerland, Switzerland, 2013), 3DOR '13, Eurographics Association, pp. 25–32.

[11] Basset, J., Wuhrer, S., Boyer, E., and Multon, F. Contact preserving shape transfer for rigging-free motion retargeting. In *Motion, Interaction and Games* (New York, NY, USA, 2019), MIG '19, Association for Computing Machinery.

[12] Belongie, S., Malik, J., and Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 4 (2002), 509–522.

[13] Ben-Chen, M., Weber, O., and Gotsman, C. Spatial deformation transfer. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2009), SCA '09, Association for Computing Machinery, p. 67–74.

[14] Biasotti, S., Cerri, A., Bronstein, A., and Bronstein, M. Recent trends, applications, and perspectives in 3d shape similarity assessment. *Computer Graphics Forum 36* (01 2016), 87–119.

[15] Bogo, F., Romero, J., Loper, M., and Black, M. J. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Piscataway, NJ, USA, June 2014), IEEE.

[16] Boscaini, D., Masci, J., Melzi, S., Bronstein, M. M., Castellani, U., and Vandergheynst, P. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Computer Graphics Forum 34*, 5 (2015), 13–23.

[17] Boscaini, D., Masci, J., Rodoià, E., and Bronstein, M. Learning shape correspondence with anisotropic convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (USA, 2016), NIPS'16, Curran Associates Inc., pp. 3197–3205.

[18] Boscaini, D., Masci, J., Rodolà, E., Bronstein, M. M., and Cremers, D. Anisotropic diffusion descriptors. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics* (Goslar Germany, Germany, 2016), EG '16, Eurographics Association, pp. 431–441.

[19] Bronstein, A. M., Bronstein, M. M., and Kimmel, R. Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences 103*, 5 (2006), 1168–1172.

[20] Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine 34*, 4 (July 2017), 18–42.

[21] Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and locally connected networks on graphs. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings* (2014).

[22] Chen, D.-Y., Tian, X.-P., Shen, Y.-T., and Ouhyoung, M. On visual similarity based 3d model retrieval. *Computer Graphics Forum* (2003).

[23] Chen, X., and Feng, J. Adaptive skeleton-driven cages for mesh sequences. *Comput. Animat. Virtual Worlds 25*, 3-4 (May 2014), 447–455.

[24] Chen, X., Feng, J., and Bechmann, D. Mesh sequence morphing. *Comput. Graph. Forum 35*, 1 (Feb. 2016), 179–190.

[25] Chen, X., Golovinskiy, A., and Funkhouser, T. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28*, 3 (Aug. 2009).

[26] Chu, H.-K., and Lin, C.-H. Example-based deformation transfer for 3d polygon models. *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 26* (03 2010), 379–391.

[27] Corman, É., Ovsjanikov, M., and Chambolle, A. Supervised descriptor learning for non-rigid shape matching. In *Computer Vision - ECCV 2014 Workshops* (Cham, 2015), L. Agapito, M. M. Bronstein, and C. Rother, Eds., Springer International Publishing, pp. 283–298.

[28] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. Deformable convolutional networks, 2017.

[29] Daz Productions, I. Daz 3D. `https://www.daz3d.com`, 2022. [Online; acessado em 29 de outubro de 2022].

[30] Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *CoRR abs/1606.09375* (2016).

[31] Dontcheva, M., Yngve, G., and Popoviundefined, Z. Layered acting for character animation. *ACM Trans. Graph. 22*, 3 (July 2003), 409–416.

[32] Eisenberger, M., Lähner, Z., and Cremers, D. Divergence-free shape correspondence by deformation. *Computer Graphics Forum 38*, 5 (2019), 1–12.

[33] Ezuz, D., Solomon, J., Kim, V. G., and Ben-Chen, M. GWCNN: A Metric Alignment Layer for Deep Shape Analysis. *Computer Graphics Forum* (2017).

[34] Gal, R., and Cohen-Or, D. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph. 25*, 1 (Jan. 2006), 130–150.

[35] Gao, L., Lai, Y.-K., Huang, Q.-X., and Hu, S.-M. A data-driven approach to realistic shape morphing. *Computer Graphics Forum 32*, 2pt4 (2013), 449–457.

