

Valguima Victória Viana Aguiar Odakura Martinez

Registro de Imagens de Documentos Antigos

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de
Mestre em Engenharia.

Durante a elaboração deste trabalho a autora recebeu apoio financeiro da CAPES

São Paulo
2002

Valguima Victória Viana Aguiar Odakura Martinez

Registro de Imagens de Documentos Antigos

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de
Mestre em Engenharia.

Área de Concentração:
Sistemas Digitais

Orientador:
Prof. Dr. Geraldo Lino de Campos

FICHA CATALOGRÁFICA

Martinez, Valguima Victoria Viana Aguiar Odakura

Registro de imagens de documentos antigos. – São Paulo: EPUSP, 2002. 79 p.

Dissertação (Mestrado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Registro de imagens – Processamento de Imagens 2. Imagens de Documentos – Processamento de Imagens I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t

Resumo

Este trabalho descreve uma técnica para alinhamento de imagens de documentos antigos. Este alinhamento é necessário como parte de um processo para remover a influência da transparência do papel, que é alta nos papéis antigos. O processo requer duas imagens, uma com fundo branco e outra com fundo preto. As imagens apresentam diferenças de aquisição que incluem translação, rotação e distorções geométricas.

A técnica de registro de imagens é formada por três etapas. A primeira é uma pré-seleção de pontos de controle. Nesta etapa, as imagens são segmentadas em linhas e a seguir as linhas são segmentadas em palavras. O início de cada palavra é considerado um ponto de controle tentativo. Este método é sujeito a erros e pode realizar seleções falsas. O segundo estágio realiza uma correspondência entre os pontos de controle extraídos das duas imagens, assegurando uma correspondência correta entre os pontos de controle. Na última etapa a segunda imagem é mapeada utilizando funções *thin plate spline*, obtendo-se uma imagem alinhada com a primeira.

Abstract

A technique to register different images of ancient documents is described. This registration is required as part of process to remove the influence of paper transparency, high on old types of paper. The process requires two images, one with white and other with black background. The acquired images are different due to translation, rotation and geometric distortions during the acquisition process.

The registration process is divided in three stages. The first is a process of control points selection. In this stage, page images are segmented first in text lines and then in words. The beginning of each word is taken as a tentative control point. This method, however, is error prone and makes false selections. The second stage refines the control point selection, assuring the true correspondence of the control points among the images. In the last stage the second image is mapped to the reference image using thin plate spline functions.

ao meu pai
Kaoru
(in memoriam)

Agradecimentos

Ao Prof. Geraldo, por ter me orientado, pela paciência e pelos ensinamentos determinantes para a conclusão desta dissertação.

A Profa. Anna, pelo apoio durante o desenvolvimento deste trabalho.

A minha família, especialmente minha mãe e meu irmão, que mesmo a quilômetros de distância nunca deixaram de estar presentes.

Aos amigos que diretamente me ajudaram Alice e Said, ouvindo meus problemas e dando sugestões. A amiga, Ana Flavia, a quem eu recorri muitas vezes para compartilhar minhas incertezas. E a todos aqueles que me ajudaram com a amizade e apoio, tornando melhor a vida aqui em São Paulo: Brown, Claus, Débora, Eliany, Fátima, Jair, Karin e Marco Aurélio.

Em especial ao meu amor, amigo, companheiro, corretor... Fábio, por ter me incentivado a fazer o mestrado e por todos os momentos que compartilhou comigo durante essa jornada. Não conseguiria expressar em palavras toda minha gratidão.

Sumário

Lista de Figuras	vii
1 Introdução	1
2 Registro de Imagens	3
2.1 O Problema	3
2.2 Registro de Imagens	5
2.2.1 Seleção dos Pontos de Controle	6
2.2.2 Correspondência de Pontos de Controle	7
2.2.3 Funções de Mapeamento	8
2.3 Registro de Imagens com Distorções Geométricas	11
2.3.1 Funções <i>Splines</i>	11
2.3.2 <i>Thin Plate Splines</i>	12
2.3.3 Cálculo dos Coeficientes das Funções TPS	13
2.3.4 Complexidade Computacional do TPS	15
3 Seleção de Pontos de Controle	17
3.1 Aplicações de Registro de Imagens de Documentos	18
3.2 Primeira Idéia	18
3.3 Pontos dos Extremos das Linhas	19
3.3.1 Segmentação da Página de Texto em Linhas	19
3.3.2 Correção da Inclinação de Imagens de Páginas de Documentos	25
3.3.3 Seleção dos Pontos de Controle dos Extremos da Linhas do Texto	30

3.3.4	Teste do Registro com Pontos de Controle dos Extremos das Linhas	30
3.4	Pontos dos Inícios das Palavras	38
3.4.1	Segmentação de Linhas de Texto em Palavras	38
3.4.2	Correspondência dos Pontos de Controle do Início das Palavras	41
3.4.3	Teste do Registro com Pontos de Controle do Início das Palavras	43
3.5	Análise dos Resultados	43
4	Remoção do Efeito da Transparência	49
4.1	Remoção do Efeito da Transparência para Imagens Binárias	50
4.2	Remoção do Efeito da Transparência por Stolfi	50
5	Conclusão	55
A	Implementação	57
A.1	Implementação em Matlab	58
	Referências Bibliográficas	75

Lista de Figuras

2.1	Imagem de um livro aberto. Os locais de possíveis distorções são: a parte central, com mancha escura, representa a junção das páginas; e paralelamente à parte central, os extremos esquerdo e direito.	5
2.2	Combinação de tipos possíveis de transformações. De cima para baixo e da esquerda para direita tem-se as seguintes transformações: globais (rígida, afim, projetiva e elástica) e locais (rígida, afim, projetiva e elástica). Figura retirada de [8].	10
3.1	Imagem de uma porção de uma página com linhas de texto. . . .	20
3.2	Histograma da imagem de uma porção de uma página com linhas de texto.	21
3.3	Linhas características da linha de texto. De cima para baixo são apresentadas as linhas ascendente, início, fim e descendente, respectivamente.	21
3.4	Linhas de texto detectadas na imagem de uma porção da página: as linhas brancas delimitam a parte média da linha de texto, ou seja, as linhas início e fim.	23
3.5	Erro na detecção das linhas de texto. A última linha de texto da imagem não foi detectada.	23
3.6	Imagem de uma porção de uma página de texto inclinada.	24
3.7	Histograma da imagem de uma porção de uma página de texto inclinada.	24
3.8	Linhas de texto detectadas na imagem de uma porção da página inclinada: as linhas brancas delimitam a parte média da linha de texto, ou seja, as linhas início e fim.	25
3.9	Imagem de uma porção da página inclinada. A região de texto entre linhas brancas corresponde a região entre as linhas ascendentes e início da média.	26

3.10	Imagem de uma porção da página sem inclinação. A região de texto entre linhas brancas corresponde a região entre as linhas ascendentes e início da média.	26
3.11	Imagem da página com inclinação corrigida.	28
3.12	Imagem de referência digitalizada com uma folha de fundo branca.	29
3.13	Imagem destino digitalizada com uma folha de fundo preta.	32
3.14	Soma das imagens de referência e destino.	33
3.15	Pontos de controle da imagem de referência.	34
3.16	Pontos de controle da imagem destino.	35
3.17	Registro da imagem destino.	36
3.18	Soma da imagem destino registrada com a imagem de referência.	37
3.19	Histograma de uma imagem de uma linha de texto.	39
3.20	Histograma de espaços em branco da imagem de uma linha de texto.	39
3.21	Segmentação da linha de texto em palavras. Cada linha vertical em branco representa o início de uma palavra.	40
3.22	Falhas na segmentação da linha de texto em palavras.	40
3.23	Pontos de controle da imagem de referência.	44
3.24	Pontos de controle da imagem destino.	45
3.25	Registro da imagem destino.	46
3.26	Soma da imagem registrada com imagem de referência.	47
4.1	Imagem com efeito da transparência.	51
4.2	Imagem com efeito da transparência removido.	52
4.3	Imagem com o efeito da transparência.	54
4.4	Imagem com o efeito da transparência removido utilizando-se a técnica desenvolvida por Stolfi [22].	54
4.5	Imagem obtida da imagem 4.4 com o filtro $y = x/0.6 - 1/3$	54

Capítulo 1

Introdução

Existe grande interesse em converter documentos de papel para um formato eletrônico, utilizável por computador. Existem muitas aplicações que realizam esta conversão de documentos como, por exemplo, reconhecimento de assinaturas, arquivamento de documentos e bibliotecas digitais.

A conversão de um documento para um formato eletrônico segue um processo de transformação. O documento é inicialmente produzido, depois manipulado ao longo de sua vida e por fim digitalizado. Na fase de preparação o documento sofre influências da qualidade do papel, da tinta e do equipamento de impressão. Durante sua manipulação, de acordo com o manuseio exercido sobre o documento, as suas folhas podem apresentar alterações, como por exemplo amassados. No momento da digitalização em um aparelho de *scanner* a imagem digital resultante pode apresentar ruídos, inclinação e distorções da página.

Dessa forma, podem ocorrer algumas perturbações na imagem do documento durante a sua obtenção. Essas perturbações devem ser minimizadas utilizando-se técnicas de processamento de imagens.

Neste trabalho estuda-se o processamento de imagens resultantes da digitalização de livros e documentos antigos, de importância cultural ou histórica. Tal tarefa contribui indubitavelmente para que obras raras possam ser melhor difundidas.

Um dos principais problemas da digitalização de livros antigos é o tipo de papel utilizado naquela época. Esse papel era mais transparente que aquele utilizado hoje e, por isso, deixava visível a impressão do verso da folha. Tal problema pode não parecer grave para leitores do documento original, uma vez que o olho humano consegue distinguir entre as marcas da folha da frente e as marcas do verso da folha. Entretanto, quando a imagem é digitalizada e exibida em um monitor, as marcas do verso da folha aparecem como ruído e dificultam

bastante a leitura, tornando necessário realizar um tratamento da imagem para eliminar tais marcas do verso da folha.

A motivação desse trabalho é então solucionar o problema da remoção do efeito da transparência do papel, ou seja, deseja-se obter uma imagem digital de um lado do documento, sem marca alguma do lado oposto.

Para remover o efeito da transparência, segundo Stolfi [22], cada página do livro deve ser digitalizada duas vezes, uma sobre um fundo branco e outra sobre fundo preto. Ao digitalizar duas vezes a mesma página do documento as imagens resultantes podem apresentar diferenças. Intuitivamente, ao digitalizar uma página do livro sobre uma folha preta, o efeito da transparência é praticamente eliminado, e ao digitalizar a mesma página sobre uma folha branca, o efeito da transparência é realçado. Essas duas imagens devem ser alinhadas e, por fim, processadas para eliminar o efeito da transparência.

Este trabalho trata principalmente do problema do alinhamento, uma vez que para eliminar os efeitos da transparência já existe um método, proposto por Stolfi, que apresenta bons resultados.

Desta forma, o objetivo deste trabalho é realizar o alinhamento de duas imagens da mesma página de um documento, que possuem diferenças devido a sua aquisição. Para solucionar o problema do alinhamento das imagens serão estudadas técnicas de registro de imagens.

No Capítulo 2 o problema do alinhamento será detalhado, especificando quais diferenças entre as imagens deseja-se minimizar. A técnica de registro de imagens será introduzida e dividida em três partes, sendo elas a seleção de pontos de controle, a correspondência desses pontos e por fim a estimativa da função de mapeamento.

A seleção dos pontos de controle será vista em detalhes no Capítulo 3. A fim de extrair pontos de controle de imagens de documentos serão estudadas técnicas de processamento de documentos. Duas técnicas para seleção de pontos de controle serão apresentadas e comparadas.

Com o problema do registro resolvido, o problema da remoção do efeito da transparência será abordado no Capítulo 4. Será descrita brevemente a técnica proposta por Stolfi e será mostrada uma imagem com o efeito da transparência removido.

Por fim, no Capítulo 5 as conclusões do trabalho serão apresentadas.

Capítulo 2

Registro de Imagens

A remoção do efeito de transparência de imagens digitalizadas de livros antigos, segundo a técnica de Stolfi [22], requer a aplicação de algumas técnicas computacionais. Essas técnicas envolvem um modelo matemático para a transparência, que utiliza o registro de imagens como uma técnica de apoio.

O modelo proposto por Stolfi requer que cada página do documento seja digitalizada por um *scanner* duas vezes, uma com uma folha de fundo branca e a outra com uma folha de fundo preta. Ao digitalizar duas vezes a mesma página do documento as imagens resultantes podem apresentar diferenças. Deseja-se então alinhar essas imagens de forma que suas diferenças sejam minimizadas. Para alinhar as imagens utiliza-se uma técnica de registro de imagens.

Na Seção 2.1 será descrito o problema da transparência e as diferenças encontradas nas imagens de entrada. Na Seção 2.2 a técnica de registro de imagens será explicada e dividida em três passos que envolvem seleção de pontos de controle na Seção 2.2.1, correspondência de pontos de controle na Seção 2.2.2 e estimativa de funções de mapeamento na Seção 2.2.3. Uma técnica para registrar imagens com distorções geométricas será mostrada na Seção 2.3. Na Seção 2.3.1 as funções *splines* serão introduzidas. Em seguida, uma técnica para registrar imagens com distorções geométricas denominada TPS será descrita na Seção 2.3.2. O cálculo dos coeficientes das funções TPS será tratado na Seção 2.3.3 e a complexidade computacional das funções TPS será avaliada e apresentada na Seção 2.3.4.

2.1 O Problema

Muitos tipos de papel são parcialmente translúcidos, o que significa que deixam passar alguma luz através de sua espessura, possibilitando que as marcas do lado oposto da folha sejam vistas. Para um leitor do documento original esse

problema não parece ser grave, uma vez que o olho humano consegue distinguir entre as marcas da frente e as marcas do verso da folha. No entanto, em imagens digitais de documentos em papel o efeito da transparência é muito aparente e não pode ser distinguido do mesmo modo que pela vista humana. O processo de visão computacional não consegue eliminar o fenômeno.

A transparência do papel descrita anteriormente é chamada de **transparência óptica**: é esta transparência que será tratada neste trabalho, uma vez que os papéis utilizados em documentos antigos são muito mais transparentes que os atualmente em uso. Existe outro tipo de transparência denominada **transparência física**, que se deve ao fato da tinta espalhar-se através da espessura do papel e aparecer no lado oposto. Embora com papéis e tintas modernos a transparência física esteja presente, seu tratamento foi excluído do escopo deste trabalho, uma vez que os papéis aqui tratados são papéis antigos, que eram relativamente mais espessos que os utilizados atualmente e este fenômeno é menos aparente.

Para resolver o problema da transparência óptica um modelo em camadas do documento foi proposto por Stolfi [22]. O modelo consiste de uma camada de papel entre duas camadas de tinta e denomina-se **modelo sanduíche**. Segundo esse modelo, cada página do documento deve ser digitalizada por um *scanner* duas vezes, uma com uma folha de fundo branca e outra com uma folha de fundo preta.

