

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

**Uma proposta de *morphing* utilizando técnicas de
interpolação de formas e média morfológica**
Faculdade de Engenharia Elétrica e de Computação

Autor: Rogério Seiji Higa

Orientador: Prof. Dr. Yuzo Iano

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Engenharia Elétrica**.

Banca Examinadora

Yuzo Iano, Dr. DECOM/FEEC/Unicamp
Evaldo Gonçalves Pelaes, Dr. UFPA
Luiz Henrique Bonani do Nascimento, Dr. CESET/Unicamp

Campinas, SP

2008

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

H533p Higa, Rogério Seiji
Uma proposta de morphing utilizando técnicas de
interpolação de formas e média morfológica / Rogério
Seiji Higa. --Campinas, SP: [s.n.], 2008.

Orientador: Yuzo Iano.
Dissertação de Mestrado - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Animação (Cinematográfica). 2. Processamento de
imagens. 3. Morfologia matemática. 4. Interpolação. 5.
Compressão de imagens. I. Iano, Yuzo. II. Universidade
Estadual de Campinas. Faculdade de Engenharia Elétrica
e de Computação. III. Título.

Título em Inglês: A morphing proposal using shape interpolation and morphological
median

Palavras-chave em Inglês: Animation (Cinematography), Image processing,
Mathematical morphology, Interpolation, Image
compression

Área de concentração: Telecomunicações e Telemática

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Evaldo Gonçalves Pelaes, Luiz Henrique Bonani do Nascimento

Data da defesa: 25/11/2008

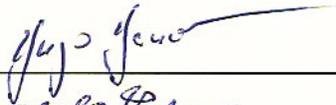
Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidato: Rogério Seiji Higa

Data da Defesa: 25 de novembro de 2008

Título da Tese: "Uma Proposta de Morphing Utilizando Técnicas de Interpolação de Formas e Média Morfológica"

Prof. Dr. Yuzo Iano (Presidente): _____ 

Prof. Dr. Evaldo Gonçalves Pelaes: _____ 

Prof. Dr. Luiz Henrique Bonani do Nascimento: _____ 

Resumo

A animação é muito utilizada na área cinematográfica para produzir efeitos visuais ou para fazer filmes inteiros. Nessa área, a busca por novas ferramentas caminha junto com a necessidade da indústria do entretenimento de sempre mostrar novidades. Tentando suprir essa demanda, este trabalho propõe uma nova ferramenta que mistura o *morphing* de imagens, já muito utilizado nesta área, com a média morfológica. O uso dessas duas técnicas permite a geração de uma sequência de imagens diferente de outras técnicas de *morphing*. Além disso, também é proposta a utilização de um algoritmo de interpolação de formas, que é utilizado junto ao *morphing* para interpolar os seus marcadores de atributos. O uso da interpolação de formas permite que dois marcadores correspondentes tenham um número diferente de pontos, e também adiciona opções de compensação de rotação entre as formas dos marcadores. Neste trabalho são mostrados os resultados obtidos com o método proposto.

Palavras-chave: *morphing*, animação, deformação de imagens, morfologia matemática, interpolação, formas, média.

Abstract

Animation is used in the cinema industry to produce visual effects or entire movies. In this area, the search for new tools comes along with the need of the entertainment industry to show new stuff all the time. To fulfill this need, this work proposes a new tool that blends the image morphing, a famous tool for visual effects, with the morphological median. The use of these two techniques provides an image sequence that's distinct from other similar tools. It is also proposed in this work the use of the shape interpolation which is used in the morphing algorithm to interpolate the features markers. The shape interpolation algorithm allows that the corresponding markers have a different number of points, it also includes a rotation compensation options for the interpolation of the shapes of the markers. In this work are shown the results obtained from the proposed method.

Keywords: morphing, animation, warping, mathematical morphology, interpolation, shapes, median.

*Aos meus pais
e irmã*

Agradecimentos

Ao meu orientador Prof. Dr. Yuzo Iano, sou grato pelos longos anos de orientação.

Ao meu pai Seiko e a minha mãe Alice que sempre me apoiaram nos momentos difíceis, e me ensinaram que a vida vai além de títulos, fronteiras e costumes.

A minha irmã Adriana, com quem sempre posso conversar sobre psicologia.

Ao resto da minha família pelo apoio durante esta jornada.

Aos professores Vicente Idalberto Becerra Sablón e Romis Ribeiro Faissol Attux pelas críticas e sugestões na leitura desta dissertação.

Aos meus companheiros do LCV, pelo apoio, dicas, sugestões e ajuda neste trabalho.

Aos meus colegas da turma EE97 e outros amigos das diversas turmas da época da graduação.

Aos demais colegas de pós-graduação, pelas críticas e sugestões.

Ao pessoal da academia pela amizade e companhia. Apesar de meu corpo ficar exausto, minha mente ficava mais calma.

À CAPES, pelo apoio financeiro.

Sumário

Lista de Figuras	xiii
Lista de Tabelas	xvii
Símbolos & Abreviaturas	xxi
Publicações	xxiii
1 Introdução	1
1.1 Motivação	2
1.2 O estado da arte	3
1.3 Organização	4
2 Interpolação de formas	7
2.1 Correspondência de pontos	8
2.1.1 Representação	9
2.1.2 Cálculo da correspondência	9
2.2 Interpolação	10
2.2.1 Interpolação linear	10
2.2.2 Interpolação por lados e ângulos	11
2.3 Métrica de polígonos	15
2.3.1 Representação	15
2.3.2 Cálculo da distância	15
2.4 Redução do número de vértices de polígonos	17
2.5 Método proposto	18
2.5.1 Utilizando a métrica para achar a correspondência inicial	20
2.5.2 Correspondência de lados	22
2.5.3 Utilizando a redução de vértices como pré-processamento	23

2.5.4	Resultados	26
2.6	Considerações finais	33
3	Morphing	35
3.1	Deformação de imagens	38
3.1.1	Mapeamento	38
3.1.2	Transformação geométrica <i>affine</i>	38
3.1.3	Deformação por malha	40
3.2	Algoritmo de deformação de imagens por campo	41
3.2.1	Transformação com múltiplos pares de segmentos de reta	44
3.2.2	Otimização	45
3.3	Interpolação das imagens deformadas	47
3.4	Considerações finais	47
4	Média de imagens utilizando morfologia matemática	49
4.1	Morfologia matemática	50
4.1.1	Imagens binárias	50
4.1.2	Imagens em tons de cinza	51
4.2	Média morfológica	53
4.2.1	Imagens coloridas	57
4.3	Função de interpolação para imagens binárias	59
4.4	Proposta de uma função de interpolação para imagens em tons de cinza	62
4.4.1	Transformada de distância	62
4.4.2	Separando a imagem em níveis	63
4.5	Proposta de controle de transição para média morfológica	64
4.5.1	Resultados	65
4.6	Considerações finais	65
5	Proposta de <i>morphing</i> utilizando interpolação de formas e média morfológica	69
5.1	Implementação	69
5.1.1	Deformação de imagens utilizando interpolação de formas	70
5.1.2	Média morfológica no lugar do <i>cross dissolve</i>	70
5.2	Resultados	71
5.2.1	Influência dos pesos sem decomposição	72
5.2.2	Influência dos pesos com decomposição	77
5.3	Seqüências geradas	79

5.4	Comparação do método proposto com o de Iwanowski e Serra	89
5.5	Vantagens e desvantagens	94
5.6	Considerações finais	95
6	Conclusões e trabalhos futuros	97
6.1	Conclusões sobre a interpolação de formas	98
6.2	Conclusões sobre a média morfológica	99
6.3	Trabalhos futuros	100
6.3.1	Interpolação de formas	100
6.3.2	Média morfológica	100
	Bibliografia	101
A	Dedução de equações	107
A.1	Dedução da equação 2.36	107
A.2	Dedução da equação 2.40	109

Lista de Figuras

2.1	Parâmetros da representação de um polígono por lados e ângulos.	12
2.2	Representação de um polígono por uma função $\Theta(s)$	15
2.3	Exemplo de um polígono representado e sua representação por uma função $\Theta(s)$. . .	16
2.4	Seqüência de redução de pontos de um polígono em forma de tubarão. A seqüência foi amostrada com uma diferença de 10 pontos entre os polígonos.	19
2.5	Detalhes da seqüência de redução de pontos de um polígono em forma de tubarão. .	19
2.6	Seqüência de redução de pontos de um polígono em forma de pato. A seqüência foi amostrada com uma diferença de 10 pontos entre os polígonos.	20
2.7	Seqüência de redução mais detalhada.	20
2.8	Processo de correspondência entre polígonos. (a) Processo de correspondência entre polígonos reduzidos. Os círculos representam os pontos dos polígonos reduzidos e os quadrados as correspondências. (b) Correspondência entre os polígonos completos utilizando a correspondências dos polígonos reduzidos como restrições. (c) Correspondência final entre os polígonos.	24
2.9	Exemplo de correspondência de lados. Um lado do polígono superior é conectado com três lados do polígono inferior.	25
2.10	Parâmetros para o cálculo de relevância dos vértices.	25
2.11	Correspondência entre um polígono em forma de dinossauro e outro em forma de tubarão.	27
2.12	Duas correspondências com parâmetros diferentes.	28
2.13	Amostragem da relação de vértices.	28
2.14	Interpolação de formas com um dançarino em duas posições diferentes.	28
2.15	Interpolação de formas de dois quadrados chanfrados sem compensação de rotação. .	29
2.16	Interpolação de formas de dois quadrados chanfrados com compensação de rotação. .	29
2.17	Interpolação entre um polígono na forma de tubarão e um na forma de pato com compensação de rotação.	31

2.18	Interpolação entre um polígono na forma de tubarão e um na forma de pato sem compensação de rotação.	31
3.1	<i>Cross dissolve</i> [1].	37
3.2	Mapeamento direto	39
3.3	Mapeamento inverso	39
3.4	Marcação de atributos utilizando malhas [1].	41
3.5	Marcação de atributos por curvas [1].	42
3.6	Parâmetros em relação a semireta na imagem de destino e na imagem de origem . . .	43
3.7	Parâmetros em relação a múltiplas semiretas	45
3.8	Posições dos pontos de teste interpolados linearmente e os calculados por deformação por campo	46
4.1	Exemplo de uso dos operadores de dilatação e erosão para imagens binárias	51
4.2	Exemplo de uso dos operadores de dilatação e erosão para imagens em tons de cinza	52
4.3	Vista em corte, como os operadores morfológicos alteram a imagem	52
4.4	Exemplo de duas imagens binárias e sua média morfológica.	54
4.5	Processo de geração de uma média morfológica	56
4.6	Comparação entre o <i>cross dissolve</i> e a média morfológica.	57
4.7	Médias morfológicas produzidas utilizando o espaço de cores RGB e YCbCr.	59
4.8	Exemplo de conjuntos binários	60
4.9	Construção da função de interpolação Int_V^V	61
4.10	Exemplos de corte na função de interpolação	61
4.11	Exemplo de uso dos operadores de dilatação e erosão para imagens em tons de cinza [2].	61
4.12	Cálculo de d_1 e d_2 através da transformada de distância	63
4.13	Exemplo de máscaras de <i>chamfer</i>	63
4.14	Diferença entre os cortes lineares e os cortes utilizando uma função linear	64
4.15	Comparação entre as imagens geradas pelas funções de transição.	66
4.16	Comparação entre as imagens geradas pelas funções de transição.	67
5.1	Teste com média morfológica sem compensação de rotação. Os subtítulos das figuras (b)-(l) são os valores de peso $[w_1, w_2]$	73
5.2	Teste com <i>cross dissolve</i> sem compensação de rotação. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$	74
5.3	Comparação de pesos para $w_1 = 1.0$ e $0.0 \leq w_2 \leq 7.0$. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$	75

5.4	Comparação de pesos sem compensação de rotação. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$	76
5.5	Comparação de pesos com compensação de rotação. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$	77
5.6	Comparação de pesos sem compensação de rotação. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$	78
5.7	Comparação de pesos com compensação de rotação. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$	79
5.8	Seqüência de <i>morphing</i> entre um avião e uma gaivota utilizando <i>cross dissolve</i>	80
5.9	Seqüência de <i>morphing</i> entre um avião e uma gaivota utilizando média morfológica.	81
5.10	Seqüência de <i>morphing</i> entre dois tipos de pão utilizando <i>cross dissolve</i>	82
5.11	Seqüência de <i>morphing</i> entre dois tipos de pão utilizando média morfológica.	83
5.12	Seqüência de <i>morphing</i> entre a Lena e a Elaine utilizando <i>cross dissolve</i>	84
5.13	Seqüência de <i>morphing</i> entre a Lena e a Elaine utilizando média morfológica.	85
5.14	Seqüência de <i>morphing</i> entre um alho e um tomate utilizando <i>cross dissolve</i>	86
5.15	Seqüência de <i>morphing</i> entre um alho e um tomate utilizando média morfológica.	87
5.16	Marcadores para o <i>morphing</i> das imagens das seqüências da Fig. 5.8 e Fig. 5.9	88
5.17	Marcadores para o <i>morphing</i> das imagens das seqüências da Fig. 5.10 e Fig. 5.11	88
5.18	Marcadores para o <i>morphing</i> das imagens das seqüências da Fig. 5.12 e Fig. 5.13	89
5.19	Marcadores para o <i>morphing</i> das imagens das seqüências da Fig. 5.14 e Fig. 5.15	89
5.20	Marcação dos atributos do método proposto e o de Iwanowski & Serra	90
5.21	Comparação entre o método proposto e o de Iwanowski & Serra	93
5.22	Comparação entre a composição por interpolação linear e por média morfológica	94

Lista de Tabelas

2.1	Distâncias entre os polígonos intermediários e os originais na Fig. 2.11.	27
2.2	Distâncias entre os polígonos intermediários e os originais na Fig. 2.14.	29
2.3	Distâncias entre os polígonos intermediários e os originais na Fig. 2.15.	29
2.4	Distâncias entre os polígonos intermediários e os originais na Fig. 2.16.	30
2.5	Distâncias entre os polígonos intermediários e os originais na Fig. 2.17.	31
2.6	Distâncias entre os polígonos intermediários e os originais na Fig. 2.18.	32
2.7	Parâmetros utilizados nas interpolações	32

Lista de Algoritmos

2.1	Cálculo da distância entre polígonos.	17
2.2	Cálculo da correspondência utilizando a função de custo.	23
2.3	Método proposto para interpolação de formas.	24
2.4	Redução do número de vértices de polígonos.	26
2.5	Método proposto para interpolação de formas com decomposição de polígonos.	26
3.1	Descrição dos passos do <i>morphing</i> por malhas.	41
3.2	Cálculo da transformação para um segmento de reta	43
3.3	Cálculo da transformação para múltiplos segmentos de reta	45
3.4	Otimização da deformação por campo	46
5.1	Método Proposto	70
5.2	Método proposto para interpolação de curvas abertas.	71

Símbolos & Abreviaturas

$\Theta(s)$	Função de representação de um polígono pelo valor do ângulo da tangente.
$\theta(t)$	Compensação de rotação.
$A \setminus B$	É equivalente a $A \cap B^C$
$A \subset B$	O conjunto A é um subconjunto de B
A^C	É o conjunto complementar de A
np	Número de pontos do polígonos reduzidos.
w_1	Peso ligado a diferença de ângulo.
w_2	Peso ligado a distribuição da correspondência.
w_{rot}	Controla a compensação de rotação.
ILM	Industrial Light & Magic
RGB	Espaço de cores formado pelos canais R (vermelho), G (verde) e B (azul).
TD(A)	Transformada de distância de A
YCbCr	Espaço de cores formado pelos canais Y (luminância), Cb e Cr (cromitância).

Publicações

1. R.S. Higa, Y. Iano. “Interpolação de formas utilizando uma métrica de polígonos e decomposição de formas”. *Revista Ciência e Tecnologia*. (aceito)
2. R.S. Higa, Y. Iano. “The Influence of the Structuring Element on the Morphological Median”. *Journal of Mathematical Imaging and Vision*. (submetido)
3. R.S. Higa, Y. Iano. “Transition Control for Morphological Interpolation”. *IEEE Transactions on Visualization and Computer Graphics*. (submetido)

Capítulo 1

Introdução

Animação vem do termo em latim *anima*, que se refere ao princípio que dá movimento ao que é vivo, o que é animado ou o que faz mover. A animação, então, pode ser definida como a tentativa de dar vida a um objeto, seja por meio de um desenho, foto ou escultura. Outro termo que também é derivado de *anima* é a palavra alma. Assim, a animação também pode ser vista como a atribuição de uma "alma" ao objeto.

Em termos práticos, a animação descreve métodos de capturar o movimento de objetos. A primeira máquina inventada que realmente criava a ilusão de movimento foi o fenacístiscopio [3]. Trata-se de uma placa circular que gira uma seqüência de imagens desenhadas com janelas de visualização que enquadram a visão do espectador nos desenhos. Mais tarde, foi criada a técnica de capturar o movimento através de uma seqüência de fotogramas, dando origem ao cinema.

A animação pode ser utilizada como uma ferramenta para transcrever o que existe em nossa mente, em forma abstrata, para uma forma concreta, permitindo desse modo o compartilhamento da informação com outras pessoas. A animação é uma poderosa ferramenta de comunicação, podendo ser utilizada no aprendizado e na transmissão de conceitos científicos e éticos.

Apesar da primeira aplicação que se pensa ao se falar em animação seja o entretenimento, a animação também tem aplicações em outras áreas, como a visualização de informação: por exemplo, mapas meteorológicos mostram as nuvens e outros indicadores de tempo na forma de animações. Outro exemplo é a arquitetura, onde um prédio pode ser visualizado antes mesmo de ser construído.

Este trabalho aborda algoritmos que tentam gerar uma animação a partir de duas imagens. Um deles é o *morphing* de imagens, que é uma técnica de animação em que uma imagem é transformada em outra através da geração de imagens intermediárias. *Morphing* também pode se referir à transformação de formas geométricas ou modelos 3D, sendo que, neste trabalho, foca-se no de imagens e de formas geométricas. O *morphing* de imagens é muito utilizado para efeitos visuais, sendo a deformação da imagem um dos seus principais componentes. Essa técnica se tornou bastante popular

com o *videoclipe "Black or White"*, de Michael Jackson.

Outra forma de gerar uma animação através de duas imagens é a média morfológica, que produz uma imagem média que leva em conta o conteúdo da imagem.

A máquina vem substituindo o homem em várias funções, normalmente em tarefas repetitivas. A criatividade é a última barreira que a máquina ainda não conseguiu vencer, pois essa envolve muito mais que a simples execução de comandos: envolve também a transmissão de sentimentos, algo que é muito difícil de descrever. O foco dessa área não é substituir o artista, e sim dar-lhe ferramentas para que o seu rendimento aumente assim como o tempo que pode ser gasto com a parte criativa.

A constante revitalização de ferramentas para efeitos visuais é necessária devido à característica humana de sempre procurar por novidades. Quando se vai ao cinema, a maioria das pessoas não quer assistir a um filme que já foi visto, mas a algo totalmente novo. O uso de certos efeitos visuais em demasia leva a fadiga, pois o efeito emocional obtido com a sua primeira utilização não é mais percebido, sendo produzido um outro ou nenhum efeito.

Podem-se comparar os efeitos visuais com o truque de um mágico. Se for feito muitas vezes e o segredo do truque é percebido, a mágica desaparece e as pessoas começam a notar as falhas do truque. Quando se assiste a um filme antigo e os efeitos visuais são analisados, nota-se que este não tem o mesmo impacto que se tinha na época de lançamento. A comparação com as técnicas atuais de efeitos visuais é inevitável, diminuindo a empolgação inicial.

Na área artística, mesmo aquilo que é considerado ruim isoladamente pode ser utilizado para montar algo interessante no conjunto. Um exemplo na música é o intervalo de segunda menor, que é extremamente irritante por causa do batimento. Esse intervalo dissonante tem um som desagradável se comparado com um intervalo consonante. Mas muitas músicas o utilizam e pode ser algo que dá o tom de suspense ou de que algo está errado. Ao utilizar esse intervalo em um acorde seu tom desagradável desaparece e acrescenta outra cor à sonoridade do acorde.

Outro exemplo é o ruído na imagem, bem presente nas transmissões analógicas e que desaparece com a vinda da TV digital. Muitos filmes simulam o ruído, seja para mostrar a perda de transmissão ou passar uma sensação de que o personagem está fora da realidade. Ou seja, algo totalmente indesejado na área de telecomunicações é uma ferramenta muito útil no cinema. Agora com as transmissões digitais esse ruído pode não existir mais, e talvez ele seja substituído por seqüências de movimento travado e cheio de blocos na imagem.

1.1 Motivação

Os métodos tradicionais de compressão de imagens se baseiam em uma transformada comprimindo a imagem na dimensão espacial. Em vídeo é utilizado, além da compressão espacial, um

preditor de movimento, e os blocos são transladados de acordo com a predição. Se fosse utilizada alguma técnica para interpolar todos os quadros no eixo temporal, isso poderia ser utilizado para melhorar a taxa de compressão. Apenas alguns parâmetros seriam guardados, e o algoritmo produziria as imagens através da interpolação dos quadros chaves. A dificuldade da interpolação de uma seqüência de imagens é que, quando existe rotação ou translação da câmera ou do objeto, a correlação entre as imagens diminui drasticamente.

Para obter uma boa seqüência de transição entre duas imagens, é necessária muita interação humana. Técnicas de *morphing* tentam inserir a intuição humana através da inserção de marcadores. Mesmo assim, até se chegar ao resultado desejado, precisa-se de muita tentativa e erro. Assim, um algoritmo que interpole automaticamente as imagens é algo muito difícil de ser implementado.

A média morfológica, que é um método de interpolação baseado em morfologia matemática, pode ser usada como um novo método para gerar a transição entre duas imagens. A característica mais interessante da média morfológica é que ela não requer a inserção de parâmetros adicionais além das imagens, sendo a geração da média um processo automático.

A seqüência gerada por essa técnica é visualmente bem interessante quando as imagens são diferentes. Mas, quando os objetos nas imagens têm muitas características em comum, o resultado fica estranho. Talvez isso aconteça porque a expectativa de transformação para imagens semelhantes é diferente para imagens distintas.

O problema da média morfológica em imagem semelhantes é o mesmo enfrentado pelo *morphing* de imagens: se as imagens não estiverem com todos os atributos alinhados, a média produzida não será agradável [1].

Iwanowisky [4] propôs utilizar técnicas de *morphing* junto com a média morfológica para melhorar os resultados. Continuando com essa idéia nesta pesquisa, foi feito um estudo de técnicas de *morphing*. O estudo foi dividido em três partes, a interpolação de formas, a deformação de imagens, e a interpolação dos valores dos *pixels*.

Juntando a interpolação de formas, a deformação de imagens e a média morfológica espera-se produzir resultados mais interessantes do que aqueles produzidos pelas técnicas de *morphing* existentes.

1.2 O estado da arte

A busca por novas formas de expressão é algo muito importante para a evolução cultural. Um exemplo dessa busca é o vestuário, toda estação uma nova coleção é apresentada. Ferramentas como o *morphing* [1] são o resultado dessa busca, mas, depois de um tempo com sua utilização popularizada, o conteúdo produzido não causa tanta comoção quanto na sua primeira aparição, essa forma de

expressão deixa de ser novidade. Novas formas de apresentar esse objeto ajudam a revitalizar o seu uso. A meta não é a substituição dos métodos e ferramentas existentes, mas sim a inserção de novas técnicas que ampliarão as opções do animador. Assim, um primeiro objetivo é criar uma ferramenta de animação que produza resultados diferentes e interessantes.

A principal técnica utilizada no *morphing* de imagens é o *warping* ou deformação de imagens. Esta técnica tenta deformar as imagens de modo que elas se encaixem uma na outra, provendo a ilusão que os objetos nas duas imagens são um só. Outra parte importante é a interpolação dos atributos marcados nas imagens, uma alternativa é utilizar técnicas de interpolação de formas [5]. Juntando essas duas técnicas é possível obter resultados no qual a interpolação de atributos tenha um comportamento mais próximo de uma animação.

A média morfológica [6, 7] é uma ferramenta interessante para a animação de seqüências de transição entre imagens, mas não funciona bem quando as imagens são semelhantes, ou seja, quando os dois objetos são muito parecidos, mas em posições diferentes. Por exemplo, olhos, boca, nariz em pessoas e roda, capô, portas e pára-choques em carros. A média morfológica não gera bons resultados se essas características não estiverem alinhadas.

Outra característica da média morfológica é que, na seqüência gerada, o começo e o final da seqüência apresentam uma transição abrupta entre as imagens. Essa característica tira a suavidade da transição. Assim, um segundo objetivo é melhorar a seqüência de imagens gerado pela média morfológica aplicando técnicas de *morphing* de imagens. Um terceiro objetivo é melhorar a transição da seqüência por médias morfológicas.

O objetivo principal desse trabalho é produzir resultados mais interessantes ao adicionar novas técnicas à média morfológica, obtendo uma seqüência de imagens que expresse a transformação de uma imagem em outra de modo suave e interessante. A idéia básica é juntar o *morphing* com a média morfológica. A primeira contribuição desse trabalho é feito aplicando uma nova técnica de interpolação de formas com o *morphing*, permitindo maior flexibilidade a ferramenta. Outra contribuição é a proposta de um método de controle de transição da média morfológica, melhorando a seqüência gerada. O resultado dessas contribuições leva a uma nova ferramenta de *morphing*, a principal contribuição desse trabalho.

1.3 Organização

No Capítulo 2, é apresentada a interpolação de formas, onde formas poligonais são interpoladas gerando formas intermediárias. Essa técnica também pode ser chamada de *morphing* de polígonos. São descritas as duas etapas da interpolação de formas, isto é, a correspondência e a interpolação dos pontos. Nesse capítulo também é proposto um método de interpolação de formas que utiliza uma

métrica de polígonos, decomposição de polígonos e correspondência de lados. Os resultados obtidos pelo método proposto são apresentados.

No Capítulo 3, é apresentado de forma breve o algoritmo de *morphing* utilizado neste trabalho. O algoritmo de *morphing* divide-se em duas etapas, sendo elas a deformação de imagens e a interpolação dos *pixels*.

O Capítulo 4 apresenta o conceito de média morfológica e as formas de aplicações dessa média para imagens binárias, em tons de cinza e coloridas. Uma função de interpolação para imagens binárias que permite obter qualquer nível de interpolação através do corte é apresentada. Apresenta-se também uma proposta de função de interpolação para imagens em tons de cinza e outra proposta de controle de transição para média morfológica.

O Capítulo 5 apresenta o método proposto de *morphing*, que combina a interpolação de formas, média morfológica, controle de transição e a deformação de imagens. Os resultados e testes do método proposto são apresentados.

No Capítulo 6, são expostas as conclusões e propostas de trabalhos futuros.

Capítulo 2

Interpolação de formas

Uma das primeiras formas de animação nasceu da tentativa do ser humano de capturar o movimento através de uma seqüência de figuras. Nessa seqüência, o objeto é representado em várias posições diferentes, cada uma delas representando um instante do movimento. Um modo simples de representar objetos é através de formas geométricas, portanto, a transição de uma forma geométrica para outra pode ser considerada a técnica mais simples de animação.

Dados dois polígonos, um representando o estado inicial e outro o estado final, a interpolação de formas gera as formas intermediárias, obtendo uma seqüência suave. O processo pode ser aplicado utilizando-se o mesmo objeto em posições diferentes ou utilizando-se dois objetos distintos. O primeiro método é chamado de animação *free form* e o segundo de *morphing* de polígonos.

A interpolação de formas pode ser aplicada tanto para o espaço tridimensional quanto para o bidimensional. No espaço 2D talvez seja mais difícil de conseguir o resultado desejado já que as formas 2D são uma projeção de objetos 3D. Assim, objetos 2D têm uma representação parcial, enquanto que no espaço 3D o objeto está inteiramente representado.

As aplicações da interpolação de formas vão desde animações de personagens bidimensionais a casamento de padrões e modelagem geométrica. Outra aplicação em computação gráfica é a eliminação de efeitos de canto em modelos 3D formados por polígonos.

Um dos primeiros trabalhos nesta área foi o de Sederberg [8], que desenvolveu um algoritmo de correspondência de pontos, baseado na interpolação dos vértices. Assim, considerando os ângulos formados pelos vértices, uma função de custo é formulada. Essa função tenta relacionar as curvas com pedaços de metal. Assim, as deformações sofridas pelo contorno são associadas às propriedades físicas dos materiais, como o alongamento e a flexão.

A interpolação de formas possui vários métodos propostos [9],[10],[11]. Em [12], Johan *et al.* propuseram um método de misturar mais de duas formas. Em [13], o polígono é dividido em subpolígonos, sendo as divisões feitas manualmente, obtendo uma interpolação mais próxima da percepção

humana.

A correspondência é feita normalmente entre pontos [8],[5], mas outras representações podem ser utilizadas, e a correspondência e interpolação são feitas nessa nova representação. Em [14], Shapira e Rappoport descrevem uma representação em forma de estrela utilizada para fazer a interpolação entre polígonos.

Os dois métodos mais utilizados são a interpolação linear e a por lados e ângulos [15]. O objetivo da correspondência de pontos é estabelecer a relação entre os dois polígonos. O objetivo da interpolação é o de determinar como o polígono de origem é transformado no polígono de destino, ou seja, determinar o caminho que os pontos irão percorrer. Também é possível realizar a interpolação de formas através de uma triangulação compatível entre dois polígonos [16][17].

Um método universal para todos os casos é praticamente impossível. Dependendo do resultado desejado, um algoritmo é mais apropriado de que outro, além das abordagens dos procedimentos aplicados em cada caso poderem ser incompatíveis. A falta de informações 3D dos objetos representados pelos contornos apresenta mais uma dificuldade. O resultado que o usuário espera obter é baseado nos objetos que ele associa a aquele contorno. Para o computador fazer a mesma tarefa, seria necessária uma biblioteca de objetos tão vasta quanto a existente na mente de um artista, e, mesmo assim, os métodos para se utilizar essas informações são bem variados. Por isso, essa área se concentra na construção de uma ferramenta, e não em um substituto para o artista. O grande desafio é encontrar um meio de se permitir o uso da criatividade na ferramenta, e, ao mesmo tempo, facilitar o trabalho do animador.