[36] Gao, L., Lai, Y.-K., Liang, D., Chen, S.-Y., and Xia, S. Efficient and flexible deformation representation for data-driven surface modeling. *ACM Trans. Graph. 35*, 5 (July 2016).

[37] Gao, L., Lai, Y.-K., Yang, J., Zhang, L.-X., Kobbelt, L., and Xia, S. Sparse data driven mesh deformation. *IEEE transactions on visualization and computer graphics* (2017).

[38] Gao, L., Yang, J., Qiao, Y.-L., Lai, Y.-K., Rosin, P. L., Xu, W., and Xia, S. Automatic unpaired shape deformation transfer. *ACM Trans. Graph. 37*, 6 (Dec. 2018).

[39] George, D., Xie, X., and KL Tam, G. 3d mesh segmentation via multi-branch 1d convolutional neural networks. *Graphical Models 96* (03 2018), 1–10.

[40] Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (June 2014), pp. 580–587.

[41] Gleicher, M. Retargetting motion to new characters. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), SIGGRAPH '98, Association for Computing Machinery, p. 33–42.

[42] Gori, M., Monfardini, G., and Scarselli, F. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* (July 2005), vol. 2, pp. 729–734 vol. 2.

[43] Groueix, T., Fisher, M., Kim, V., Russell, B., and Aubry, M. Unsupervised cycle-consistent deformation for shape matching. *Computer Graphics Forum 38* (07 2019).

[44] Groueix, T., Fisher, M., Kim, V. G., Russell, B., and Aubry, M. 3d-coded : 3d correspondences by deep deformation. In *ECCV* (2018).

[45] Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., and Aubry, M. A papier-mache approach to learning 3d surface generation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), 216–224.

[46] Guo, K., Zou, D., and Chen, X. 3d mesh labeling via deep convolutional neural networks. *ACM Trans. Graph. 35*, 1 (Dec. 2015), 3:1–3:12.

[47] Halimi, O., Litany, O., Rodola, E., Bronstein, A., and Kimmel, R. Unsupervised learning of dense shape correspondence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (06 2019), pp. 4365–4374.

[48] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 770–778.

[49] Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. L. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH '01, Association for Computing Machinery, p. 203–212.

[50] Hinton, G. E., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput. 18*, 7 (July 2006), 1527–1554.

[51] Hsu, E., Gentry, S., and Popoviundefined, J. Example-based control of human motion. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2004), SCA '04, Eurographics Association, p. 69–77.

[52] Huang, G.-B., Wang, D., and Lan, Y. Extreme learning machines: A survey. *IJMLC* (01 2011), 1–16.

[53] Huang, H., Kalogerakis, E., Chaudhuri, S., Ceylan, D., Kim, V. G., and Yumer, E. Learning local shape descriptors from part correspondences with multi-view convolutional networks, 2017.

[54] Huang, H., Kalogerakis, E., and Marlin, B. Analysis and synthesis of 3d shape families via deep-learned generative models of surfaces. In *Proceedings of the Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2015), SGP '15, Eurographics Association, pp. 25–38.

[55] Kalogerakis, E., Averkiou, M., Maji, S., and Chaudhuri, S. 3d shape segmentation with projective convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 6630–6639.

[56] Kalogerakis, E., Hertzmann, A., and Singh, K. Learning 3d mesh segmentation and labeling. In *ACM SIGGRAPH 2010 Papers* (New York, NY, USA, 2010), SIGGRAPH 10, ACM, pp. 102:1–102:12.

[57] Katz, S., and Tal, A. Hierarchical mesh decomposition using fuzzy clustering and cuts. In *ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), SIGGRAPH '03, Association for Computing Machinery, p. 954–961.

[58] Kavan, L., Sloan, P.-P., and O Sullivan, C. Fast and Efficient Skinning of Animated Meshes. *Computer Graphics Forum* (2010).

[59] Khamparia, A., and Singh, K. M. A systematic review on deep learning architectures and applications. *Expert Systems 36*, 3 (2019), e12400. e12400 EXSY-Jul-18-241.R3.

[60] Kim, V. G., Lipman, Y., and Funkhouser, T. Blended intrinsic maps. *ACM Trans. Graph. 30*, 4 (July 2011).