Ao digitalizar duas vezes a mesma página do documento, as imagens resultantes podem apresentar diferenças em relação à inclinação e à posição no plano. Esse processo faz com que as coordenadas dos pontos em uma imagem não coincidam com as coordenadas na outra imagem, ou seja, o mesmo par de coordenadas nas duas imagens não represente a mesma porção do documento. Deseja-se então alinhar essas imagens de forma que suas coordenadas sejam correspondentes.

Para que o alinhamento seja possível há ainda uma outra diferença presente nas imagens que precisa ser tratada. Como as imagens resultam de livros, que em geral são volumosos, as imagens podem apresentar distorções. Dado um livro volumoso aberto, a distorção pode aparecer no meio do livro, mais precisamente em uma porção na junção da página do lado esquerdo com a página do lado direito. A distorção pode ainda ocorrer no extremo esquerdo da página esquerda e no extremo direito da página direita. A Figura 2.1 ilustra o local onde as distorções mencionadas aparecem na imagem de páginas de livros.

Uma vez que esses livros são exemplares antigos de importante valor cultural, não é sensato pensar os livros em um *scanner* para que a imagem resultante não apresente tais distorções. Uma alternativa é utilizar um *scanner* planetário. Em tal aparelho o livro fica aberto sobre uma base e sobre ele existe um sistema móvel que possibilita a digitalização sem contato do aparelho com



Figura 2.1: Imagem de um livro aberto. Os locais de possíveis distorções são: a parte central, com mancha escura, representa a junção das páginas; e paralelamente à parte central, os extremos esquerdo e direito.

as folhas do livro. O fato de não haver contato com as folhas do livro facilita a digitalização de exemplares volumosos e frágeis. Esse tipo de *scanner* possui um mecanismo que permite eliminar as distorções causadas pela espessura do livro e pela curvatura da lombada. Isto entretanto não soluciona o problema de posicionamento e eventuais deformações da folha quando são realizadas duas operações de digitalização por página, com fundos branco e preto.

Assim, um primeiro passo para resolver o problema da transparência é, dadas duas imagens da mesma página do documento, alinhá-las de maneira que as distorções sejam ajustadas. Após solucionar o problema do alinhamento, a eliminação do efeito da transparência do papel pode ser obtida utilizando a técnica desenvolvida por Stolfi.

2.2 Registro de Imagens

O registro de imagens é uma tarefa comum em aplicações de processamento de imagem e visão computacional. O registro é utilizado para coincidir duas ou mais imagens obtidas de diferentes sensores, em diferentes tempos ou de diferentes pontos focais. Um grande número de métodos de registro de imagens pode ser encontrado na literatura, como por exemplo em [4] e [8]. Qualquer dos métodos produz um conjunto de equações que transformam as coordenadas de cada ponto de uma imagem em coordenadas do ponto correspondente na outra imagem. A tarefa do registro é, então, determinar como transformar a primeira imagem de forma que ela coincida com a segunda.

São consideradas duas imagens da mesma cena que apresentam diferenças devido ao processo de aquisição, representando a imagem de referência e a imagem destino. O **registro de imagens** pode ser definido como sendo um mapeamento entre as duas imagens de forma que essas diferenças sejam minimizadas. As funções de transformação entre as imagens são f_x e f_y como segue

$$u = f_x(x, y) \quad e \quad (2.1)$$

$$v = f_y(x, y). \quad (2.2)$$

Tais funções relacionam as coordenadas da imagem de referência na forma (x, y) com as coordenadas da imagem destino na forma (u, v) .

O processo de registro de imagens pode ser dividido em três passos: seleção dos pontos de controle, correspondência desses pontos e estimativa da função de mapeamento. No primeiro deles, são determinados dois conjuntos de pontos de controle. Ou seja, são determinados dois subconjuntos de pontos, um da imagem I_1 e outro da imagem I_2 , denotados por C_1 e C_2 . No segundo, os pontos em C_1 devem ser correspondentes aos pontos em C_2 , isto é, para cada ponto em C_1 que representa uma porção da imagem existe um ponto em C_2 que representa a mesma porção da imagem. Esses pontos são chamados de **pontos de controle**. No terceiro passo, esses pontos correspondentes são utilizados para estimar uma função de mapeamento que possa relacionar os pontos restantes nas imagens. A função de mapeamento é escolhida entre os vários tipos de transformações. As seções seguintes tratam da seleção e correspondência dos pontos de controle e dos vários tipos de transformações.

2.2.1 Seleção dos Pontos de Controle

A determinação dos pontos de controle correspondentes nas imagens é uma tarefa muito importante no processo de registro. A seleção dos pontos é feita a partir de características das imagens. É necessário que a quantidade de pontos de controle nas imagens seja suficiente para caracterizar adequadamente as distorções de interesse.

Pontos de controle podem ser intrínsecos ou extrínsecos, de acordo com Brown [4]. Pontos de controle extrínsecos são marcas na imagem que são colocadas na cena especificamente para o propósito do registro e são facilmente identificados. Pontos de controle intrínsecos são determinados a partir da imagem, de forma manual ou automática. Pontos de controle manuais são reconhecidos através da intervenção humana. Para que tais pontos sejam reconhecidos, há a necessidade de um especialista no assunto, como por exemplo, um médico para reconhecer estruturas anatômicas em uma imagem médica. Os pontos de controle intrínsecos localizados automaticamente utilizam-se de características da imagem.

Barnea e Silverman [1] encontraram pontos de controle em imagens que possuem apenas diferenças de translação selecionando um número de janelas, localizando as janelas na outra imagem e usando os centros de janelas correspondentes como pontos de controle. Por janela de uma imagem I de tamanho $n \times n$, entende-se um subconjunto da imagem I , denotado por J , de tamanho $m \times m$, com $m < n$.

Se linhas retas podem ser encontradas nas imagens, as intersecções dessas linhas e os pontos de fim de linha resultam em pontos de controle, conforme Stockman et al. [21]. Esta técnica pode ser utilizada para imagens com diferenças de translação e rotação.

Se as imagens podem ser segmentadas e bordas de regiões fechadas podem ser determinadas, os centros de gravidade das regiões produzem pontos de controle, de acordo Goshtasby [10]. Esta técnica pode ser utilizada para imagens com diferenças de translação, rotação e escala.

2.2.2 Correspondência de Pontos de Controle

Em seguida ao passo da determinação do conjunto de características, as características em uma imagem devem ser correspondidas com as características na outra imagem. Para pontos de controle identificados manualmente, as tarefas de encontrar os pontos e determinar a correspondência podem ser feitas simultaneamente. Para pontos que são encontrados automaticamente, a correspondência pode ser determinada baseada nas propriedades dos pontos. Existem muitos métodos para a correspondência automática dos pontos de controle, entre eles o de *clustering*, de distância entre pontos e métodos de relaxamento.

Na técnica de *clustering* em [21], todos os pares possíveis de características da imagem de referência com a imagem destino são determinados. Para cada par possível de características correspondidas são calculados os parâmetros das transformações de translação, rotação e escala. Correspondências corretas tendem a formar um *cluster*. Os valores dos parâmetros correspondendo ao *cluster* mais denso são tomados como parâmetros da transformação, que pode corresponder os dois conjuntos de pontos de controle.

No método de distância entre pontos, uma matriz de distância entre os pontos de controle é utilizada para determinar a similaridade entre os pontos. Lavine et al. [13] representam a matriz na forma de uma árvore geradora mínima, em que os pontos definem os vértices do grafo e as distâncias entre os pontos definem os pesos nas arestas. No trabalho de Wong e Salay [25], uma árvore representa toda correspondência possível entre os pontos de controle da imagem destino e da imagem de referência. Uma função de custo é definida para uma correspondência de tamanho n como a soma das diferenças entre to-

das as distâncias entre pontos correspondentes. Uma correspondência entre as imagens é encontrada procurando na árvore pelo mapeamento de custo mínimo com o máximo número de pontos que se correspondem dentro de uma dada tolerância.

Técnicas de relaxamento iterativamente determinam valores para cada mapeamento possível, de forma a garantir que os valores permaneçam consistentes. A solução é encontrada quando esses valores convergem.

Há na literatura muitos métodos para seleção e correspondência de pontos de controle automáticos. Stockman et al. [21] utilizaram intersecções de linhas como pontos de controle e determinaram a correspondência entre eles por uma técnica de *clustering*. Barnea e Silverman [1] utilizaram centros de janelas como pontos de controle e por correlação das imagens estabeleceram a correspondência entre eles. Goshtasby et al. [10] utilizaram centros de gravidade de regiões fechadas como pontos de controle e por um processo de relaxamento determinaram os pontos correspondentes.

2.2.3 Funções de Mapeamento

Sejam I_1 a imagem de referência e I_2 a imagem destino. Sejam também (x, y) uma coordenada de I_1 e (u, v) uma coordenada de I_2 . A distorção geométrica entre as imagens pode ser descrita por:

$$u = \alpha(x, y) \quad \text{e} \quad (2.3)$$

$$v = \beta(x, y), \quad (2.4)$$

onde α e β são funções desconhecidas.

Lembrando que um conjunto de pontos de controle é um subconjunto de pontos da imagem, sejam (x_i, y_i) e (u_i, v_i) , para todo $i = 1, \dots, n$, as posições de n pontos de controle correspondentes nas duas imagens, representando a imagem de referência e a imagem destino, respectivamente. A correspondência entre os pontos é estabelecida de forma que

$$u_i = \alpha(x_i, y_i) \quad \text{e} \quad (2.5)$$

$$v_i = \beta(x_i, y_i), \quad (2.6)$$

para todo $i = 1, \dots, n$.

Deseja-se determinar as funções de mapeamento f_x e f_y de modo que o registro de imagens seja representado por

$$u = f_x(x, y) \quad \text{e} \quad (2.7)$$

$$v = f_y(x, y), \quad (2.8)$$

da forma mais precisa possível, de maneira que as distâncias $\|f_x - \alpha\|$, $\|f_y - \beta\|$ sejam mínimas, de acordo com alguma norma. Isto é, deseja-se encontrar as funções f_x e f_y para mapear os pontos de controle correspondentes de maneira que

$$u_i = f_x(x_i, y_i) \quad \text{e} \quad (2.9)$$

$$v_i = f_y(x_i, y_i), \quad (2.10)$$

para todo $i = 1, \dots, n$, com erro mínimo.

Para determinar corretamente a transformação a ser utilizada é necessário identificar os tipos de distorção que as imagens possuem. Essas distorções podem incluir ruído no sensor, mudanças perspectivas do ponto focal do sensor, mudanças no objeto tais como movimentos, deformações ou crescimento, mudanças na luz e atmosfera incluindo sombras e nuvens, e diferentes sensores.

Em geral, os métodos de registro são desenvolvidos para alinhar imagens com várias possíveis distorções e não é óbvio para que tipos de problemas cada método é mais adequado. A diversidade de problemas e suas aplicações fez com que se desenvolvessem muitos métodos independentes para o registro. Por causa dessa grande variedade de métodos é difícil classificar e comparar as técnicas. Brown [4] fez um apanhado dos métodos existentes e van den Elsen et al. [8] fizeram uma revisão dos métodos voltada para a área médica.

Transformações

A característica fundamental de qualquer técnica de registro de imagem é o tipo de transformação espacial utilizado para sobrepor duas imagens. Uma **transformação** é um mapeamento de localizações de pontos em uma imagem para novas localizações em outra imagem.

As transformações utilizadas para alinhar duas imagens podem ser globais ou locais. Uma transformação é **global** quando é dada por uma única equação que mapeia a imagem inteira. Já uma transformação **local** mapeia a imagem diferentemente dependendo da localização espacial e é muito mais difícil de expressar sucintamente. As transformações mais comuns, segundo Van Den Elsen et al. [8], são rígida, afim, projetiva, perspectiva e elástica. Estas transformações podem ser globais ou locais conforme ilustra a Figura 2.2. Cada uma delas será descrita brevemente a seguir.

A **transformação rígida** é usada para imagens obtidas com movimento do sensor ou objeto, em que as imagens mantêm suas formas e tamanhos. Nesse tipo de transformação, a distância entre dois pontos quaisquer na primeira imagem é preservada quando são mapeados para a segunda imagem. Essa transformação pode ser decomposta em rotação, translação e escala.

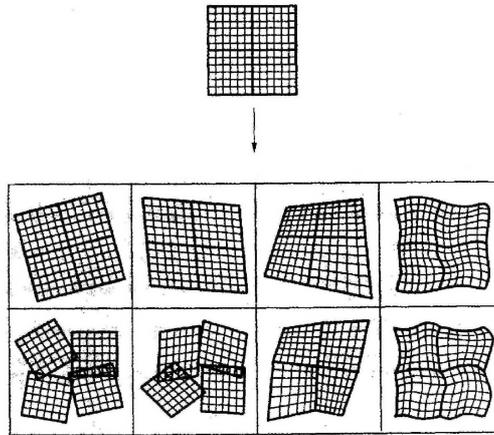


Figura 2.2: Combinação de tipos possíveis de transformações. De cima para baixo e da esquerda para direita tem-se as seguintes transformações: globais (rígida, afim, projetiva e elástica) e locais (rígida, afim, projetiva e elástica). Figura retirada de [8].

A **transformação afim** é mais geral que a rígida e é utilizada quando imagens da cena são vistas do mesmo ângulo, mas de diferentes posições. Nesse tipo de transformação uma linha reta na primeira imagem é mapeada para uma linha reta na segunda, mantendo o paralelismo.

A **transformação projetiva**, e de maneira mais geral a transformação perspectiva, são usadas para imagens cujas distorções ocorrem devido à projeção de objetos de distâncias variadas do sensor no plano da imagem. O paralelismo entre uma linha reta na primeira imagem mapeada para uma linha reta na segunda nem sempre é mantido.

A **transformação elástica** é a operação de transformação mais geral. Essa transformação pode mapear linhas retas em curvas. Uma classe de funções dessa transformação é a classe de transformações polinomiais. A transformação polinomial é uma das transformações globais mais geral e pode ser utilizada em vários tipos de distorção.

Neste trabalho, como o processo de manuseio das folhas de um livro pode produzir deformações diferentes em diferentes partes da folha, decidiu-se pela utilização da transformação elástica, que permite corrigir esse tipo de deformação, além da translação e rotação que ocorrem ao serem realizadas duas digitalizações com fundos diferentes.

2.3 Registro de Imagens com Distorções Geométricas

Devido ao fato de algumas imagens apresentarem distorções complexas como, por exemplo, projeção bidimensional de objetos tridimensionais, movimentos de objetos incluindo efeitos de oclusão e as deformações de objetos elásticos, existe a necessidade de encontrar métodos capazes de registrar imagens com tais distorções. Os métodos que solucionam esse problema exploram modelos elásticos.

A aplicação da teoria da elasticidade conta com a utilização de leis físicas que governam a deformação do objeto que está sob a ação de forças externas, bem como propriedades intrínsecas ao objeto. Formas estáticas mostradas por um número de corpos deformados, tais como borracha, mola, papel e roupa, podem ser modeladas de forma precisa através da simulação de propriedades físicas como pressão e rigidez.