Dentro dessas duas etapas da interpolação de formas, a correspondência entre as formas e a interpolação dos atributos, o método proposto neste trabalho utiliza uma métrica de polígonos para melhorar a correspondência inicial e aplica a correspondência de lados no lugar da de pontos. Utiliza-se para interpolação o método proposto por Sederberg *et al.* em [15], por lados e ângulos.

O critério para uma boa interpolação é que a transformação deve ser visualmente suave. Efeitos não naturais, como auto-intersecção, encolhimento e distorção de polígonos intermediários devem ser evitados. O mais importante é que os polígonos intermediários devem ter características dos dois polígonos, ou seja, do de origem e do de destino. Essas são as metas buscadas neste trabalho.

2.1 Correspondência de pontos

Essa é a etapa mais importante na interpolação de formas. Suponha, por exemplo, uma forma humana. Neste caso, ela controla a transformação de uma parte representando a cabeça no polígono de origem em outra parte no polígono de destino que também representa a cabeça. Se a correspondência for adequada, essas duas partes serão correlacionadas, mas, se nessa etapa, o algoritmo utilizado

não for apropriado, a cabeça poderá se transformar, por exemplo, no pé. Apesar de existir um valor artístico nesse tipo de transformação, dependendo do resultado esperado, isso é indesejado.

A representação mais comum de uma figura geométrica é por uma lista de pontos. A correspondência de pontos faz com que cada elemento em uma lista tenha o seu correspondente na outra. Com isso, cada ponto da figura geométrica terá um ponto correspondente na outra figura, ou seja, uma correspondência de um para um.

A correspondência é calculada utilizando-se uma função de custo. Essa função leva em conta as tangentes da figura geométrica e a distância da diagonal do gráfico de correspondência dos pontos. Nesta seção descreve-se a função de custo utilizada por Johan *et al.* em [5].

2.1.1 Representação

O polígono é uma figura geométrica constituída por segmentos de retas que formam uma figura. Sua descrição pode ser feita através da lista de pontos que definem os segmentos de retas.

$$P = [P_0, P_1, \dots, P_n] \quad (2.1)$$

$$(2.2)$$

onde n é o número de pontos do polígono e $P_i = (x_i, y_i)$. Note que, para o polígono ser fechado, $P_0 = P_n$.

2.1.2 Cálculo da correspondência

Sejam dados dois polígonos P^S e P^T com n_S e n_T pontos, respectivamente. Primeiro, calcula-se a tangente em cada um dos pontos da figura geométrica, utilizando-se da fórmula:

$$\vec{P}_i^* = \frac{P_{i+1}^* - P_{i-1}^*}{\|P_{i+1}^* - P_{i-1}^*\|} \quad (2.3)$$

onde P^* representa tanto P^S quanto P^T . Como $P_0 = P_n$ o cálculo da tangente utiliza P_{n-1} e P_1 nos lugares de P_{i-1} e P_{i+1} .

Define-se a operação Z como:

$$Z(\vec{A}, \vec{B}) = A_x B_y - A_y B_x \quad (2.4)$$

O custo em relação a tangente da curva é dado por:

$$\text{angle}(P_i^*) = \frac{1}{2} \arccos(\overrightarrow{P_{i-1}^*} \cdot \overrightarrow{P_{i+1}^*}) \text{sign}(Z(\overrightarrow{P_{i+1}^*}, \overrightarrow{P_{i-1}^*})) \quad (2.5)$$

onde a função $\text{sign}()$ é definida como:

$$\text{sign}(x) = \begin{cases} -1 & \text{para } x < 0 \\ 1 & \text{para } x \geq 0 \end{cases} \quad (2.6)$$

O custo total é calculado pela equação seguinte.

$$\text{cost}(i, j) = w_1 |\text{angle}(P_i^S) - \text{angle}(P_j^T)| + w_2 \left| \frac{i}{n_S} - \frac{j}{n_T} \right| \quad (2.7)$$

A Equação (2.5) leva em conta os ângulos das tangentes que faz com que pontos com valores de tangentes próximos sejam correlacionados. O termo $|i/n_S - j/n_T|$ na Equação (2.7) se refere à distância da diagonal no grafo de programação de dinâmica. A aplicação desse termo resulta na redução da ocorrência de múltiplos pontos ligados a apenas um ponto na outra figura geométrica. Os pesos w_1 e w_2 controlam o balanço dessas duas características.

A correspondência é obtida minimizando-se a função de custo em relação aos pontos:

$$\min \sum_{i=1}^{n_S} \text{cost}(i, j) \quad (2.8)$$

Ao final do processo, cada índice i estará ligado a um índice j , sendo que o polígono é reescrito de acordo com a correspondência obtida. Os pontos que tiverem mais de um ponto correspondente são repetidos na nova descrição do polígono.

2.2 Interpolação

Para a interpolação, assume-se que os polígonos tenham o mesmo número de pontos e lados. Esse problema é resolvido pelo processo de correspondência de pontos entre polígonos, sendo que, no final do processo, cada ponto de um polígono tem um ponto correspondente no outro.

2.2.1 Interpolação linear

O método mais simples de interpolar um polígono é a interpolação linear dos pontos. Dados dois polígonos P^A e P^B , a interpolação seria calculada utilizando-se a Equação (2.9). Essa equação supõe

que os dois polígonos tenham o mesmo número de pontos.

$$P_i^I = (1 - t) P_i^A + t P_i^B \quad (2.9)$$

Por ser um processo simples, a interpolação linear pode apresentar certos problemas dependendo das formas e das posições das figuras. Um problema que pode ocorrer é que, durante a interpolação, o ângulo interno formado por duas retas pode passar pelo zero e ficar com valores negativos. Geometricamente, isso significa que a parte de dentro da figura, em algum momento da interpolação, transformou-se na parte de fora, o que não é desejado e não condiz com a topologia da figura. A interpolação linear também pode causar o cruzamento de lados não consecutivos. Esse problema também é conhecido como auto-cruzamento do polígono.

2.2.2 Interpolação por lados e ângulos

Um meio de resolver o problema de auto-cruzamento de lados é mudar a representação da figura geométrica. A representação de pontos é modificada para a de comprimentos de lados e ângulos, sendo que esses novos parâmetros são interpolados. Essa técnica de interpolação foi introduzida em [15].

Considera-se que os dois polígonos têm o mesmo número de pontos, n . Então o número de lados é igual a $m = n - 1$, lembrando que para o polígono ser fechado tem-se $P_0 = P_n$.

A lista de comprimento de lados é calculada por:

$$L_i^* = |P_{i+1}^* - P_i^*| \quad (2.10)$$

onde $i = 1, \dots, m$.

A lista de ângulos é calculada por:

$$\theta_i^* = \arccos \left(\frac{P_{i+2}^* - P_{i+1}^*}{L_{i+1}} \cdot \frac{P_{i+1}^* - P_i^*}{L_i} \right) \text{sign} \left(Z \left(\frac{P_{i+2}^* - P_{i+1}^*}{L_{i+1}}, \frac{P_{i+1}^* - P_i^*}{L_i} \right) \right) \quad (2.11)$$

A Equação (2.14) e Equação (2.15) descrevem a interpolação para a nova representação da figura geométrica.

$$P_0^I = (1 - t) P_0^A + t P_0^B \quad (2.12)$$

$$\alpha_0^I = (1 - t) \alpha_0^A + t \alpha_0^B \quad (2.13)$$

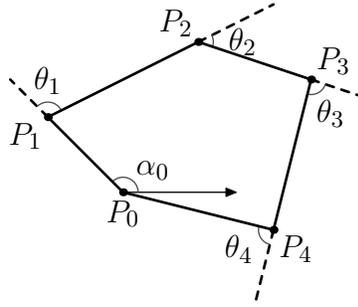


Fig. 2.1: Parâmetros da representação de um polígono por lados e ângulos.

$$L_i^I = (1 - t) L_i^A + t L_i^B \quad (2.14)$$

$$\theta_i^I = (1 - t) \theta_i^A + t \theta_i^B \quad (2.15)$$

O parâmetro α_0 é o ângulo do primeiro lado da lista com o eixo x . Como essa é uma representação relativa, o único parâmetro que a junta à representação com as coordenadas globais é o termo α_0 e o primeiro ponto do polígono P_0^* . A posição global intermediária é calculada através da interpolação linear desse primeiro ponto. A Fig. 2.1 ilustra os parâmetros da representação nos polígonos.

Ajuste no comprimento dos lados

Uma das desvantagens da interpolação por lados e ângulos é que, nos polígonos intermediários, o ponto inicial e final podem não coincidir, ou seja, $P_0 \neq P_n$. Como essa interpolação não garante que o resultado entre duas curvas fechadas resulte em uma curva fechada, um algoritmo de correção é utilizado para garantir o fechamento do polígono.

O comprimento do lado deve ser ajustado reescrevendo a Equação (2.14) como:

$$L_i^I = (1 - t) L_i^A + t L_i^B + S_i \quad (2.16)$$

onde $i = 1, \dots, m$.

Se dois lados tiverem o mesmo comprimento nos dois polígonos, a lógica seria que durante a interpolação o comprimento permanecesse constante. Então, S_i deve ser proporcional a $\|L_{A_i} - L_{B_i}\|$.

$$L_{AB_i} = \max(\|L_{A_i} - L_{B_i}\|, L_{tol}) \quad (2.17)$$

onde $L_{tol} = 0,001$ é utilizado para evitar divisões por zero. Para melhorar a leitura das equações, utiliza-se o termo L_{A_i} no lugar de L_i^A , assim como L_{B_i} no lugar de L_i^B .

O objetivo é calcular S_i para todos os lados, de tal forma a minimizar a função:

$$f(S_1, S_2, \dots, S_m) = \sum_{i=0}^m \frac{S_i^2}{L_{AB_i}^2} \quad (2.18)$$

A minimização está sujeita as restrições:

$$\varphi_1(S_1, S_2, \dots, S_m) = \sum_{i=0}^m [(1-t)L_{A_i} + tL_{B_i} + S_i] \cos \alpha_i = 0 \quad (2.19)$$

$$\varphi_2(S_1, S_2, \dots, S_m) = \sum_{i=0}^m [(1-t)L_{A_i} + tL_{B_i} + S_i] \sin \alpha_i = 0 \quad (2.20)$$

onde α_i são os ângulos entre o eixo x e os vetores $\overrightarrow{P_i P_{i+1}}$.

As equações (2.19) e (2.20) garantem que o polígono será fechado. A Equação (2.19) restringe o deslocamento sobre o eixo x , ao percorrer todos os pontos do polígono. O deslocamento total deve ser zero, já que P_0 deve ser igual a P_n . A Equação (2.20) garante a mesma condição para a coordenada y .

$$\alpha_i = \alpha_{i-1} + \theta_i \quad (2.21)$$

Resolve-se pelo método de Lagrange a equação a seguir:

$$\Phi(\lambda_1, \lambda_2, S_1, S_2, \dots, S_m) = f + \lambda_1 \varphi_1 + \lambda_2 \varphi_2 \quad (2.22)$$

onde λ_1 e λ_2 são os multiplicadores.

Do sistema de equações:

$$\begin{cases} \frac{\partial \Phi}{\partial S_i} = \frac{2 S_i}{L_{AB_i}^2} + \lambda_1 \cos \alpha_i + \lambda_2 \sin \alpha_i = 0 \\ \sum_{i=0}^m [(1-t)L_{A_i} + tL_{B_i} + S_i] \cos \alpha_i = 0 \\ \sum_{i=0}^m [(1-t)L_{A_i} + tL_{B_i} + S_i] \sin \alpha_i = 0 \end{cases} \quad (2.23)$$

Obtêm-se:

$$\begin{cases} E\lambda_1 + F\lambda_2 = U \\ F\lambda_1 + G\lambda_2 = V \end{cases} \quad (2.24)$$

onde os termos E , F , G , U e V são descritos como:

$$E = \sum_{i=0}^m L_A^2 B_i \cos^2 \alpha_i \quad (2.25)$$

$$F = \sum_{i=0}^m L_A^2 B_i \sin \alpha_i \cos \alpha_i \quad (2.26)$$

$$G = \sum_{i=0}^m L_A^2 B_i \sin^2 \alpha_i \quad (2.27)$$

$$U = 2 \left\{ \sum_{i=0}^m [(1-t) L_{A_i} + t L_{B_i}] \cos \alpha_i = 0 \right\} \quad (2.28)$$

$$V = 2 \left\{ \sum_{i=0}^m [(1-t) L_{A_i} + t L_{B_i}] \sin \alpha_i = 0 \right\} \quad (2.29)$$

Resolvendo o sistema, obtêm-se os valores de λ_1 e λ_2 . A condição de $EG - F^2 \neq 0$ deve ser satisfeita para evitar uma divisão por zero.

$$\lambda_1 = \left| \begin{array}{cc|cc} U & F & E & F \\ V & G & F & G \end{array} \right| \quad (2.30)$$

$$\lambda_2 = \left| \begin{array}{cc|cc} E & U & E & F \\ F & V & F & G \end{array} \right| \quad (2.31)$$

Assim, os valores de ajuste dos comprimentos de lados são dados por:

$$S_i = -\frac{1}{2} L_{AB_i}^2 (\lambda_1 \cos \alpha_i + \lambda_2 \sin \alpha_i) \quad (2.32)$$

Utilizando a Equação (2.15), Equação (2.16) e Equação (2.21), as novas coordenadas podem ser calculadas por:

$$\begin{aligned} x_i &= x_{i-1} + L_{i-1} \cos \alpha_{i-1} \\ y_i &= y_{i-1} + L_{i-1} \sin \alpha_{i-1} \end{aligned} \quad (2.33)$$

Esse algoritmo só funciona se as pontas das curvas estiverem próximas, o que se enquadra exatamente com o problema gerado pela interpolação de lados e ângulos. Assim pode-se assumir que esse algoritmo garante que os polígonos gerados pela interpolação sejam fechados.

2.3 Métrica de polígonos

A interpolação entre dois polígonos tenta manter as partes similares e transformar suavemente as partes diferentes. Uma métrica de polígonos é útil para quantificar a diferença entre dois polígonos, provendo parâmetros de como correlacioná-los.

Nesta seção é apresentada uma métrica sugerida por Arkin *et al.* em [18]. Essa métrica é utilizada na correspondência de polígonos do método proposto descrito na Seção 2.5.

2.3.1 Representação

O polígono é representado por uma função que descreve o valor da tangente em um ponto que percorre todo o perímetro do polígono, como mostrado na Fig. 2.2. A representação do polígono é feita por uma função $\Theta(s)$, que representa o valor do ângulo da tangente pelo perímetro s . É importante que os polígonos sejam percorridos sempre no mesmo sentido para que a comparação seja condizente. Neste trabalho adotou-se o sentido anti-horário. O perímetro do polígono é normalizado, ou seja, o polígono é representado no intervalo $[0,1]$. A função é considerada periódica, isto é, a função continua percorrendo o polígono indefinidamente. Assim a condição $\Theta(s + 1.0) = \Theta(s) + 2\pi$ deve ser satisfeita para representar um polígono fechado.

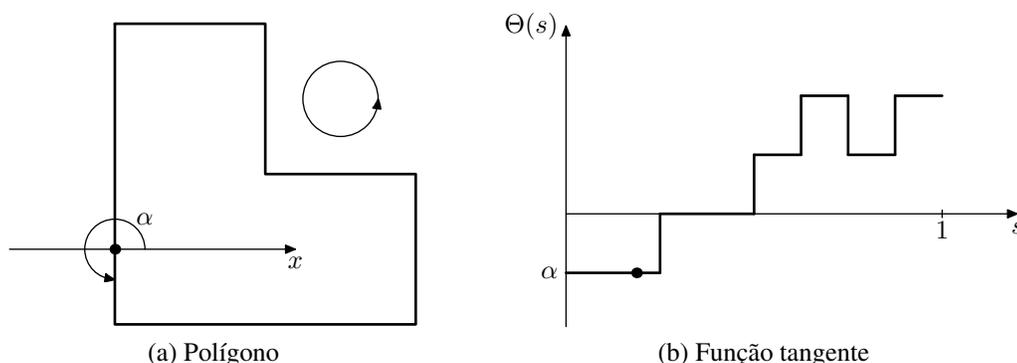


Fig. 2.2: Representação de um polígono por uma função $\Theta(s)$.

A Fig. 2.3 mostra um exemplo de representação de polígono.

2.3.2 Cálculo da distância

Utilizando a representação descrita na subseção anterior, define-se a distância entre dois polígonos A e B como sendo a diferença entre as suas funções de representação:

$$D_p^{A,B} = \left(\int_0^1 |\Theta_A(s) - \Theta_B(s)|^p ds \right)^{\frac{1}{p}} \quad (2.34)$$

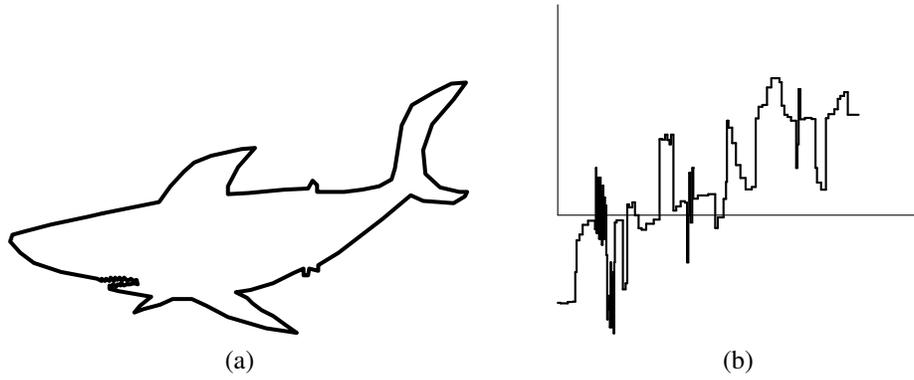


Fig. 2.3: Exemplo de um polígono representado e sua representação por uma função $\Theta(s)$.

onde $\Theta(s)$ é a função de representação do polígono e p controla o tipo de distância utilizado.

Definindo a distância como a diferença de duas funções de representação, não se garante um valor de distância único. A Equação (2.34) descreve uma distância que depende do ponto de início dos polígonos e da rotação entre eles. Como a função de representação $\Theta(s)$ pode deslocar-se no eixo s dependendo do ponto por qual se começa a descrever o polígono, o valor da distância será afetado por esse deslocamento. Assim, dois polígonos idênticos, mas com pontos P_0 diferentes, produzem valores de distância diferentes. Outro fator que influencia o valor da distância é o ângulo de rotação entre os polígonos. Se dois polígonos idênticos tiverem um ângulo de rotação entre si, o valor da distância não será zero. Isso se deve à característica da função de representação $\Theta(s)$ que descreve o ângulo em relação ao eixo x . Assim, a representação depende da posição do polígono.

Uma boa métrica deve fornecer o mesmo valor de distância para o parâmetro que está sendo medido, neste caso a similaridade. Ou seja, não importa que os polígonos estejam em posições diferentes, a distância deve medir as diferenças de similaridade e não de posição.

Para obter uma métrica que não seja afetada por esses fatores, a Equação (2.35) apresenta uma fórmula de distância que procura o valor mínimo considerando todas as variações de rotação e de ponto de início do polígono.

$$\begin{aligned} \|\Theta_A - \Theta_B\|_p &= \min_{\theta \in \mathfrak{R}, t \in [0,1]} D_p^{A,B}(t, \theta) \\ &= \min_{\theta \in \mathfrak{R}, t \in [0,1]} \left(\int_0^1 |\Theta_A(s+t) - \Theta_B(s) + \theta|^p ds \right)^{\frac{1}{p}} \end{aligned} \quad (2.35)$$

onde t representa a variação do ponto de início e θ a rotação entre polígonos.

A rotação entre os polígonos pode ser descrita em função da variação do ponto de início t , como mostra a Equação (2.36).

$$\theta(t) = \int_0^1 \Theta_B(s) ds - 2\pi t - \int_0^1 \Theta_A(s) ds \quad (2.36)$$

O algoritmo utilizado para calcular a métrica varia o ponto de início do polígono. Assim $P_0^{*rot} = P_i^*$. A distância entre os polígonos é o menor valor aferido entre todas as variações de P_0^{rot} . O Algoritmo 2.1 descreve o cálculo de distância.

Algoritmo 2.1: Cálculo da distância entre polígonos.

$D_{min} = \infty$;

para cada ponto P_i^A **do polígono A faça**

para cada ponto P_j^B **do polígono B faça**

 Rotacione o polígono A tal que, $P^{Arot} = [P_i^A, \dots, P_{N-1}^A, P_0^A, \dots, P_i^A]$;

 Rotacione o polígono B tal que, $P^{Brot} = [P_j^B, \dots, P_{M-1}^B, P_0^B, \dots, P_j^B]$;

u =comprimento de P_0^A até P_i^A normalizado pelo comprimento total do polígono A;

v =comprimento de P_0^B até P_j^B normalizado pelo comprimento total do polígono B;

$t = v - u$;

 Converta para a representação $\Theta_A(s)$ e $\Theta_B(s)$;

 Calcule $\theta(t)$;

 Calcule $D = \left(\int_0^1 |\Theta_A(s+t) - \Theta_B(s) + \theta|^p ds \right)^{\frac{1}{p}}$;

se $D < D_{min}$ **então**

$D = D_{min}$;

$P_{i_{min}}^A = P_i^A$;

$P_{j_{min}}^B = P_j^B$;

fim

fim

fim

Distância= D_{min} ;

2.4 Redução do número de vértices de polígonos

O reconhecimento de formas e de quão diferentes elas são entre si é uma das habilidades que facilitam ao ser humano fazer interpolações. Assim, o uso de um algoritmo que consiga descrever uma forma básica através de poucos pontos pode ser utilizado para a correspondência entre polígonos. O algoritmo descrito aqui foi proposto por Latecki *et al.* em [19].

Contornos retirados de imagens digitais são distorcidos devido ao ruído de digitalização e erros de segmentação. Para obter o contorno desejado, é necessária a eliminação dessas distorções, mas ao mesmo tempo, preservando a aparência perceptual suficiente para o reconhecimento do objeto.

Um modo de se fazer isso é aproximando o contorno por outro que tenha uma aparência perceptual similar.

Qualquer curva digital pode ser considerada um polígono sem perda de informações, desde que o número de vértices para representá-lo seja suficiente. Assim, o estudo da redução de vértices para formas poligonais é válido para qualquer contorno.

A idéia básica da redução de pontos de um polígono é simples. Em cada passo, um par consecutivo de segmentos de linha s_1, s_2 é substituído por um único segmento de linha unindo as pontas de $s_1 \cup s_2$.

O segredo desse algoritmo é a ordem em que esses segmentos de linha são substituídos. Essa substituição é feita de acordo com uma medida de relevância descrito na Equação (2.37).

$$K(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{l(s_1) + l(s_2)} \quad (2.37)$$

onde $\beta(s_1, s_2)$ é o ângulo no vértice comum entre os segmentos s_1 e s_2 e l é o comprimento do segmento de reta normalizado em relação ao comprimento do polígono inteiro.

Quanto maior for o valor de $K(s_1, s_2)$, maior será a contribuição dos segmentos s_1 e s_2 para o formato da curva. Assim, retirando os segmentos menos relevantes do polígono, reduzem-se o número de pontos, mas a semelhança visual com o original é mantida.

Na Fig. 2.4 e Fig. 2.6 são mostrados dois exemplos de redução de pontos, sendo um polígono em forma de tubarão e outro em forma de pato, com diferença de 10 pontos entre cada decomposição. A Fig. 2.5 e Fig. 2.7 ilustram a redução ponto a ponto. Nelas, percebe-se que mesmo com poucos pontos a forma básica ainda é preservada.

2.5 Método proposto

O método proposto utiliza a métrica de polígonos para achar a correspondência inicial, e utiliza redução de vértices de polígonos, que decompõe os polígonos em formas mais básicas, tentando melhorar o processo de correspondência. A idéia de decompor o polígono já foi utilizada para interpolação de formas por Yang *et al.* em [13], mas a decomposição foi feita de modo manual. Outra proposta é a utilização da correspondência de lados no lugar da de pontos, visando evitar o aparecimento de lados de comprimento zero na interpolação. Essa característica é desejável para a utilização nos algoritmos de deformação de imagens, descrito no Capítulo 3. A função de custo utilizada é similar à descrita na Subseção 2.1.2.

A contribuição desse método se restringe à parte de correspondência de polígonos. Na parte de interpolação, é utilizado o método descrito na Subseção 2.2.2, por lados e ângulos.

Nas seções seguintes são utilizados um polígono de origem, P^A , com M pontos e outro de destino,

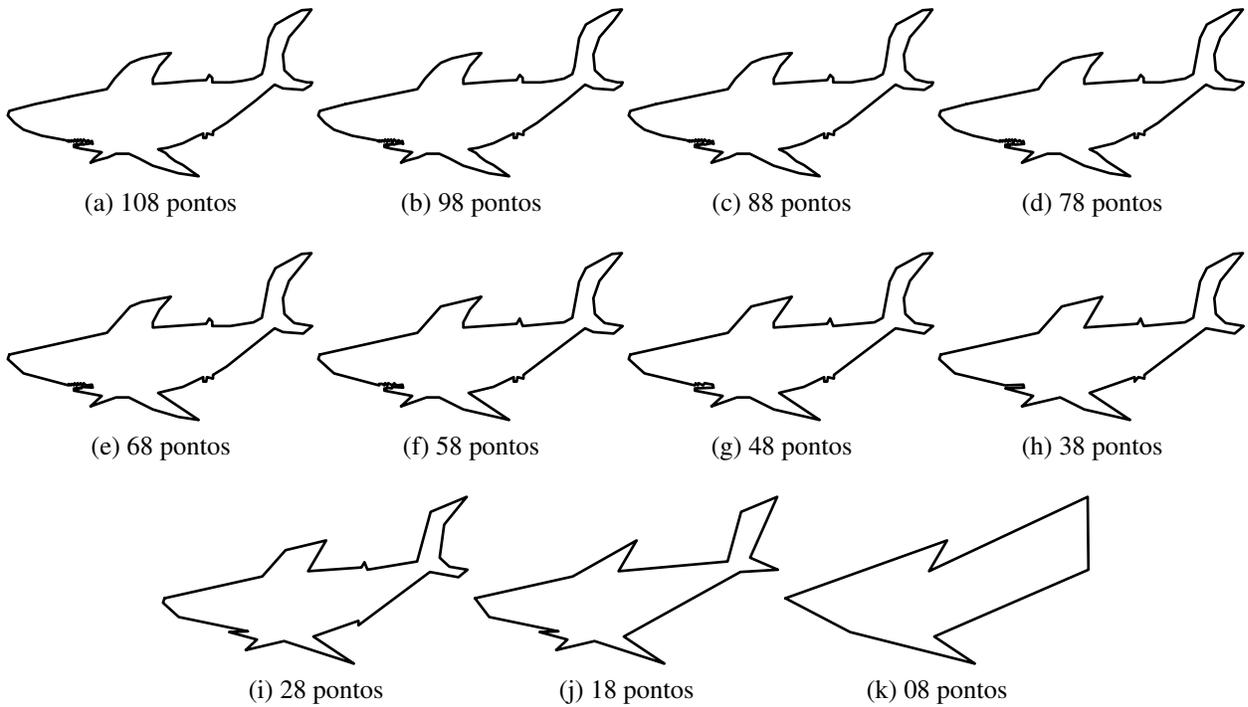


Fig. 2.4: Sequência de redução de pontos de um polígono em forma de tubarão. A sequência foi amostrada com uma diferença de 10 pontos entre os polígonos.

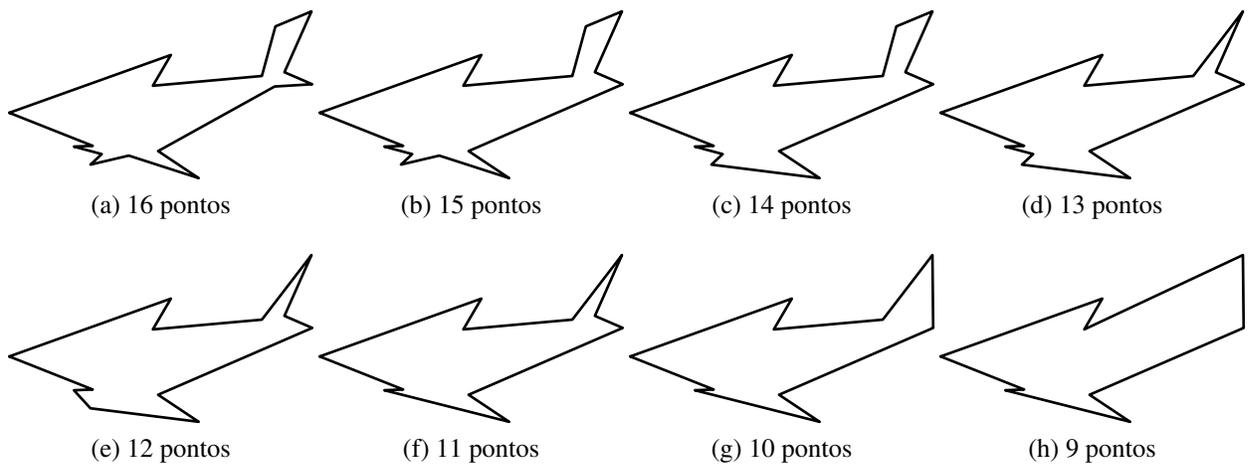


Fig. 2.5: Detalhes da sequência de redução de pontos de um polígono em forma de tubarão.