[61] Kingma, D. P., and Welling, M. Auto-encoding variational bayes. *Proceedings of the International Conference on Learning Representations abs/1312.6114* (2013).

[62] Kircher, S., and Garland, M. Progressive multiresolution meshes for deforming surfaces. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2005), SCA '05, Association for Computing Machinery, p. 191–200.

[63] Kovnatsky, A., Bronstein, M. M., Bresson, X., and Vandergheynst, P. Functional correspondence by matrix completion, 2014.

[64] Kry, P., James, D., and Pai, D. Eigenskin: Real time large deformation character skinning in hardware. *SCA* (07 2002).

[65] Kulpa, R., Multon, F., and Arnaldi, B. Morphology-independent representation of motions for interactive human-like animation. In *Eurographics* (Dublin,/Ireland, France, Aug. 2005), E. A. for Computer Graphics, Ed., M. Alexa, J. Marks. http://www.eg.org/.

[66] Laga, H., Guo, Y., Tabia, H., Fisher, R., and Bennamoun, M. *Global Shape Descriptors*. John Wiley & Sons, Ltd, 2018, ch. 4, pp. 65–91.

[67] Lai, K., Bo, L., Ren, X., and Fox, D. A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE International Conference on Robotics and Automation* (2011), pp. 1817–1824.

[68] Landreneau, E., and Schaefer, S. Simplification of articulated meshes. *Computer Graphics Forum 28*, 2 (2009), 347–353.

[69] Lee, A. W. F., Dobkin, D., Sweldens, W., and Schröder, P. Multiresolution mesh morphing. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 343–350.

[70] Levie, R., Monti, F., Bresson, X., and Bronstein, M. M. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *CoRR abs/1705.07664* (2017).

[71] Li, Q., Liu, S., Hu, L., and Liu, X. Shape correspondence using anisotropic chebyshev spectral cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020).

[72] Li, Q., Liu, S., Hu, L., and Liu, X. Shape correspondence using anisotropic chebyshev spectral cnns. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition,CVPR 2020, Seattle, WA, USA, June 13-19, 2020* (2020), Computer Vision Foundation / IEEE, pp. 14646–14655.

[73] Li, T., Lin, Q., Bao, Y., and Li, M. Atss-net: Target speaker separation via attention-based neural network, 2020.

[74] Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., and Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection, 2020.

[75] Lian, Z., Godil, A., and Sun, X. Visual similarity based 3d shape retrieval using bag-of-features. In *2010 Shape Modeling International Conference* (2010), pp. 25–36.

[76] Lim, I., Gehre, A., and Kobbelt, L. Identifying style of 3d shapes using deep metric learning. In *Proceedings of the Symposium on Geometry Processing* (Goslar Germany, Germany, 2016), SGP '16, Eurographics Association, pp. 207–215.

[77] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. Microsoft coco: Common objects in context, 2014.

[78] Litany, O., Remez, T., Rodolà, E., Bronstein, A., and Bronstein, M. Deep functional maps: Structured prediction for dense shape correspondence. In *2017 IEEE International Conference on Computer Vision (ICCV)* (Oct 2017), pp. 5660–5668.

[79] Litman, R., Bronstein, A., Bronstein, M., and Castellani, U. Supervised learning of bag-of-features shape descriptors using sparse coding. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2014), SGP '14, Eurographics Association, pp. 127–136.

[80] Litman, R., and Bronstein, A. M. Learning spectral descriptors for deformable shape correspondence. *IEEE Trans. Pattern Anal. Mach. Intell. 36*, 1 (Jan. 2014), 171–180.

[81] Liu, P.-C., Wu, F.-C., Ma, W.-C., Liang, R.-H., and Ouhyoung, M. Automatic animation skeleton construction using repulsive force field. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications* (USA, 2003), PG '03, IEEE Computer Society, p. 409.

[82] Liu, R., Zhang, H., Shamir, A., and Cohen-Or, D. A part-aware surface metric for shape analysis. *Computer Graphics Forum 28*, 2 (2009), 397–406.

[83] Liu, Z., Mucherino, A., Hoyet, L., and Multon, F. Surface based motion retargeting by preserving spatial relationship. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games* (New York, NY, USA, 2018), MIG '18, Association for Computing Machinery.