Vários métodos de registro de imagem associam a idéia de modelos elásticos com a modelagem de imagens distorcidas. Em geral, a imagem é modelada como um corpo elástico e a medida de similaridade entre os pontos em ambas as imagens moldam o corpo em questão.

Um modelo elástico para o alinhamento de imagens é o *Thin Plate Spline* – *TPS* [9]. O modelo TPS é um conjunto de funções que modelam uma placa de metal infinita sob deformação e que podem ser consideradas uma combinação linear de um plano e uma superfície *spline* geral. Os resultados mais precisos em registro de imagens com distorções geométricas locais foram obtidos usando as funções de mapeamento de superfície *spline*, segundo Barrodale et al. [2]. Por essa razão, o método de registro de imagens utilizado nesse trabalho será o TPS. Na próxima seção será apresentada uma visão de funções *splines* e na seção seguinte, o uso de TPS em alinhamento de imagens.

2.3.1 Funções Splines

Uma função *spline* pode ser vista como uma régua elástica que pode ser deformada de maneira que passe através de um dado conjunto de pontos (x_i, y_i) , para $i = 1, \dots, m$. Seja $a = x_0 < x_1 < \dots < x_m = b$ uma subdivisão do intervalo $[a, b]$. Uma função *spline* de grau p que passa pelos pontos x_i , para $i = 1, \dots, m$ é uma função S com as seguintes propriedades:

- em cada subintervalo $[x_i, x_{i+1}]$, para $i = 0, 1, \dots, m-1$, $S(x)$ é um polinômio de grau p .
- $S(x)$ e sua primeira derivada são contínuas no intervalo $[a, b]$.

Funções *splines* podem ser utilizadas em muitos problemas em que uma aproximação polinomial sobre um intervalo inteiro é insuficiente. Maiores detalhes sobre funções *spline* podem ser obtidos em [5].

2.3.2 Thin Plate Splines

A utilização de *Thin Plate Splines* como funções de registro de imagens deformadas foi proposta por Goshtasby em [9]. Seu método é formulado como um problema de ajuste de superfície em que as superfícies obtidas representam as componentes de uma função de mapeamento e caracterizam as distorções geométricas locais entre as imagens.

Uma função *thin plate spline* é vista como uma folha fina de algum material não maleável com extensão infinita. Quando pontos de controle específicos são dispostos na placa, a mesma sofre uma deformação de tal modo que minimiza a energia total de deformação, E , aplicada pela transformação.

Funções TPS são formadas através da combinação linear de funções radiais, o que significa que os valores das funções são obtidos pela diferença, ou distância, de coordenadas do ponto em que a função deve ser avaliada para as coordenadas de pontos de referência. A medida que se afasta dos pontos de referência, a função apresenta um comportamento quase linear. Assim, embora sejam funções de interpolação globais, as influências de deformações locais tendem a diminuir a medida que se afastam do ponto de deformação.

A função base radial para geração da TPS é definida pela seguinte expressão:

$$g(x, y) = r^2 \log r^2, \quad (2.11)$$

onde $r = \sqrt{x^2 + y^2}$ é a distância cartesiana à origem. Maiores detalhes a respeito da formulação teórica da função TPS podem ser obtidos em [3].

O problema de encontrar as funções de mapeamento f_x e f_y utilizando TPS pode ser formulado como um problema de interpolação de superfícies de maneira que as superfícies obtidas representam as componentes f_x e f_y da função de mapeamento e caracterizam as distorções geométricas entre as imagens.

Desta forma, dados dois conjuntos de pontos tridimensionais

$$S = \{(x_i, y_i, u_i): \text{ para } i = 1, \dots, n\} \text{ e} \quad (2.12)$$

$$Q = \{(x_i, y_i, v_i): \text{ para } i = 1, \dots, n\}, \quad (2.13)$$

quer-se encontrar superfícies suaves $f_x(x, y)$ e $f_y(x, y)$. A superfície $f_x(x, y)$ deve passar por todos os pontos em S e a superfície $f_y(x, y)$ deve passar por todos os pontos em Q .

As superfícies desejadas podem ser encontradas através das expressões

$$f_x(x, y) = a_0 + a_1x + a_2y + \sum_{i=1}^n F_i r_i^2 \log r_i^2 \quad e \quad (2.14)$$

$$f_y(x, y) = b_0 + b_1x + b_2y + \sum_{i=1}^n G_i r_i^2 \log r_i^2, \quad (2.15)$$

onde $r_i^2 = (x - x_i)^2 + (y - y_i)^2$.

Na formulação da função *thin plate spline*, na Equação 2.14, a componente linear da transformação é dada pelos coeficientes a_0 , a_1 e a_2 . Essa formulação pode ser interpretada como sendo a sobreposição de uma superfície de deformação sobre o plano $a_0 + a_1x + a_2y$ que define uma transformação afim. A formulação da Equação 2.15 é interpretada da mesma maneira.

Os coeficientes a_0 , a_1 , a_2 e F_i , para $i = 1, \dots, n$, são determinados pela substituição dos n pontos de controle na Equação (2.14) e solucionando o sistema de $(n + 3)$ equações lineares a seguir

$$f(x_i, y_i) = u_i, \quad \text{para } i = 1, \dots, n, \quad (2.16)$$

$$\sum_{i=1}^n F_i = 0, \quad (2.17)$$

$$\sum_{i=1}^n F_i x_i = 0, \quad (2.18)$$

$$\sum_{i=1}^n F_i y_i = 0. \quad (2.19)$$

A superfície $f_x(x, y)$ obtida representa a primeira componente da função de mapeamento. A superfície $f_y(x, y)$, que representa a segunda componente, é determinada de maneira similar.

Depois que os coeficientes foram determinados e as duas funções de mapeamento f_x e f_y desejadas foram encontradas, então é possível registrar as imagens. Para efetuar a transformação aplicam-se as funções f_x e f_y a cada ponto da imagem destino.

2.3.3 Cálculo dos Coeficientes das Funções TPS

Os coeficientes a_0 , a_1 , a_2 e F_i , para $i = 1, \dots, n$, da Equação 2.14 e os coeficientes b_0 , b_1 , b_2 e G_i , para $i = 1, \dots, n$, da Equação 2.15 podem ser encontrados solucionando o sistema de equações lineares dado por $Ax = b$, onde:

$$A = \begin{bmatrix} 0 & r_{2,1}^2 \log r_{2,1}^2 & \dots & r_{n,1}^2 \log r_{n,1}^2 & 1 & x_1 & y_1 \\ r_{1,2}^2 \log r_{1,2}^2 & 0 & \dots & r_{n,2}^2 \log r_{n,2}^2 & 1 & x_2 & y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ r_{1,n}^2 \log r_{1,n}^2 & r_{2,n}^2 \log r_{2,n}^2 & \dots & 0 & 1 & x_n & y_n \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 \\ x_1 & x_2 & \dots & x_n & 0 & 0 & 0 \\ y_1 & y_2 & \dots & y_n & 0 & 0 & 0 \end{bmatrix}, \quad (2.20)$$

$$x = \begin{bmatrix} F_1 & G_1 \\ F_2 & G_2 \\ \vdots & \vdots \\ F_n & G_n \\ a_0 & b_0 \\ a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}, \quad (2.21)$$

e

$$b = \begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \\ \vdots & \vdots \\ u_n & v_n \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \quad (2.22)$$

A inversa da matriz A pode ser determinada uma vez que se sabe que A não é singular. A inversa pode ser obtida utilizando eliminação gaussiana. Assim, para solucionar o sistema de equações lineares acima basta resolver a seguinte expressão:

$$x = A^{-1}b. \quad (2.23)$$

Desta forma, a matriz x fornece os valores dos coeficientes desejados das funções TPS. Para maiores detalhes sobre a computação dos coeficientes e da energia de deformação consultar Costa e Cesar [7].

Dependendo dos conjuntos de pontos de controle selecionados, a energia de deformação aplicada sobre a superfície é modificada. Sejam as matrizes T e W dadas a seguir:

$$T = \begin{bmatrix} 0 & r_{2,1}^2 \log r_{2,1}^2 & \dots & r_{n,1}^2 \log r_{n,1}^2 \\ r_{1,2}^2 \log r_{1,2}^2 & 0 & \dots & r_{n,2}^2 \log r_{n,2}^2 \\ \vdots & \vdots & \ddots & \vdots \\ r_{1,n}^2 \log r_{1,n}^2 & r_{2,n}^2 \log r_{2,n}^2 & \dots & 0 \end{bmatrix}. \quad (2.24)$$

$$W = \begin{bmatrix} F_1 & F_2 & \dots & F_n \\ G_1 & G_2 & \dots & G_n \end{bmatrix}^T. \quad (2.25)$$

A energia de deformação E pode ser obtida de maneira que,

$$E = \text{trace}\{W^T T W\}, \quad (2.26)$$

onde trace é a operação de soma dos elementos da diagonal da matriz.

No caso em que a matriz A não é singular, a sua inversa não pode ser determinada e outra solução para o sistema de equações lineares $Ax = b$ deve ser encontrada. Uma solução possível é a utilização de *singular value decomposition* – SVD.

Pelo método de SVD, dado um sistema de equações lineares $Ax = b$, a matriz A quadrada pode ser decomposta em $A = UWV^T$ onde U , W e V são matrizes quadradas do tamanho de A . A solução do sistema é então dada por $x = VW^{-1}(U^T b)$. Para uma descrição completa do método mencionado, ver [19]. A solução do sistema usando decomposição por valor único foi empregada por Hill para registro de imagens médicas com TPS em [12].

Neste trabalho, o cálculo dos coeficientes das funções TPS solucionando o sistema $Ax = b$ será efetuado utilizando SVD, uma vez que nos testes realizados com imagens de documentos foram encontradas matrizes singulares.

2.3.4 Complexidade Computacional do TPS

O tempo necessário para a transformação de uma imagem deformada utilizando TPS é igual à soma entre o tempo de computação dos coeficientes e o tempo da transformação geométrica na imagem destino.

Para a análise do tempo de computação as operações de soma, subtração, multiplicação e logaritmo serão consideradas operações básicas, todas contribuindo com o mesmo custo no cálculo do tempo. Consideram-se as imagens com o total de m pontos e n pontos de controle.

Se a matriz A não é singular, o cálculo dos coeficientes das funções TPS é efetuado solucionando $x = A^{-1}b$. Desta maneira, o tempo gasto para o cálculo

dos coeficientes das funções é a soma do tempo gasto para calcular a inversa da matriz A e o tempo para multiplicar $A^{-1}b$.

A determinação da inversa de uma matriz A utilizando eliminação gaussiana requer $4n^3/3 - n/3$ multiplicações e divisões e $4n^3/3 - 3n^2/2 + n/6$ adições e subtrações para solucionar o sistema linear de n equações, conforme Burden e Faires em [5]. A multiplicação de $A^{-1}b$ requer $2n^2$ multiplicações e $2n^2 - 2n$ somas.

Desta forma, o tempo gasto para computação dos coeficientes é:

$$\frac{4n^3}{3} - \frac{n}{3} + \frac{4n^3}{3} - \frac{3n^2}{2} + \frac{n}{6} + 2n^2 + 2n^2 - 2n = \frac{16n^3 - 33n^2 - 13n}{6} = \theta(n^3). \quad (2.27)$$

Isso significa que a complexidade de tempo para o cálculo dos coeficientes é $\theta(n^3)$.

No caso em que a matriz A é singular, o cálculo dos coeficientes das funções TPS é realizado utilizando SVD. O tempo gasto para efetuar essa computação não foi mencionado nas referências consultadas e o seu cálculo foge do escopo deste trabalho.

O tempo gasto para aplicação da transformação geométrica em toda a imagem para uma componente é igual a aplicação da Equação (2.14) m vezes, ou seja,

$$m(9n + 4) = 9mn + 4m. \quad (2.28)$$

Como é necessário aplicar a transformação para as duas componentes, tem-se

$$18mn + 8m = \theta(mn). \quad (2.29)$$

Isso significa que a complexidade de tempo para a transformação da imagem é $\theta(mn)$.

Capítulo 3

Seleção de Pontos de Controle

O primeiro passo para a solução do problema de registro de imagens é a seleção dos pontos de controle. Para o problema do registro de imagens de documentos foi estudada uma forma de seleção de pontos de controle baseada em técnicas de processamento de documentos. Neste capítulo serão abordadas duas técnicas utilizadas para seleção dos pontos de controle, testadas neste trabalho, juntamente com o relato dos resultados obtidos com a aplicação de cada uma delas.

As imagens utilizadas para testes neste trabalho foram obtidas do livro *Sermão no Auto da Fé em Coimbra* de Mendonça [16] e foram digitalizadas utilizando um aparelho de *scanner* de mesa, com resolução 300×300 pixels por polegada, intensidade de 8 bits por pixel, com 256 níveis de cinza e dimensões 507×736 pixels para imagem de uma página.

Na Seção 3.1 serão apresentados alguns trabalhos que utilizam registro de imagens de documentos. Na Seção 3.2 uma primeira idéia a respeito da seleção de pontos de controle será mostrada. Uma outra idéia para seleção dos pontos será apresentada na Seção 3.3 em que são extraídos pontos dos extremos das linhas de texto. Para que essa seleção seja possível é necessário segmentar o texto em linhas e corrigir a inclinação da imagem da página, assuntos que serão tratados nas Seções 3.3.1 e 3.3.2, respectivamente. A seleção dos pontos de controle propriamente dita será feita na Seção 3.3.3. Por fim, um teste dessa técnica será realizado na Seção 3.3.4. Outra maneira de selecionar pontos de controle será abordada na Seção 3.4 em que são extraídos pontos do início de cada palavra do texto. Essa técnica requer a segmentação de linhas de texto em palavras, que será mostrada na Seção 3.4.1. A correspondência dos pontos de controle do início das palavras se faz necessária e será mostrada na Seção 3.4.2. Em seguida, um teste dessa técnica será apresentado na Seção 3.4.3. Por fim, na Seção 3.5 será apresentada uma análise dos resultados.

3.1 Aplicações de Registro de Imagens de Documentos

Não existem muitas aplicações de registro de imagens de documentos presentes na literatura. Entre as aplicações encontradas estão o mosaico de documentos e o reconhecimento de caracteres.

No trabalho de Zappalá et al. [26] utiliza-se registro de imagens na aplicação do mosaico de documentos. Dadas várias imagens de um documento, cada uma representando porções diferentes do mesmo, deseja-se unir as imagens de forma a obter a imagem do documento inteiro. No entanto, anterior a esse procedimento, as imagens precisam ser alinhadas. O ângulo de inclinação da imagem é estimado e esta é girada até que não esteja mais inclinada. Em seguida, as imagens são transformadas em imagens binárias e então segmentadas em uma hierarquia de colunas, linhas e palavras. A correspondência entre as imagens é feita percorrendo linha a linha duas imagens e comparando o tamanho das palavras. Zappalá et al. consideraram que as imagens não possuem distorções.

No reconhecimento de caracteres em [14] por Leung et al., uma imagem é o padrão e a outra contém o caracter a ser reconhecido. As imagens são transformadas em imagens binárias e então um método de afinamento é aplicado para que os caracteres sejam representados por linhas finas. Os pontos de controle são extraídos como pontos de interseção e fim de linhas. Então aplica-se uma técnica de registro para verificar se há correspondência.