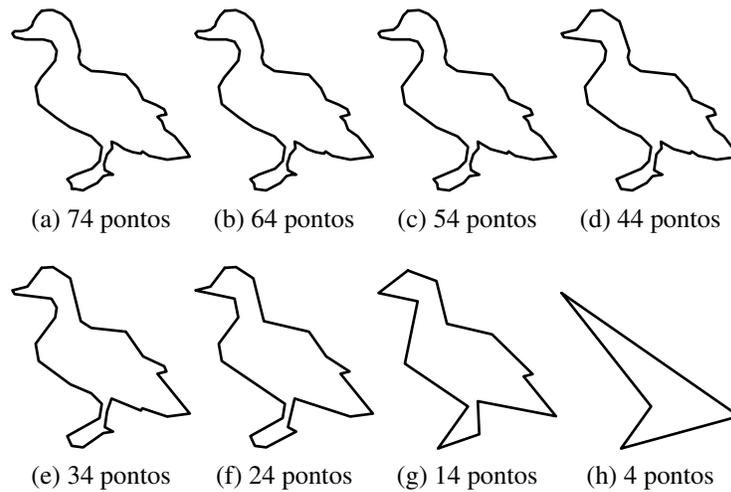


Fig. 2.6: Seqüência de redução de pontos de um polígono em forma de pato. A seqüência foi amostrada com uma diferença de 10 pontos entre os polígonos.

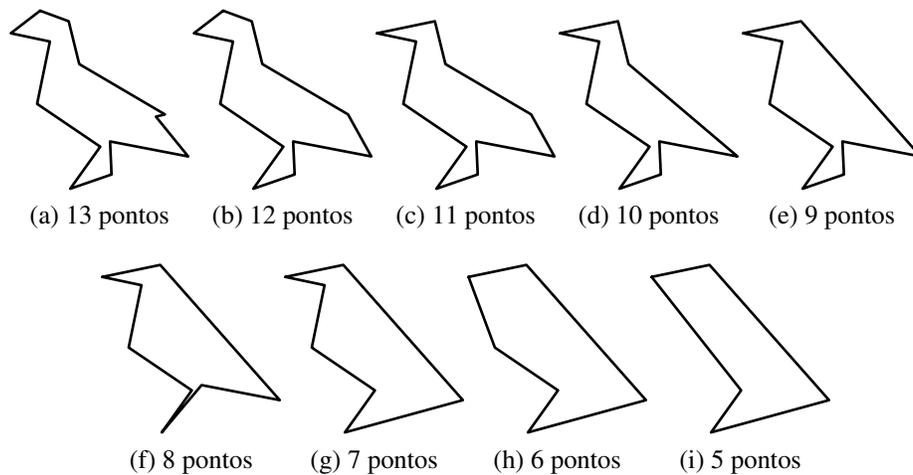


Fig. 2.7: Seqüência de redução mais detalhada.

P^B com N pontos.

2.5.1 Utilizando a métrica para achar a correspondência inicial

A métrica pode ser utilizada para se ter uma idéia quantitativa de distância entre os polígonos. Essa medida é utilizada para fazer a correspondência inicial, onde a primeira correspondência é feita pelo algoritmo de distância entre polígonos. Depois, no cálculo do resto da correspondência, os polígonos são tratados como curvas abertas. A idéia do algoritmo de distância é escolher o melhor ponto P_0 para os dois polígonos. Assim, tem-se já uma correspondência do ponto inicial. Um dos

efeitos esperados é obter assim uma melhor correlação.

Ao encontrar o valor de distância entre os polígonos, encontra-se também o valor de t . Esse valor representa o deslocamento entre as funções $\Theta(s)$, e também define os pontos iniciais dos polígonos.

Considera-se que, ao variar o ponto de início do polígono, isso seja equivalente a deslocar a função por um valor u . Então, as variações de ponto no polígono rotacionado $P^{A_{rot}}$ podem ser representadas na função de representação como:

$$P_0^{A_{rot}} = P_i^A \rightarrow \Theta_A(s - u_i) \quad (2.38)$$

onde u_i é o comprimento entre P_i^A e P_0^A normalizado pelo perímetro do polígono P^A .

O mesmo pode ser feito para P_B :

$$P_0^{B_{rot}} = P_i^B \rightarrow \Theta_B(s - v_i) \quad (2.39)$$

onde v_i é o comprimento entre P_i^B e P_0^B normalizado pelo perímetro do polígono P^B .

A relação entre o termo t e os pontos de início é dada por:

$$t = v_i - u_i \quad (2.40)$$

Na prática, é mais simples variar o ponto de início de descrição e calcular t a partir da Equação (2.40), como descrito no Algoritmo 2.1.

O uso da compensação de rotação na correspondência inicial depende de parâmetros da função de correspondência. Neste trabalho, é utilizado $p = 2$ na fórmula de distância. Assim, a distância entre polígonos é descrita por:

$$\min_{\theta \in \mathbb{R}, t \in [0,1]} \left(\int_0^1 |\Theta_A(s+t) - \Theta_B(s) + w_{rot} \theta(t)|^2 ds \right)^{\frac{1}{2}} \quad (2.41)$$

onde w_{rot} controla o uso do termo $\theta(t)$ que compensa a rotação entre polígonos. Esse termo está relacionado com o processo de correspondência descrito na Subseção 2.5.2.

A função de representação do polígono pode ser descrita de forma discreta, listando o valor do ângulo tangente e o comprimento do lado normalizado pelo comprimento total do polígono.

$$\Theta(s) \Rightarrow \begin{cases} \Theta(i) = [\Theta(0), \dots, \Theta(n)] \\ C(i) = [C(0), \dots, C(n)] \end{cases} \quad (2.42)$$

onde $\Theta(i)$ e $C(i)$ são os valores do ângulo e do comprimento do lado i , respectivamente.

Esses parâmetros se relacionam com os parâmetros descritos na Subseção 2.2.2 da seguinte forma:

$$\Theta_A(i) = \alpha_i^A \quad (2.43)$$

$$C_A(i) = \frac{L_i^A}{\sum_{i=1}^M (L_i^A)} \quad (2.44)$$

Após os pontos de início dos polígonos serem obtidos através do algoritmo de distância, os polígonos são reescritos para começar a partir desses pontos. O resto da correspondência é feito tratando os polígonos como curvas abertas.

2.5.2 Correspondência de lados

Como dito anteriormente, uma condição necessária para a interpolação é que os polígonos ou curvas tenham o mesmo número de pontos e lados. Essa equalização na topologia é feita a partir da correspondência, onde é determinada a relação de pontos entre os dois polígonos.

Normalmente, é utilizado a correspondência de pontos, onde os pontos dos polígonos são correlacionados. Mas, utilizando os lados no lugar dos pontos, pode-se fazer uma correlação que impede a aparição de lados de comprimento zero, adaptando o algoritmo de interpolação para a aplicação em algoritmos de deformação de imagens. Isso é uma característica muito importante, pois os algoritmos de deformação de imagens, quando dois pontos estão correlacionados a um único ponto, têm a tendência de produzir dobras na imagem deformada.

A correspondência é feita utilizando a função de custo descrita na Equação (2.45), sendo que a representação utilizada é descrita na Equação (2.42). A correspondência tenta relacionar os lados que são similares na inclinação. A compensação de rotação pode ser utilizada na correspondência para obter resultados mais interessantes.

Função de custo

A função de custo é baseada na diferença entre os ângulos dos lados dos polígonos e também na distribuição da correspondência. A diferença dos ângulos $\Theta_A(i) - \Theta_B(j)$ faz com que os lados que estão com a mesma inclinação sejam correlacionados. Já o termo $|i/M - j/N|$ não permite que a correspondência fique concentrada em um ponto ou região. Por exemplo, todos os lados do polígono de origem poderiam corresponder a um único lado do polígono de destino, produzindo um resultado indesejado.

$$custo(i, j) = w_1 |\Theta_A(i) - \Theta_B(j) - w_{rot} \theta| + w_2 \left| \frac{i}{M} - \frac{j}{N} \right| \quad (2.45)$$

onde w_1 e w_2 são os pesos na diferença de ângulos e distribuição respectivamente e w_{rot} faz com que a função de custo compense ou não a rotação entre os polígonos. Se w_{rot} é igual a 1, a função de custo tenta compensar a rotação entre os polígonos. Se é igual a 0 ela tenta fazer a correspondência não considerando a diferença de rotação entre os polígonos. A variável θ é a mesma variável $\theta(t)$ da Equação (2.36).

A correspondência é obtida quando a função da Equação (2.45) for minimizada. Para isso calcula-se uma matriz de custo total de acordo com o Algoritmo 2.2. A correspondência é dada pela lista $cmín$, onde o lado C_i^A corresponde ao lado C_j^B .

Algoritmo 2.2: Cálculo da correspondência utilizando a função de custo.

```

para cada elemento  $[i, j]$  da matriz  $CustoTotal$  faça
  |  $CustoTotal(i, j) = custo(i, j) + \min(custo(i - 1, j), custo(i, j - 1), custo(i - 1, j - 1))$ ;
fim
Faça  $[i, j] = [M, N]$ ;
enquanto  $[i, j]$  não é o início da matriz faça
  | Adicione  $[i, j]$  a lista  $cmín$ ;
  | Escolha  $[i_{min}, j_{min}]$  entre os elementos  $[i - 1, j], [i, j - 1], [i - 1, j - 1]$  da matriz
  |  $CustoTotal$  que tenha o menor valor;
  | Faça  $[i, j] = [i_{min}, j_{min}]$ ;
fim

```

Um exemplo de correspondência pode ser visto na Fig. 2.8c. A linha tracejada é o caminho de menor custo calculado pelo Algoritmo 2.2. O processo completo do método proposto é descrito no Algoritmo 2.3. Nesse algoritmo, o caminho não pode ter um passo para baixo seguido de um passo para a direita, nem um passo para a direita seguido de um para baixo. Esse caminho pode ser representado como uma curva de 90 graus na Fig. 2.8c. A correspondência gerada por esse caminho impossibilita a divisão dos polígonos, pois um grupo de lados é correlacionado com outro grupo de lados a partir de um lado, não ficando claro a qual grupo esse lado pertence.

Depois de a correspondência ser obtida, os polígonos são divididos de modo que os dois tenham o mesmo número de lados. A Fig. 2.9 ilustra como a divisão entre os lados correspondentes é feita. Se um lado do polígono A corresponder a três lados do polígono B , o lado do polígono A é dividido em três partes proporcionais aos comprimentos dos três lados do polígono B .

2.5.3 Utilizando a redução de vértices como pré-processamento

Às vezes, a correspondência baseada apenas em uma função de custo não leva a resultados satisfatórios. Utilizando a idéia de decompor o polígono em partes para a correspondência, aplica-se

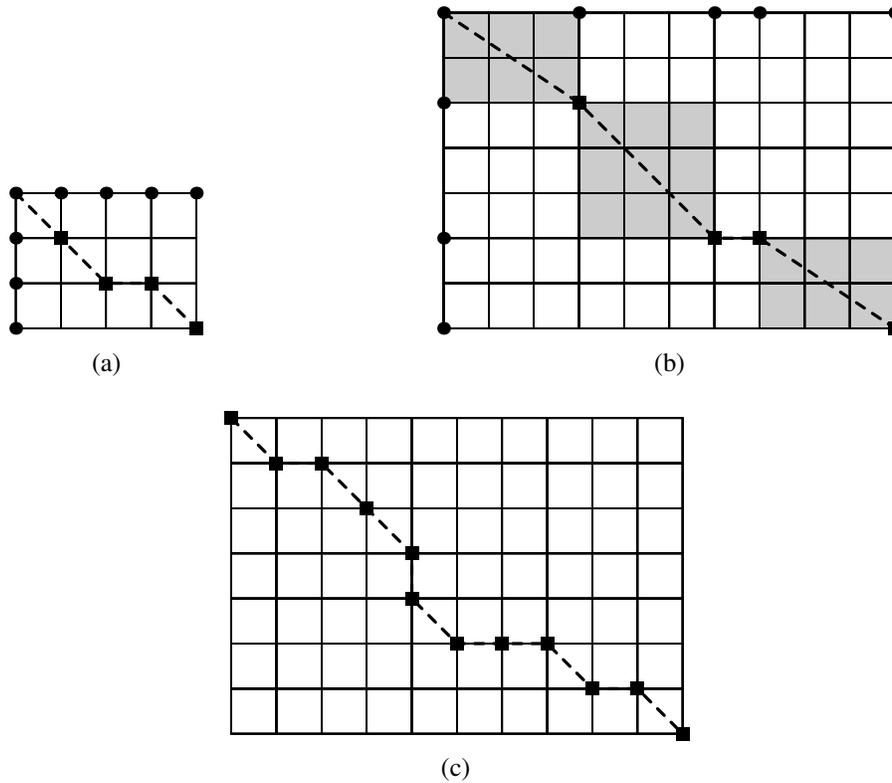


Fig. 2.8: Processo de correspondência entre polígonos. (a) Processo de correspondência entre polígonos reduzidos. Os círculos representam os pontos dos polígonos reduzidos e os quadrados as correspondências. (b) Correspondência entre os polígonos completos utilizando a correspondências dos polígonos reduzidos como restrições. (c) Correspondência final entre os polígonos.

Algoritmo 2.3: Método proposto para interpolação de formas.

início

Encontre θ e os pontos $P_{i_{min}}^A$ e $P_{j_{min}}^B$ através da métrica de polígonos;
 Reescreva o polígono A para $P^A = [P_i^A, \dots, P_{N-1}^A, P_0^A, \dots, P_i^A]$ com $i = i_{min}$;
 Reescreva o polígono B para $P^B = [P_j^B, \dots, P_{M-1}^B, P_0^B, \dots, P_j^B]$ com $j = j_{min}$;
 Correlacione os polígonos através da função de custo;
 Divida os polígonos de acordo com a correlação;

para cada quadro da animação faça

| Interpole os lados e ângulos de acordo com um parâmetro de transição t ;

fim

fim

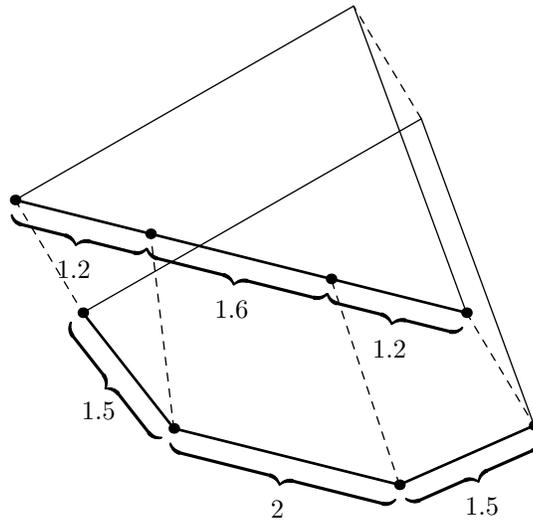


Fig. 2.9: Exemplo de correspondência de lados. Um lado do polígono superior é conectado com três lados do polígono inferior.

aqui o algoritmo de redução de vértices de polígonos da Subseção 2.4 que tenta achar os pontos mais relevantes e que expressem as características principais da forma original. Esse pré-processamento é aplicado antes da correspondência inicial pela métrica.

$$K(P_n) = \frac{\beta_n \|P_n - P_{n-1}\| \|P_{n+1} - P_n\|}{\|P_n - P_{n-1}\| + \|P_{n+1} - P_n\|} \quad (2.46)$$

onde β_n é o ângulo entre os segmentos $\overrightarrow{P_{n-1}P_n}$ e $\overrightarrow{P_nP_{n+1}}$.

Os parâmetros utilizados na Equação (2.46) são mostrados na Fig. 2.10.

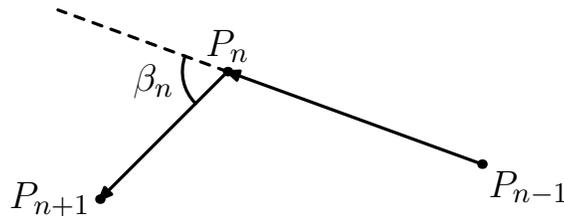


Fig. 2.10: Parâmetros para o cálculo de relevância dos vértices.

Dois parâmetros são inseridos no algoritmo para indicar a decomposição dos polígonos, um para o polígono A , np_1 , e um para o polígono B , np_2 . O processo de decomposição é descrito no Algoritmo 2.4.

A Fig. 2.8a representa o algoritmo de correspondência feito para os polígonos com número de pontos reduzidos. Na Fig. 2.8b, a área sombreada representa os subproblemas de correspondência,

Algoritmo 2.4: Redução do número de vértices de polígonos.

enquanto número de pontos maior que np **faça**
 para cada ponto P_n **faça**
 | Calcule $K(P_n)$;
 fim
 Encontre o valor mínimo de K , K_{min} ;
 Retire o ponto P correspondente a K_{min} ;
fim

descritos no Algoritmo 2.5.

Algoritmo 2.5: Método proposto para interpolação de formas com decomposição de polígonos.

início
 Obtenha os polígonos reduzidos $P^{A_{base}}$ e $P^{B_{base}}$ utilizando o algoritmo de redução de pontos;
 Encontre θ e os pontos $P_{i_{min}}^{A_{base}}$ e $P_{j_{min}}^{B_{base}}$ através da métrica de polígonos;
 Reescreva o polígono A para $P^{A_{base}} = [P_i^{A_{base}}, \dots, P_{N-1}^{A_{base}}, P_0^{A_{base}}, \dots, P_i^{A_{base}}]$ com $i = i_{min}$;
 Reescreva o polígono B para $P^{B_{base}} = [P_j^{B_{base}}, \dots, P_{M-1}^{B_{base}}, P_0^{B_{base}}, \dots, P_j^{B_{base}}]$ com $j = j_{min}$;
 Calcule a correspondência de $P^{A_{base}}$ e $P^{B_{base}}$ através da função de custo;
 Utilize a correspondência calculada para $P^{A_{base}}$ e $P^{B_{base}}$ nos pontos correspondentes em P^A e P^B ;
 Divida em subproblemas de correspondência como na Fig. 2.8b. Onde as extremidades são os pontos já correlacionados. ;
 Divida os polígonos de acordo com a correspondência obtida;
 para cada quadro da animação **faça**
 | Interpole os lados e ângulos de acordo com um parâmetro de transição t ;
 fim
fim

2.5.4 Resultados

As formas geradas pelo método proposto dependem muito da configuração da função de custo. A Fig. 2.11 mostra uma boa escolha de parâmetros. Já a Fig. 2.12 mostra outras escolhas que não geraram resultados adequados.

Na Tab. 2.1, Tab. 2.2, Tab. 2.4, Tab. 2.3, Tab. 2.5 e Tab. 2.6, Θ_{orig} é o polígono de origem e Θ_{dest} o polígono de destino. Nessas tabelas, manteve-se a notação decimal por pontos e não por vírgulas para facilitar a correspondência com a programação em *software*.

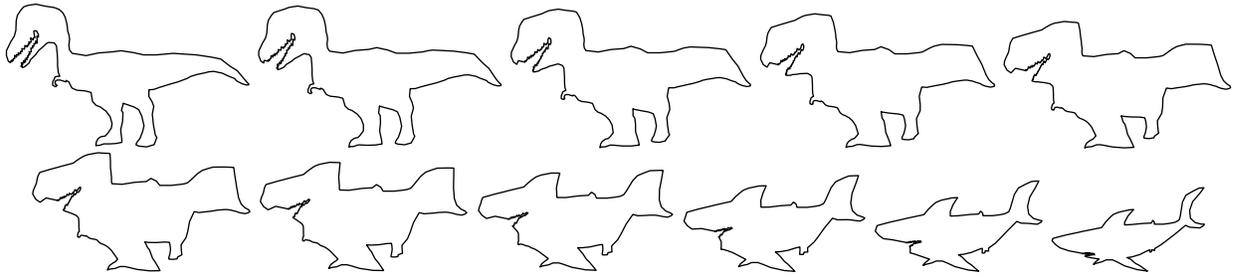


Fig. 2.11: Correspondência entre um polígono em forma de dinossauro e outro em forma de tubarão.

Tab. 2.1: Distâncias entre os polígonos intermediários e os originais na Fig. 2.11.

t	Distância de Θ_{orig}	Distância de Θ_{dest}
0.00	0.0000	1.5369
0.10	0.7915	1.4452
0.20	1.0140	1.3907
0.30	1.1690	1.3676
0.40	1.2797	1.3554
0.50	1.3906	1.3657
0.60	1.4780	1.3176
0.70	1.4482	1.2013
0.80	1.4402	1.0482
0.90	1.4921	0.7750
1.00	1.5369	0.0000

Os parâmetros da função de custo w_1 e w_2 são relacionais, isto é, o que realmente influencia na correspondência é a relação w_2/w_1 . Para $w_2 = 0$, a correspondência será feita tendo apenas como referência os valores dos ângulos dos lados. Para $w_1 = 0$, a correspondência é feita tentando manter a relação de um lado para um lado, ou seja, evitando que dois lados sejam relacionados a um lado.

A Fig. 2.13 mostra a correspondência de pontos entre os polígonos. A seqüência da Fig. 2.14 foi gerada com os parâmetros $np1 = 17$, $np2 = 18$, $w_1 = 1.0$, $w_2 = 2.2$ e $w_{rot} = 0$. Esses parâmetros são os únicos que produzem bons resultados para estes polígonos: valores fora dessa faixa produzem resultados visualmente estranhos. Esses polígonos destacam a utilidade da decomposição do polígono através da redução de vértices. Sem essa decomposição, não foi possível achar valores de w_1 e w_2 que produzissem uma forma intermediária satisfatória.

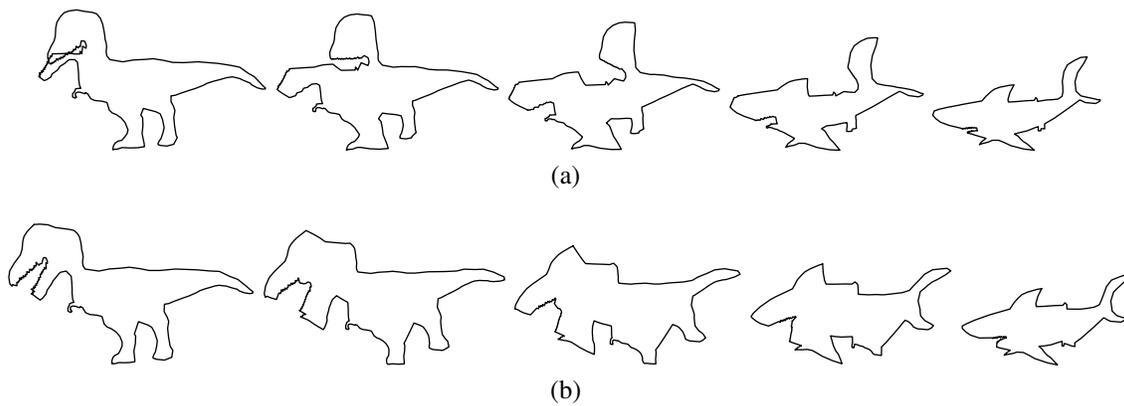


Fig. 2.12: Duas correspondências com parâmetros diferentes.

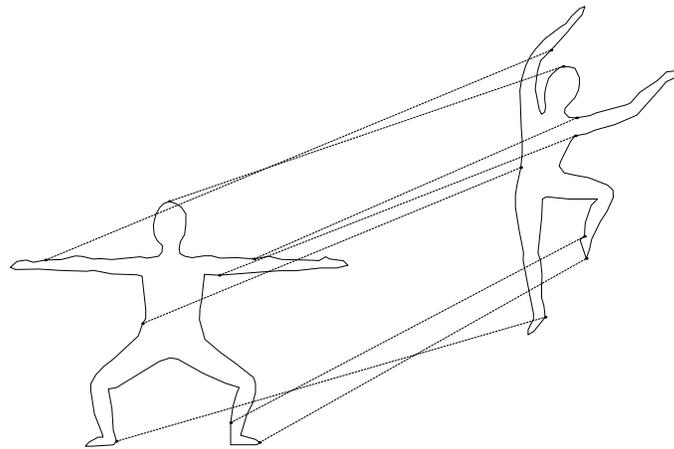


Fig. 2.13: Amostragem da relação de vértices.

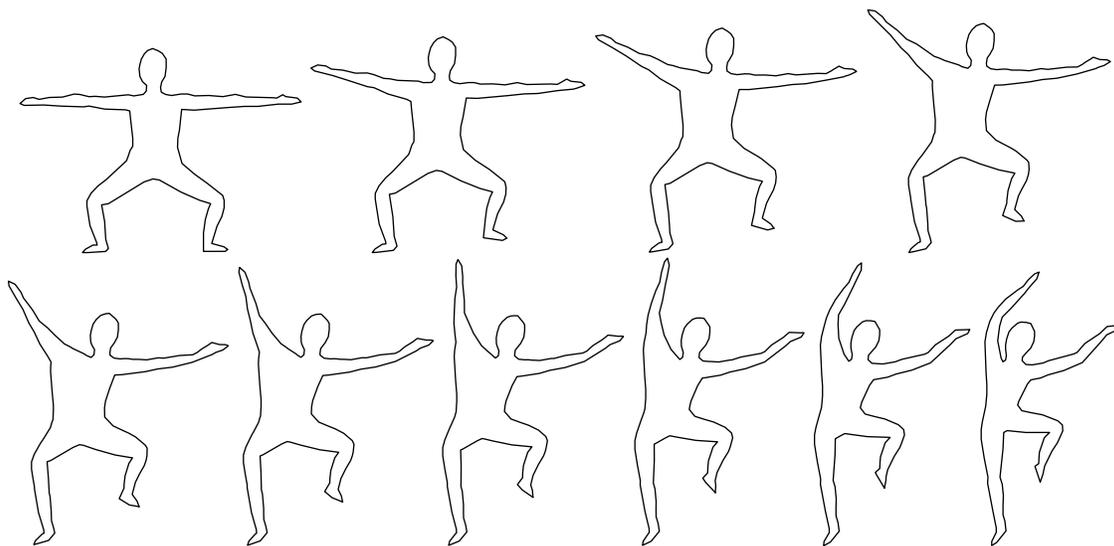


Fig. 2.14: Interpolação de formas com um dançarino em duas posições diferentes.

Tab. 2.2: Distâncias entre os polígonos intermediários e os originais na Fig. 2.14.

t	Distância de Θ_{orig}	Distância de Θ_{dest}
0.00	0.0000	1.2719
0.11	0.2863	1.1975
0.22	0.4201	1.1000
0.33	0.5565	0.9864
0.44	0.6966	0.8659
0.56	0.8343	0.7358
0.67	0.9605	0.5938
0.78	1.0772	0.4475
0.89	1.1810	0.2813
1.00	1.2719	0.0000

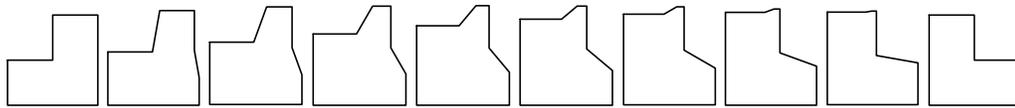


Fig. 2.15: Interpolação de formas de dois quadrados chanfrados sem compensação de rotação.

Tab. 2.3: Distâncias entre os polígonos intermediários e os originais na Fig. 2.15.

t	Distância de Θ_{orig}	Distância de Θ_{dest}
0.00	0.0000	0.0000
0.11	0.4291	0.4291
0.22	0.6068	0.6068
0.33	0.7354	0.7354
0.44	0.7748	0.7748
0.56	0.6802	0.6802
0.67	0.5718	0.5718
0.78	0.4485	0.4485
0.89	0.3031	0.3031
1.00	0.0000	0.0000

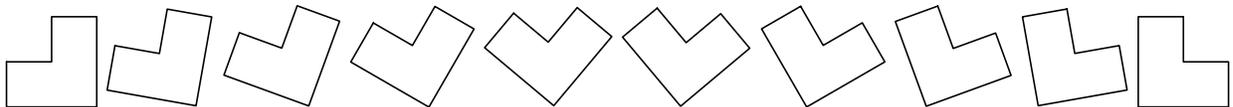


Fig. 2.16: Interpolação de formas de dois quadrados chanfrados com compensação de rotação.

Tab. 2.4: Distâncias entre os polígonos intermediários e os originais na Fig. 2.16.

t	Distância de Θ_{orig}	Distância de Θ_{dest}
0.00	0.0000	0.0000
0.11	0.0000	0.0000
0.22	0.0000	0.0000
0.33	0.0004	0.0004
0.44	0.0000	0.0000
0.56	0.0000	0.0000
0.67	0.0000	0.0000
0.78	0.0000	0.0000
0.89	0.0000	0.0000
1.00	0.0000	0.0000

A Fig. 2.15 mostra a interpolação de dois quadrados chanfrados. As formas intermediárias produzidas, apesar de não serem desagradáveis, não são exatamente a seqüência mais suave, como a Tab. 2.3 mostra. As formas interpoladas possuem valores de distância dos polígonos originais maiores do que a distância entre os polígonos de origem e de destino.

A interpolação vista na Fig. 2.16 tem os mesmos polígonos de origem e de destino da Fig. 2.15, mas as formas interpoladas estão mais parecidas com as originais, como mostra a Tab. 2.4. Os dois polígonos originais representam a mesma forma, sendo a diferença apenas a rotação entre elas. A correspondência da Fig. 2.15 não considerou a rotação, enquanto que a da Fig. 2.16 considerou-a devido ao parâmetro w_{rot} ter ativado a compensação de rotação. Essa situação mostra um uso da compensação de rotação.

Nota-se na Tab. 2.4 que todos os valores de distância são zero, isto é, durante toda a interpolação o polígono permanece o mesmo. Para $t = 0.33$ a distância entre os polígonos é de 0.0004. Isso é explicado pelo fato do algoritmo utilizado para fechar os polígonos poder distorcer a forma, mas essa distorção não é perceptível visualmente.

Outra forma de utilizar a compensação de rotação é para produzir transformações mais interessantes como mostram a Fig. 2.17 e Fig. 2.18. Na Fig. 2.17 a transformação tem uma aparência mais dinâmica, enquanto, na Fig. 2.18, a aparência é mais estática.

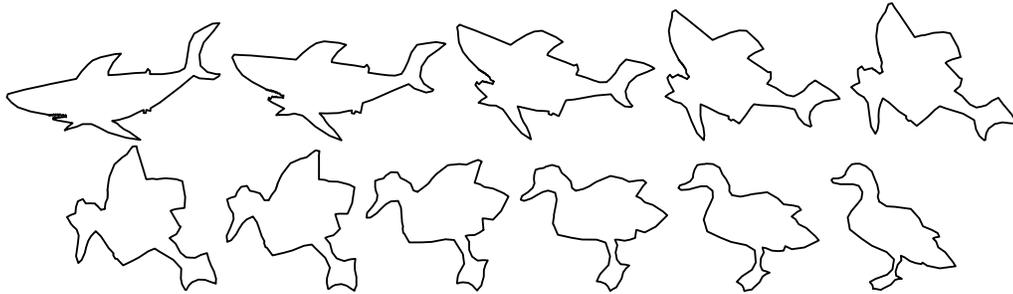


Fig. 2.17: Interpolação entre um polígono na forma de tubarão e um na forma de pato com compensação de rotação.