[84] Long, J., Zhang, N., and Darrell, T. Do convnets learn correspondence? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1* (Cambridge, MA, USA, 2014), NIPS'14, MIT Press, pp. 1601–1609.

[85] Lun, Z., Kalogerakis, E., and Sheffer, A. Elements of style: Learning perceptual shape style similarity. *ACM Trans. Graph. 34*, 4 (July 2015), 84:1–84:14.

[86] Maron, H., Galun, M., Aigerman, N., Trope, M., Dym, N., Yumer, E., Kim, V. G., and Lipman, Y. Convolutional neural networks on surfaces via seamless toric covers. *ACM Trans. Graph. 36*, 4 (July 2017), 71:1–71:10.

[87] Masci, J., Boscaini, D., Bronstein, M., and Vandergheynst, P. Shapenet: Convolutional neural networks on non-euclidean manifolds. *Infosciende EPFL scientific publications* (2015).

[88] Masci, J., Boscaini, D., Bronstein, M. M., and Vandergheynst, P. Geodesic convolutional neural networks on riemannian manifolds. *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)* (2015), 832–840.

[89] Meagher, D. Geometric modeling using octree encoding. *Computer Graphics and Image Processing 19*, 2 (1982), 129 – 147.

[90] Medalha, A., Pagliosa, L., Paiva, A., and Pagliosa, P. Least-squares morphing of dynamic meshes. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (Oct 2017), pp. 23–30.

[91] Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., and Bronstein, M. M. Geometric deep learning on graphs and manifolds using mixture model cnns. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 5425–5434.

[92] Nash, C., and Williams, C. K. I. The shape variational autoencoder: A deep generative model of part-segmented 3d objects. *Comput. Graph. Forum 36*, 5 (Aug. 2017), 1–12.

[93] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. Shape distributions. *ACM Trans. Graph. 21*, 4 (Oct. 2002), 807–832.

[94] Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., and Guibas, L. Functional maps: A flexible representation of maps between shapes. *ACM Trans. Graph. 31*, 4 (July 2012), 30:1–30:11.

[95] Pelillo, M., Bulo, S. R., Torsello, A., Albarelli, A., and Rodola, E. A game-theoretic approach to pairwise clustering and matching. In *Similarity-Based Pattern Analysis and Recognition*. Springer, 2013.

[96] Qi, C. R., Su, H., Niessner, M., Dai, A., Yan, M., and Guibas, L. J. Volumetric and multi-view cnns for object classification on 3d data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 5648–5656.

[97] Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (USA, 2017), NIPS'17, Curran Associates Inc., pp. 5105–5114.

[98] Qi Charles, R., Su, H., Kaichun, M., and Guibas, L. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (07 2017), pp. 77–85.

[99] Redmon, J. Yolov5 documentation. `https://docs.ultralytics.com/`, 2015. [Online; acessado em 29 de outubro de 2022].

[100] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection, 2015.

[101] Reid, M. A. Graded rings and birational geometry. In *Proceedings of algebraic Symposium* (K. Ohno, 2000), pp. 1–72.

[102] Riegler, G., Ulusoy, A. O., and Geiger, A. Octnet: Learning deep 3d representations at high resolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 6620–6629.

[103] Rodolà, E., Bulò, S. R., Windheuser, T., Vestner, M., and Cremers, D. Dense non-rigid shape correspondence using random forests. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2014), CVPR '14, IEEE Computer Society, pp. 4177–4184.

[104] Rodolà, E., Torsello, A., Harada, T., Kuniyoshi, Y., and Cremers, D. Elastic net constraints for shape matching. In *ICCV* (2013), IEEE Computer Society, pp. 1169–1176.

[105] Rodrigues, R. S. V., Morgado, J. F. M., and Gomes, A. J. P. Part-based mesh segmentation: A survey. *Computer Graphics Forum 37*, 6 (2018), 235–274.

[106] Rostami, R., Bashiri, F. S., Rostami, B., and Yu, Z. A survey on data-driven 3d shape descriptors. *Computer Graphics Forum* (09 2018).

[107] Roufosse, J.-M., and Ovsjanikov, M. Unsupervised deep learning for structured shape matching. *ArXiv abs/1812.03794v3* (2019).