Uma vez que é necessário realizar o registro de imagens de documentos foram estudadas técnicas de processamento de imagens de documentos para extrair pontos de controle. Um apanhado geral de tais técnicas foi elaborado por Tang et al. [24] e por O’Gorman e Kasturi [18].

3.2 Primeira Idéia

Uma maneira de selecionar pontos de controle para imagens de documentos é através do afinamento de cada caracter e da utilização dos pontos de interseção e fim de linhas como pontos de controle. No entanto, essa prática resulta em um número muito grande de pontos. Supondo, por exemplo, que cada página possua 30 linhas e cada linha contenha 40 caracteres, e ainda que cada caracter contribua com 6 pontos, tem-se o total de $30 \times 40 \times 6 = 7.200$ pontos de controle por página. Embora essa maneira pareça ideal pois descreve bem as distorções caracter por caracter, foi descartada pois é computacionalmente impraticável, como visto na Seção 2.3.4, já que o cálculo dos coeficientes gastaria $(7.200)^3$ operações.

3.3 Pontos dos Extremos das Linhas

Uma outra forma de selecionar pontos de controle é utilizar para cada linha de texto os pontos extremos nas suas margens esquerda e direita. Isso significa que cada linha de texto contribui com dois pontos de controle. Supondo o mesmo exemplo anterior, se a página possui 30 linhas, o número de pontos de controle será 60, que representa um número razoável em termos do número de operações computacionais realizadas, conforme descrito na Seção 2.3.4. A seleção de pontos de controle adotada não necessita da etapa de correspondência dos pontos, uma vez que os pontos selecionados nas duas imagens já são correspondentes.

A seleção de pontos de controle acima foi realizada supondo que as páginas do documento possuem somente linhas de texto, sem figuras. Se a página contiver figuras, a seleção de pontos de controle empregada não funciona. Para o caso de páginas com figuras, deve-se antes segmentar a imagem de forma que as partes textual e não textual sejam separadas. Então pode-se aplicar a técnica descrita a seguir para a parte que contém somente texto.

Como as imagens utilizadas neste trabalho para testes possuem 256 níveis de cinza, deve-se calcular o histograma de cada imagem e encontrar um limiar para cada uma delas. Os pixels da imagem que tiverem um valor menor ou igual ao limiar correspondente serão chamados de pixels em preto e os demais de pixels do fundo da página.

3.3.1 Segmentação da Página de Texto em Linhas

Para extração dos pontos de controle como pontos dos extremos das linhas é necessária a identificação das linhas como um passo inicial. Essa tarefa será executada utilizando uma técnica semelhante àquela empregada por Muge et al. [17].

A segmentação da imagem em linhas de texto começa com a computação de um vetor de número de pixels em preto por linha. Baseado nesse vetor, uma operação é efetuada envolvendo a computação de um limiar para extrair as linhas de texto. O vetor de número de pixels em preto é uma operação simples que percorre as linhas da imagem e para cada linha armazena o número de pixels em preto que aparecem na linha. O Algoritmo 1 descreve tal operação. Assim, no final dessa operação, um vetor dos valores obtidos constitui um histograma. Na Figura 3.1 é apresentada uma imagem com linhas de texto e na Figura 3.2 a representação do seu histograma.

Algoritmo 1 Vetor de número de pixels em preto por linha

Entrada: Imagem de entrada I , número de linhas lin e número de colunas col de I , limiar extraído do histograma de níveis de cinza.

Saída: $histo(i)$, vetor de número de pixels em preto por linha para $i = 1, \dots, lin$.

```

para  $i = 1$  até  $lin$  faça
   $cont \leftarrow 0$ 
  para  $j = 1$  até  $col$  faça
    se  $I(i, j) \leq \text{limiar}$  então
       $cont \leftarrow cont + 1$ 
    fim-se
  fim-para
   $histo(i) \leftarrow cont$ 
fim-para
devolva  $histo$ 

```

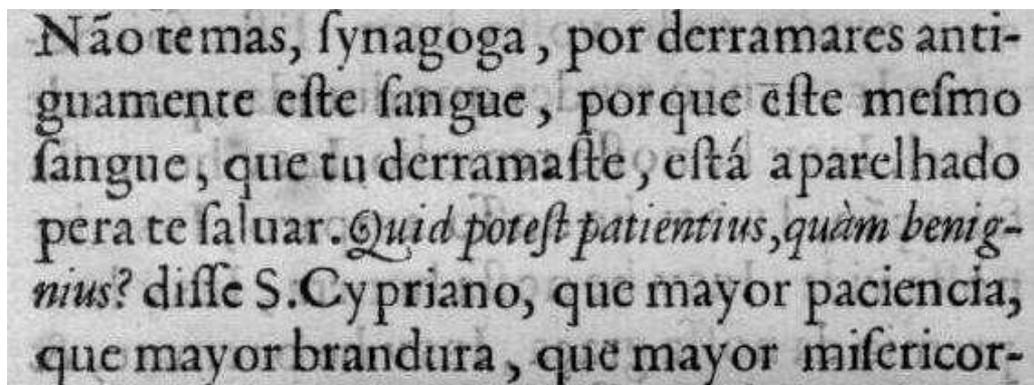


Figura 3.1: Imagem de uma porção de uma página com linhas de texto.

Ao observar a Figura 3.2 é possível intuitivamente identificar as linhas de texto. Percorrendo o histograma verticalmente encontram-se alternadamente picos e vales. Os picos representam as linhas de texto e os vales os espaços entre linhas.

Quando se quer identificar linhas de texto, precisa-se antes definir qual informação caracteriza uma linha. Segundo Muge et al. [17], são necessárias quatro características representadas por quatro linhas sendo elas a linha ascendente, início da linha média, fim da linha média e a linha descendente.

Essas linhas são necessárias devido à natureza do texto sendo analisado. As letras das linhas do texto podem ser classificadas em ascendente, descendente ou média, de acordo com o espaço que utilizam da linha. As letras ascendentes são letras como o b ou o d que têm uma haste ascendente e ocupam a parte

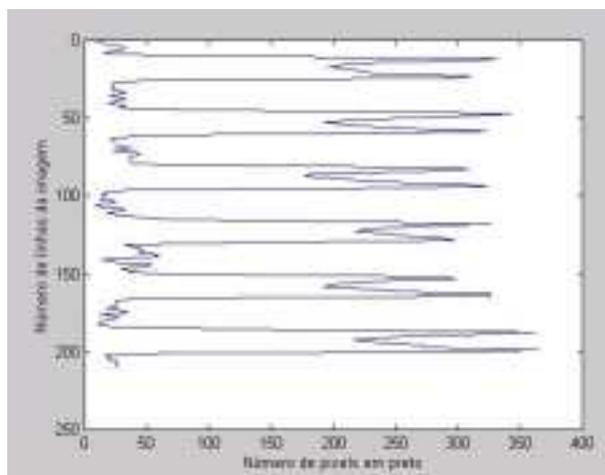


Figura 3.2: Histograma da imagem de uma porção de uma página com linhas de texto.

superior da linha, letras médias são como o *a* ou o *m* que ocupam a parte central da linha e letras descendentes são letras como o *g* ou o *p* que possuem haste descendente e ocupam a parte inferior da linha.

A Figura 3.3 ilustra as linhas necessárias para caracterizar uma linha de texto. As linhas início da média e fim da média serão mencionadas no texto, a partir deste ponto, como linhas de início e fim, apenas para tornar suas menções mais breves.

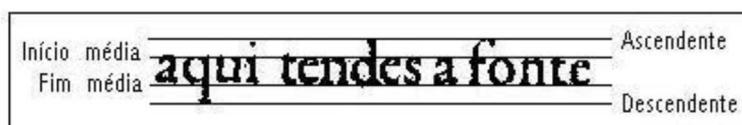


Figura 3.3: Linhas características da linha de texto. De cima para baixo são apresentadas as linhas ascendente, início, fim e descendente, respectivamente.

Geralmente, essas linhas características seguem aproximadamente o seguinte critério:

$$(\text{ascendente} - \text{início}) = (\text{início} - \text{fim}) = (\text{fim} - \text{descendente}), \quad (3.1)$$

onde cada elemento da expressão é dado em alguma escala ou medida.

Para computar a localização de uma linha de texto, uma medida estatística chamada valor médio m de pixels por linha é introduzida. Sua computação é realizada utilizando a seguinte expressão:

$$m = \sum_{i=1}^{\text{lin}} \text{histo}_i / \text{lin}, \quad (3.2)$$

onde lin é o tamanho do vetor de número de pixels em preto por linha e histo_i , com $i = 1, \dots, \text{lin}$, é cada elemento desse vetor.

Sendo conhecido o limiar m , percorre-se o vetor de número de pixels por linha para identificação da linha início e linha fim. A segmentação da imagem em linhas de texto pode ser vista no Algoritmo 2.

Algoritmo 2 Segmentação da imagem em linhas de texto

Entrada: $\text{histo}(i)$ vetor de número de pixels em preto por linha de tamanho lin , m média do número de pixels em preto por linha.

Saída: linhas , matriz $n \times 2$, onde n é o número de linhas de texto. A primeira coluna da matriz representa as coordenadas das linhas início e a segunda coluna as coordenadas das linhas fim.

```

n ← 0
flag ← verdadeiro
para i = 1 até lin faça
    se  $\text{histo}(i) > m$  e flag =verdadeiro então
        n ← n + 1
        inicio(n) ← i
        flag ← falso
    senão
        se flag =falso então
            fim(n) ← i
            flag ← verdadeiro
    fim-se
fim-se
fim-para
devolva linhas

```

A técnica descrita foi utilizada na imagem da porção da página de texto da Figura 3.1 e as linhas de texto detectadas pelo algoritmo podem ser observadas na Figura 3.4.

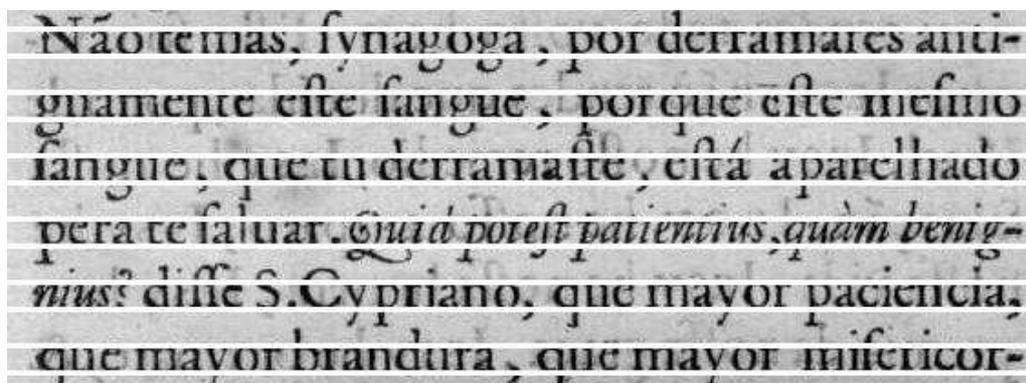


Figura 3.4: Linhas de texto detectadas na imagem de uma porção da página: as linhas brancas delimitam a parte média da linha de texto, ou seja, as linhas início e fim.

Esta técnica deixa algumas linhas sem identificação, como é o caso de linhas de texto de comprimento menor que a média. A Figura 3.5 ilustra o caso em que a última linha de texto da imagem não foi detectada. Uma maneira de solucionar esse problema é reaplicar o mesmo algoritmo para as porções da imagem que não contribuíram com nenhuma linha.

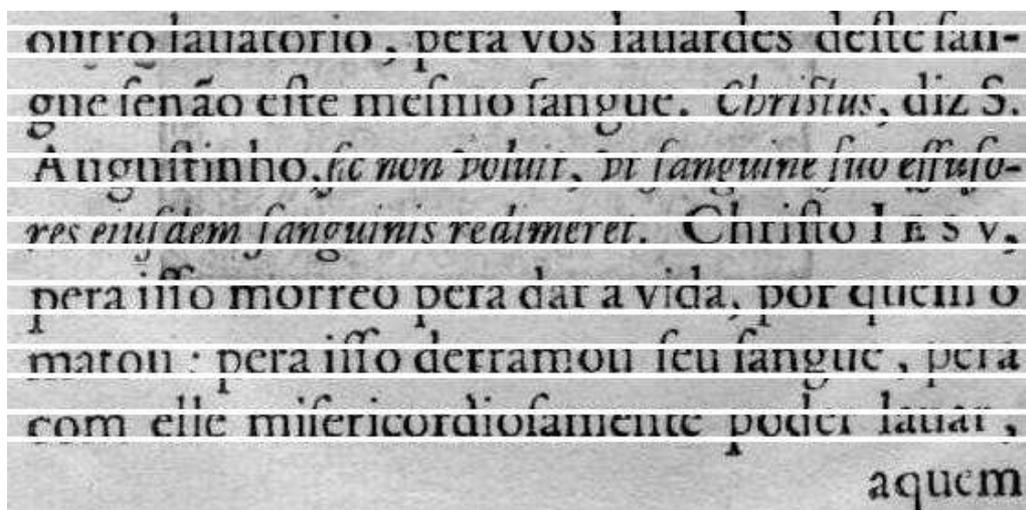


Figura 3.5: Erro na detecção das linhas de texto. A última linha de texto da imagem não foi detectada.

Na presença de inclinação na imagem da página do documento, as linhas de texto não mantêm sua aparência distintiva no vetor de número de pixels em preto por linha. O histograma da imagem da página inclinada da Figura 3.6 pode ser visto na Figura 3.7.

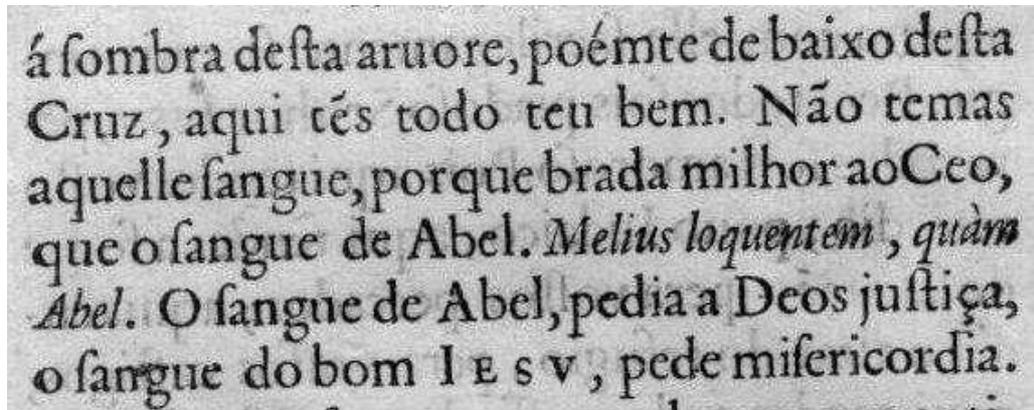


Figura 3.6: Imagem de uma porção de uma página de texto inclinada.