Tab. 2.5: Distâncias entre os polígonos intermediários e os originais na Fig. 2.17.

t	Distância de Θ_{orig}	Distância de Θ_{dest}
0.00	0.0000	1.2140
0.10	0.5351	1.1747
0.20	0.7234	1.1342
0.30	0.8622	1.0734
0.40	0.9743	1.0081
0.50	1.0460	0.9300
0.60	1.1024	0.8323
0.70	1.1458	0.7128
0.80	1.1665	0.5670
0.90	1.1720	0.3920
1.00	1.2140	0.0000

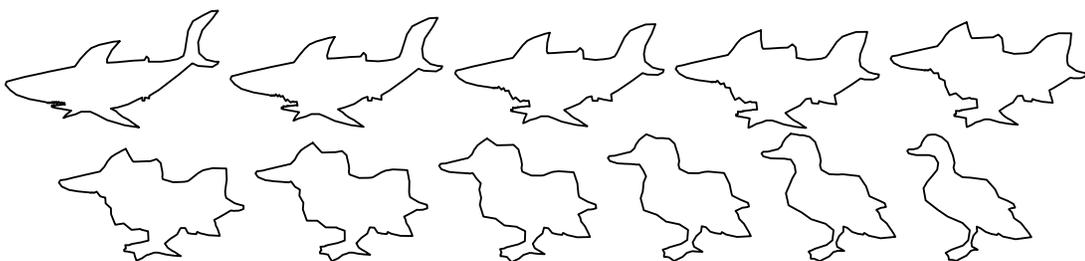


Fig. 2.18: Interpolação entre um polígono na forma de tubarão e um na forma de pato sem compensação de rotação.

Tab. 2.6: Distâncias entre os polígonos intermediários e os originais na Fig. 2.18.

t	Distância de Θ_{orig}	Distância de Θ_{dest}
0.00	0.0000	1.2140
0.10	0.6810	1.2098
0.20	0.9489	1.1377
0.30	1.1089	1.0872
0.40	1.2052	1.1122
0.50	1.2679	1.1837
0.60	1.3052	1.1343
0.70	1.3157	0.9758
0.80	1.2269	0.7969
0.90	1.1915	0.5636
1.00	1.2140	0.0000

Tab. 2.7: Parâmetros utilizados nas interpolações

Figura	w_1	w_2	w_{rot}	np_1	np_2
Fig. 2.11	2.00	1.00	0	17	18
Fig. 2.12a	1.00	1.00	0	-	-
Fig. 2.12b	1.00	1.00	0	9	9
Fig. 2.14	1.00	2.20	0	17	18
Fig. 2.16	1.00	1.00	1	-	-
Fig. 2.15	1.00	0.00	0	-	-
Fig. 2.17	1.00	0.00	1	-	-
Fig. 2.18	1.00	5.00	0	-	-

Vantagens e desvantagens

Esse método procura utilizar uma métrica de polígonos para melhorar a correspondência. Os resultados obtidos foram visualmente iguais aos de outros métodos. Entretanto, a vantagem está no termo $\theta(t)$ proveniente da métrica, que permite que a correspondência tenha uma aparência mais tradicional até uma aparência mais abstrata, produzindo resultados inesperados.

A desvantagem está na complexidade do algoritmo de distância entre polígonos, que eleva o tempo de processamento. Outra desvantagem é que a interpolação está trabalhando apenas com o contorno, enquanto a visão humana para formas considera mais relevante a área.

2.6 Considerações finais

Neste capítulo, foram descritos os dois componentes principais da interpolação de formas, a correspondência e a interpolação.

As principais contribuições do método proposto foram a implementação de uma forma de decomposição de polígonos automática e a compensação de rotação na interpolação de formas. Esses dois fatores podem ser utilizados para controlar a forma intermediária gerada.

Em estudos futuros espera-se implementar um algoritmo de triangulação compatível entre os dois polígonos correlacionados. Interpolando a triangulação é possível produzir formas intermediárias que levam em conta a área da forma.

Capítulo 3

Morphing

O *morphing* é uma poderosa ferramenta de animação para efeitos visuais. O seu uso pode ser visto em muitas produções de cinema e de televisão. O termo *morphing* é derivado de metamorfose de imagem, e se refere ao efeito obtido através do processamento de imagens que gera uma seqüência que mostra a transformação de um objeto em outro.

A metamorfose entre duas ou mais imagens é uma técnica muito útil, utilizada para propósitos educacionais ou de entretenimento. As técnicas anteriores ao *morphing* digital de imagens incluíam tomadas inteligentes, como a de um personagem se transformando enquanto corre pela floresta passando por trás de várias árvores, e o *cross dissolve* ótico, onde uma imagem é escurecida enquanto outra ao mesmo tempo é iluminada, assim a imagem de uma pessoa em roupas normais é apagada e ao mesmo tempo outra imagem é iluminada mostrando uma pessoa com todas as modificações de maquiagem, trajes e objetos. Vários filmes de terror clássicos mostram tais processos.

O método mais simples de transformar uma imagem digital em outra é através do *cross dissolve*, mostrado na Fig. 3.1. Nesse método a cor de cada *pixel* é interpolada no tempo, partindo do valor da primeira imagem até o valor correspondente na segunda. É um método de transição de imagens muito utilizado na edição de vídeos. Mas o seu uso para mostrar uma metamorfose não é tão eficiente, já que não ocorre uma mistura total das imagens e visualmente a transição se parece mais com duas imagens transparentes uma sobre a outra. Esse efeito também é chamado de efeito fantasma. Outro motivo que contribui para a sua ineficiência é que o seu uso já foi consagrado como uma transição que demonstra um salto no tempo ou mudança de lugar, ou seja, uma linguagem visual já está estabelecida. Apesar disso, o *cross dissolve* foi muito utilizado para mostrar uma metamorfose, mas toda uma pré-produção tinha que ser feita. A pessoa que iria se transformar e a criatura tinham que ter as mesmas proporções e os enquadramentos das imagens tinham que ser os mesmos. Assim, tinha-se duas imagens com a pessoa e a criatura alinhada na mesma posição geométrica.

O processo de *morphing* pode ser separado em duas partes. A primeira parte utiliza técnicas de

deformação de imagens, mais popularmente conhecida como *warping*, e a segunda utiliza a interpolação das cores entre duas imagens, o *cross dissolve*. A deformação de imagens é utilizada para obter as condições ideais onde o *cross dissolve* funciona bem. Essa condição se dá quando as duas imagens a serem transformadas têm seus contornos alinhados, não produzindo o efeito fantasma. Tentar alinhar os contornos através da pré-produção dos atores, além de ser trabalhoso, não permite um alinhamento perfeito, devido tanto a pequenas diferenças nas proporções dos personagens a serem transformados quanto ao posicionamento das câmeras e dos atores.

A idéia do algoritmo é especificar uma deformação que leve os contornos da primeira imagem para a posição dos contornos da segunda, e leve os contornos da segunda imagem para a posição dos contornos da primeira imagem. Durante o decorrer da metamorfose, a primeira imagem é gradualmente deformada e apagada, enquanto a segunda imagem que está totalmente deformada aparece gradualmente retornando os contornos ao estado original. As primeiras imagens da seqüência gerada serão mais parecidas com a primeira imagem, enquanto as últimas serão mais parecidas com a segunda. A imagem que fica exatamente na metade da seqüência normalmente indica a qualidade das outras imagens. Se a qualidade dessa imagem for boa, provavelmente a qualidade da seqüência também será boa.

A metamorfose entre duas imagens começa com um animador estabelecendo a correspondência entre elas com pares de primitivas de atributos. Essas primitivas podem ser pontos de uma malha de controle, segmentos de reta, curvas ou pontos. Cada primitiva especifica um atributo ou ponto de referência. A correspondência de atributos é então utilizada para calcular funções de mapeamento que definem a relação espacial entre todos os pontos nas duas imagens. Funções de transição são utilizadas para interpolar as posições dos atributos durante a seqüência de transformação. Uma vez que as imagens tenham sido deformadas para alinhamento em posições intermediárias, a interpolação linear das cores é utilizada para gerar as imagens intermediárias.

Entre as partes do *morphing*, o *cross dissolve* é um processo simples. A grande dificuldade está em obter as imagens deformadas.

Existem vários métodos de *morphing* [1]. O método mais antigo é o por malha, em que pontos formam uma grade sobre as imagens. Associando as linhas da grade com o contorno dos objetos nas imagens, faz-se uma correspondência entre as duas imagens, isto é, define-se qual parte da imagem de origem e de destino se transformarão uma na outra.

Outro método parecido com o de malhas é o de triangulações [17]. Mas, nesses métodos, os pontos são definidos pelo usuário e depois o algoritmo tenta achar a melhor triangulação entre as duas imagens, tornando a interface mais intuitiva.

Tentando melhorar a interface com o animador, o *morphing* por campo [20] introduziu a inserção de marcadores através de segmentos de reta. Outro modo de marcação é por pontos. Nesse caso,

utiliza-se de funções de base radial para deformar as imagens.



Fig. 3.1: *Cross dissolve* [1].

Uma variação do *morphing* entre duas imagens é o *morphing* de vista [21],[22],[23], que combina características de síntese de visão e *morphing* de imagens. Nessa técnica, no lugar de duas imagens com pessoas diferentes, são utilizadas duas imagens com perspectivas diferentes de um mesmo ambiente. Assim o algoritmo faz uma interpolação movendo de uma perspectiva para outra.

O *morphing* também pode ser utilizado para ajudar na visualização de informação como mostrado em [24] e [25].

Várias técnicas tentam promover um mapeamento um-para-um [26], evitando dobramentos na imagem. Técnicas utilizando quadrados mínimos móveis são apresentadas [27],[28]. Apesar das técnicas de computação gráfica 3D estarem bem evoluídas, o *morphing* 2D ainda é muito utilizado, principalmente devido ao custo de modelagem dos cenários em 3D.

Uma outra forma de *morphing* que surgiu através da morfologia matemática utiliza operadores morfológicos para gerar uma imagem média. A produção de uma seqüência de imagens é feita através da aplicação do operador novamente sobre as imagens médias geradas. Esse processo é repetido até que se chegue ao número de imagens desejado, produzindo assim uma seqüência de transformação suave.

O *morphing* e a morfologia matemática são duas técnicas diferentes. Apesar de as duas trabalharem com imagens, a morfologia matemática se refere a um conjunto de operadores enquanto que o *morphing* se refere ao resultado produzido. A morfologia matemática também pode ser utilizada para fazer uma seqüência de metamorfose como será explicado no Capítulo 4. Uma vantagem desse tipo de *morphing* é que não é preciso especificar nada além das imagens de origem e de destino: a seqüência é obtida através da interpolação das características morfológicas da imagem. Apesar de o efeito ser interessante, as características do objeto em uma imagem não se transformam nas características de outro objeto como no *morphing* que utiliza deformações de imagens. Por exemplo, se as imagens de origem e de destino tiverem duas pessoas, mas se as posições delas forem muito diferentes, o efeito vai ser uma desaparecendo em um lugar e outra aparecendo em outro lugar.

O algoritmo utilizado para este trabalho foi o *morphing* por campo otimizado apresentado na Seção 3.2.

3.1 Deformação de imagens

A deformação de imagens, mais conhecido como *warping*, é o processo de manipular digitalmente uma imagem de tal maneira que as formas presentes na imagem sejam deformadas de acordo com marcadores inseridos. Essa é uma grande área do processamento de imagens que trata de transformações geométricas. O *warping* tem uma grande importância no avanço dos efeitos visuais. O interesse por essa área data desde 1960, quando foi introduzido para correções geométricas em imagens de monitoramento remoto. Desde então, teve um grande crescimento com aplicações nas áreas de imagens médicas, de visão computacional e de computação gráfica.

A deformação de imagens é a principal técnica empregada no *morphing* digital de imagens. No *morphing*, os atributos das duas imagens são alinhados para que a transição entre elas seja a mais suave possível. O resultado será tanto melhor quanto mais o conjunto de atributos estiver alinhado. Esse alinhamento é feito utilizando a deformação de imagens.

A deformação de imagens consiste em mapear pontos entre as duas imagens. Primeiramente define-se um mapeamento que consiste em transferir cada *pixel* na imagem de origem para o *pixel* correspondente na imagem de destino.

Dados dois marcadores, um na imagem de origem indicando os atributos a serem deformados e outro indicando o lugar para onde esses atributos têm que ser deslocados, a deformação de imagens promove o mapeamento entre essas duas imagens a partir desses marcadores.

3.1.1 Mapeamento

Existem duas maneiras de aplicar as deformações na imagem. A primeira, chamada de mapeamento direto, varre os *pixels* da imagem de origem e os copia para a posição apropriada na imagem de destino. A Fig. 3.2 mostra como é feito o mapeamento direto. A segunda é o mapeamento inverso, onde os *pixels* da imagem de destino são varridos e o *pixel* correspondente na imagem de origem é amostrado. O mapeamento inverso é mostrado na Fig. 3.3.

A vantagem do mapeamento inverso é que todos os *pixels* na imagem de destino são preenchidos. Já no mapeamento direto pode ocorrer lacunas que deverão ser preenchidas através da interpolação dos *pixels* vizinhos.

3.1.2 Transformação geométrica *affine*

A forma mais simples de calcular o mapeamento da deformação é por transformações geométricas *affine*, onde as coordenadas são mapeadas através de transformações de translação, rotação e escalamento. Os termos x_0 e y_0 representam as coordenadas originais, e x e y as transformadas.

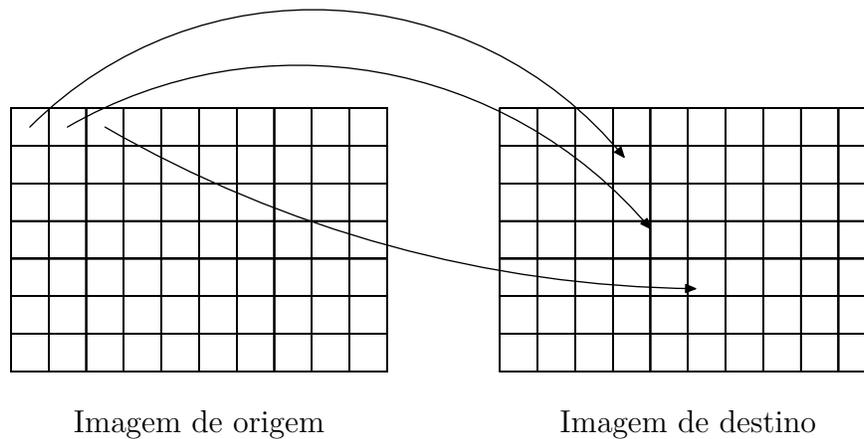


Fig. 3.2: Mapeamento direto

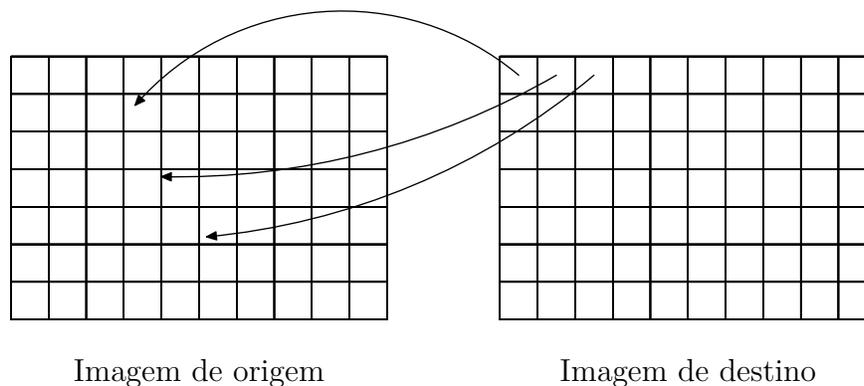


Fig. 3.3: Mapeamento inverso

As transformações são aplicadas por meio de operações matriciais, nas quais são utilizadas coordenadas homogêneas. No sistema de coordenadas homogêneo, um espaço geométrico de n dimensões é representado por um sistema de $n+1$ dimensões. Assim, considera-se esse espaço geométrico como uma projeção de um sistema de $n+1$ dimensões. Por exemplo, no caso 2D, as coordenadas homogêneas são representadas por: $[x \ y \ w]$. Quando projetadas no espaço geométrico, essas coordenadas homogêneas correspondem às seguintes coordenadas no sistema cartesiano: $[x/w \ y/w]$. No caso de $w = 1$ as coordenadas homogêneas e cartesianas são as mesmas, isto é, o ponto está exatamente no plano de projeção. Esse tipo de coordenada possibilita que as transformações geométricas sejam representadas por operações matriciais.

A transformação de translação é obtida da operação matricial descrita na Equação(3.1), onde t_x é

a translação no eixo x e t_y a translação no eixo y .

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \quad (3.1)$$

A Equação (3.2) representa a transformação de rotação, onde θ é ângulo de rotação.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\text{sen} \theta & 0 \\ \text{sen} \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \quad (3.2)$$

A transformação de escalamento é obtida através da Equação (3.3), onde s_x é o escalamento no eixo x , e s_y é o escalamento no eixo y .

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \quad (3.3)$$

Se o mapeamento for direto, x e y representam as coordenadas na imagem de destino. Se o mapeamento for inverso, x e y representam as coordenadas na imagem de origem.

3.1.3 Deformação por malha

O uso da deformação de imagens por malha foi desenvolvido pela ILM (Industrial Light & Magic) para o filme *Willow* em 1988. Tornou-se padrão para esse tipo de efeito em filmes posteriores. Para ilustrar o processo, considere uma imagem com uma malha de pontos por cima. As imagens de origem e de destino são definidas como I_s e I_t , respectivamente. A imagem de origem tem a malha M_s e a de destino M_t . Essas malhas marcam os pontos importantes nas respectivas imagens. As linhas do nariz, boca e olhos na imagem de origem estão associadas na malha M_s e também são associados na malha M_t para a imagem de destino. As malhas M_s e M_t são utilizadas para fazer o mapeamento dos *pixels* entre as imagens. As malhas são topologicamente equivalentes, ou seja, sem discontinuidades nem dobramentos. Outra condição assumida é de que as bordas são fixas, isto é, o mapeamento não vai além dos limites das imagens.

As duas imagens são distorcidas e mapeadas para uma posição intermediária, fazendo com que seus atributos fiquem alinhados. Aplicando-se o *cross dissolve*, obtém-se a imagem final, que é a composição entre as duas imagens. Todos os quadros na seqüência de *morphing* são o produto de

quatro passos, conforme descrito no Algoritmo 3.1.

Algoritmo 3.1: Descrição dos passos do *morphing* por malhas.

para cada quadro f faça

 Aplique a interpolação linear nas malhas M_s e M_t gerando a malha M_i ;

 Deforme I_s para I_1 , usando M_i e M_s ;

 Deforme I_t para I_2 , usando M_i e M_t ;

 Aplique o *cross dissolve* nas duas imagens resultantes I_1 e I_2 ;

fim

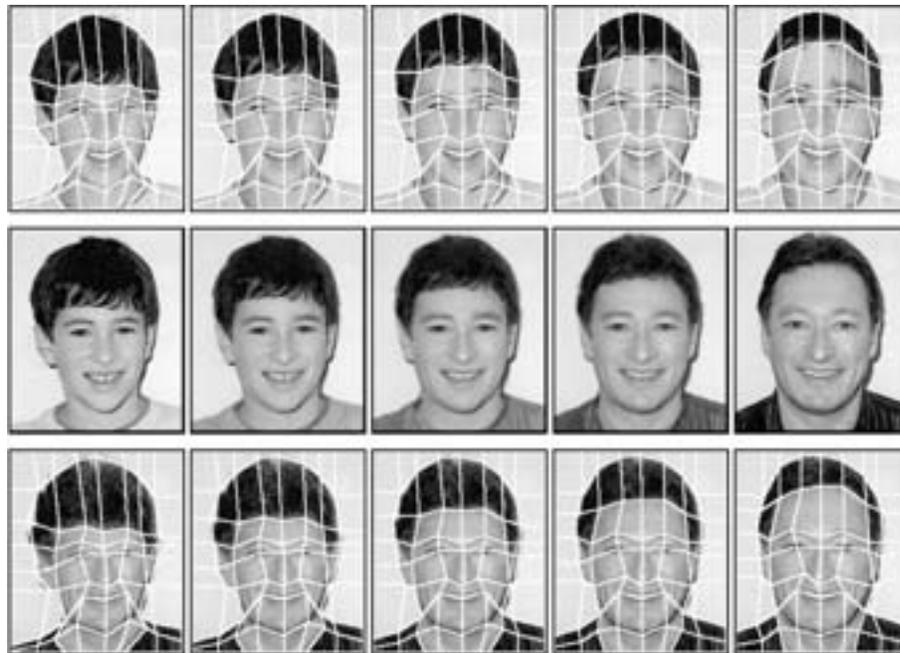


Fig. 3.4: Marcação de atributos utilizando malhas [1].

3.2 Algoritmo de deformação de imagens por campo

Os ajustes dos atributos na deformação por malha são trabalhosos, pois todos os pontos da malha devem ser correlacionados. A deformação por campo, desenvolvida por Beier e Neely, surgiu para facilitar a interface, utilizando pares de linhas como marcadores de correspondência. Esse algoritmo é descrito em [20], sendo uma das características destacadas o fato de que o algoritmo usa menos pontos do que o *morphing* por malha. Especificando apenas os atributos desejados, esse algoritmo se mostrou mais intuitivo para o usuário. O nome dado para esse algoritmo foi de *morphing* por campo.

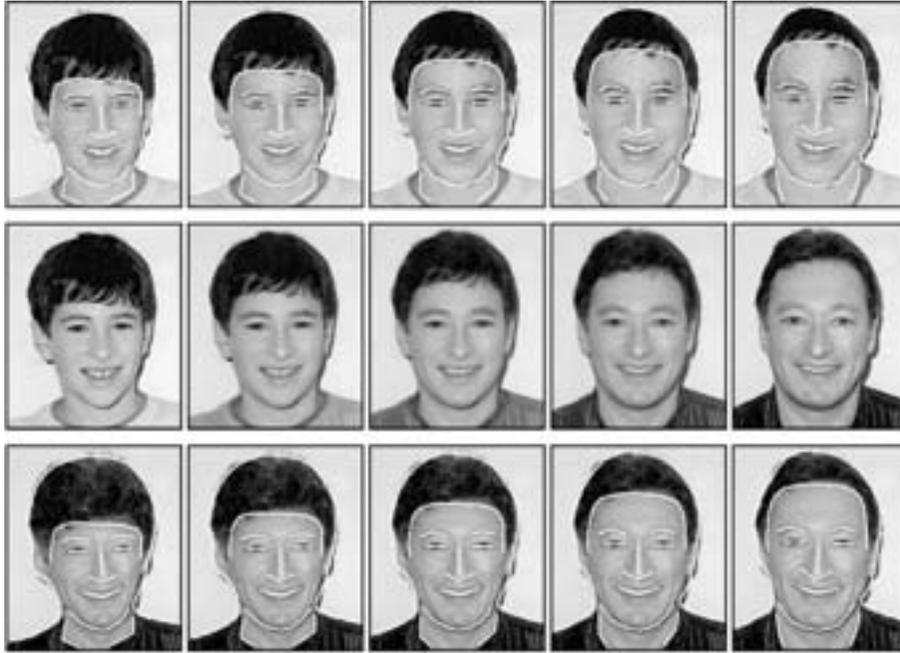


Fig. 3.5: Marcação de atributos por curvas [1].

Na deformação por campo, um par de segmentos de reta, uma relativa à imagem de origem e outra à imagem de destino, define um mapeamento de uma imagem para outra. Utiliza-se o mapeamento inverso para a amostragem de *pixels*.

Um par de segmentos de reta correspondentes nas imagens de origem e de destino mapeiam da coordenada do *pixel* da imagem de destino X para a coordenada do *pixel* da imagem de origem X' . O segmento de reta \overline{PQ} na imagem de destino corresponde ao segmento de reta $\overline{P'Q'}$ na imagem de origem.

Dado um ponto X , calcula-se a sua posição em relação ao segmento de reta \overline{PQ} .

$$u = \frac{(X - P) \cdot (Q - P)}{\|Q - P\|^2} \quad (3.4)$$

$$v = \frac{(X - P) \cdot \text{Perpendicular}(Q - P)}{\|Q - P\|} \quad (3.5)$$

O valor de u é a posição ao longo do segmento de reta, v é a distância de X do segmento de reta PQ . Se o *pixel* está entre P e Q , o valor de u varia de 0 a 1. Fora do segmento de reta os valores extrapolam essa faixa. O valor de v é a distância perpendicular do *pixel* X ao segmento de reta PQ .

A função $\text{Perpendicular}()$ retorna o vetor perpendicular e com mesmo tamanho que o vetor de entrada. A implementação dessa função pode retornar o vetor perpendicular à direita ou à esquerda, desde que a função sempre retorne a mesma perpendicular para todas as entradas.

Com as coordenadas relativas, calcula-se a posição relativa ao segmento de reta $\overline{P'Q'}$.

$$X' = P' + u(Q' - P') + v \frac{\text{Perpendicular}(Q' - P')}{\|Q' - P'\|} \quad (3.6)$$

Um modo de implementar a função $\text{Perpendicular}()$ é descrito pela Equação (3.7).

$$\text{Perpendicular}(\vec{m}) = (m_y, -m_x) \quad (3.7)$$

onde $\vec{m} = (m_x, m_y)$.

Na Fig. 3.6, X' é a posição onde a imagem de origem é amostrada correspondendo ao *pixel* X na imagem de destino. Essa posição X' está a uma distância v do segmento de reta $\overline{P'Q'}$ e a uma altura u ao longo desse segmento de reta.

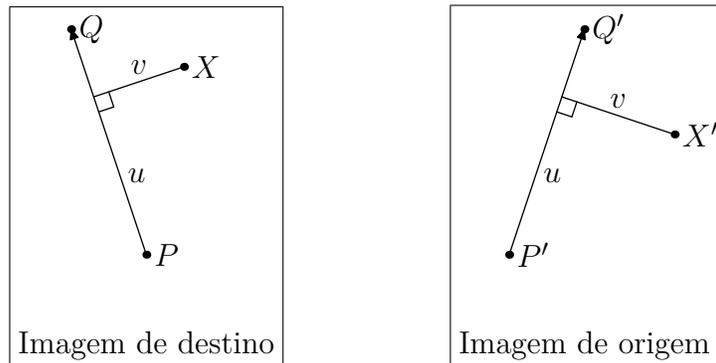


Fig. 3.6: Parâmetros em relação a semireta na imagem de destino e na imagem de origem

A transformação para apenas um par de segmentos de reta é descrita no Algoritmo 3.2.

Algoritmo 3.2: Cálculo da transformação para um segmento de reta

para cada *pixel* X na imagem de destino **faça**
 | Ache os parâmetros u, v correspondentes;
 | Ache X' na imagem de origem utilizando u e v ;
 | $\text{ImagemdeDestino}(X) = \text{ImagemdeOrigem}(X')$;
fim

O algoritmo transforma cada coordenada de *pixel* por meio da rotação, translação e escalamento, transformando assim a imagem toda. Todos os *pixels* sobre o segmento de reta são copiados sobre o segmento de reta na imagem de destino. Como u é normalizado com o comprimento do segmento de reta, e a distância v não é, a imagem é escalada na direção do segmento de reta proporcionalmente ao seu comprimento. O escalamento ocorre somente na direção do segmento de reta.

Para apenas um segmento de reta, esse algoritmo tem resultados parecidos com a transformação *affine*, onde a diferença de posição entre os segmentos de reta indica a translação, a inclinação entre os segmentos a rotação e a diferença de tamanho o escalamento. O escalamento, apesar de ser restrito ao eixo do segmento de reta, é mais intuitivo do que os parâmetros da transformação *affine*, onde os escalamentos são definidos no eixo x e y .

3.2.1 Transformação com múltiplos pares de segmentos de reta

Pares múltiplos de segmentos de reta especificam transformações mais complexas através do cálculo de uma ponderação das transformações das coordenadas para cada segmento. Uma posição X'_i é calculada para cada segmento de reta, e calcula-se também o deslocamento $\vec{D}_i = X'_i - X$, que é a diferença entre a posição do *pixel* na imagem de origem e de destino. Uma média ponderada é calculada a partir desses deslocamentos.

O peso é determinado pela distância de X até o segmento de reta. Esse deslocamento médio é relativo. Logo, para se ter a coordenada na imagem, o deslocamento médio é adicionado às coordenadas do *pixel* X para determinar a posição X' , onde a imagem de origem é amostrada. O caso de um único segmento de reta é tratado como um caso especial da transformação com múltiplos segmentos de reta.

O peso associado a cada segmento de reta deve ser mais forte quando o ponto X estiver exatamente sobre o segmento de reta, e mais fraco quanto mais distante. Assim, o peso é inversamente proporcional à distância do ponto X ao segmento de reta.

Os pesos são calculados a partir da equação a seguir:

$$peso = \left(\frac{compr^p}{a + dist} \right)^b \quad (3.8)$$

onde *compr* é o comprimento da reta, *dist* é a distância do ponto X ao segmento de reta e a, b, p são constantes que podem ser utilizadas para modificar o efeito dos segmentos de reta.

Se a tem um valor um pouco acima de zero, e se a distância do *pixel* ao segmento de reta for zero, o peso é quase infinito. Valores maiores suavizarão o *warping*, mas irão diminuir a precisão de controle. O parâmetro b controla como os pesos dos diferentes segmentos de reta se comportam com a distância. Se b for muito grande, apenas os *pixels* próximos às retas serão influenciados. Se for zero, todos os *pixels* serão afetados igualmente. Valores de b no intervalo de $[0.5, 2]$ são os recomendados. O valor de p varia de 0 a 1. Se for zero, todos os segmentos de reta têm o mesmo peso. Se for diferente de zero, o peso é proporcional ao comprimento do segmento de reta.