[108] Roufosse, J.-M., Sharma, A., and Ovsjanikov, M. Unsupervised deep learning for structured shape matching. In *The IEEE International Conference on Computer Vision (ICCV)* (October 2019), pp. 1617–1627.

[109] Rousseeuw, P., and Driessen, K. A fast algorithm for the minimum covariance determinant estimator. *Technometrics 41* (08 1999), 212–223.

[110] Rustamov, R. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Symposium on Geometry Processing* (01 2007), pp. 225–233.

[111] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *Trans. Neur. Netw. 20*, 1 (Jan. 2009), 61–80.

[112] scikit-learn developers. Gaussian mixture models. `https://scikit-learn.org/stable/modules/mixture.html#expectation-maximization`, 2022. [Online; acessado em 29 de outubro de 2022].

[113] Shapira, L., Shalom, S., Shamir, A., Cohen-Or, D., and Zhang, H. Contextual part analogies in 3d objects. *International Journal of Computer Vision 89*, 2-3 (2010), 309–326.

[114] Shelhamer, E., Long, J., and Darrell, T. Fully convolutional networks for semantic segmentation. *PAMI* (2016).

[115] Shi, B., Bai, S., Zhou, Z., and Bai, X. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Processing Letters 22*, 12 (2015), 2339–2343.

[116] Shu, Z., Qi, C., Xin, S., Hu, C., Wang, L., Zhang, Y., and Liu, L. Unsupervised 3d shape segmentation and co-segmentation via deep learning. *Comput. Aided Geom. Des. 43*, C (Mar. 2016), 39–52.

[117] Sinha, A., Bai, J., and Ramani, K. Deep learning 3d shape surfaces using geometry images. In *Computer Vision – ECCV 2016* (10 2016), vol. 9910, Springer, Cham, pp. 223–240.

[118] Sinha, A., Unmesh, A., Huang, Q., and Ramani, K. Surfnet: Generating 3d shape surfaces using deep residual networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017), pp. 791–800.

[119] Sorkine, O., and Cohen-Or, D. Least-squares meshes. In *Proceedings of the Shape Modeling International 2004* (Washington, DC, USA, 2004), SMI '04, IEEE Computer Society, pp. 191–199.

[120] Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)* (Washington, DC, USA, 2015), ICCV '15, IEEE Computer Society, pp. 945–953.

[121] Sumner, R. W., and Popoviundefined, J. Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 399–405.

[122] Sun, J., Ovsjanikov, M., and Guibas, L. A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2009), SGP 09, Eurographics Association, pp. 1383–1392.

[123] Tan, Q., Gao, L., Lai, Y.-K., and Xia, S. Variational autoencoders for deforming 3d mesh models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (06 2018), pp. 5841–5850.

[124] Tangelder, J., and Veltkamp, R. A survey of content based 3d shape retrieval methods. In *Proceedings Shape Modeling Applications, 2004.* (2004), pp. 145–156.

[125] Teichmann, M., and Teller, S. Assisted articulation of closed polygonal models. In *ACM SIGGRAPH 98 Conference Abstracts and Applications* (New York, NY, USA, 1998), SIGGRAPH '98, Association for Computing Machinery, p. 254.

[126] Thiery, J.-M., Guy, E., Boubekeur, T., and Eisemann, E. Animated mesh approximation with sphere-meshes. *ACM Trans. Graph. 35*, 3 (May 2016).

[127] TurboSquid, I. Free3D. `https://free3d.com/`, 2022. [Online; acessado em 29 de outubro de 2022].

[128] van Kaick, O., Zhang, H., Hamarneh, G., and Cohen-Or, D. A survey on shape correspondence. *Computer Graphics Forum 30*, 6 (2011), 1681–1707.

[129] Wade, L., and Parent, R. E. Automated generation of control skeletons for use in animation. *Vis. Comput. 18*, 2 (Apr. 2002), 97–110.

[130] Wang, P., Gan, Y., Zhang, Y., and Shui, P. 3d shape segmentation via shape fully convolutional networks. *Computers & Graphics* (02 2017).

[131] Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y., and Tong, X. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph. 36*, 4 (July 2017), 72:1–72:11.