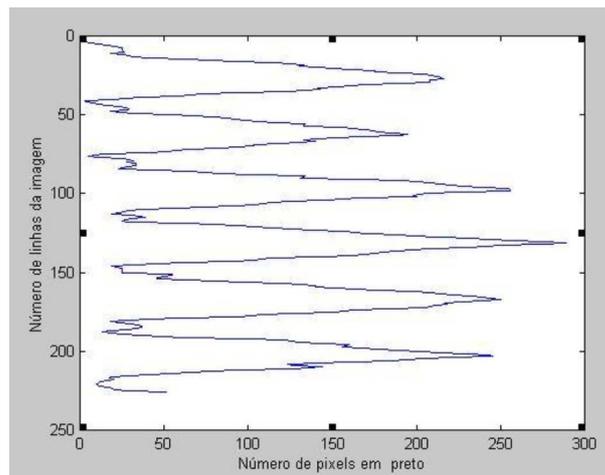


Figura 3.7: Histograma da imagem de uma porção de uma página de texto inclinada.

Observando o histograma é fácil ver que a transição entre picos e vales não identifica a linha fim e a linha início de cada linha de texto. Aplicando a técnica de segmentação de linhas na imagem da página inclinada não se obtêm as informações corretas das linhas, como ilustra a Figura 3.8.

Como a segmentação da página em linhas não funciona para imagens de páginas inclinadas, verificou-se a necessidade de corrigir as possíveis diferenças de inclinação anteriormente ao processo de segmentação em linhas. Na Seção 3.3.2 será apresentada uma técnica para correção da inclinação de páginas de documentos.

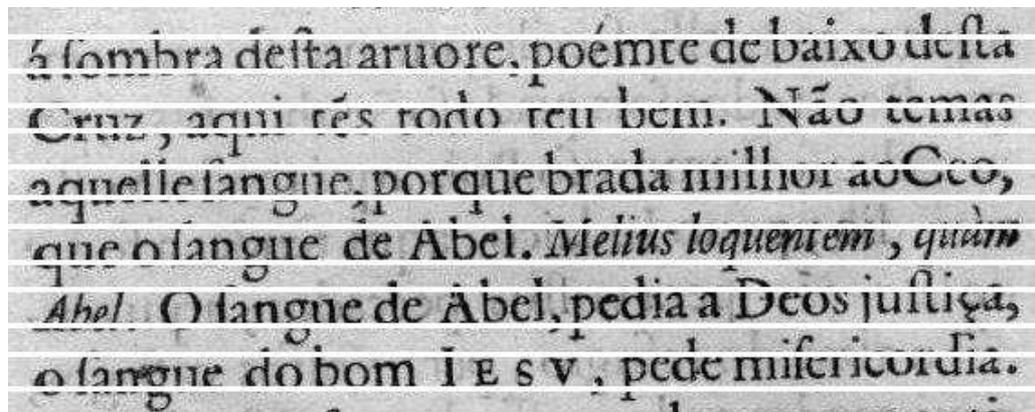


Figura 3.8: Linhas de texto detectadas na imagem de uma porção da página inclinada: as linhas brancas delimitam a parte média da linha de texto, ou seja, as linhas início e fim.

3.3.2 Correção da Inclinação de Imagens de Páginas de Documentos

Para solucionar o problema da inclinação de páginas será apresentado um método iterativo. Para corrigir a inclinação da imagem, primeiro é necessário analisá-la e verificar se a mesma encontra-se ou não inclinada. Se a imagem estiver inclinada deseja-se saber qual o sentido da inclinação, se horário ou anti-horário.

A tarefa de verificar se a imagem da página está ou não inclinada foi realizada recorrendo-se às informações da página obtidas através da segmentação em linhas. Ao observar a região entre a linha ascendente e a linha do início da média de duas páginas, sendo uma com inclinação e a outra sem, como ilustram as Figuras 3.9 e 3.10, respectivamente, é possível notar que na imagem inclinada a quantidade de pixels em preto na região é maior que na imagem sem inclinação.

A razão disso ocorrer é que na imagem da página sem inclinação a região analisada possui apenas pixels em preto correspondentes às hastes ascendentes das letras, enquanto que na imagem da página inclinada encontra-se na mesma região parte da letra média. Com a escolha apropriada de um limiar pode-se verificar se há ou não inclinação da página, apenas com a contagem do número de pixels.

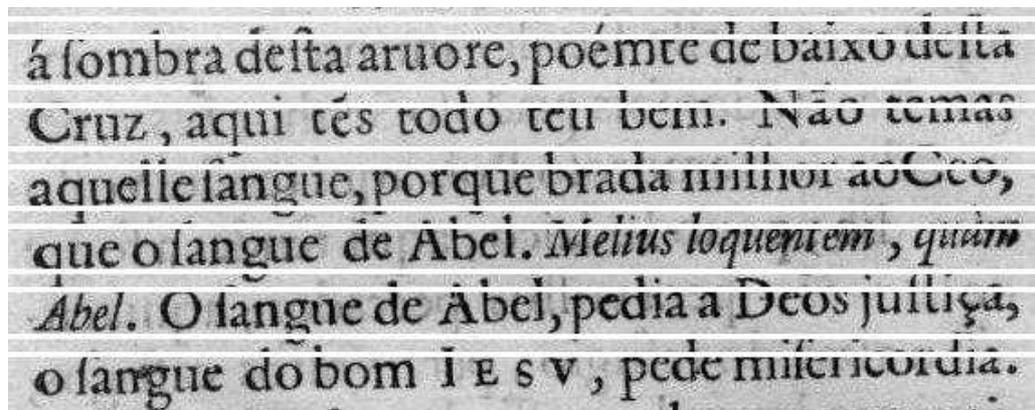


Figura 3.9: Imagem de uma porção da página inclinada. A região de texto entre linhas brancas corresponde a região entre as linhas ascendentes e início da média.

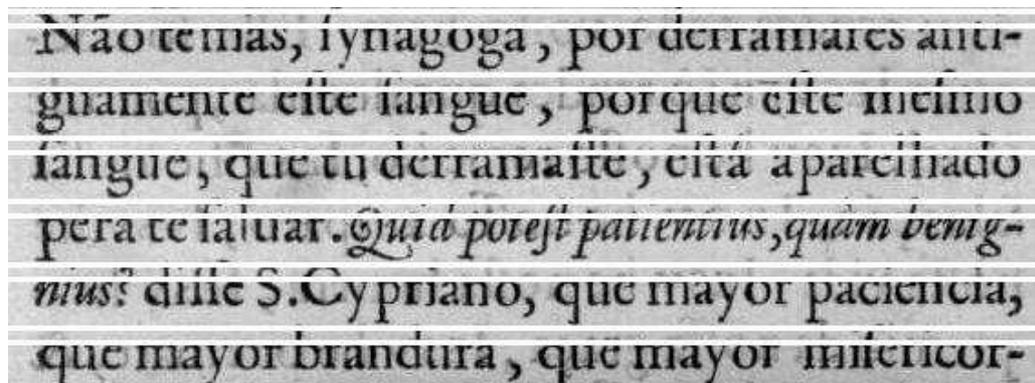


Figura 3.10: Imagem de uma porção da página sem inclinação. A região de texto entre linhas brancas corresponde a região entre as linhas ascendentes e início da média.

A descoberta do sentido da inclinação é realizada baseada na análise do número de pixels em preto na região que compreende a linha ascendente e a linha do início da média. Se a imagem está inclinada no sentido horário, um número maior de pixels em preto se concentrará no lado direito da região. Caso contrário, se a imagem estiver inclinada no sentido anti-horário, a concentração de pixels em preto se dará no lado esquerdo da mesma região. O Algoritmo 3 a seguir descreve essa operação.

Algoritmo 3 Verificação do sentido da inclinação da imagem

Entrada: I – Imagem que se quer verificar possível inclinação. ini e fim linhas ascendente e início da média da primeira linha de texto.

Saída: 0 – inclinação no sentido horário, 1 – inclinação no sentido anti-horário, 2 – sem inclinação.

```

contd  $\leftarrow$  0
conte  $\leftarrow$  0
para  $i = ini$  até  $fim$  faça
  para  $j = 1$  até  $col/2$  faça
    se  $I(i, j) < limiar$  então
      conte  $\leftarrow$  conte + 1
    fim-se
  fim-para
  para  $j = col/2$  até 1 faça
    se  $I(i, j) < limiar$  então
      contd  $\leftarrow$  contd + 1
    fim-se
  fim-para
   $s \leftarrow 2$ 
  se conte > tam então
     $s \leftarrow 1$ 
  fim-se
  se contd > tam então
     $s \leftarrow 0$ 
  fim-se
devolva  $s$ 

```

Uma vez que se constatou que a imagem está inclinada, e qual o sentido da inclinação pode-se iterativamente girar a imagem de um ângulo θ , previamente escolhido, no sentido correto até que não haja mais inclinação na imagem da página. As Figuras 3.11 e 3.12 mostram a imagem de uma página com a correção da inclinação e imagem inclinada, respectivamente.

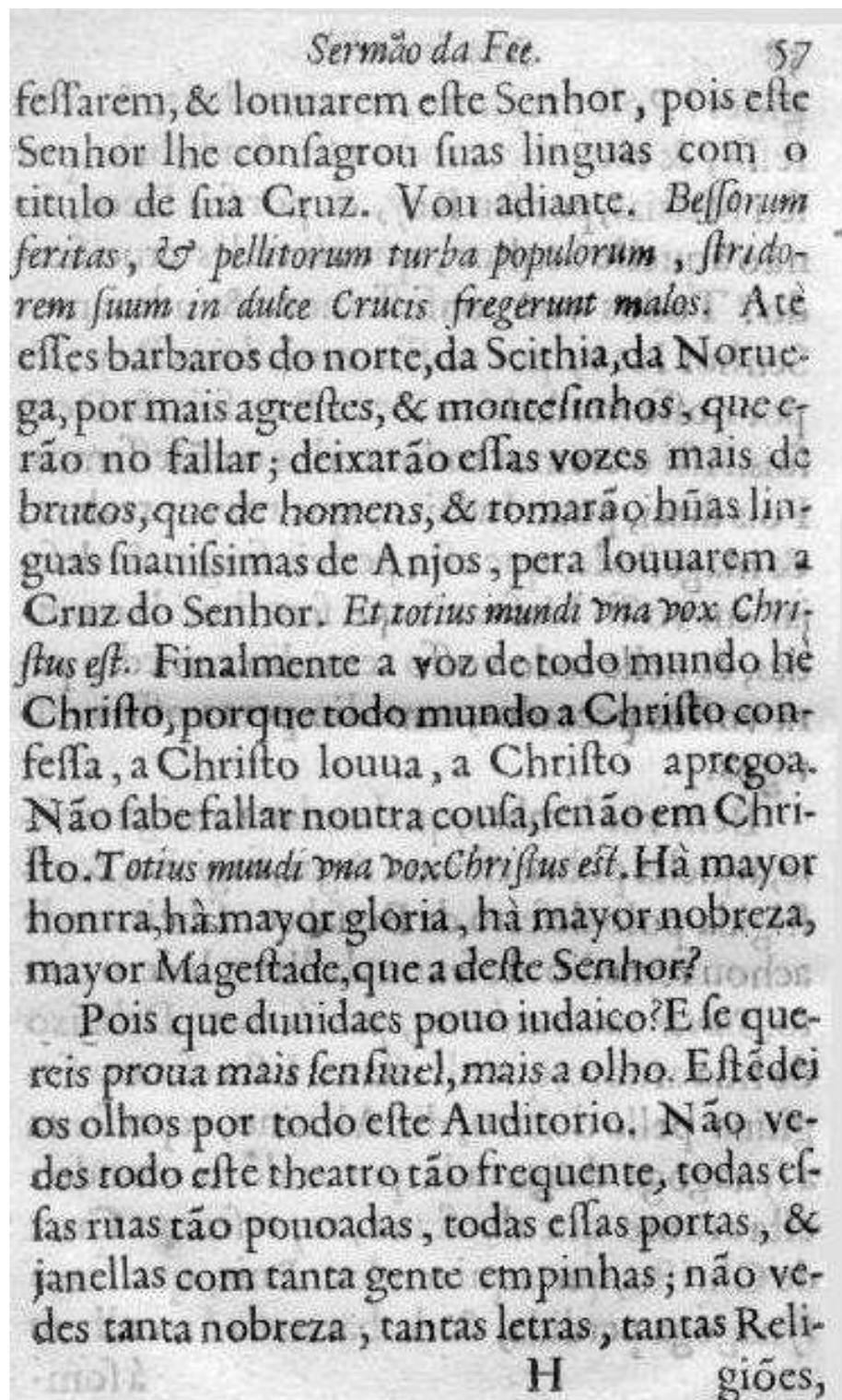


Figura 3.11: Imagem da página com inclinação corrigida.

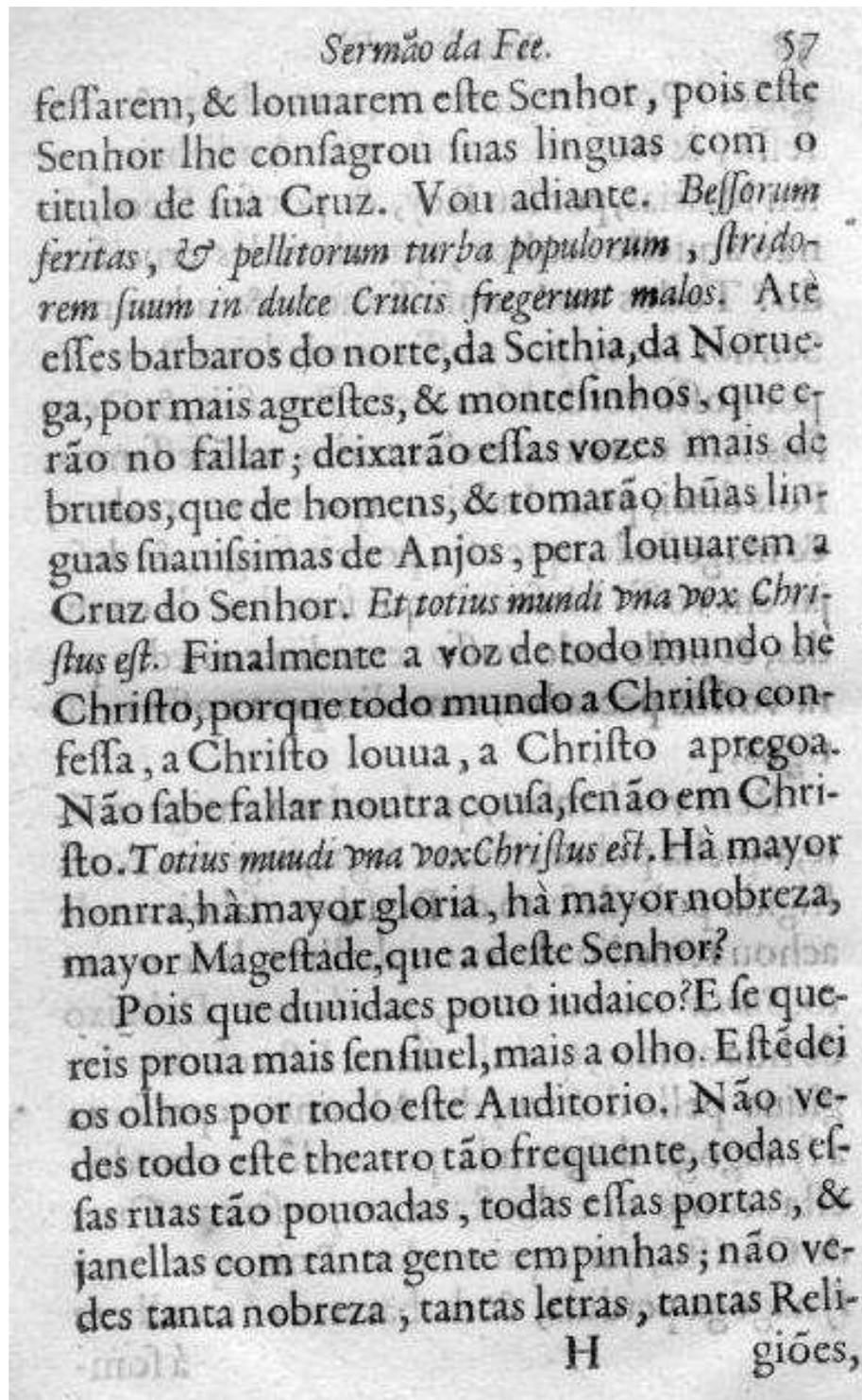


Figura 3.12: Imagem de referência digitalizada com uma folha de fundo branca.