Na Fig. 3.7, X' é a posição para amostrar a imagem de origem para o *pixel* X na imagem de destino. Essa posição é a média ponderada da diferença entre os dois *pixels* X'_1 e X'_2 e o *pixel* X .

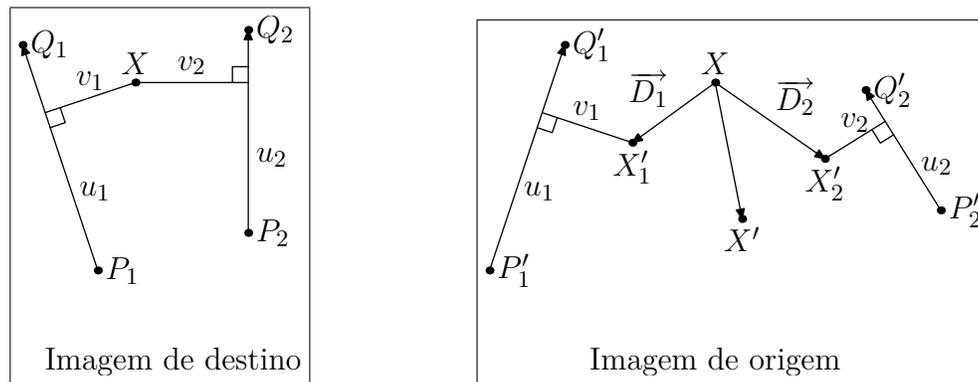


Fig. 3.7: Parâmetros em relação a múltiplas semiretas

O Algoritmo 3.3 descreve como é feito o mapeamento para múltiplos pares de segmentos de reta.

Algoritmo 3.3: Cálculo da transformação para múltiplos segmentos de reta

```

para cada pixel  $X$  na imagem de destino faça
   $\overrightarrow{D_{SOMA}} = (0,0)$ ;
   $somapesos = 0$ ;
  para cada segmento  $P_i Q_i$  faça
    Calcule  $u, v$  com relação a  $P_i Q_i$ ;
    Calcule  $X'_i$  baseado em  $u, v$  e  $P_i Q_i$ ;
    Calcule o deslocamento  $\overrightarrow{D}_i = X_i - X$ ;
     $dist =$  menor distância entre  $X$  e  $P_i Q_i$ ;
     $peso = (compr^p / a + dist)^b$ ;
     $\overrightarrow{D_{SOMA}} + = \overrightarrow{D}_i * peso$ ;
     $somapesos + = peso$ ;
  fim
   $X' = X + \overrightarrow{D_{SOMA}} / somapesos$ ;
  ImagemdeDestino( $X$ ) = ImagemdeOrigem( $X'$ );
fim

```

3.2.2 Otimização

Karam *et al.* propõem em [29] uma otimização no algoritmo de Beier e Neely. A parte mais complexa do algoritmo de campo é o cálculo da coordenada para vários segmentos de reta.

Foi proposto que a imagem fosse dividida por uma malha, e nos pontos dessa malha fossem calculadas as coordenadas de mapeamento pelo algoritmo de deformação por campo. As coordenadas

entre os pontos da malha são então calculadas por interpolação bilinear. Dentro dessa região, são escolhidos pontos de teste. O valor da coordenada obtida pelo algoritmo de deformação por campo e os calculados por interpolação bilinear são comparados. Caso o erro em alguma coordenada seja maior do que um valor limite, a malha é subdividida nessa região e o processo volta a se repetir. No Algoritmo 3.4, é descrito esse procedimento.

Algoritmo 3.4: Otimização da deformação por campo

Divida a imagem em uma grade de pontos;

Calcule as coordenadas dos pontos pelo algoritmo de deformação por campo;

repita

Subdivida a grade em pontos de teste;

para cada ponto de teste faça

Calcule a coordenada do ponto de teste pelo algoritmo de deformação por campo;

Calcule o *erro* entre a coordenada real e a calculada por interpolação bilinear;

se erro > limite então

Subdivida a grade adotando os pontos de teste como pontos da grade;

Condparada=falso;

fim

senão

Condparada=verdadeiro;

fim

fim

até Condparada=verdadeiro ;

Calcule o resto das coordenadas por interpolação bilinear;

No pior caso, esse algoritmo calcula todos os pontos pelo algoritmo original de deformação por campos. A Fig. 3.8 mostra os pontos de teste do algoritmo.

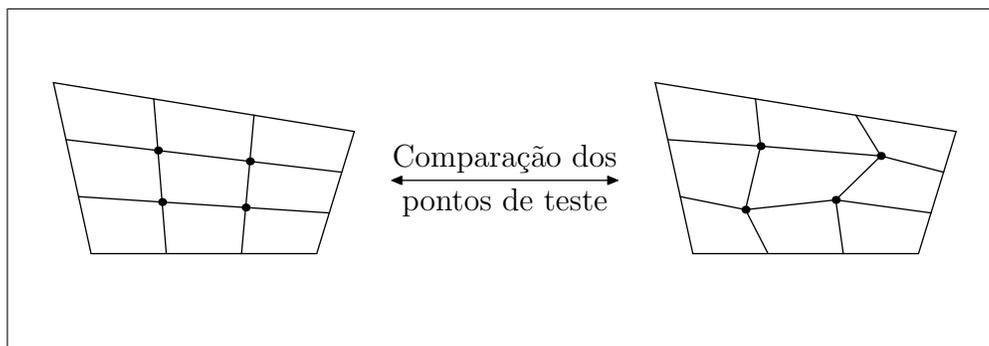


Fig. 3.8: Posições dos pontos de teste interpolados linearmente e os calculados por deformação por campo

3.3 Interpolação das imagens deformadas

Mais conhecido como *cross dissolve* na indústria, a interpolação linear é o método mais utilizado para mesclar duas imagens. O *cross dissolve* foi uma das primeiras técnicas utilizadas para a metamorfose, mas, para esse efeito ser alcançado, a filmagem tinha que alinhar os objetos que iriam ser transformados. Por exemplo, a transformação de um homem em um lobisomem era filmado alinhando o ator que iria se transformar com outro fantasiado. O *cross dissolve* era aplicado posicionando cuidadosamente os atores.

Sejam a I_1 a imagem de origem deformada para uma posição intermediária, e I_2 a imagem de destino deformada para uma posição intermediária. O *cross dissolve* é então aplicado sobre as duas imagens, gerando um quadro do *morphing*.

$$I_I(x, y) = (1 - t) I_1(x, y) + t I_2(x, y) \quad (3.9)$$

3.4 Considerações finais

Neste capítulo, foi visto um resumo das técnicas tradicionais de *morphing* de imagens. Essa técnica tenta colocar em uma ferramenta de pós-produção as técnicas utilizadas no cinema para a metamorfose de personagens.

A utilização de marcadores por malha e por segmentos de reta difere muito conforme o modo de pensar do artista. Utilizando-se uma malha como marcador, todos os pontos devem ser correlacionados. Utilizando-se o *morphing* por campo, apenas os atributos que o animador achar necessários é que são inseridos. Quanto mais intuitiva for a ferramenta, melhor tende a ser o resultado final.

Capítulo 4

Média de imagens utilizando morfologia matemática

Neste capítulo, apresenta-se uma média diferente da obtida pela interpolação linear, chamada de média morfológica. Enquanto no *cross dissolve* cada *pixel* é independente, isto é, o valor médio do *pixel* entre as duas imagens não depende de outros *pixels*, na média morfológica a média dependerá do valor do *pixel* e dos valores vizinhos a ele. Morfologia se refere à estrutura geométrica da imagem, ou seja, a forma do objeto contido na imagem. Assim, morfologia matemática tem uma visão total da figura, e não apenas uma visão individual dos *pixels*.

A média morfológica foi introduzida por Serra [6] para imagens binárias. Beucher [7] aplicou a média para imagens em tons de cinza e mosaicos. Iwanowski estendeu a média para imagens coloridas em [4], além de outras contribuições em [30],[2] e [31]. A média morfológica é um método de se calcular uma imagem intermediária baseado na forma da imagem.

A interpolação ocorre de maneira automática sem qualquer interação com o usuário. Para a maioria das imagens, o resultado é bem interessante. Essa média pode ser utilizada como uma nova ferramenta para transição de imagens a ser utilizada em edição de vídeos [32].

Aplicações da média morfológica para reconstruir a estrutura 3D de um dente foi apresentada em [33]. Devido às várias propostas de média morfológica, Iwanowski propôs um interpolador morfológico universal em [34].

A vantagem da média morfológica é que ela é uma interpolação que leva em conta o contexto da imagem, mas a desvantagem é que, se os objetos estiverem muito deslocados, ela não funciona. Isso ocorre por causa da restrição da média morfológica da intersecção entre as imagens. Essa não pode ser nula, sendo que a razão para isso está ligada ao modo de como a média é construída. Neste capítulo, essa restrição será explicada mais detalhadamente.

4.1 Morfologia matemática

A morfologia matemática utiliza as propriedades geométricas das imagens para procurar informações. Utilizando conceitos da teoria de conjuntos, a morfologia matemática analisa as imagens de acordo com a forma contida na imagem. Primeiramente, a morfologia foi aplicada em imagens binárias e posteriormente estendida para imagens em tons de cinza[35]. Estudos estão sendo feitos para a aplicação em imagens coloridas [36, 37], onde o desafio se encontra em como tratar as três componentes de cores.

A morfologia matemática é muito utilizada em reconhecimento de padrões e reconstrução de imagens[38].

4.1.1 Imagens binárias

Dois operadores básicos da morfologia matemática são a erosão e a dilatação. A erosão se baseia na intersecção do conjunto de pontos transladados por um elemento estruturante. A dilatação é a operação complementar da erosão baseado na união dos pontos.

A erosão binária é definida como:

$$X \ominus B = \bigcap_{y \in B} X_y \quad (4.1)$$

A erosão é a intersecção de todas as translações do conjunto definido pelo elemento estruturante B . O elemento estruturante elementar B normalmente é um círculo de tamanho variado. Quanto maior o tamanho, tanto maior será o efeito do operador. O conjunto resultante é menor do que o conjunto de entrada X . Visualmente, pode-se associar o efeito com a erosão da borda do objeto representado pelo conjunto de *pixels*.

A dilatação binária é definida como:

$$X \oplus B = \bigcup_{y \in B} X_y \quad (4.2)$$

A dilatação, sendo uma operação complementar da erosão, é definida pela união das translações do conjunto pelo elemento estruturante. O efeito visual provocado por este operador é o da borda crescendo ou dilatando. O conjunto resultante é maior do que o conjunto X . A Fig. 4.1 mostra exemplos de aplicações de operadores morfológicos binários.



Fig. 4.1: Exemplo de uso dos operadores de dilatação e erosão para imagens binárias

4.1.2 Imagens em tons de cinza

A morfologia matemática para imagens em tons de cinza continua utilizando a analogia com conjuntos. A diferença é que não são conjuntos planos 2D, e sim conjuntos 3D. Como cada *pixel* possui um valor que vai de 0 a 255, esse valor é considerado como a altura da imagem. Assim, uma imagem em tons de cinza pode ser tratada como uma figura topológica.

A erosão em tons de cinza[31] pode ser escrita como:

$$E = \epsilon(F) \Leftrightarrow \forall p \in P : E(p) = \min_{q \in V(p)} \{F(p + q)\} - h \quad (4.3)$$

onde E é a imagem erodida, ϵ o operador de erosão, P o domínio da imagem, $V(p)$ é a vizinhança de p definido pelo elemento estruturante e h a altura do elemento estruturante.

A dilatação em tons de cinza é definida como:

$$D = \delta(F) \Leftrightarrow \forall p \in P : D(p) = \max_{q \in V(p)} \{F(p + q)\} + h \quad (4.4)$$

onde D é a imagem dilatada e δ o operador de dilatação. A Fig. 4.2 mostra exemplos de aplicações de operadores morfológicos em tons de cinza.

Os operadores morfológicos podem ser implementados como uma máscara que percorre a imagem. Essa máscara é definida pelo elemento estruturante. Para cada ponto da imagem, a máscara é somada aos elementos da imagem e o valor máximo, no caso da dilatação, ou mínimo, no caso da erosão, é atribuído à imagem de saída. Na Fig. 4.3, é mostrada uma visão em corte de um conjunto representando uma imagem em tons de cinza. A linha tracejada representa a dilatação e a linha ponto-tracejada representa a erosão.

Desde que o elemento estruturante seja convexo e possa ser decomposto, a operação de dilatação



Fig. 4.2: Exemplo de uso dos operadores de dilatação e erosão para imagens em tons de cinza

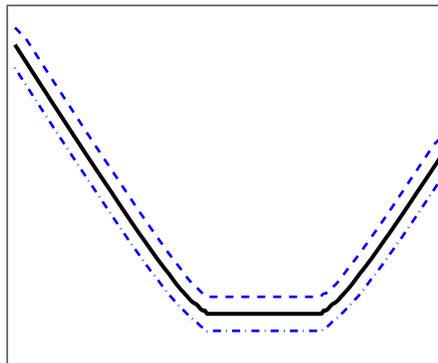


Fig. 4.3: Vista em corte, como os operadores morfológicos alteram a imagem

e de erosão por um elemento estruturante de tamanho n pode ser dividida em várias operações de tamanho unitário. Assim, as seguintes relações são válidas:

$$E = \varepsilon(A)^{(n)} = \overbrace{\varepsilon(\varepsilon(\dots \varepsilon(A) \dots))}^{n \text{ vezes}} \quad (4.5)$$

$$D = \delta(A)^{(n)} = \overbrace{\delta(\delta(\dots \delta(A) \dots))}^{n \text{ vezes}} \quad (4.6)$$

onde E é a imagem erodida, D a imagem dilatada e n o tamanho do elemento estruturante.

Assim, pode-se decompor qualquer operação por um elemento estruturante de tamanho n em n operações morfológicas.

4.2 Média morfológica

Para explicar o conceito de média morfológica [4], precisa-se considerar primeiro os conjuntos de zonas de influência. Se R_1, R_2, \dots, R_n são conjuntos disjuntos, sem elementos em comum, então a zona de influência de R_i é a região de pontos que estão mais próximos do conjunto R_i do que de qualquer outro conjunto. Por exemplo, quando apenas dois conjuntos estão envolvidos, R e S , com S contido em R , $S \in R$, a zona de influência de S com respeito a R^C é chamado de zona de influência de R dentro de S . Então:

$$IZ_R(S) = \{x : d(x, S) < d(x, R^C)\} \quad (4.7)$$

onde $d(x, A)$ define a distância geodésica [39] entre o ponto x e o conjunto A , R^C é o conjunto complemento de R . Considera-se a distância geodésica como o menor caminho entre dois pontos dentro de um conjunto. No caso em questão, é considerado o conjunto $R \cap S^C$. A zona de influência definida na Equação (4.7) representa um conjunto médio entre dois conjuntos, um contido no outro.

$$M(R, S) = IZ_R(S) \quad (4.8)$$

Foi provado em [6] que os conjuntos médios satisfazem a seguinte representação:

$$M(R, S) = \bigcup_{\forall \lambda} (S \oplus \lambda B) \cap (R \ominus \lambda B) \quad (4.9)$$

onde $\oplus \lambda B$ é uma dilatação de tamanho λ e $\ominus \lambda B$ é uma erosão de tamanho λ . Os dois operadores utilizam o elemento estruturante elementar B .

No caso de uma inclusão parcial de um conjunto X em Y , sendo que a intersecção entre eles é não nula, $X \cap Y \neq \emptyset$. O conjunto médio de X e Y é introduzido como a zona de influência de $X \cap Y$ em $X \cup Y$:

$$M(X, Y) = IZ_{(X \cup Y)}(X \cap Y) \quad (4.10)$$

A Equação (4.10) descreve um novo conjunto, localizado entre dois conjuntos iniciais. Então, da Equação (4.9), obtém-se que:

$$M(X, Y) = \bigcup_{\forall \lambda} \{[(X \cap Y) \oplus \lambda B] \cap [(X \cup Y) \ominus \lambda B]\} \quad (4.11)$$

A Equação (4.11) é a base da média morfológica. A restrição de que a média morfológica funciona apenas para imagens com intersecção não nula vem do processo de construção da média. O operador de dilatação é aplicado sobre a intersecção. Quanto maior a dimensão do elemento estruturante tanto maior a dilatação da intersecção, sendo que para $\lambda \rightarrow \infty$, o objeto ocupará toda a imagem.

Como o operador de erosão é o complementar da dilatação, o efeito contrário ocorre, portanto quando $\lambda \rightarrow \infty$, o conjunto se tornará vazio. Essa característica garante a convergência do algoritmo, pois a intersecção de um objeto dilatando e a de um objeto erodindo por um elemento estruturante de dimensão crescente, tende ao conjunto vazio. Se a intersecção for vazia ou a união for igual ao conjunto universo, os operadores morfológicos não conseguirão modificar os conjuntos, portanto não convergindo para a média morfológica. Na Fig. 4.4, são mostrados dois conjuntos com intersecção não nula e a média morfológica resultante.

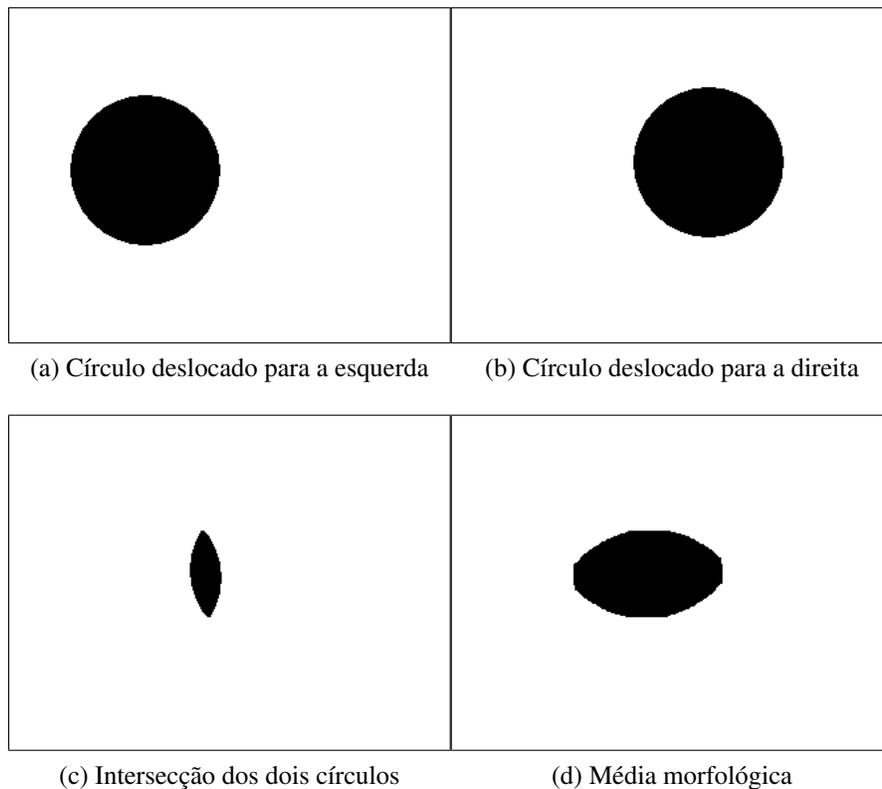


Fig. 4.4: Exemplo de duas imagens binárias e sua média morfológica.

Utilizando Equação (4.11), a média pode ser estendida para imagens em tons de cinza, substituindo os operadores binários por seus operadores equivalentes para tons de cinza, \cup por \sup , \cap por \inf , sendo que \sup e \inf são operadores que retornam os valores de máximo e mínimo, respectivamente.

$$M(X, Y) = \sup_{\forall \lambda} \{ [(X \inf Y) \oplus \lambda B] \inf [(X \sup Y) \ominus \lambda B] \} \quad (4.12)$$

A partir da Equação (4.12), observa-se que a média nada mais é que uma união de conjuntos crescentes. Utilizando as relações da Equação (4.6) e da Equação (4.5), a média pode então ser

calculada aplicando os operadores de erosão e dilatação, iterativamente.

Considerando os seguintes conjuntos:

$$z_0 = \inf(f, g) \quad (4.13)$$

$$w_0 = \sup(f, g) \quad (4.14)$$

$$m_0 = \inf(f, g) \quad (4.15)$$

Os valores iterativos são computados como:

$$z_i = \delta(z_{i-1}) \quad (4.16)$$

$$w_i = \epsilon(w_{i-1}) \quad (4.17)$$

$$m_i = \sup[\inf(z_i, w_i), m_{i-1}] \quad (4.18)$$

As iterações são feitas até que:

$$m(f, g) = m_\infty = m_i \quad (4.19)$$

Na Fig. 4.5 é ilustrado o processo iterativo de construção da média morfológica.

A média morfológica pode ser usada para produzir um *morphing* entre duas imagens, obtendo-se resultados bem melhores do que o *cross dissolve*, que é apenas uma média linear entre as imagens. Na Fig. 4.6, pode-se ver a diferença entre as médias. No *cross dissolve* das duas imagens, apesar de estarem transparentes, ainda pode-se notar as duas montanhas presentes na imagem. Mas na média morfológica, apenas uma montanha é notada na imagem média. A desvantagem da média morfológica é a perda das texturas das imagens originais.

A média morfológica é uma nova maneira de interpolar as imagens. Para imagens bem diferentes, quando não se tem nenhuma expectativa sobre a seqüência, o resultado se mostra bem interessante. Já quando as imagens são muito parecidas, sendo que se espera que a maioria das características seja mantida, o resultado não é tão satisfatório. Se as imagens apresentarem muito deslocamento entre os atributos, o algoritmo de média morfológica não consegue interpolar esses atributos. Como a média morfológica trabalha basicamente nas bordas do objeto, os atributos da imagem de origem serão mesclados com os atributos mais próximos da imagem de destino. Por isso, este trabalho propõe a utilização de técnicas de *morphing* para tentar melhorar a média morfológica.

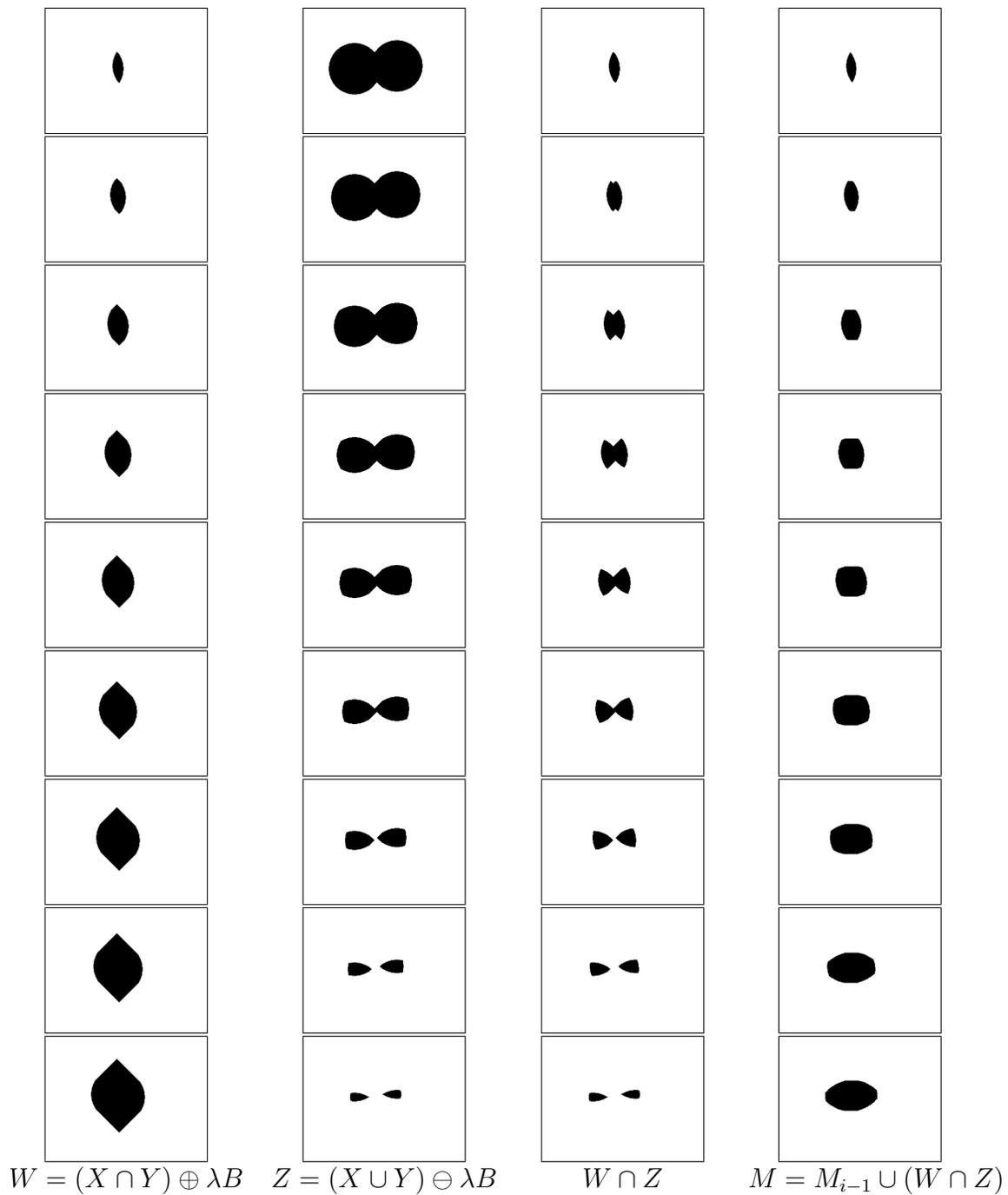


Fig. 4.5: Processo de geração de uma média morfológica

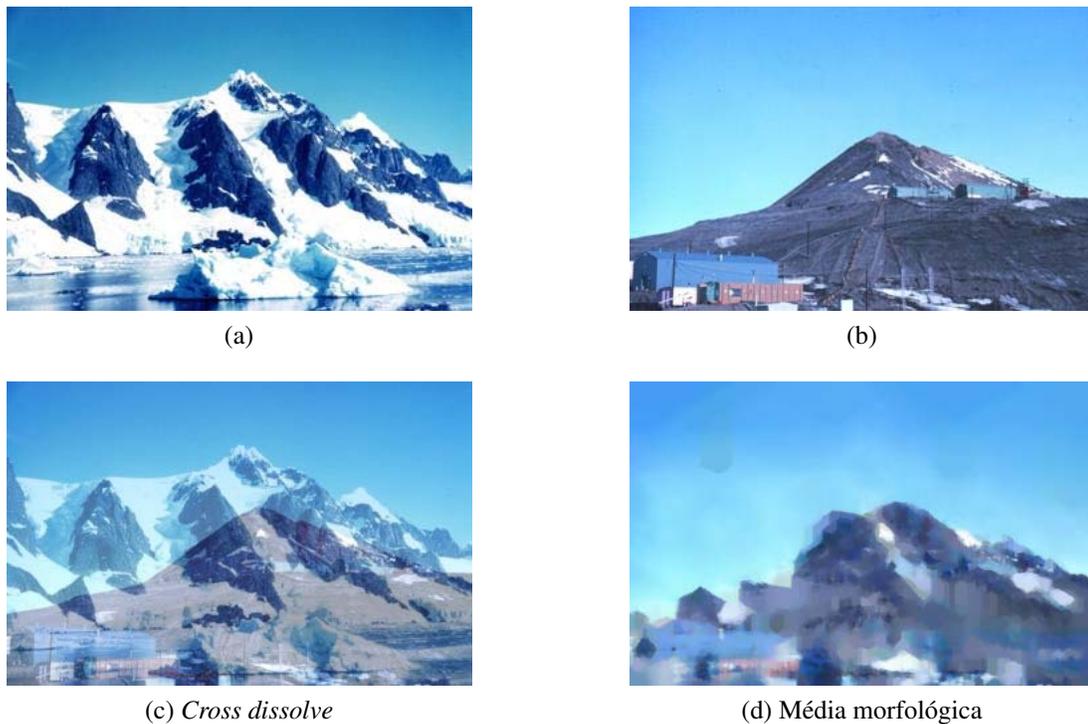


Fig. 4.6: Comparação entre o *cross dissolve* e a média morfológica.

4.2.1 Imagens coloridas

Iwanowski [4] estende a média morfológica para imagens coloridas através da comparação de um vetor ordenado. A aplicação dos operadores morfológicos é feita utilizando a ordenação lexográfica para comparar os três canais do espaço de cores RGB (*Red, Green, Blue*), isto é, primeiro é comparado o valor do canal R, depois do G, e por último o canal B. O olho humano é mais sensível à luminância, portanto, transformando a imagem RGB para um formato que isole esse canal, pode-se melhorar a qualidade da imagem resultante priorizando esse canal.

Tentando não modificar os operadores morfológicos, propõem-se dois métodos para aplicar a média morfológica em imagens coloridas. Um método utiliza o espaço de cores RGB e outro o espaço YCbCr (*Luminance, Color Difference Blue, Color Difference Red*). Não há um consenso de como aplicar os operadores morfológicos em imagens coloridas. O método proposto aqui visa facilitar a geração da média.

Espaço de cores RGB

O espaço de cores RGB é formado pelas três cores primárias: vermelho, verde e azul. A proposta para a utilização da média morfológica no espaço RGB é aplicar a média em cada canal de cor independentemente. Assim, cada canal é tratado como uma imagem em tons de cinza.

$$R_{interpolado} = M(R_A, R_B) \quad (4.20)$$

$$G_{interpolado} = M(G_A, G_B) \quad (4.21)$$

$$B_{interpolado} = M(B_A, B_B) \quad (4.22)$$

onde $M(\cdot)$ é o operador de média morfológica e R, G e B os canais vermelho, verde e azul das imagens.

O tratamento independente dos canais de cores é utilizado também pelo MATLAB.