[132] Wang, R. Y., Pulli, K., and Popoviundefined, J. Real-time enveloping with rotational regression. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA, 2007), SIGGRAPH '07, Association for Computing Machinery, p. 73–es.

[133] Wang, W., Ceylan, D., Mech, R., and Neumann, U. 3dn: 3d deformation network. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019), pp. 1038–1046.

[134] Wei, L., Huang, Q., Ceylan, D., Vouga, E., and Li, H. Dense human body correspondences using convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition,CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* (2016), pp. 1544–1553.

[135] Welchman, A., Deubelius, A., Conrad, V., Bülthoff, H., and Kourtzi, Z. 3d shape perception from combined depth cues in human visual cortex. *Nature neuroscience 8* (07 2005), 820–7.

[136] Xie, Z., Xu, K., Liu, L., and Xiong, Y. 3d shape segmentation and labeling via extreme learning machine. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2014), SGP '14, Eurographics Association, pp. 85–95.

[137] Xie, Z., Xu, K., Shan, W., Liu, L., Xiong, Y., and Huang, H. Projective feature learning for 3d shapes with multi-view depth images. *Comput. Graph. Forum 34*, 7 (Oct. 2015), 1–11.

[138] Xu, K., Kim, V. G., Huang, Q., Mitra, N., and Kalogerakis, E. Data-driven shape analysis and processing. In *SIGGRAPH ASIA 2016 Courses* (New York, NY, USA, 2016), SA '16, ACM, pp. 4:1–4:38.

[139] Xu, K., Kim, V. G., Huang, Q., Mitra, N., and Kalogerakis, E. Data-driven shape analysis and processing. In *SIGGRAPH ASIA 2016 Courses* (New York, NY, USA, 2016), SA '16, Association for Computing Machinery.

[140] Xufeng Han, Leung, T., Jia, Y., Sukthankar, R., and Berg, A. C. Matchnet: Unifying feature and metric learning for patch-based matching. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015), pp. 3279–3286.

[141] Yang, J., Gao, L., Lai, Y.-K., Rosin, P. L., and Xia, S. Biharmonic deformation transfer with automatic key point selection. *Graphical Models 98* (2018), 1 – 13.

[142] Yang, Y., Feng, C., Shen, Y., and Tian, D. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (June 2018), pp. 206–215.

[143] Yi, L., Su, H., Guo, X., and Guibas, L. J. Syncspeccnn: Synchronized spectral CNN for 3d shape segmentation. *CoRR abs/1612.00606* (2016).

[144] Yifan, W., Aigerman, N., Kim, V., Chaudhuri, S., and Sorkine-Hornung, O. Neural cages for detail-preserving 3d deformations, 2019.

[145] Yin, K., Chen, Z., Huang, H., Cohen-Or, D., and Zhang, H. Logan: Unpaired shape transform in latent overcomplete space. *ACM Trans. Graph. 38*, 6 (Nov. 2019).

[146] Yin, L., Guo, K., Zhou, B., and Zhao, Q. 3d shape co-segmentation via sparse and low rank representations. *Science China Information Sciences 61* (05 2018).

[147] Zagoruyko, S., and Komodakis, N. Learning to compare image patches via convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015), pp. 4353–4361.

[148] Zhang, H., Sheffer, A., Cohen-Or, D., Zhou, Q., van Kaick, O., and Tagliasacchi, A. Deformation-driven shape correspondence. In *Proceedings of the Symposium on Geometry Processing* (Goslar, DEU, 2008), SGP '08, Eurographics Association, p. 1431–1439.

[149] Zhang, H., Wang, Y., Dayoub, F., and Sünderhauf, N. Varifocalnet: An iou-aware dense object detector, 2020.

[150] Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., Li, M., and Smola, A. Resnest: Split-attention networks, 2020.

[151] Zhirong Wu, Song, S., Khosla, A., Fisher Yu, Linguang Zhang, Xiaoou Tang, and Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015), pp. 1912–1920.

[152] Zhu, Z., Wang, X., Bai, S., Yao, C., and Bai, X. Deep learning representation using autoencoder for 3d shape retrieval. *CoRR abs/1409.7164* (2014).

[153] Zuffi, S., and Black, M. J. The stitched puppet: A graphical model of 3d human shape and pose. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015), pp. 3537–3546.