3.3.3 Seleção dos Pontos de Controle dos Extremos da Linhas do Texto

Uma vez que a inclinação da imagem da página foi corrigida, a imagem está pronta para ser segmentada em linhas, aplicando a técnica descrita anteriormente. Depois da imagem já estar devidamente segmentada em linhas, falta selecionar os dois pontos dos extremos das linhas.

Para a seleção dos pontos dos extremos das linhas, percorre-se a imagem de cima para baixo, ou seja, no intervalo entre a linha início e fim, em busca do primeiro pixel em preto. Essa busca é efetuada duas vezes, uma vez partindo da primeira coluna da imagem e seguindo em direção à última coluna, o que corresponde ao ponto de controle do extremo esquerdo da página. Na outra vez, o percurso é executado da última coluna para a primeira, correspondendo ao ponto de controle do extremo direito da página.

A coordenada x dos pontos de controle é dada pela coordenada x no intervalo entre as linhas início e fim que contém o primeiro pixel em preto encontrado. A coordenada y do ponto de controle é dada pela coordenada y da imagem que contém o primeiro pixel em preto encontrado.

Dessa forma, as coordenadas dos pixels selecionados são armazenadas no conjunto de pontos de controle $C = \{(x_i, y_i) \mid i = 1, \dots, n\}$, onde n é duas vezes o número de linhas de texto. Portanto, cada linha de texto contribui com dois pontos de controle. A seleção de pontos de controle dos extremos da linha pode ser vista no Algoritmo 4.

3.3.4 Teste do Registro com Pontos de Controle dos Extremos das Linhas

Um teste da técnica descrita na seção anterior foi realizado utilizando como imagens de entrada duas imagens da mesma página de um livro antigo. A Figura 3.12 ilustra a imagem de referência obtida com uma folha de fundo branca e a Figura 3.13 mostra a imagem destino obtida com uma folha de fundo preta. A soma das imagens de entrada pode ser vista na Figura 3.14, em que se percebe que uma imagem possui uma inclinação diferente em relação a outra. A Figura 3.11 é utilizada como imagem de referência, uma vez que tal imagem é a Figura 3.12 sem inclinação. Nas Figuras 3.15 e 3.16 pode-se observar os pontos de controle selecionados, marcados com pontos brancos destacados. Finalmente, a Figura 3.17 mostra a imagem destino registrada utilizando funções *thin plate spline* e a Figura 3.18 ilustra a soma da imagem registrada com a imagem de referência.

Algoritmo 4 Seleção de pontos de controle dos extremos das linhas

Entrada: Imagem da qual se quer extrair os pontos de controle. Vetoe linhas, que é obtido do Algoritmo 2.

Saída: pc – conjunto de pontos de controle da imagem I.

```

l ← 0
para k = 1 até lin faça
  flag ← verdadeiro
  para j = 1 até col faça
    para i = linhas(k, 1) até linhas(k, 2) faça
      se  $I(i, j) \leq \text{limiar}$  e flag = verdadeiro então
        l ← l + 1
        pc(l, 1) ← i
        pc(l, 2) ← j
        flag ← falso
      fim-se
    fim-para
  fim-para
fim-para
para k = 1 até lin faça
  flag ← verdadeiro
  para j = col até 1 faça
    para i = linhas(k, 1) até linhas(k, 2) faça
      se  $I(i, j) \leq \text{limiar}$  e flag = verdadeiro então
        l ← l + 1
        pc(l, 1) ← i
        pc(l, 2) ← j
        flag ← falso
      fim-se
    fim-para
  fim-para
fim-para
devolva pc

```

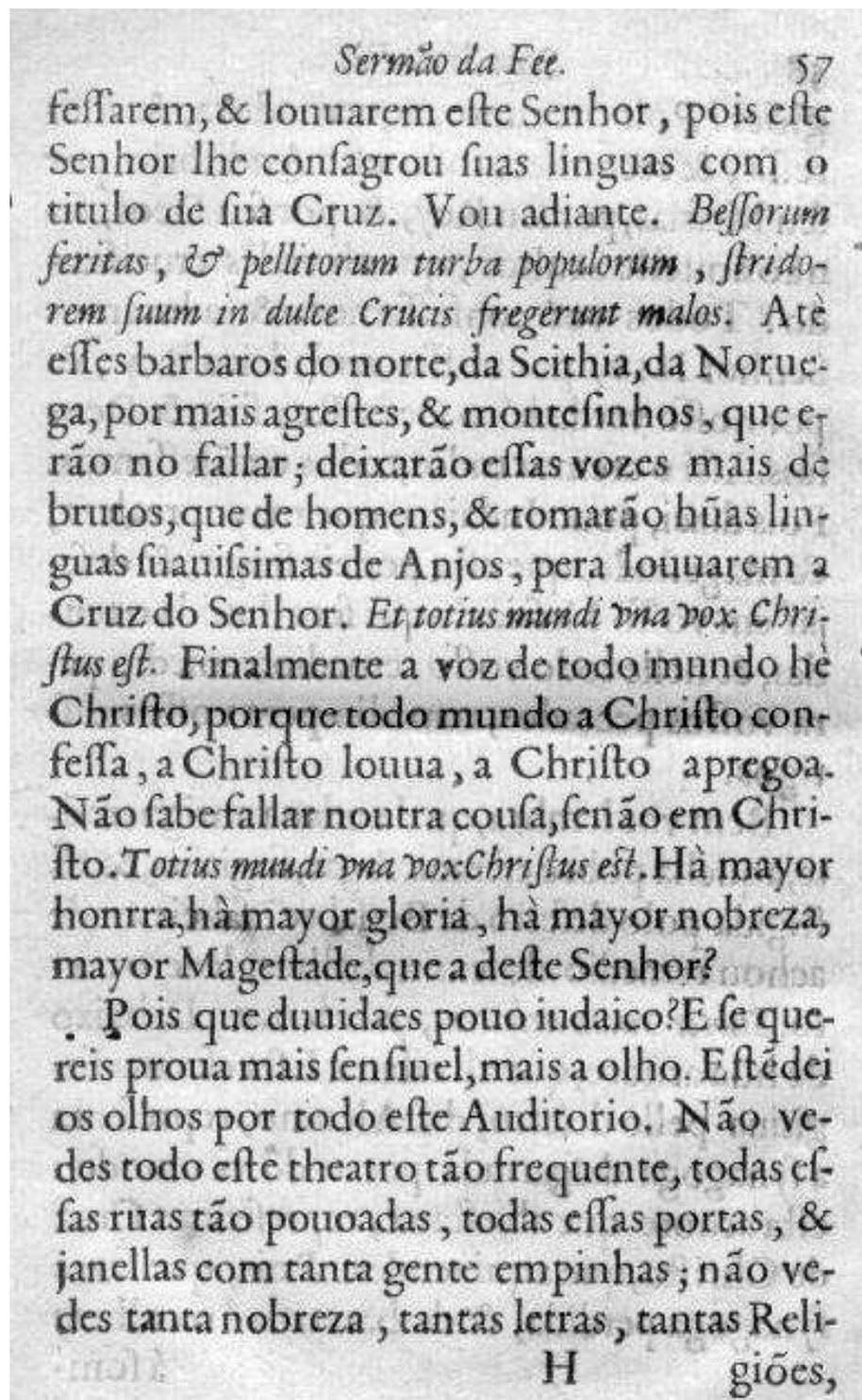


Figura 3.13: Imagem destino digitalizada com uma folha de fundo preta.

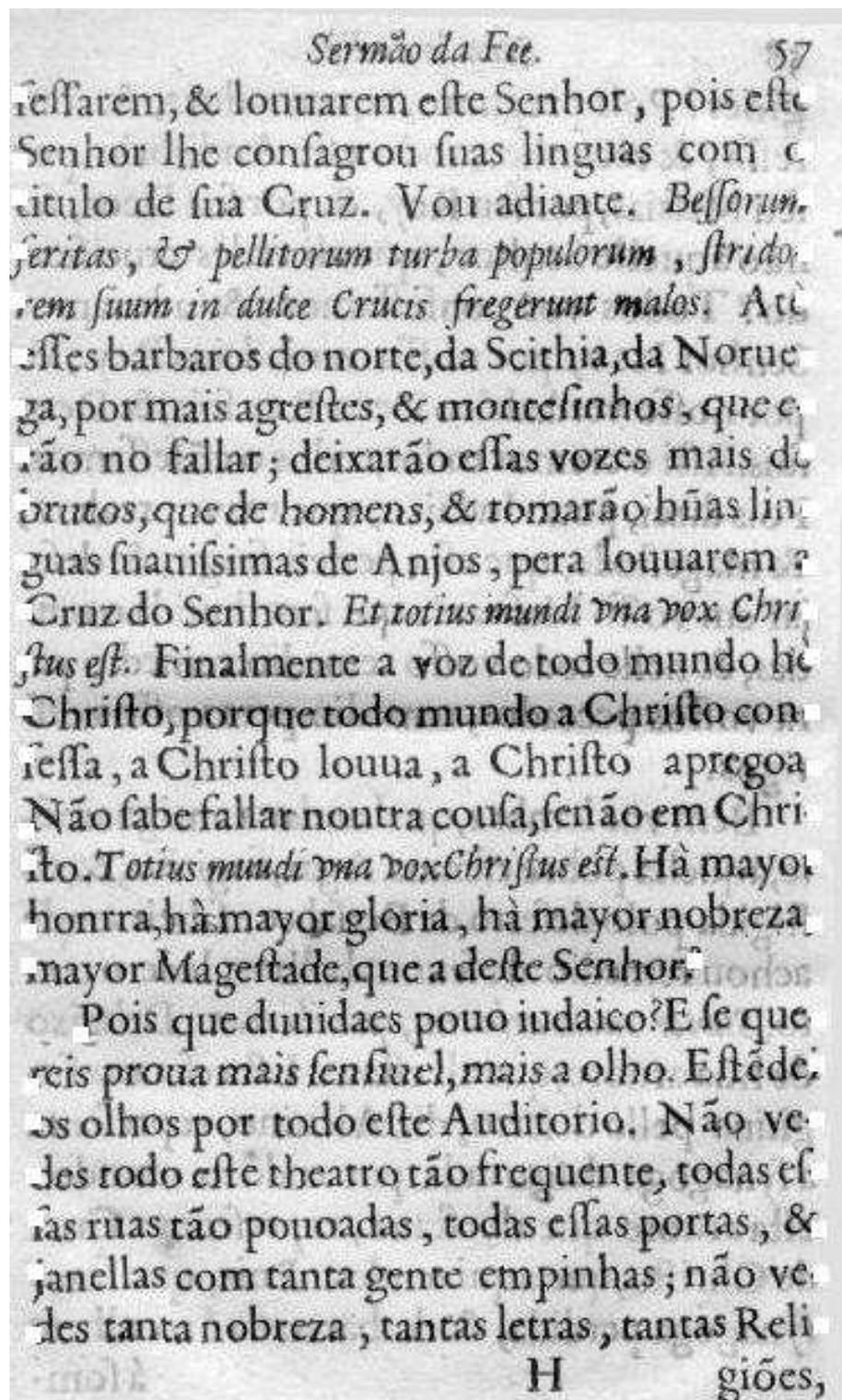


Figura 3.15: Pontos de controle da imagem de referência.

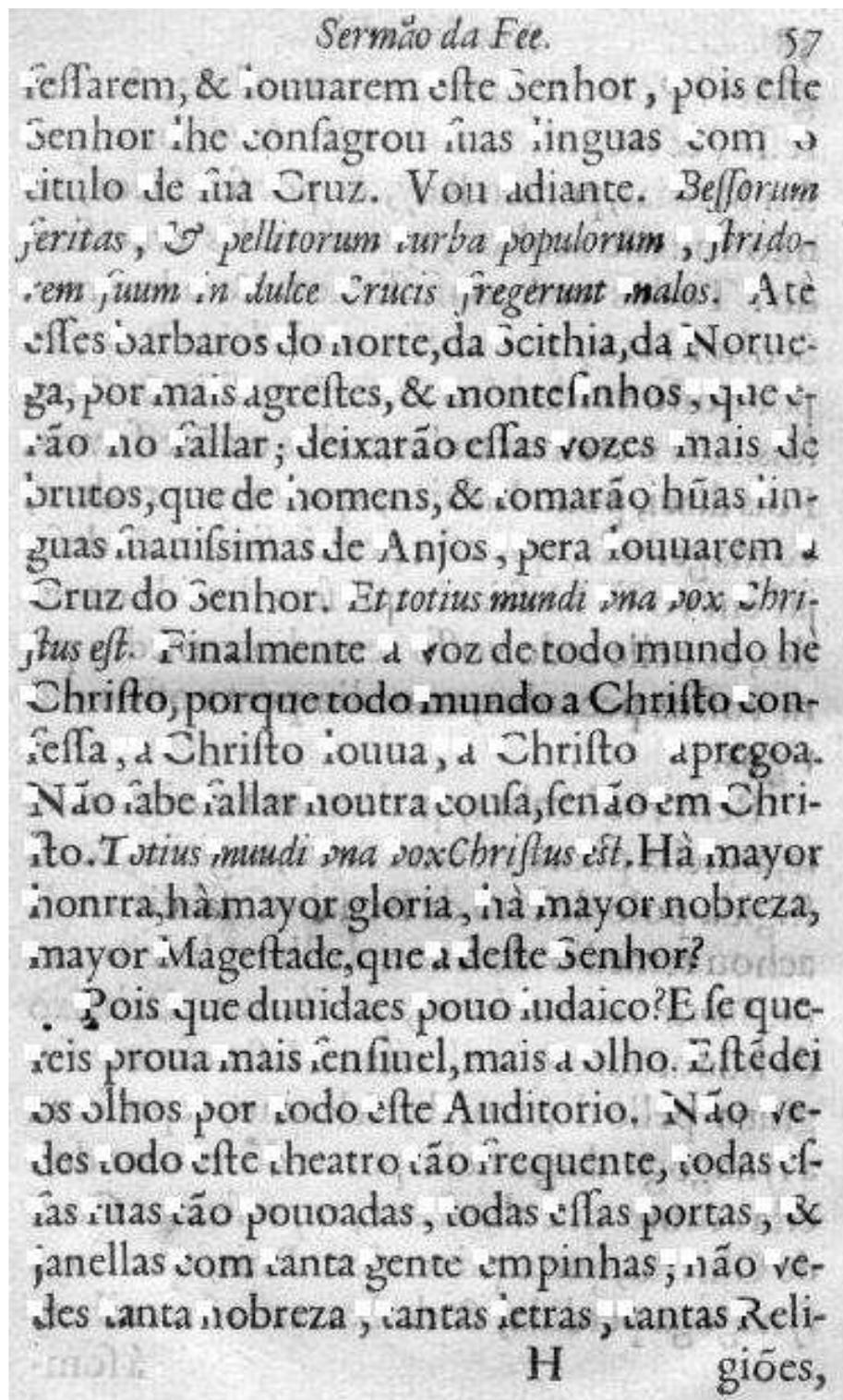


Figura 3.16: Pontos de controle da imagem destino.