Espaço de cores YCbCr

O espaço de cores YCbCr possui um canal de luminância e dois canais de cores formados pelos sinais diferença de cor azul e vermelho. O canal de luminância é equivalente à imagem em tons de cinza, onde a informação de brilho e contraste da imagem estão concentrados. Os canais Cb e Cr guardam as componentes de cor. Com as imagens no formato YCbCr a seguinte estratégia é utilizada: como o canal Y é o mais importante para a visão humana, ele será interpolado pela média morfológica enquanto que nos canais Cb e Cr será utilizado a interpolação linear.

$$Y_{interpolado} = M(Y_A, Y_B) \quad (4.23)$$

$$Cb_{interpolado} = (1 - t) Cb_A + t Cb_B \quad (4.24)$$

$$Cr_{interpolado} = (1 - t) Cr_A + t Cr_B \quad (4.25)$$

onde $M(\cdot)$ é o operador de média morfológica, Y, Cb e Cr são as componentes do espaço de cores.

Apesar de ser mais rápido do que o método no espaço de cores RGB, pois aplica a média morfológica em apenas um canal, esse método produz cores menos vivas. Na Fig. 4.7 pode-se ver a diferença entre as médias produzidas nos espaços de cores RGB e YCbCr.

Outros espaços de cores

Foi também testado o espaço de cores sugerido em [40], mas o resultado não foi muito diferente de se aplicar diretamente sobre o canal RGB. Esse espaço foi desenvolvido focado na adição de cores, mas como a média morfológica trata os canais independentemente através de uma média e não da adição de cores, não há efeito sobre o resultado. Esse espaço de cores sugerido por Gossett *et al.* é um espaço de cores RGB distorcido.



Fig. 4.7: Médias morfológicas produzidas utilizando o espaço de cores RGB e YCbCr.

4.3 Função de interpolação para imagens binárias

Em [2], foi proposta uma função de interpolação para imagens binárias, sendo assim possível obter qualquer nível de interpolação sem ter que gerar níveis intermediários.

Para gerar uma função de interpolação considerando dois conjuntos $U \subset V$, deve-se calcular primeiramente duas distâncias geodésicas auxiliares definidas em $V \setminus U$, onde $V \setminus U$ é equivalente a $A \cap B^C$.

A distância do ponto $x \in V \setminus U$ ao conjunto U é expressa utilizando a função geodésica de distância d_1 e é obtida através da dilatação geodésica de U com a máscara V . A segunda distância geodésica d_2 descreve a distância do ponto x até a borda de V e é computada através de dilatações geodésicas de V^C em U^C .

A função de distância relativa é utilizada como uma função de interpolação definida como:

$$Int_U^V = \begin{cases} 0 & em U \\ \frac{d_1(p)}{d_1(p) + d_2(p)} & em V \setminus U \\ 1 & em V^c \end{cases} \quad (4.26)$$

Essa equação define a distância morfológica de U até V .

Para o computador, o zero (0) representa a cor preta e o um (1) representa a cor branca. Apesar disso, na Fig. 4.8, está-se representando o um como preto e o zero como branco, devido à associação que se faz em termos de conjunto branco sendo vazio e preto sendo um elemento. Para utilizar essa representação, é importante lembrar de inverter os valores no processamento. Se isso não for feito, o resultado gerado não será o desejado, pois as operações estarão invertidas, sendo que a dilatação produzirá uma erosão e a erosão uma dilatação.

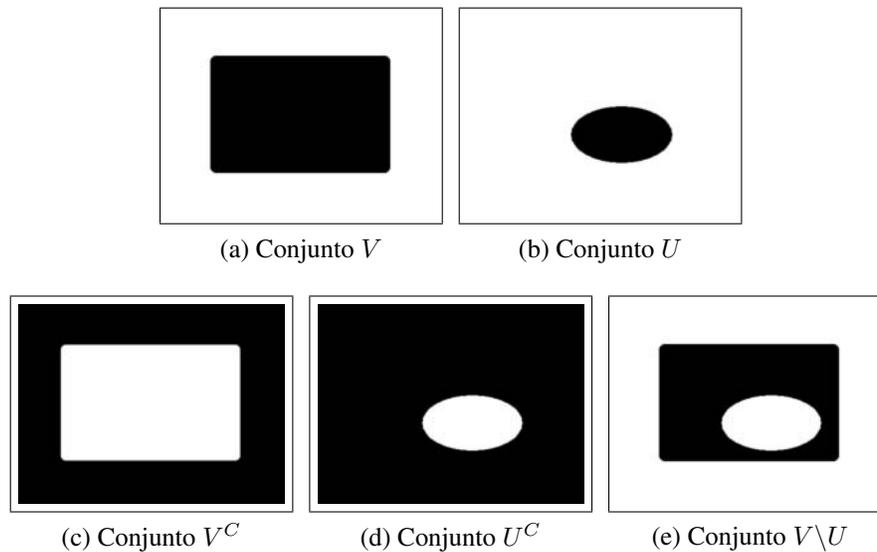


Fig. 4.8: Exemplo de conjuntos binários

A Fig. 4.9 apresenta um exemplo de construção de uma função de interpolação. A Fig. 4.9a, Fig. 4.9b, Fig. 4.9c mostram as regiões onde cada parte da função de interpolação é definida.

O conjunto interpolado entre U e V em nível α é obtido através do corte da função (4.27).

$$Int_U^V(\alpha) = \{p | Int_U^V(p) \leq \alpha\} \quad (4.27)$$

onde α está no intervalo $[0, 1]$.

Na Fig. 4.10, é ilustrado o corte de nível da função de interpolação da Fig. 4.9f. Um exemplo de conjuntos com inclusão parcial interpolados por essa técnica é visto na Fig. 4.11.

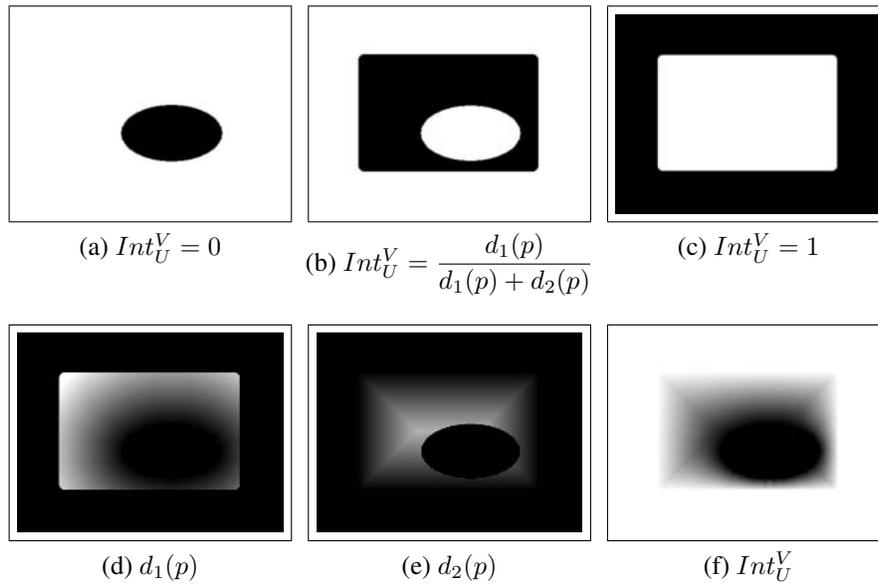


Fig. 4.9: Construção da função de interpolação Int_U^Y .

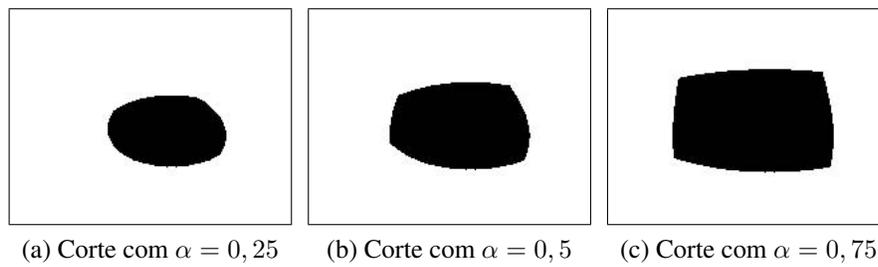


Fig. 4.10: Exemplos de corte na função de interpolação

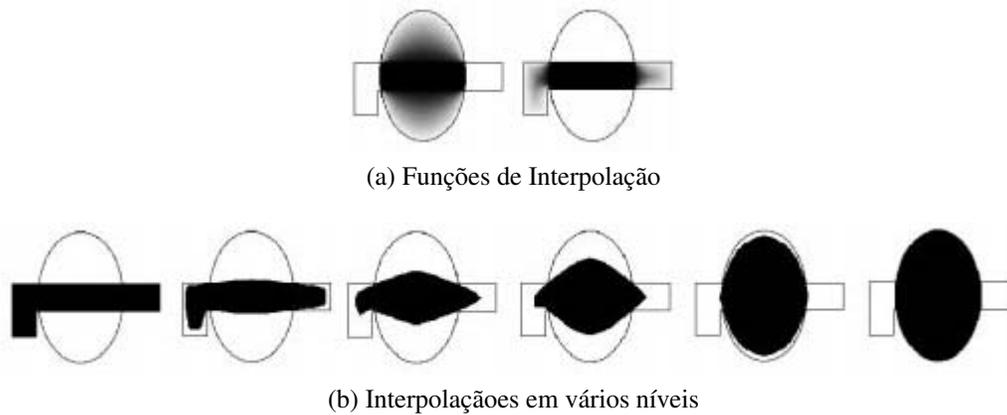


Fig. 4.11: Exemplo de uso dos operadores de dilatação e erosão para imagens em tons de cinza [2].

4.4 Proposta de uma função de interpolação para imagens em tons de cinza

A proposta de criar uma função de interpolação pode ser estendida para imagens em tons de cinza. Há duas formas de fazer isso. A primeira é considerando um cubo 3D de *pixels*, sendo que esse método é apenas uma extensão natural para morfologia matemática em tons de cinza, mas tem a desvantagem de consumir muita memória. Outro modo é dividir a imagem em camadas, ou seja, cada nível de tom de cinza seria uma camada. Isso economiza memória, mas a desvantagem é que a função de interpolação não é armazenada. Devido ao alto consumo de memória, para este trabalho, optou-se por implementar a segunda solução.

4.4.1 Transformada de distância

Em uma imagem binária, pode-se representar se o *pixel* pertence ou não a um conjunto. A transformada de distância computa a distância de uma imagem binária guardando os valores de distância de cada *pixel* em relação ao conjunto na própria coordenada do *pixel*. Assim, após a transformada, os valores dos *pixels* serão a distância do *pixel* mais próximo pertencente ao conjunto.

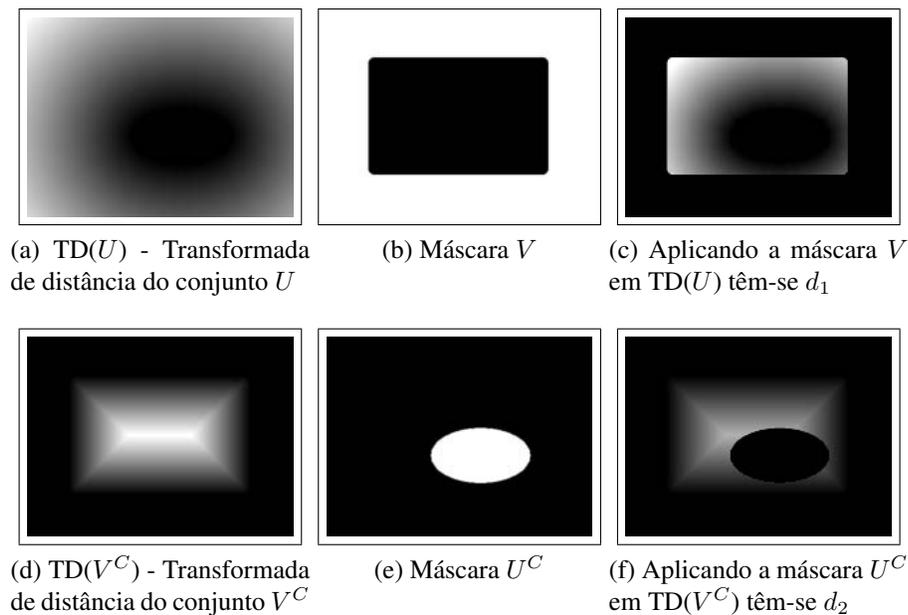
Visto que operações morfológicas têm íntima ligação com transformadas de distância, pode-se substituir o cálculo da média por uma transformada de distância. A distância d_1 e d_2 na Equação (4.26) pode ser obtida aplicando uma transformada de distância TD sobre U e V^C , respectivamente, e aplicando logo após a máscara V sobre $TD(U)$ e U^C sobre $TD(V^C)$. Onde U^C e V^C são os conjuntos complemento de U e V , respectivamente. A função de interpolação é calculada por:

$$\frac{d_1(p)}{d_1(p) + d_2(p)} \quad (4.28)$$

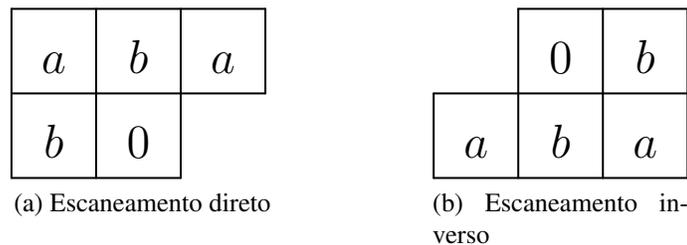
A Fig. 4.12 mostra o processo para calcular a função de interpolação utilizando a transformada de distância.

Define-se TD como sendo um transformada de distância de uma imagem binária. A transformada de distância mais simples é a de *chamfer*, onde uma máscara percorre a imagem duas vezes. Uma no sentido direto calculando as distâncias dos *pixels* à esquerda e acima, e outra no sentido inverso calculando as distâncias dos *pixels* a direita e abaixo.

Antes de aplicar a máscara deve ser feito um pré-processamento, para os *pixels* pertencentes ao conjunto é atribuído o valor 0, e para o resto ∞ . Na Fig. 4.13a, a máscara é usada para percorrer a imagem na ordem direta, de cima para baixo e da esquerda para a direita. Em cada ponto da imagem, os valores da máscara são somados ao da imagem e o valor mínimo entre eles é atribuído ao *pixel*. Aplica-se a máscara da Fig. 4.13b no sentido inverso. Após essas duas passagens a imagem resultante

Fig. 4.12: Cálculo de d_1 e d_2 através da transformada de distância

será a transformada de distância da imagem binária inicial. Os valores da máscara para se calcular uma aproximação da distância euclidiana são $a = 1.41$ e $b = 1$.

Fig. 4.13: Exemplo de máscaras de *chamfer*.

4.4.2 Separando a imagem em níveis

Normalmente, cada *pixel* na imagem guarda 8 bits de informação de cor ou luminância, podendo representar 256 níveis no total. Assim, uma função de interpolação para cada nível é gerada. Cada camada será calculada independentemente e a função de interpolação de cada camada poderá liberar memória para a função de interpolação da camada seguinte, gerando assim a economia de memória.

A desvantagem desse método é que a função de interpolação não fica armazenada e, portanto, para cada seqüência que se quiser gerar, é preciso calcular a função novamente.

4.5 Proposta de controle de transição para média morfológica

Uma característica observada utilizando uma divisão linear na função de interpolação é a impressão de que tanto no começo quanto no final da seqüência de imagens, a transição é mais rápida do que no meio. Assim, a velocidade da transformação parece desacelerar até o meio da seqüência, acelerando do meio até o final.

Esse fenômeno também foi notado por Iwanowski em [31]. Para minimizar esse efeito, foi proposto que fossem geradas imagens médias entre as imagens onde o erro fosse maior. O erro era dado pelo absoluto da diferença de valores entre os *pixels*.

A explicação para esse tipo de fenômeno é que, como os valores dos *pixels* são tratados como distâncias, ao gerar imagens com cortes lineares na função de interpolação, é equivalente ao movimento linear uniforme, isto é, com velocidade constante. Observando algumas teorias de controle de transição para animação [41], observa-se que esse tipo de transição não é recomendada, pois o resultado não é agradável. Uma solução é um movimento que vai acelerando no começo e desacelerando no final, também chamado de *easy in easy out*. Na prática, isso quer dizer que a velocidade inicial é zero e vai gradativamente subindo até o ápice no meio da seqüência, e vai gradativamente diminuindo no final. Utilizando essas diretivas, consegue-se um bom resultado que dá a impressão de uma transformação mais suave e uniforme.

A solução proposta é a utilização de uma função *cos* como controle de transição. Essa função coloca mais imagens no começo e no final da interpolação. Assim a passagem da imagem de origem para a seqüência fica mais suave, assim como a transição da seqüência para a imagem de destino.

$$u = \frac{(1 - \cos(t * \pi))}{2}; \quad (4.29)$$

Utilizando a Equação (4.29) converte-se um parâmetro de corte t , que dividiria linearmente a função de interpolação, por uma função proporcional ao *cos*. A Fig. 4.14 mostra as diferenças entre as duas abordagens. Observe que, utilizando a função cossenoidal, mais imagens perto da imagem original são geradas.

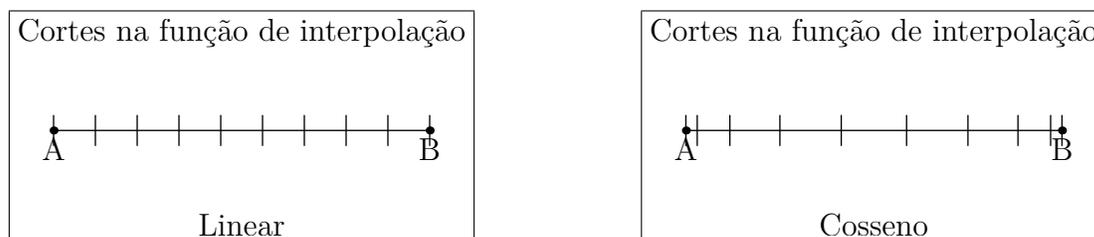


Fig. 4.14: Diferença entre os cortes lineares e os cortes utilizando uma função cossenoidal

4.5.1 Resultados

Na Fig. 4.15, na segunda imagem das duas seqüências, pode-se perceber a diferença. Na primeira seqüência que pega as interpolações linearmente com o parâmetro de corte, a segunda imagem já é bem diferente da original. Já no método proposto a imagem ainda mantém muita semelhança com a original. O efeito do método proposto pode ser observado melhor em forma de animação. Na Fig. 4.16, são mostrados mais dois exemplos utilizando a função de transição proposta e comparando com a linear.

Esse controle de transição só é possível através da função de interpolação estendida para imagens em tons de cinza. O mesmo princípio pode ser aplicado através da geração de médias como proposto por Iwanowski, mas o controle não vai ser tão preciso. Além disso, se o objetivo for a aproximação de uma função cossenoidal, quadros gerados terão que ser jogados fora, desperdiçando processamento.

4.6 Considerações finais

Neste capítulo foi vista a média morfológica. Essa média tem algumas vantagens sobre a média obtida através da interpolação linear. Uma delas é a eliminação do efeito de transparência, onde as duas imagens originais ainda são reconhecidas na imagem média. Na média morfológica, o efeito é mais sólido, mas a desvantagem é que a definição das bordas é perdida, produzindo um efeito de borramento.

Foi apresentada também uma extensão para a função de interpolação de imagens binárias para imagens em tons de cinza. Essa extensão foi feita cortando cada nível da imagem e construindo a função de interpolação binária equivalente para cada nível.

A função de interpolação foi utilizada para implementar uma proposta de controle de transição para média morfológica. Essa transição se mostrou mais suave do que a produzida por cortes equidistantes.

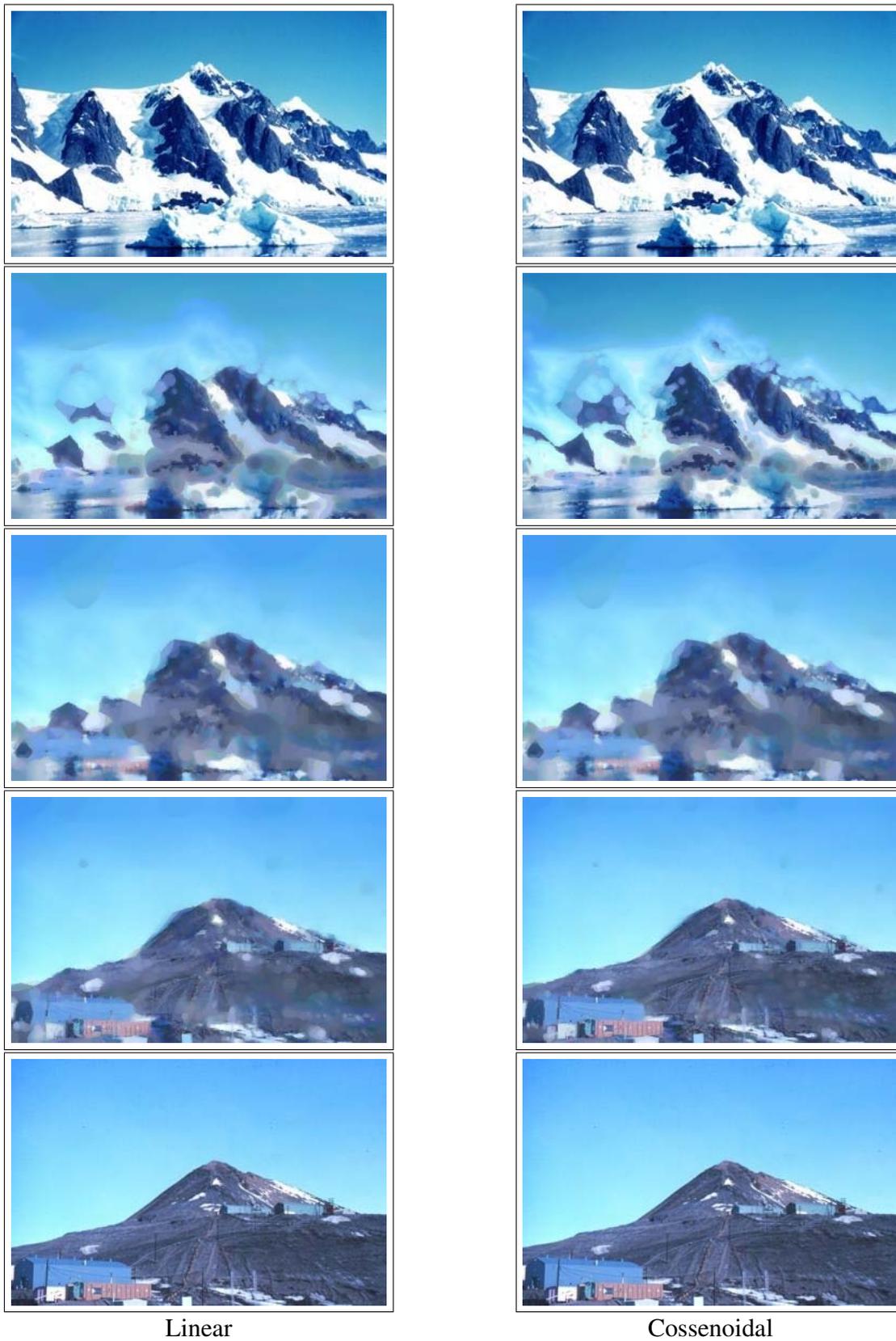


Fig. 4.15: Comparação entre as imagens geradas pelas funções de transição.

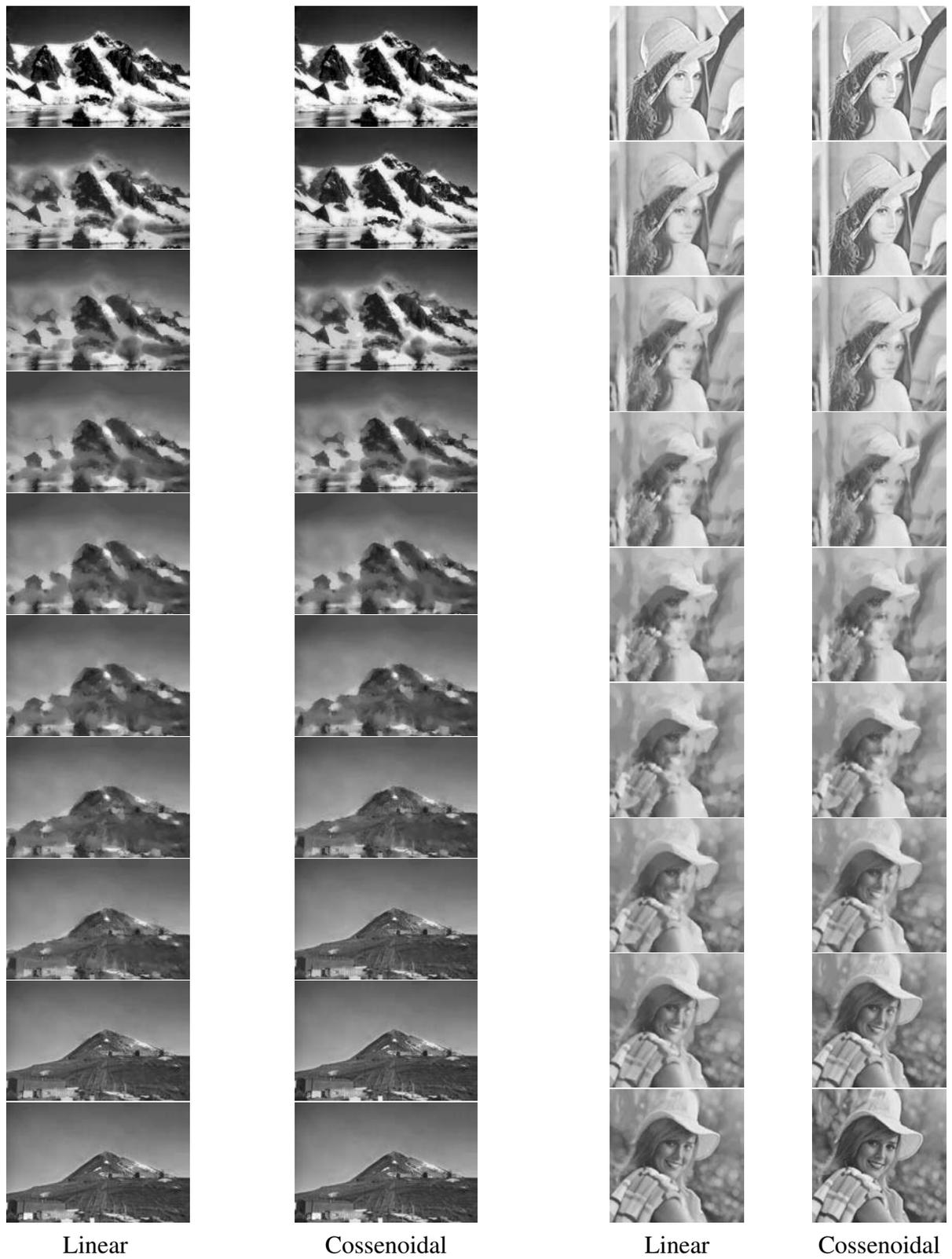


Fig. 4.16: Comparação entre as imagens geradas pelas funções de transição.

Capítulo 5

Proposta de *morphing* utilizando interpolação de formas e média morfológica

Um dos problemas do *morphing* tradicional com o *cross dissolve* é que se os atributos não estiverem exatamente alinhados, surge o efeito fantasma. Isso acontece onde as duas imagens originais aparecem de forma transparente na seqüência de metamorfose.

Como visto no Capítulo 4, a média morfológica produz imagens mais sólidas do que o *cross dissolve*, mas os resultados também não são interessantes se os atributos estiverem muito deslocados. Seguindo a idéia do *morphing* de alinhar as imagens antes da aplicação do *cross dissolve*, em [4] foi proposto que a deformação de imagens fosse utilizada para alinhar os atributos melhorando o resultado da média morfológica.

A proposta descrita neste capítulo é dividida em duas partes. A primeira parte utiliza o algoritmo de interpolação de formas proposto na Seção 2.5 do Capítulo 2 para interpolar os marcadores das imagens. Os marcadores interpolados são então utilizados para deformar as imagens com o algoritmo de deformação por campo otimizado.

A segunda parte consiste em utilizar a deformação de imagens proposta na primeira parte junto com a média morfológica. A média morfológica é utilizada com a função de interpolação proposta na Seção 4.4 e com o controle de transição proposto na Seção 4.5.

A diferença desse novo processo para outros métodos de *morphing* é que o *cross dissolve* é substituído pela média morfológica.

5.1 Implementação

A primeira parte do método proposto é implementada juntando o método proposto para interpolação de formas e deformação de imagens por campo. A segunda parte utiliza a média morfológica

com o resultado obtido na primeira parte.

No Algoritmo 5.1 é descrito o método proposto para o *morphing* de imagens juntando as duas partes propostas.

Algoritmo 5.1: Método Proposto

para cada *quadro* faça

Interpole os marcadores utilizando a interpolação de formas;

Aplique a deformação de imagens com o algoritmo de deformação por campos otimizado;

Mescle as duas imagens utilizando a média morfológica;

fim

5.1.1 Deformação de imagens utilizando interpolação de formas

A primeira proposta é a utilização da interpolação de formas para gerar os marcadores da deformação de imagens em posições intermediárias. Para calcular o mapeamento entre as imagens é utilizado o algoritmo de deformação por campo otimizado.

Normalmente, utiliza-se mais de um par de marcadores para a deformação das imagens. Então para mais de um par de curvas, o processo de interpolação de formas é aplicado sobre cada par individualmente.

O algoritmo descrito no Capítulo 2 na Seção 2.5 é utilizado para marcadores fechados. Para curvas abertas só existem duas possibilidades de correspondência inicial: ou a correspondência é feita como as linhas já estão listadas, ou na listagem inversa. Testando essas duas possibilidades na função de custo, a correspondência é escolhida pelo menor custo. Assim, pode-se utilizar o algoritmo de interpolação de formas proposto, onde a correspondência inicial não é feita pela métrica de polígonos, mas manualmente.

Com os marcadores interpolados pelo algoritmo de interpolação de formas proposto, as imagens são deformadas utilizando-se os marcadores interpolados. Como cada polígono possui o mesmo número de lados, a aplicação direta do algoritmo de deformação por campo, descrita na Seção 3.2, pode ser utilizada.