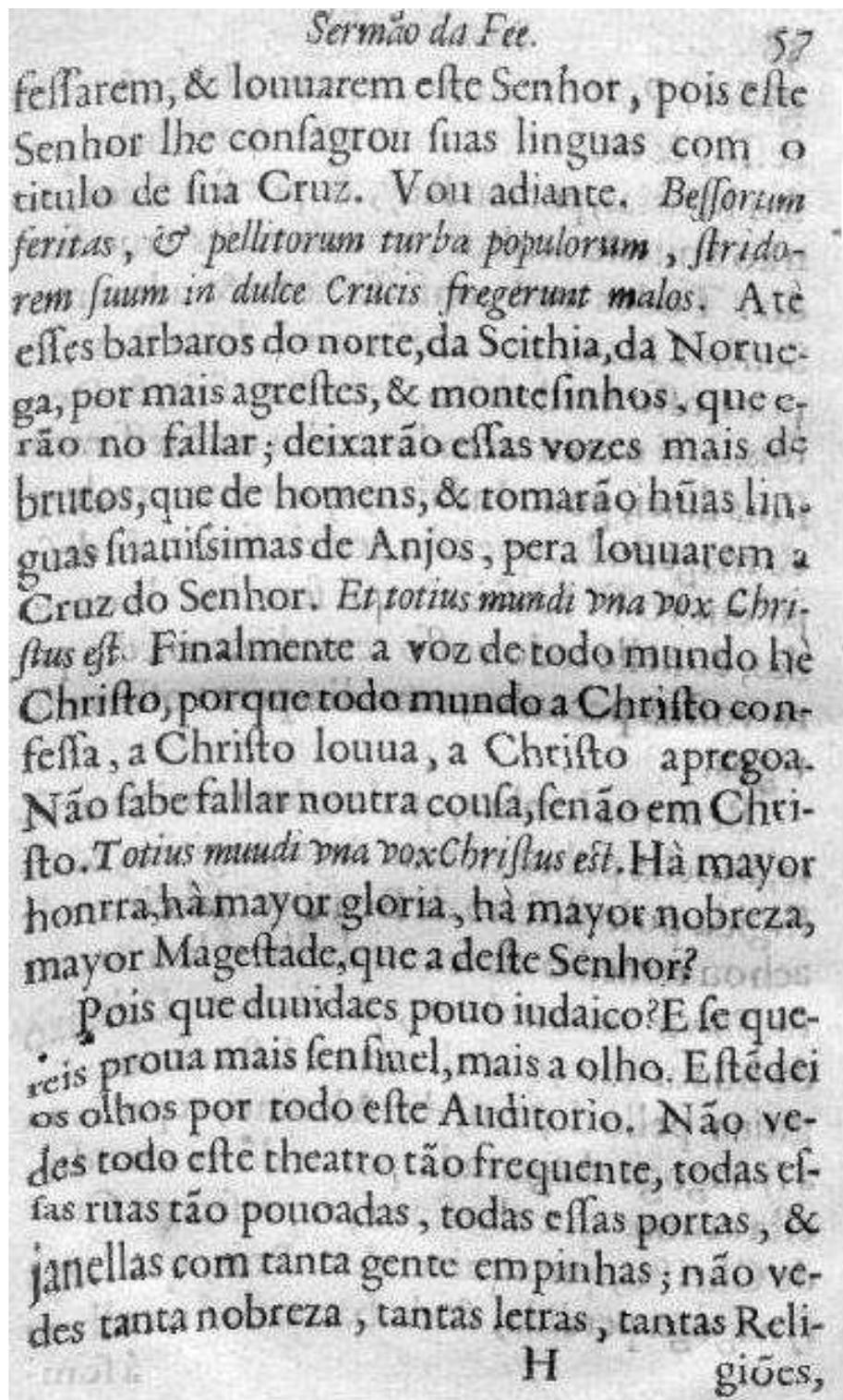


Figura 3.17: Registro da imagem destino.

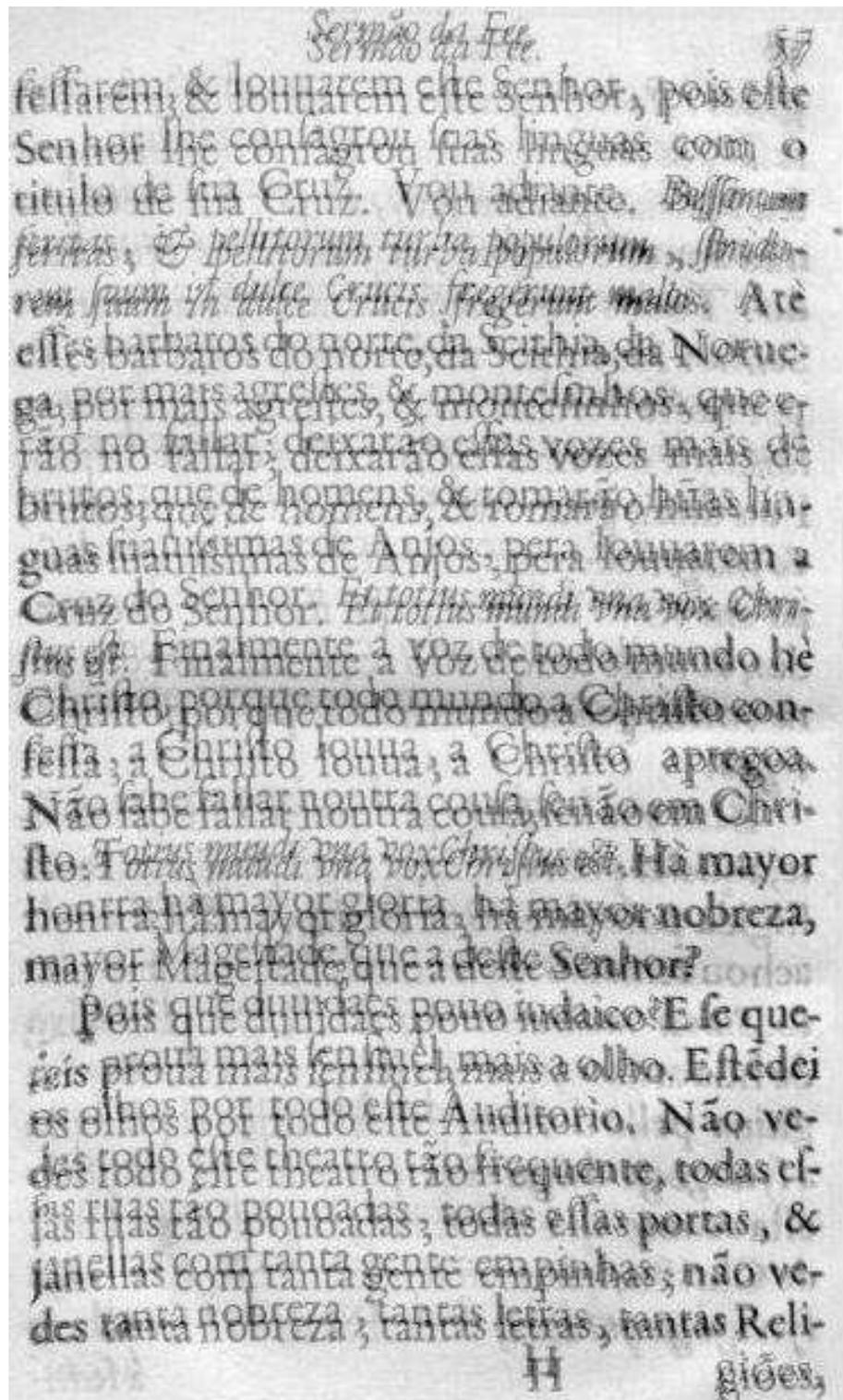


Figura 3.18: Soma da imagem destino registrada com a imagem de referência.

Analisando o teste, concluiu-se que a técnica empregada para seleção de pontos de controle foi insuficiente. Pela Figura 3.18 percebe-se uma área nítida nos extremos das linhas, ou seja, locais em que se extraiu pontos de controle, e percebe-se uma área manchada no interior do texto, ou seja, local da imagem do qual não se escolheram pontos de controle.

3.4 Pontos dos Inícios das Palavras

Uma vez que pontos de controle dos extremos das linhas não foram suficientes para modelar as distorções de interesse para utilização da técnica de registro de imagens, foi vista a necessidade de encontrar não só pontos dos extremos das linhas, mas também pontos internos às linhas de texto.

Uma maneira de extrair pontos do interior do texto é identificando palavras dentro de cada linha de texto. Então, para cada palavra da página, seleciona-se um ponto de controle. Prosseguindo com o mesmo exemplo da Seção 3.3, em que cada página possui 30 linhas e supondo que cada linha contenha 6 palavras, o número de pontos de controle selecionados será 180.

3.4.1 Segmentação de Linhas de Texto em Palavras

Para solucionar o problema de segmentar um texto em palavras, uma técnica foi desenvolvida por Muge et al. [17]. A técnica baseia-se em uma suposição do espaçamento do texto. Por observação de documentos, assume-se que o espaço que separa dois caracteres é menor que o espaço que separa duas palavras. Em imagens de documentos esse espaço é medido em pixels.

Para detecção de palavras, assume-se também que as linhas de texto da imagem já foram identificadas, sendo a técnica de segmentação de palavras aplicada para cada uma das linhas obtidas.

Assim, o primeiro passo para a segmentação é o cálculo do vetor de número de pixels em preto por coluna de cada imagem, lembrando que agora uma imagem é uma linha de texto. O vetor de número de pixels em preto é quase igual ao anterior, ou seja, para cada coluna da imagem computa-se o número de pixels em preto presentes, conforme ilustra o Algoritmo 5. Na Figura 3.19 tem-se uma linha de texto e a representação do seu histograma.

Algoritmo 5 Vetor de número de pixels em preto por coluna

Entrada: Imagem de entrada I , número de linhas lin e número de colunas col de I , limiar extraído do histograma de níveis de cinza.

Saída: $histo(i)$, vetor de número de pixels em preto por coluna.

```

para  $j = 1$  até  $col$  faça
   $cont \leftarrow 0$ 
  para  $i = 1$  até  $lin$  faça
    se  $I(i, j) \leq \text{limiar}$  então
       $cont \leftarrow cont + 1$ 
    fim-se
  fim-para
   $histo(j) \leftarrow cont$ 
fim-para
devolva  $histo$ 

```

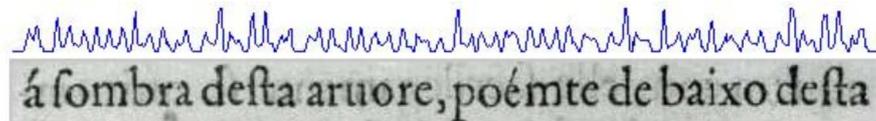


Figura 3.19: Histograma de uma imagem de uma linha de texto.

A partir do histograma do Algoritmo 5 é possível construir o histograma de comprimentos de espaços em branco que estão presentes em cada linha. O histograma de espaços em branco da linha da Figura 3.19 pode ser visto na Figura 3.20.

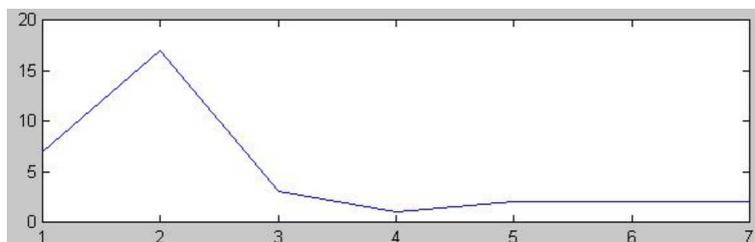


Figura 3.20: Histograma de espaços em branco da imagem de uma linha de texto.

Analisando o histograma da Figura 3.20 pode-se considerar que seu lado esquerdo corresponde ao espaço entre caracteres, ou seja, tamanhos pequenos de espaço em branco, e que seu lado direito corresponde a espaços entre palavras, isto é, tamanhos maiores de espaço.

Através do histograma escolhe-se um limiar apropriado para divisão em palavras. No exemplo da Figura 3.20, o limiar escolhido foi $l = 8$. O resultado final da segmentação pode ser visto na Figura 3.21.

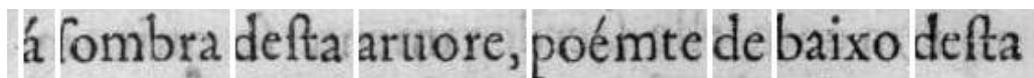


Figura 3.21: Segmentação da linha de texto em palavras. Cada linha vertical em branco representa o início de uma palavra.

O algoritmo para segmentação em palavras pode apresentar algumas falhas se a suposição inicial de espaçamento não é verificada. Como os documentos aqui tratados são antigos, podem possuir irregularidades no formato que podem levar a falhas do algoritmo. No entanto, como a técnica de segmentação é aplicada neste trabalho com o intuito de extrair pontos de controle e não necessariamente identificar todas as palavras do texto, a perda de algumas palavras não será prejudicial para seu desenvolvimento.

Durante os testes, observou-se que pelo fato do espaçamento entre as palavras ser irregular, o algoritmo em alguns casos dividiu algumas palavras e em outros não encontrou algumas palavras. Um exemplo em que as palavras não foram corretamente identificadas pode ser visto na Figura 3.22. Observe nesse exemplo que o espaçamento entre as palavras em itálico é menor que o espaçamento entre as demais.