5.1.2 Média morfológica no lugar do *cross dissolve*

A segunda proposta é a utilização de média morfológica no lugar do *cross dissolve*. Uma dos problemas enfrentados pela média morfológica é que, se os atributos forem muito distantes um do outro, a imagem resultante não produz um resultado aceitável. Isso ocorre devido à característica

Algoritmo 5.2: Método proposto para interpolação de curvas abertas.

início

Calcule o custo c_1 da correlação dos polígonos P^A e P^B através da função de custo;
 Reescreva o polígono B para $P^{B_{inv}} = [P_M^B, P_{M-1}^B, \dots, P_1^B, P_0^B]$;
 Calcule o custo c_2 da correlação dos polígonos P^A e $P^{B_{inv}}$ através da função de custo;

se $c_1 < c_2$ **então**

| Correlacione os polígonos P^A e P^B através da função de custo;

fim

senão

| Correlacione os polígonos P^A e $P^{B_{inv}}$ através da função de custo;

fim

Divida os polígonos de acordo com a correlação;

para cada quadro da animação faça

| Interpole os lados e ângulos de acordo com um parâmetro de transição t ;

fim

fim

da média morfológica de interpolar as bordas, e como outras bordas ficam no meio dos atributos correspondentes, o resultado é diferente do esperado.

Iwanowski propôs a utilização de deformação de imagens por malha para melhorar os resultados da média morfológica [4]. O mesmo princípio é aplicado aqui substituindo a deformação de imagens por malhas pela deformação por campo utilizando interpolação de formas.

Controle de transição

Neste trabalho utilizam-se cortes lineares equidistantes para o controle de transição dos marcadores, enquanto, para a média morfológica, é utilizada a função cossenoidal, descrita na Seção 4.5.

Para utilizar o controle de transição cossenoidal na média morfológica é necessário também o uso da função de interpolação para imagens em tons de cinza. Assim, para cada par de imagens deformadas uma nova função de interpolação deve ser calculada, pois as imagens de entrada são diferentes para cada quadro. Isso aumenta a complexidade do algoritmo, mas os resultados são bem melhores do que a seqüência gerada pelas médias.

5.2 Resultados

Nesta seção, são apresentados os resultados do método proposto, tendo sido feitas algumas combinações de parâmetros para testar a influência dos pesos. A primeira combinação foi a variação dos pesos w_1 e w_2 sem decompor os polígonos e depois com decomposição.

Na última parte comparam-se os resultados obtidos pelo método proposto e os resultados obtidos por Iwanowski e Serra [4].

5.2.1 Influência dos pesos sem decomposição

Para descobrir o comportamento da ferramenta para diferentes valores de pesos, os testes foram feitos com duas variações. Uma com a compensação de rotação desligada e outra com a compensação ligada.

Foram geradas imagens para valores de w_1 e w_2 espaçados igualmente de 2, 5. A Fig. 5.1 mostra os resultados. Observa-se que as imagens Fig. 5.1b, Fig. 5.1c e Fig. 5.1d são idênticas, assim como a Fig. 5.1e e Fig. 5.1i. Isso ocorre porque a influência dos pesos na função de custo é relacional, isto é, o que os pesos controlam é qual termo será mais forte. Assim, para a primeira seqüência citada, w_1 é igual à zero anulando completamente a influência do termo de diferença de ângulos. Já para a segunda seqüência o termo w_2 é igual à zero, retirando o termo que tenta manter uma distribuição uniforme.

Na Fig. 5.2 é mostrado o mesmo teste feito na Fig. 5.1, só que utilizando *cross dissolve* no lugar da média morfológica. Pode-se observar que para as imagens geradas com a média morfológica tem-se uma sensação de solidez maior que as geradas com *cross dissolve*. O efeito de transparência evidencia mais o dobramento das imagens provocado pelo algoritmo de deformação de imagens, enquanto que a média morfológica suaviza esse efeito.

Para tentar descobrir mais sobre a influência dos pesos foram geradas imagens com valores de peso mais próximos. O peso w_1 foi fixado em 1.0 e w_2 foi variado de 0.0 a 7.0, como mostrado na Fig. 5.3.

Na Fig. 5.4, percebe-se que o termo de diferença de ângulos w_1 tem uma grande influência sobre a interpolação gerada. Foi preciso um valor para o peso w_2 40 vezes maior do que w_1 para se chegar a um resultado parecido com o obtido para $w_1 = 0$.

Assim, estima-se que, para valores de $w_1 = 1.0$, a zona de ajuste de w_2 é de 0.0 até 40.0. Para a faixa de [0.0, 7.0], o resultado produzido é inesperado, e pode ser utilizado para gerar efeitos de distorção nas imagens. A faixa de [7.0, 40.0] é onde o ajuste é mais linear e produz médias condizentes com o esperado. Também se tem a opção de igualar $w_1 = 0$, eliminando a influência da diferença de ângulos, o que não é recomendado já que apenas ajustando w_2 já se consegue resultados semelhantes.



Fig. 5.1: Teste com média morfológica sem compensação de rotação. Os subtítulos das figuras (b)-(l) são os valores de peso $[w_1, w_2]$.



Fig. 5.2: Teste com *cross dissolve* sem compensação de rotação. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$.



Fig. 5.3: Comparação de pesos para $w_1 = 1.0$ e $0.0 \leq w_2 \leq 7.0$. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$.

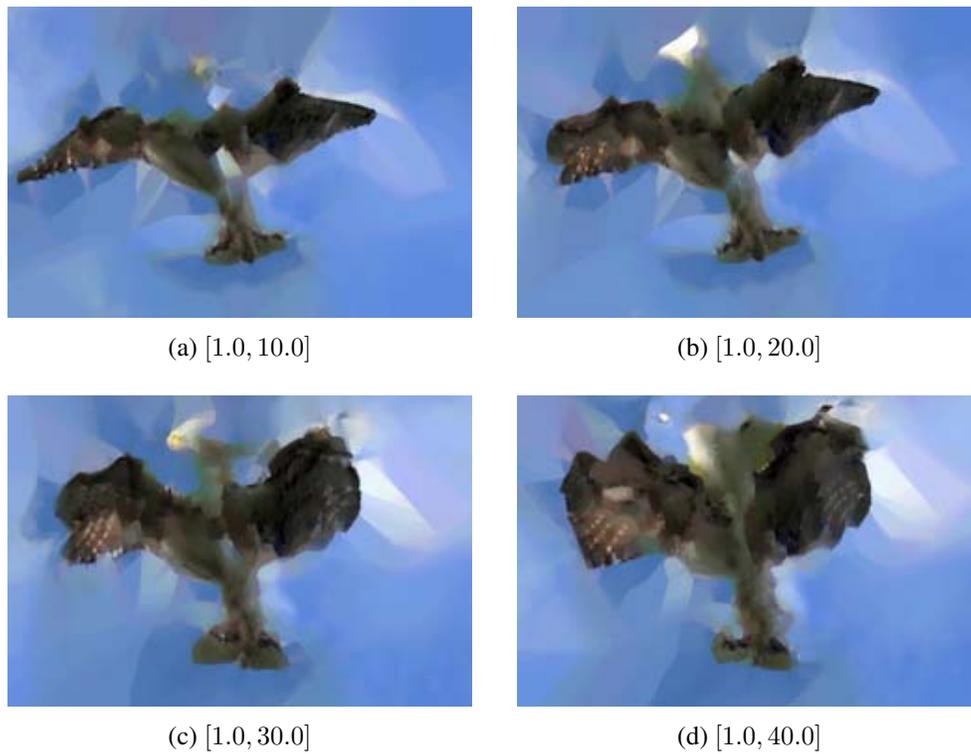


Fig. 5.4: Comparação de pesos sem compensação de rotação. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$.

Na Fig. 5.5, pode-se notar que a compensação de rotação melhora as imagens interpoladas, mas, ao mesmo tempo, diminui a influência dos pesos, limitando o controle sobre o resultado.



Fig. 5.5: Comparação de pesos com compensação de rotação. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$.

5.2.2 Influência dos pesos com decomposição

Os mesmos testes feitos na subseção anterior são feitos utilizando a decomposição de polígonos. Os polígonos foram decompostos em 6 partes, isto é, reduziu-se o número de pontos de cada polígono para 7, e foi aplicado o algoritmo de interpolação de formas descrito no Algoritmo 2.5. O número de pontos do polígono é diferente do número de partes, pois, na representação de pontos, o primeiro e o último são os mesmos pontos geometricamente.

Observando a Fig. 5.6 e a Fig. 5.7, pode-se notar que a influência dos pesos é quase nula, sendo

que a imagem gerada é sempre a mesma.

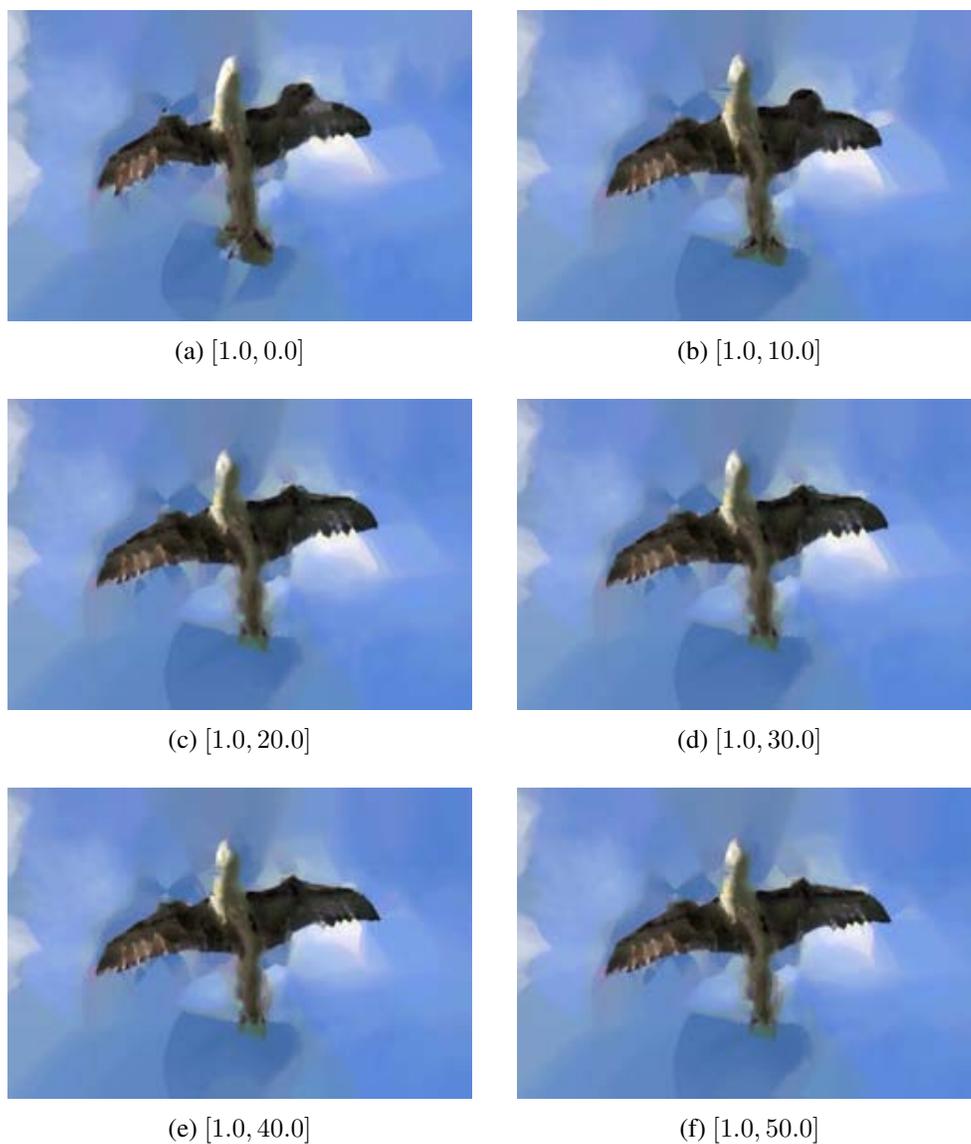


Fig. 5.6: Comparação de pesos sem compensação de rotação. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$.

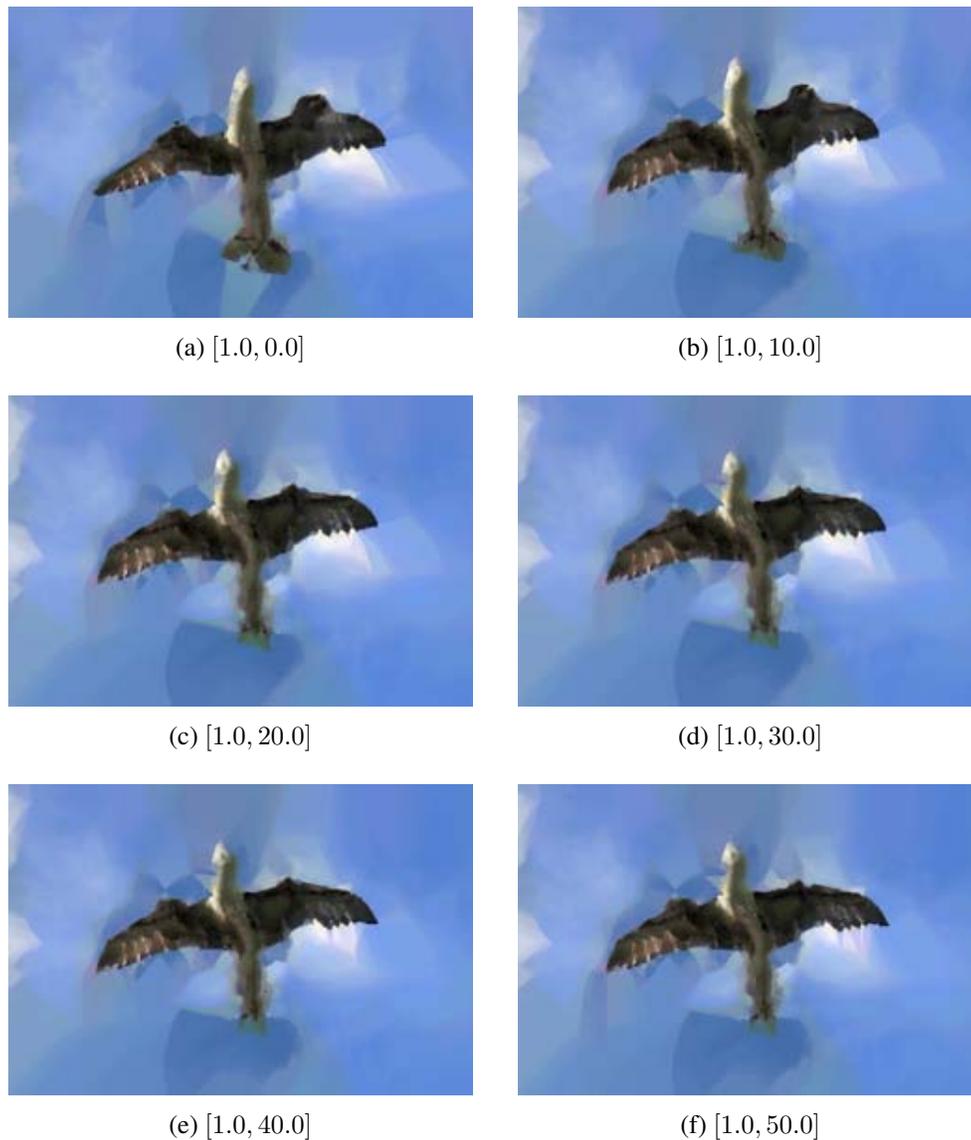


Fig. 5.7: Comparação de pesos com compensação de rotação. Os subtítulos das figuras são os valores de peso $[w_1, w_2]$.

Pelos testes, chegou-se à conclusão de que o conjunto de valores $w_1 = 1.0, w_2 = 10.0, np_1 = 7, np_2 = 7$ e $w_{rot} = 1$ produz os melhores resultados.

5.3 Seqüências geradas

Utilizando o conjunto de valores descritos na seção anterior, foram geradas algumas seqüências de imagens. Na Fig. 5.8 e Fig. 5.9, pode-se ver que as deformações obtidas pelo método proposto se comportaram como o esperado, com a asa do avião se transformando na asa da gaivota e o nariz

do avião se transformando na cabeça da gaivota. Também se pode notar como a média morfológica compensa as distorções enquanto o *cross dissolve* não consegue eliminá-las.

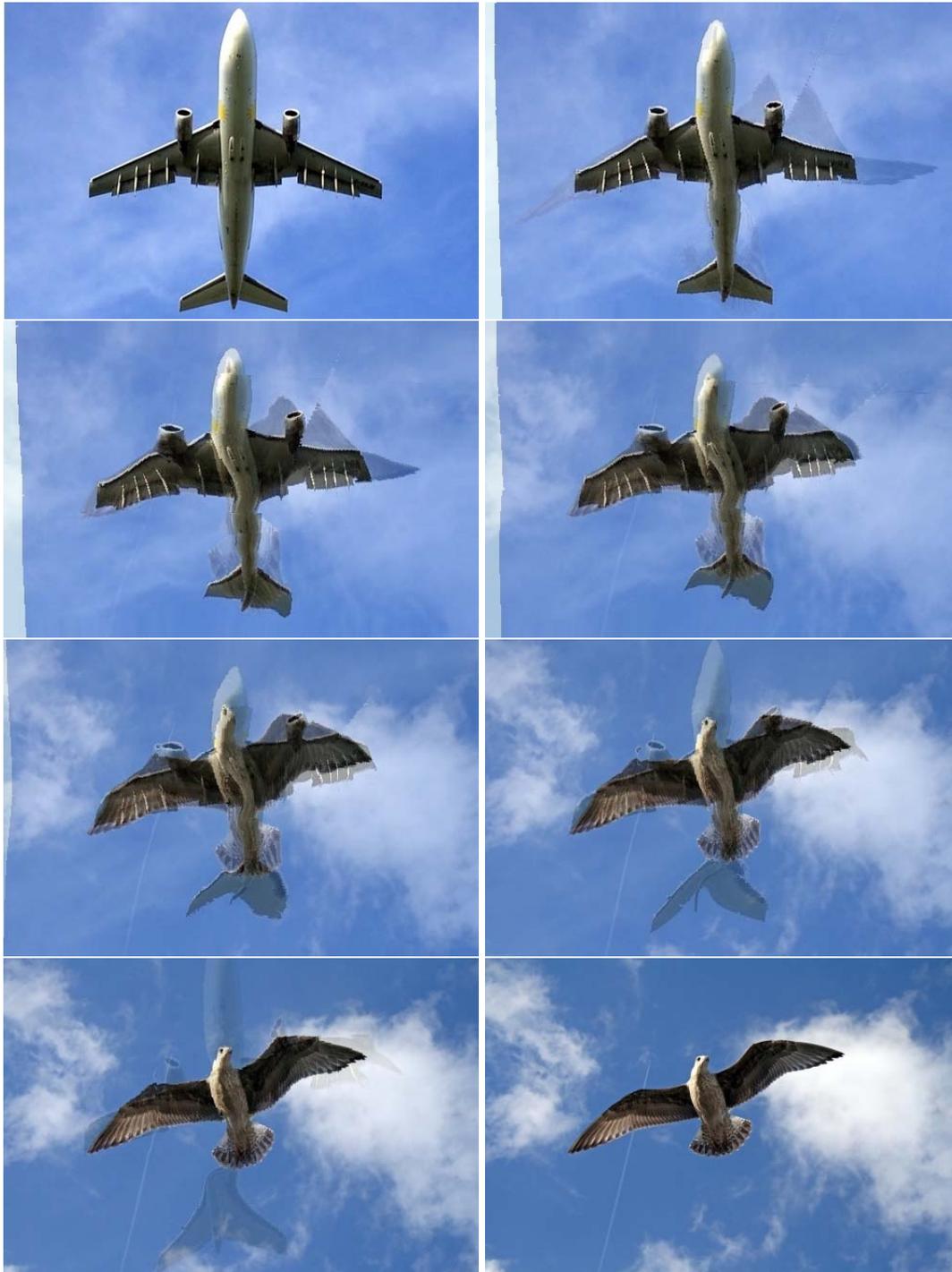


Fig. 5.8: Sequência de *morphing* entre um avião e uma gaivota utilizando *cross dissolve*.



Fig. 5.9: Seqüência de *morphing* entre um avião e uma gaiivota utilizando média morfológica.

Nas seqüências da Fig. 5.10 e Fig. 5.11, observa-se que a borda é melhor mesclada utilizando a média morfológica.



Fig. 5.10: Sequência de *morphing* entre dois tipos de pão utilizando *cross dissolve*.



Fig. 5.11: Sequência de *morphing* entre dois tipos de pão utilizando média morfológica.

Nas seqüências da Fig. 5.12 e Fig. 5.13, pode-se notar alguns artefatos provocados pela deformação de imagens: isso é causado devido à excessiva quantidade de marcadores. São os marcadores de contorno que entram em conflito com os dos olhos e boca, provocando pequenos dobramentos.



Fig. 5.12: Seqüência de *morphing* entre a Lena e a Elaine utilizando *cross dissolve*.



Fig. 5.13: Seqüência de *morphing* entre a Lena e a Elaine utilizando média morfológica.

No caso da Fig. 5.14 e Fig. 5.15, a interpolação linear obteve um resultado melhor apesar das distorções. Isso ocorre devido às bordas dessas imagens serem bem definidas, evidenciando a característica de borrimento da média morfológica.

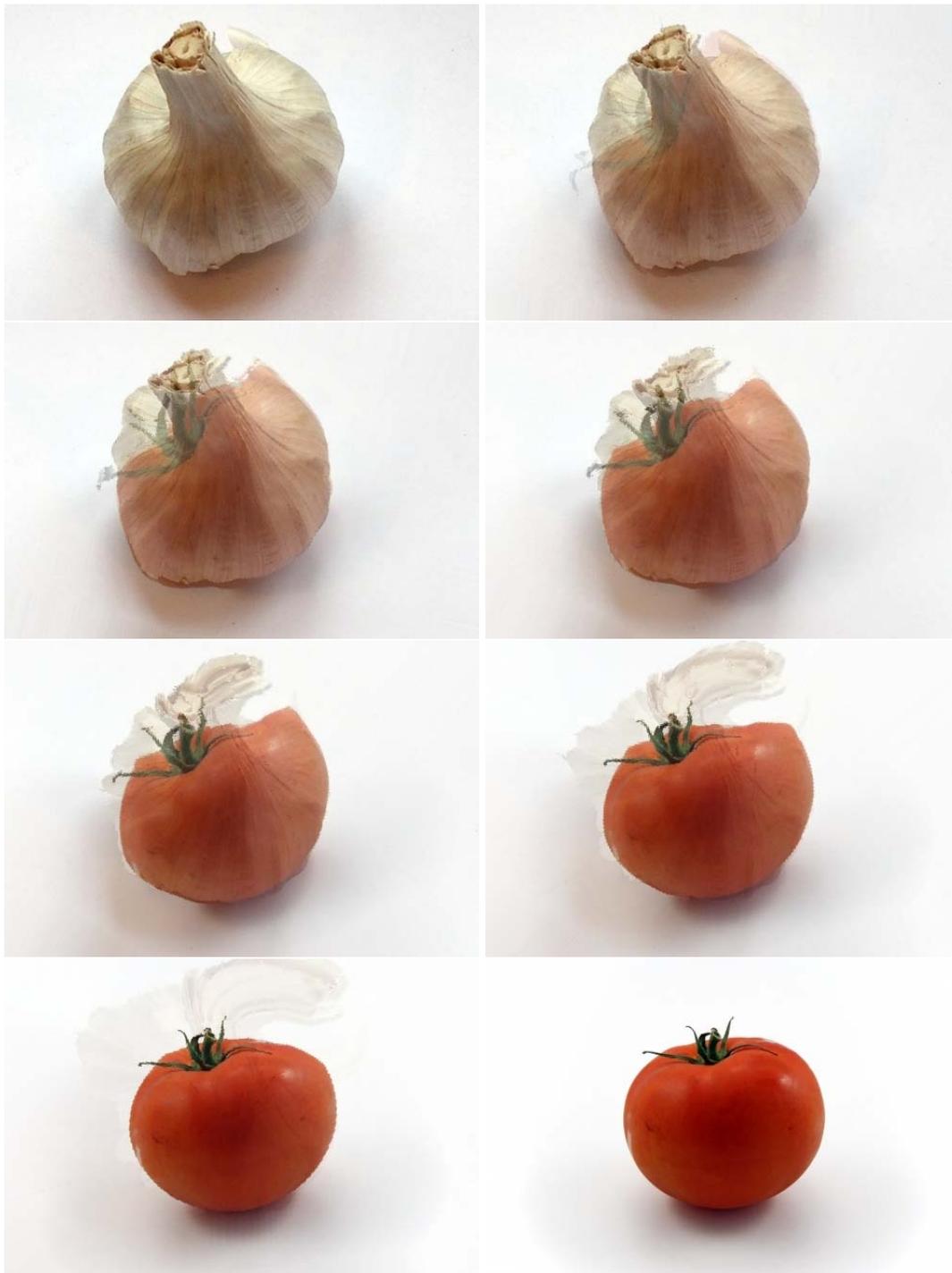


Fig. 5.14: Sequência de *morphing* entre um alho e um tomate utilizando *cross dissolve*.

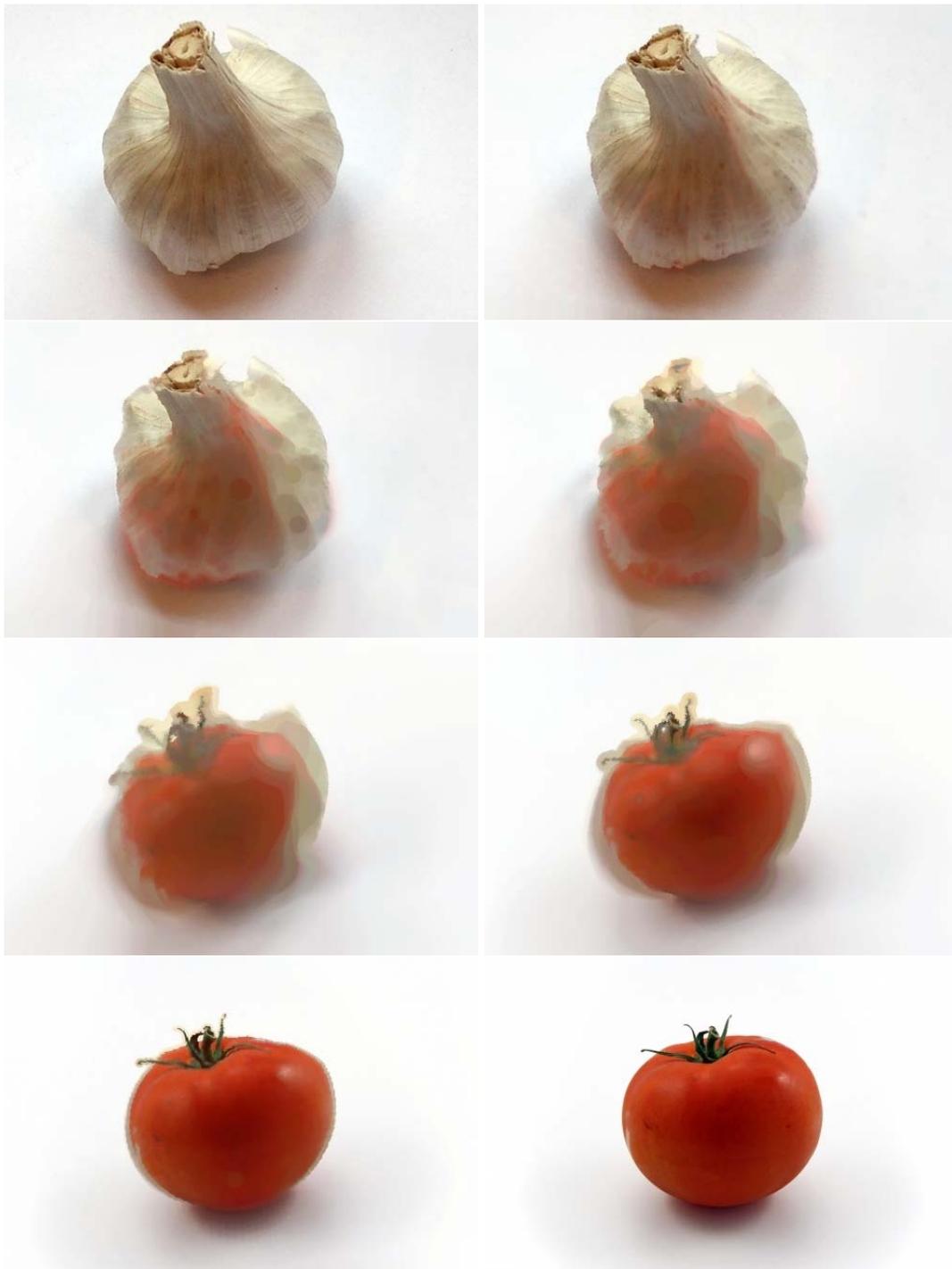


Fig. 5.15: Seqüência de *morphing* entre um alho e um tomate utilizando média morfológica.

As seqüências obtidas utilizando o conjunto de valores propostos foram boas. Assim, esse método pode ser utilizado sem muitos ajustes, mas, ao mesmo tempo, provendo a flexibilidade caso necessária. Os parâmetros do algoritmo de interpolação de formas influenciam mais os polígonos de curvas

fechadas, pois em curvas abertas há muita pouca liberdade de correspondência.

A Fig. 5.16, Fig. 5.17, Fig. 5.18 e Fig. 5.19 mostram os marcadores utilizados no *morphing*. Esses marcadores foram gerados manualmente.



Fig. 5.16: Marcadores para o *morphing* das imagens das seqüências da Fig. 5.8 e Fig. 5.9

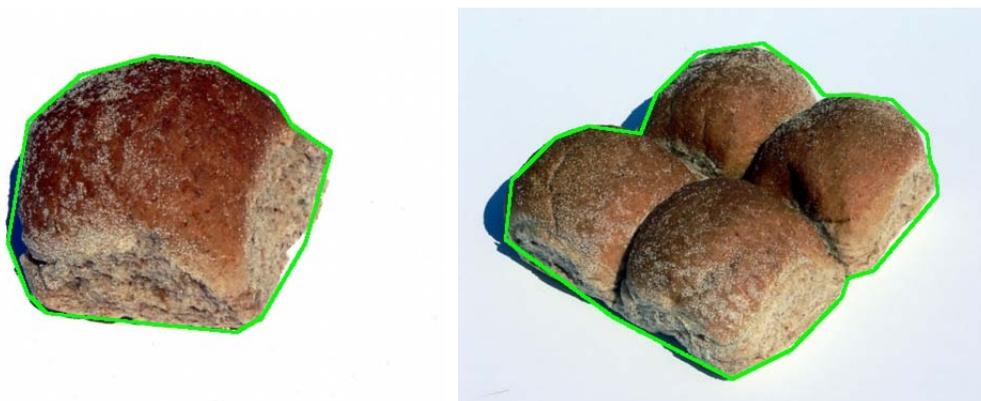


Fig. 5.17: Marcadores para o *morphing* das imagens das seqüências da Fig. 5.10 e Fig. 5.11



Fig. 5.18: Marcadores para o *morphing* das imagens das seqüências da Fig. 5.12 e Fig. 5.13

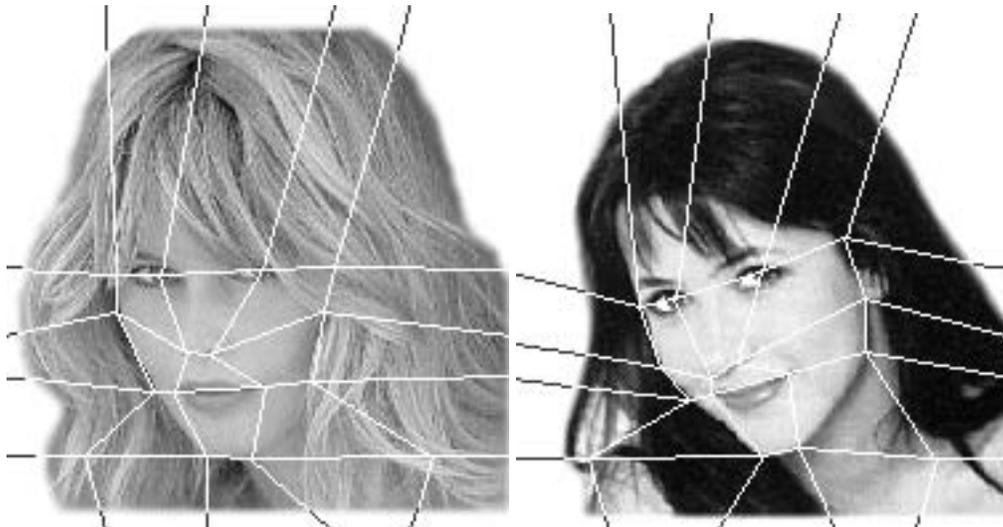


Fig. 5.19: Marcadores para o *morphing* das imagens das seqüências da Fig. 5.14 e Fig. 5.15

5.4 Comparação do método proposto com o de Iwanowski e Serra

Em [4], Iwanowski e Serra também utilizam a deformação de imagens para melhorar a média morfológica. Mas a seqüência de imagens é por médias devido a restrição da média morfológica. Assim, a primeira imagem gerada pelo algoritmo de Iwanowski e Serra é a imagem central da seqüência. O processo consiste em deformar as imagens para o meio e aplicar a média morfológica. A próxima imagem da seqüência é gerada deformando a imagem de origem e a imagem central, ou seja, diferentemente do método proposto e do tradicional, onde apenas as imagens originais são deformadas, o método de Iwanowski e Serra deformam as imagens da seqüência. Isso quer dizer que esse método tem os mesmos problemas de controle de transição da média morfológica citado na Seção 4.5.

Na Fig. 5.20 pode-se ver a diferença entre as duas formas de inserção de marcadores, por malhas e por linhas. Essa diferença, devido a quantidade de pontos, pode se traduzir em algumas horas de trabalho. Esse caso demonstra a vantagem do *morphing* por campo que insere apenas os atributos que se quer correlacionar.



(a) Iwanowski



(b) Proposto

Fig. 5.20: Marcação dos atributos do método proposto e o de Iwanowski & Serra

Na Fig. 5.21, pode-se ver a vantagem de se utilizar o controle de transição cossenoidal, enquanto que a seqüência produzida por Iwanowski tem uma transição abrupta, o método proposto neste trabalho parece mais suave durante toda a seqüência.



(a) Iwanowski & Serra

(b) Proposto



(c) Iwanowski & Serra

(d) Proposto



(e) Iwanowski & Serra

(f) Proposto

Fig. 5.21: Comparação entre o método proposto e o de Iwanowski & Serra

Comparando os resultados, notam-se duas vantagens. A primeira é que a correspondência entre as imagens utiliza apenas 3 linhas como marcadores, simplificando o processo de correspondência. A segunda é que o método proposto de controle de transição provou ser eficiente para suavizar a transição inicial e final, produzindo uma seqüência mais suave.



Fig. 5.22: Comparação entre a composição por interpolação linear e por média morfológica

Pela Fig. 5.22, percebe-se o quanto o efeito produzido pelo *cross dissolve* e pela média morfológica são diferentes. Enquanto a interpolação linear produz um efeito fantasma, o da média produz um efeito de aquarela. Um problema a ser resolvido é o das perdas das texturas na média morfológica. Na Fig. 5.22, na parte do cabelo produzido pela média morfológica apesar de a borda estar bem mesclada, o interior está sem textura, enquanto, no *cross dissolve*, ainda percebe-se a textura, mas a borda não foi bem mesclada.

5.5 Vantagens e desvantagens

O motivo de se deformar a imagem no *morphing* é o alinhamento dos atributos. Isso é feito através da inserção de marcadores, mas esse processo é bem trabalhoso e para um bom resultado é necessário muita precisão. Utilizando a morfologia matemática, que já produz bons resultados quando as imagens estão apenas um pouco deslocadas, o alinhamento das características não precisa ser tão exato, facilitando assim o trabalho do animador.

O uso de técnicas de interpolação de formas possibilita que não apenas linhas, mas pequenos

contornos, sejam correlacionados. Isso melhora a interface, deixando a ferramenta mais intuitiva, não sendo necessário se restringir apenas a correspondência de retas.

O uso do método proposto tem a vantagem de ser mais robusto, pois as imperfeições da correspondência manual são eliminadas pela média morfológica. Isso faz com que essa ferramenta fique mais acessível para usuários amadores que querem apenas brincar com as fotos obtendo assim bons resultados mesmo com marcações mais grosseiras.

Uma desvantagem é a complexidade imposta pela geração da função de interpolação para cada uma das imagens deformadas. A média morfológica tem uma desvantagem computacional muito grande com relação ao *cross dissolve*, pois enquanto que o último trabalha apenas com cada par de *pixels* individualmente a cada iteração, a média morfológica trabalha sempre com todos os *pixels* da imagem. Outra desvantagem é o efeito de borramento produzido pela média morfológica que não produz bons resultados em imagens com bordas muito evidentes.

5.6 Considerações finais

Neste capítulo, foi apresentada uma proposta que combina a interpolação de formas, controle de média morfológica e deformação de imagens. Esses três fatores levam a uma ferramenta que produz imagens originais e permitem certa flexibilidade com o ajuste de parâmetros.

Utilizando a interpolação de formas foi possível produzir marcadores de maneira mais intuitiva, além de produzir formas intermediárias que correspondem ao esperado.

A média morfológica conseguiu melhorar os resultados para os casos onde as bordas não são perfeitamente alinhadas, diminuindo assim a precisão necessária para a inserção de marcadores e consequentemente o trabalho do usuário.

O controle de transição proposto para a média morfológica também melhorou os resultados em comparação com a produção de seqüência por médias.

Capítulo 6

Conclusões e trabalhos futuros

O método proposto está longe de ser a ferramenta perfeita, apesar de melhorar a mistura das imagens. Essa mistura tem certas características que podem ser indesejáveis para o usuário, por exemplo, o efeito de borrachamento da imagem, mas mesmo assim a seqüência de metamorfoses produzida está de acordo com o resultado geral desejado.

O mais importante para essa área que trabalha com outros fatores além da lógica é a diversidade. A ferramenta proposta provê mais uma opção para quem deseja gerar uma seqüência de metamorfoses, entregando resultados diferentes de outras ferramentas.

Outro fator que mostra a importância da diversidade é que as pessoas são atraídas pela novidade. A maioria delas vai ao cinema para ver filmes nunca antes vistos, ou seja, não se vai ao cinema ver algo que já se sabe o que vai acontecer. Assim quanto maior a variedade de recursos, maior será a probabilidade de surpreender o espectador.

No método proposto foi utilizada a interpolação de formas e a média morfológica em conjunto com a deformação de imagens, formando assim uma ferramenta de *morphing* de imagens.

A interpolação de formas possui cinco parâmetros que podem ser utilizados para modificar a seqüência de imagens geradas. Por meio de testes foi possível encontrar um conjunto de valores que gera seqüências satisfatórias para a maioria das imagens.

Utilizando a interpolação de formas em conjunto com a deformação de imagens por campo otimizado, foi possível implementar uma interface mais intuitiva onde o usuário se preocupa apenas em relacionar as formas correspondentes entre as imagens. O número de pontos e o caminho deles durante a interpolação são resolvidos pelo algoritmo de interpolação de formas.

O uso da média morfológica no lugar do *cross dissolve* resultou em imagens mais sólidas, onde o efeito de transparência é eliminado. A média morfológica também eliminou problemas com a deformação de imagens quando partes do objeto são distorcidas, por exemplo, na Fig. 5.8 e Fig. 5.9 pode-se notar que mesmo onde o resultado é muito distorcido, que a média morfológica produz

resultados desejáveis. São resultados diferentes do esperado, que é um objeto metade avião e metade gaivota, mas bons o suficiente para serem utilizados em uma transição de imagens que não represente uma metamorfose. Mesmo esses resultados mais estranhos podem ser utilizados em vídeos musicais, por exemplo, onde muitos efeitos de distorção de imagens são utilizados.

Apesar de toda a representação na computação gráfica caminhar para 3D, a forma final onde o produto desse processo é visualizado ainda é 2D, isto é, as imagens são vistas em perspectiva do mundo virtual 3D. O estudo de técnicas de processamento de imagens como o *warping* e a média morfológica são muito importantes para a área de computação gráfica. A aplicação vai desde a utilização em texturas para modelos 3D até a pós-produção de imagens e vídeos.

A principal ferramenta utilizada neste trabalho foi a média morfológica. Acredita-se que essa ferramenta ainda tem muito campo a ser explorado, tanto nas aplicações como na melhoria de suas características.

6.1 Conclusões sobre a interpolação de formas

Interpolar formas geométricas é um processo complexo. O resultado nem sempre é o esperado. Uma das razões é que as formas geométricas são reconhecidas pelo ser humano como projeções de objetos tridimensionais. Assim, o resultado esperado da interpolação é baseado em informações que não estão inclusas nos modelos bidimensionais.

A correspondência inicial obtida pela métrica, apesar de aumentar a complexidade do algoritmo de correspondência, cria um parâmetro importante para a correspondência que é a compensação de rotação, o que não seria possível se apenas a função de custo fosse utilizada. Além disso, a correspondência inicial é feita com base na similaridade global e não de modo local como na utilizada na função de custo, onde se analisa apenas os valores dos ângulos dos lados.

Utilizando o algoritmo de redução de pontos de polígonos foi possível descobrir os pontos importantes para percepção humana. Uma decomposição do polígono foi feita a partir desses pontos mais relevantes. Os lados entre os pontos mais relevantes são tratados como um só lado, e assim a correspondência mais relevante é feita através desses pontos. Dessa forma, a correspondência obtida tem maiores chances de atingir o resultado esperado. A correspondência dos lados entre os pontos mais relevantes é feito com a restrição de que a correspondência entre os pontos mais relevantes não seja alterada. O processo se resume em fazer uma correspondência mais grosseira com os pontos mais relevantes e depois completar a correspondência utilizando todos os lados. Esse processo tenta imitar o modo como o ser humano realiza a correspondência entre dois objetos, onde primeiro analisa-se a forma mais geral e depois os detalhes.

Os resultados obtidos pelo método proposto tentam interpolar as formas a partir de informações

geométricas, diferente do método humano que tenta capturar um objeto a partir de mais informações, não só a geométrica. Apesar de ser diferente do que um ser humano faria, essa ferramenta pode ser utilizada para obter resultados diferentes, onde o objetivo é criar uma transformação impressionante em que a lógica não se aplica, como em filmes de fantasia. Um exemplo é uma cena de uma série de televisão onde um padre junta as duas mãos para o alto formando a cabeça do monstro, e o resto do corpo aparece a partir da cabeça.

O método proposto trabalhou apenas na primeira parte da interpolação de formas que é a correspondência. Pode-se ainda melhorar o método trabalhando na parte de interpolação dos atributos.

6.2 Conclusões sobre a média morfológica

A média morfológica produz uma interpolação visualmente interessante e que mescla melhor as imagens se comparada com o *cross dissolve*, que é uma simples interpolação linear.

A forma de uso de operadores morfológicos para imagens coloridas ainda não está bem definida, mas aplicando separadamente em cada canal de cor RGB os resultados foram satisfatórios. Os futuros resultados dos estudos de operadores morfológicos para imagens coloridas podem ser aplicados na média morfológica, e espera-se que isso a torne uma ferramenta ainda mais atrativa.

A função de interpolação possibilita um melhor controle das médias morfológicas geradas. A extensão dessa função para imagens de cinza é essencial para o desenvolvimento dessa ferramenta. Neste trabalho, essa extensão foi feita dividindo a imagem em 256 níveis e gerando uma função de interpolação para cada uma delas. A desvantagem é que essa função de interpolação não é armazenada. Outra opção é a extensão dos operadores morfológicos para estruturas tridimensionais, mas como dito na Seção 4.4 essa opção consome muita memória. A memória consumida chega a 512 vezes o tamanho da imagem, pois além dos 256 níveis de tons de cinza, ainda tem que se gerar duas funções de interpolação no caso de conjuntos parcialmente contidos como descrito na Seção 4.3. Então para imagens de alta resolução, a memória requerida é muito grande.

A armazenagem da função de interpolação é útil para gerar a seqüência entre duas imagens estáticas. Mas utilizando a técnica de *morphing* as imagens interpoladas não são as mesmas, isto é, para cada quadro da seqüência as duas imagens de entrada são diferentes. As imagens de entrada para a seqüência de *morphing* são as imagens deformadas pelo algoritmo de deformação de imagens. Assim a armazenagem da função de interpolação não é vantajosa, pois a função de interpolação gerada para duas imagens deformadas só servirá para aquele quadro. Para cada quadro uma nova função de interpolação tem que ser gerada.

O controle de transição é um fator muito importante para a animação. O uso desse controle na média morfológica proporcionou evidentes melhorias na seqüência gerada. Uma seqüência gerada

com cortes lineares na função de interpolação gera no começo e no final da seqüência de imagens uma transição abrupta, mas ao utilizar uma função que coloque mais quadros no começo e no final da seqüência, o resultado é mais suave.

6.3 Trabalhos futuros

Os métodos propostos aqui ainda têm alguns tópicos que podem ser estudados. As duas principais áreas trabalhadas foram a interpolação de formas e a média morfológica. Nas subseções seguintes são citadas algumas sugestões de estudos futuros para cada área.

6.3.1 Interpolação de formas

A interpolação entre as formas no método proposto é feita utilizando a interpolação por lados e ângulos. Estudos futuros podem substituir essa interpolação pela interpolação de triangulações, onde são geradas triangulações compatíveis entre as duas formas e os atributos interpolados são os triângulos. Esse método também tem a vantagem de poder ser utilizado para deformar imagens, onde o mapeamento é feito nos triângulos.

Outro estudo que pode ser feito é explorar mais a compensação de rotação na interpolação de formas. No método proposto existem duas opções. A compensação é feita ou não é feita, mas poder-se-ia se utilizar um parâmetro somado a essa compensação. Essa opção seria um ângulo definido pelo usuário que afetaria como as formas são correspondidas.

Uma outra sugestão é utilizar a medida de relevância utilizada para a redução de pontos dos polígonos na triangulação. Assim, permite-se uma triangulação mais significativa para polígonos.

6.3.2 Média morfológica

A média morfológica depende dos operadores básicos da morfologia matemática. Ou seja, da erosão e da dilatação, que por sua vez dependem do elemento estruturante. Estudos preliminares descobriram que a forma e o tamanho possuem grande influência na média gerada. Assim, estudos mais aprofundados de como utilizar esse elemento poderiam contribuir ainda mais para o desenvolvimento dessa ferramenta.

Outro estudo que poderia ser feito é testar outros espaços de representação para a imagem como feito em [42], que utilizou um espaço polar logarítmico obtendo resultados relevantes.

Bibliografia

- [1] George Wolberg. Recent advances in image morphing. In *CGI '96: Proceedings of the 1996 Conference on Computer Graphics International*, page 64, Washington, DC, USA, 1996. IEEE Computer Society.
- [2] Marcin Iwanowski. Generalized morphological mosaic interpolation and its application to computer-aided animations. In *CAIP '01: Proceedings of the 9th International Conference on Computer Analysis of Images and Patterns*, pages 493–501, London, UK, 2001. Springer-Verlag.
- [3] Kit Laybourne. *The Animation Book: A complete guide to animated filmmaking—from flip-books to sound cartoons to 3-D animation*. Three Rivers Press, New York, New York, new digital edition, 1998.
- [4] Marcin Iwanowski and Jean Serra. Morphological interpolation and color images. In *ICIAP '99: Proceedings of the 10th International Conference on Image Analysis and Processing*, page 50, Washington, DC, USA, 1999. IEEE Computer Society.
- [5] Henry Johan, Yuichi Koiso, and Tomoyuki Nishita. Morphing using curves and shape interpolation techniques. In *PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications*, page 348, Washington, DC, USA, 2000. IEEE Computer Society.
- [6] Jean Serra. Hausdorff distances and interpolations. In *ISMM '98: Proceedings of the fourth international symposium on Mathematical morphology and its applications to image and signal processing*, pages 107–114, Norwell, MA, USA, 1998. Kluwer Academic Publishers.
- [7] Serge Beucher. Sets, partitions and functions interpolations. In *ISMM '98: Proceedings of the fourth international symposium on Mathematical morphology and its applications to image and signal processing*, pages 307–314, Norwell, MA, USA, 1998. Kluwer Academic Publishers.
- [8] Thomas W. Sederberg and Eugene Greenwood. A physically based approach to 2-D shape

- blending. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA, 1992. ACM Press.
- [9] S. Cohen, G. Elber, and R. Bar-Yehuda. Matching of freeform curves. *Computer Aided Design*, 29(5):369–378, 1997.
- [10] Michela Mortara and Michela Spagnuolo. Similarity measures for blending polygonal shapes. *Computers & Graphics*, 25(1):13–27, 2001.
- [11] Alon Efrat, Sariel Har-Peled, Leonidas J. Guibas, and T. M. Murali. Morphing between polylines. In *SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 680–689, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.
- [12] Henry Johan and Tomoyuki Nishita. Blending multiple polygonal shapes. In *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 434, Washington, DC, USA, 2003. IEEE Computer Society.
- [13] Wenwu Yang, Jieqing Feng, Xiaogang Jin, Qunsheng Peng, and Robin Forrest. 2-d shape blending based on visual feature decomposition. In *Proc. of Computer Animation and Social Agents 2004*, pages 139–146, July 2004.
- [14] Michal Shapira and Ari Rappoport. Shape blending using the star-skeleton representation. *IEEE Computer Graphics and Applications*, 15(2):44–50, March 1995.
- [15] Thomas W. Sederberg, Peisheng Gao, Guojin Wang, and Hong Mu. 2-d shape blending: an intrinsic solution to the vertex path problem. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 15–18, New York, NY, USA, 1993. ACM Press.
- [16] Tatiana Surazhsky, Vitaly Surazhsky, Gill Barequet, and Ayellet Tal. Blending polygonal shapes with different topologies. *Computers & Graphics*, 25(1):29–39, 2001.
- [17] Craig Gotsman and Vitaly Surazhsky. Guaranteed intersection-free polygon morphing. *Computers & Graphics*, 25(1):67–75, 2001.
- [18] Esther M. Arkin, L. Paul Chew, David P. Huttenlocher, Klara Kedem, and Joseph S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. In *SODA '90: Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 129–137, Philadelphia, PA, USA, 1990. Society for Industrial and Applied Mathematics.

- [19] Longin Jan Latecki and Rolf Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000.
- [20] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 35–42, New York, NY, USA, 1992. ACM Press.
- [21] J. Xiao, C. Rao, and M. Shah. View interpolation for dynamic scenes. In *Proceedings of EUROGRAPHICS*, 2002.
- [22] Jiangjian Xiao and Mubarak Shah. Two-frame wide baseline matching. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 603, Washington, DC, USA, 2003. IEEE Computer Society.
- [23] Jiangjian Xiao and Mubarak Shah. From images to video: View morphing of three images. In *Proceedings of the Vision, Modeling, and Visualization Conference 2003 (VMV 2003)*, pages 495–502, München, Germany, November 2003. Aka GmbH.
- [24] Wolfgang Müller and Marc Alexa. Using morphing for information visualization. In *NPIV '98: Proceedings of the 1998 workshop on New paradigms in information visualization and manipulation*, pages 49–52, New York, NY, USA, 1998. ACM.
- [25] Derek F. Reilly and Kori M. Inkpen. Map morphing: making sense of incongruent maps. In *GI '04: Proceedings of Graphics Interface 2004*, pages 231–238, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.
- [26] Seung-Yong Lee, Kyung-Yong Chwa, and Sung Yong Shin. Image metamorphosis using snakes and free-form deformations. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 439–448, New York, NY, USA, 1995. ACM.
- [27] Scott Schaefer, Travis McPhail, and Joe Warren. Image deformation using moving least squares. *ACM Trans. Graph.*, 25(3):533–540, 2006.
- [28] Ligang Liu Min Meng. Sketching image morphing using moving least squares. In *The 3rd Korea-China Joint Conference on Geometric and Visual Computing*, August 2007.

- [29] Hussien Karam, A. Hassanien, and M. Nakajima. Feature-based image metamorphosis optimization algorithm. In *VSMM '01: Proceedings of the Seventh International Conference on Virtual Systems and Multimedia (VSMM'01)*, page 553, Washington, DC, USA, 2001. IEEE Computer Society.
- [30] Marcin Iwanowski and Jean Serra. The morphological-affine object deformation. In *Mathematical Morphology and Its Applications to Image and Signal Processing*, pages 81–90. Kluwer Academic Publishers, 2000.
- [31] Marcin Iwanowski and Jean Serra. Image morphing based on morphological interpolation combined with linear filtering. *International Journal of WSCG*, 1(10):233–239, 2002.
- [32] Chen Chun Bu Jia-jun, Zhao Chuan-yu. A framework to create video-transition effects. *Journal of Zhejiang University - Science A*, 6(1):58–65, August 2005.
- [33] Adrian G. Bors, Lefteris Kechagias, and Ioannis Pitas. Binary morphological shape-based interpolation applied to 3d tooth reconstruction. *IEEE Trans. Med. Imaging*, 21(2):100–108, 2002.
- [34] Marcin Iwanowski. Universal morphological interpolator. In IEEE Signal Processing Society, editor, *Proceedings of International Conference on Image Processing ICIP'05, Genova, Italy*, volume II, pages 978–981, September 2005.
- [35] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., Orlando, FL, USA, 1983.
- [36] Allan Hanbury, Jean Serra, and Centre De Morphologie Mathématique. Mathematical morphology in the $l^*a^*b^*$ colour space. Technical report, Centre de Morphologie Mathématique, Ecole des Mines de Paris, 2001.
- [37] Allan Hanbury and Jean Serra. Mathematical morphology in the hls colour space. In *Proceedings of the British Machine Vision Conference 2001*, pages 451–460. BMVA, 2001.
- [38] Edmond J. Breen, Ronald Jones, and Hugues Talbot. Mathematical morphology: A useful set of tools for imageanalysis. *Statistics and Computing*, 10(2):105–120, 2000.
- [39] Pierre Soille. Spatial distributions from contour lines: An efficient methodology based on distance transformations. *Journal of Visual Communication and Image Representation*, 2(2):138–150, June 1991.
- [40] Nathan Gossett and Baoquan Chen. Paint inspired color mixing and compositing for visualization. *Information Visualization, IEEE Symposium on*, 0:113–118, 2004.

- [41] Rick Parent. *Computer Animation Algorithms and Techniques*, pages 86–97. Morgan Kauffman Publishers, 2002.
- [42] G. Flandrin M.A. Luengo-Oroz, J. Angulo and J. Klossa. Mathematical morphology in polar-logarithmic coordinates. application to erythrocyte shape analysis. In *Pattern Recognition and Image Analysis*, volume 3523, pages 199–206. Springer Berlin / Heidelberg, 2005.
- [43] Alexander Kort. Computer aided inbetweening. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 125–132, New York, NY, USA, 2002. ACM Press.
- [44] Tomoyuki Nishita, Thomas W. Sederberg, and Masanori Kakimoto. Ray tracing trimmed rational surface patches. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 337–345, New York, NY, USA, 1990. ACM Press.
- [45] Tomoyuki Nishita and Henry Johan. A scan line algorithm for rendering curved tubular objects. In *PG '99: Proceedings of the 7th Pacific Conference on Computer Graphics and Applications*, page 92, Washington, DC, USA, 1999. IEEE Computer Society.
- [46] Thowas W. Sederberg and Tomoyuki. Curve intersection using bezier clipping. *Computer Aided Design*, 22(9):538–549, 1990.
- [47] Minglun Gong and Yee-Hong Yang. Genetic-based multiresolution color image segmentation. In *Proceedings of Vision Interface*, pages 71–80, Ottawa, ON, Canada, June 2001.
- [48] Tong-Yee Lee, Young-Ching Lin, Leeween Lin, and Yung-Nien Sun. Fast feature-based metamorphosis and operator design. *Comput. Graph. Forum*, 17(3):15–22, 1998.
- [49] Kevin Novins and James Arvo. The morphological cross-dissolve. In *SIGGRAPH '99: ACM SIGGRAPH 99 Conference abstracts and applications*, page 257, New York, NY, USA, 1999. ACM Press.
- [50] Borut Zalik. Merging a set of polygons. *Computers & Graphics*, 25(1):77–88, 2001.
- [51] Vitaly Surazhsky and Craig Gotsman. Controllable morphing of compatible planar triangulations. *ACM Trans. Graph.*, 20(4):203–231, 2001.
- [52] Francisco de Assis Zampiroli. *Transformada de Distância por Morfologia Matemática*. Doutorado em engenharia elétrica, Faculdade de Engenharia Elétrica e de Computação – Universidade Estadual de Campinas (UNICAMP), 2003.

- [53] H.J.A.M. Heijmans. Theoretical aspects of gray-level morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):568–582, 1991.

Apêndice A

Dedução de equações

A.1 Dedução da equação 2.36

Reescreve-se o termo $D_p^{A,B}(t, \theta)$ da equação 2.35 como:

$$h(t_1, t_2, \theta) = \int_0^1 |f(s+t) - g(s) + \theta|^2 ds \quad (\text{A.1})$$

$$h(t, \theta) = \int_0^1 (f(s+t) - g(s) + \theta)^2 ds \quad (\text{A.2})$$

Aplicando a derivada em A.2, obtem-se:

$$\frac{\partial h(t, \theta)}{\partial \theta} = \int_0^1 2(\theta + f(s+t) - g(s)) ds \quad (\text{A.3})$$

(A.4)

Procura-se θ que minimize a função $h(t_1, t_2, \theta)$. Então:

$$\frac{\partial h(t_1, t_2, \theta)}{\partial \theta} = 0 \quad (\text{A.5})$$

$$\int_0^1 2(\theta + f(s+t) - g(s))ds = 0 \quad (\text{A.6})$$

$$\int_0^1 2\theta ds + \int_0^1 2f(s+t) - g(s)ds = 0 \quad (\text{A.7})$$

$$2\theta = -2 \int_0^1 f(s+t) - g(s)ds \quad (\text{A.8})$$

$$\theta = \int_0^1 g(s) - f(s+t)ds \quad (\text{A.9})$$

$$(\text{A.10})$$

Substituindo $s = s + t$ no termo $\int_0^1 f(s+t)$, obtem-se:

$$\int_0^1 f(s+t)ds = \int_t^{t+1} f(s)ds \quad (\text{A.11})$$

$$= \int_t^1 f(s)ds + \int_1^{t+1} f(s)ds \quad (\text{A.12})$$

$$(\text{A.13})$$

Lembrando que a função de representação é periódica, e a relação $f(s+1.0) = f(s) + 2\pi$ é válida, tem-se:

$$\int_1^{t+1} f(s)ds = \int_0^t (f(s) + 2\pi)ds \quad (\text{A.14})$$

Substituindo na equação A.13, obtem-se:

$$\int_0^1 f(s+t)ds = \int_0^t (f(s) + 2\pi)ds + \int_t^1 f(s)ds \quad (\text{A.15})$$

$$= \int_0^1 f(s)ds + 2\pi \quad (\text{A.16})$$

Substituindo A.16 em A.10, tem-se:

$$\theta = \int_0^1 g(s) - \left(\int_0^1 f(s)ds + 2\pi \right) \quad (\text{A.17})$$

$$\int_0^1 g(s) - \int_0^1 f(s)ds - 2\pi \quad (\text{A.18})$$

A.2 Dedução da equação 2.40

A seguir deduz-se a equação 2.40 a partir das equações 2.35, 2.38 e 2.39, tem-se:

$$\Theta_A(s + t) - \Theta_B(s) = \Theta_A(f - u_i) - \Theta_B(f - v_i) \quad (\text{A.19})$$

Da equação A.19, obtem-se:

$$\begin{cases} s + t = f - u_i \\ s = f - v_i \end{cases} \quad (\text{A.20})$$

Substituindo s , tem-se:

$$f - v_i + t = f - u_i \quad (\text{A.21})$$

$$t = v_i - u_i \quad (\text{A.22})$$