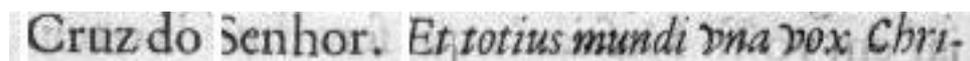


Figura 3.22: Falhas na segmentação da linha de texto em palavras.

O conjunto de pontos de controle da imagem $C = \{(x_i, y_i) \mid i = 1, \dots, n\}$, onde n é duas vezes o número de linhas de texto, é formado na coordenada x pela linha da imagem que corresponde à linha início e na coordenada y pela coluna do início de uma palavra.

Como a segmentação é utilizada com o objetivo de extrair pontos de controle, há a preocupação que dadas duas imagens da mesma página de um documento, seus conjuntos de pontos de controle se correspondam. Devido a falhas do algoritmo já comentadas anteriormente, os conjuntos de pontos extraídos das duas imagens são diferentes, ou seja, possuem tamanhos diferentes. Desta forma, torna-se necessário realizar a tarefa de correspondência dos conjuntos de pontos encontrados.

3.4.2 Correspondência dos Pontos de Controle do Início das Palavras

Vários trabalhos, encontrados na literatura, trataram do problema da correspondência dos conjuntos de pontos controle. Neste trabalho, procura-se uma forma se corresponder os pontos de controle que possua menor custo computacional possível. Seja n o tamanho do maior conjunto de pontos de controle. Uma técnica baseada em relaxação, com complexidade $O(n^6)$, foi desenvolvida por Cheng [6]. Goshtasby e Stockman [11] apresentaram uma técnica que utiliza fecho convexo e tem tempo de execução $O(n^3)$. As melhores técnicas encontradas têm complexidade $O(n^2 \log n)$. Uma delas, desenvolvida por Wong [25] utiliza busca de um mapeamento de custo mínimo em uma árvore e a outra, por Lavine et al. [13] emprega árvore geradora mínima.

Como a extração de pontos é feita linha por linha, pode-se fazer a correspondência dos pontos também desta forma. A aplicação de uma técnica de correspondência para os pontos de cada linha, ao invés de para o conjunto inteiro de pontos diminui consideravelmente o tempo de computação. Por exemplo, suponha uma página com 25 linhas, cada linha com aproximadamente 7 palavras. Neste caso, existem 175 palavras por página. Se for utilizado o algoritmo de Cheng [6] para o conjunto de pontos de tamanho 175, gasta-se $175^6 = 3 \times 10^{13}$ computações. No entanto, se a mesma técnica for aplicada para cada linha, têm-se $7^6 \times 25 = 3 \times 10^6$ computações, que representa uma economia da ordem de 10^7 computações.

A idéia da correspondência linha por linha pode ser aplicada, neste trabalho, para qualquer técnica de correspondência existente, reduzindo consideravelmente a complexidade computacional desta tarefa. No entanto, observando a forma de seleção dos pontos de controle, feita linha por linha, e aproveitando a idéia de correpondência da mesma forma, uma nova maneira de efetuar a correspondência pode ser empregada. Continuando a idéia da correspondência linha por linha, levando em consideração que a seleção de pontos também é feita linha por linha e o conjunto de pontos encontrados para cada linha está ordenado pela coordenada y , isto é, pela coluna da imagem em que se inicia uma palavra, a correspondência dos pontos de controle pode ser feita em tempo linear.

Um método para correspondência de pontos de controle utilizando as características descritas acima será proposto a seguir. A idéia do algoritmo é, dados dois conjuntos de pontos de controle, um de cada linha, percorrê-los comparando pares de pontos. Caso a distância entre um par de pontos esteja dentro de um limiar, tais pontos são correspondidos e então passa-se para os pontos seguintes. Caso contrário, verifica-se qual dos conjuntos possui a coordenada y de menor valor, o próximo ponto desse conjunto é tomado e então repete-se o procedimento. O limiar utilizado é escolhido de acordo com a tolerância permitida para cada par correspondente. Essa idéia é implementada no Algoritmo 6, que descreve o método de correspondência.

Algoritmo 6 Correspondência de pontos de controle

Entrada: p_1, p_2 conjuntos de pontos de entrada, n_1, n_2 tamanhos dos conjuntos de pontos p_1 e p_2 , respectivamente, limiar para semelhança.

Saída: pc_1 e pc_2 , conjuntos de pontos de controle correspondidos.

```

k ← 0
enquanto i ≤ n1 e j ≤ n2 faça
    dist ← p1(i, 2) − p2(j, 2)
    se dist ≤ limiar então
        k ← k + 1
        pc1(k, 1) ← p1(i, 1)
        pc1(k, 2) ← p1(i, 2)
        pc2(k, 1) ← p2(j, 1)
        pc2(k, 2) ← p2(j, 2)
    senão se p1(i, 2) < p2(j, 1) então
        i ← i + 1
    senão
        j ← j + 1
    fim-se
fim-enquanto
devolva pc1, pc2

```

O método de correspondência de pontos de controle proposto é específico para a seleção de pontos empregada neste trabalho, não sendo útil para outras aplicações. No entanto, essa foi a melhor solução encontrada, com tempo de execução linear no conjunto de pontos, isto é, proporcional ao número de pontos de controle existentes.

3.4.3 Teste do Registro com Pontos de Controle do Início das Palavras

Novamente foi realizado o teste de registro de imagens usando TPS, desta vez utilizando como pontos de controle os pontos dos inícios das palavras. As imagens de entrada continuam sendo as Figuras 3.11 e 3.13. Nas Figuras 3.23 e 3.24 estão os pontos de controle selecionados, pontos dos inícios das palavras, marcados com pontos brancos. O registro das imagens pode ser visto na Figura 3.25 e a Figura 3.26 mostra a soma da imagem de referência com a imagem registrada.

Ao analisar o resultado do registro, concluiu-se que a técnica para seleção dos pontos de controle empregada foi satisfatória. Em relação ao teste anterior, ilustrado na Figura 3.18, o registro apresentou considerável melhora comparativa no interior do texto. Na seção seguinte, uma análise dos resultados será apresentada.

3.5 Análise dos Resultados

Uma maneira de medir o resultado do registro de imagens encontrado é através da soma dos quadrados das diferenças entre as imagens. Essa técnica foi utilizada por Takeuchi e Hebert [23] para comparar as imagens após o registro. Vale ressaltar que nos trabalhos estudados não foram mencionadas formas de avaliar o registro.

Sejam I_1 e I_2 as imagens de referência e destino, respectivamente. Seja I_r a imagem destino registrada. Pode-se calcular a soma dos quadrados das diferenças, E^2 , para as imagens I_1 e I_2 , que representa as diferenças entre as imagens antes do registro, como segue:

$$E_{1,2}^2 = \sum_{x=1,y=1}^{\text{lin,col}} [I_1(x,y) - I_2(x,y)]^2, \quad (3.3)$$

onde lin é o número de linhas da imagem e col é o número de colunas. Calcula-se novamente E^2 , desta vez para as imagens I_1 e I_r , que representa as diferenças entre as imagens após o registro, como segue:

$$E_{1,r}^2 = \sum_{x=1,y=1}^{\text{lin,col}} [I_1(x,y) - I_r(x,y)]^2. \quad (3.4)$$

A comparação entre os valores de $E_{1,2}^2$ e $E_{1,r}^2$ torna possível avaliar a melhora na diferença entre a imagem de referência e a imagem obtida com o registro. Deseja-se que o valor de $E_{1,r}^2$ seja mínimo, ou seja, próximo a 0.

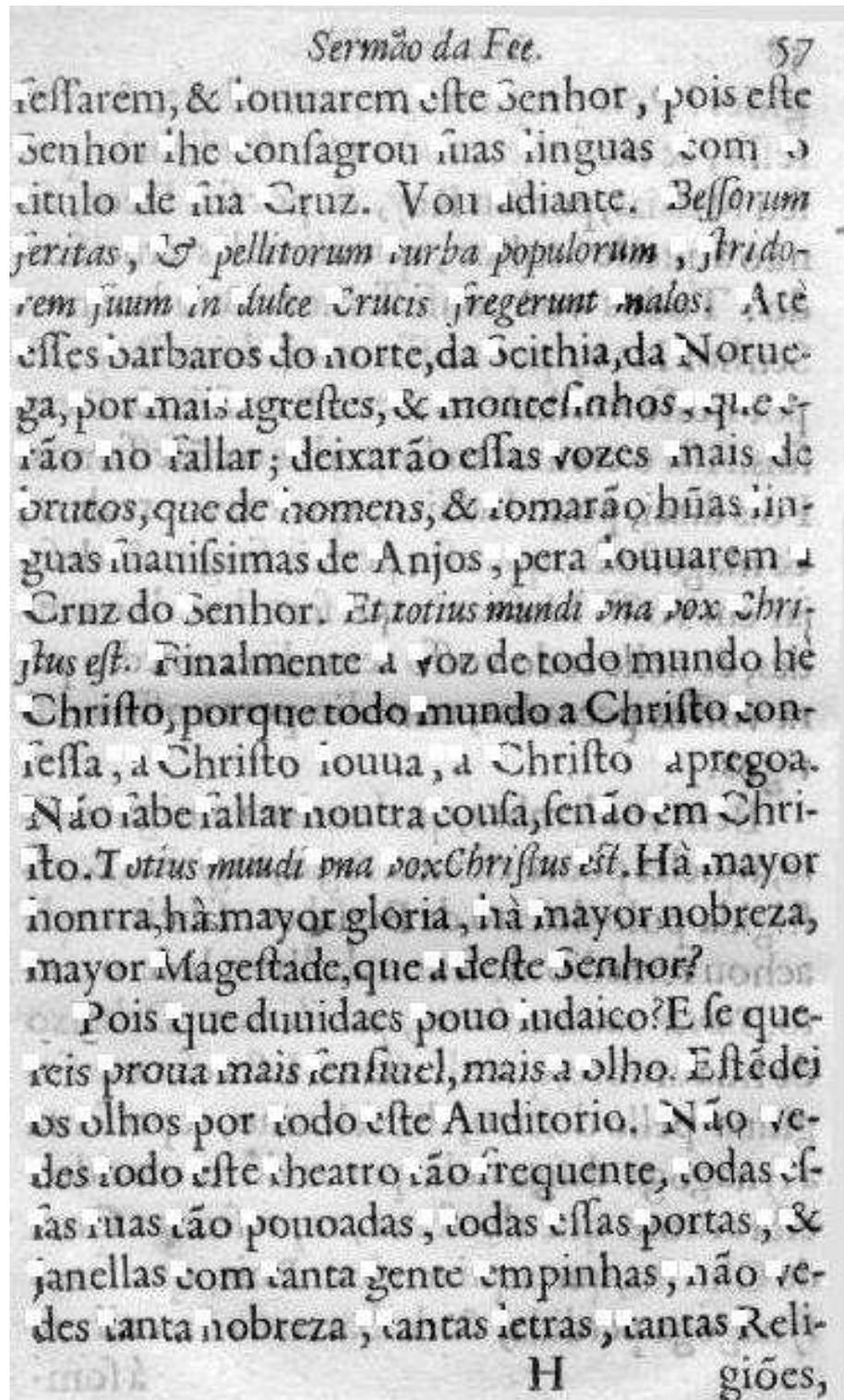


Figura 3.23: Pontos de controle da imagem de referência.

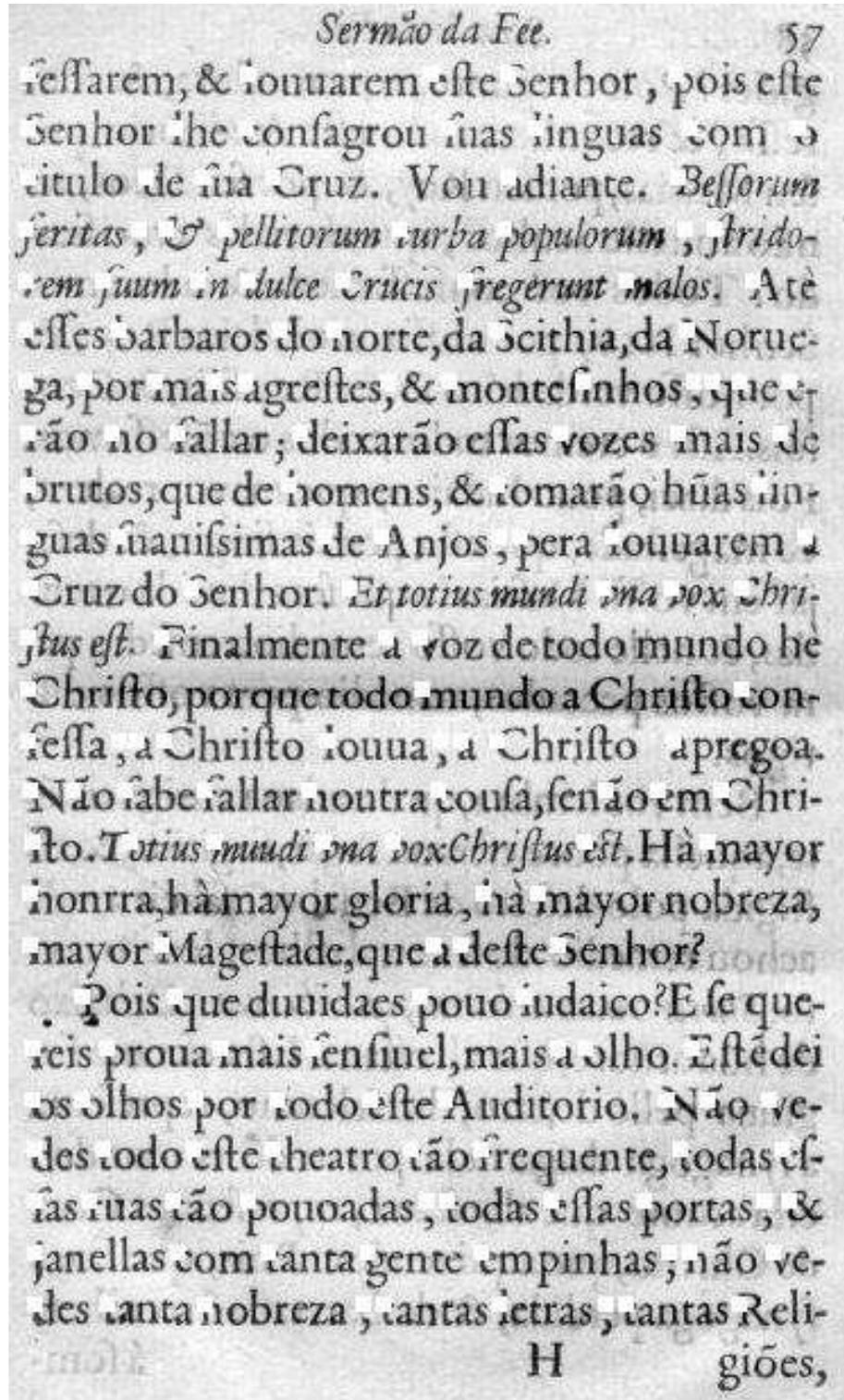


Figura 3.24: Pontos de controle da imagem destino.

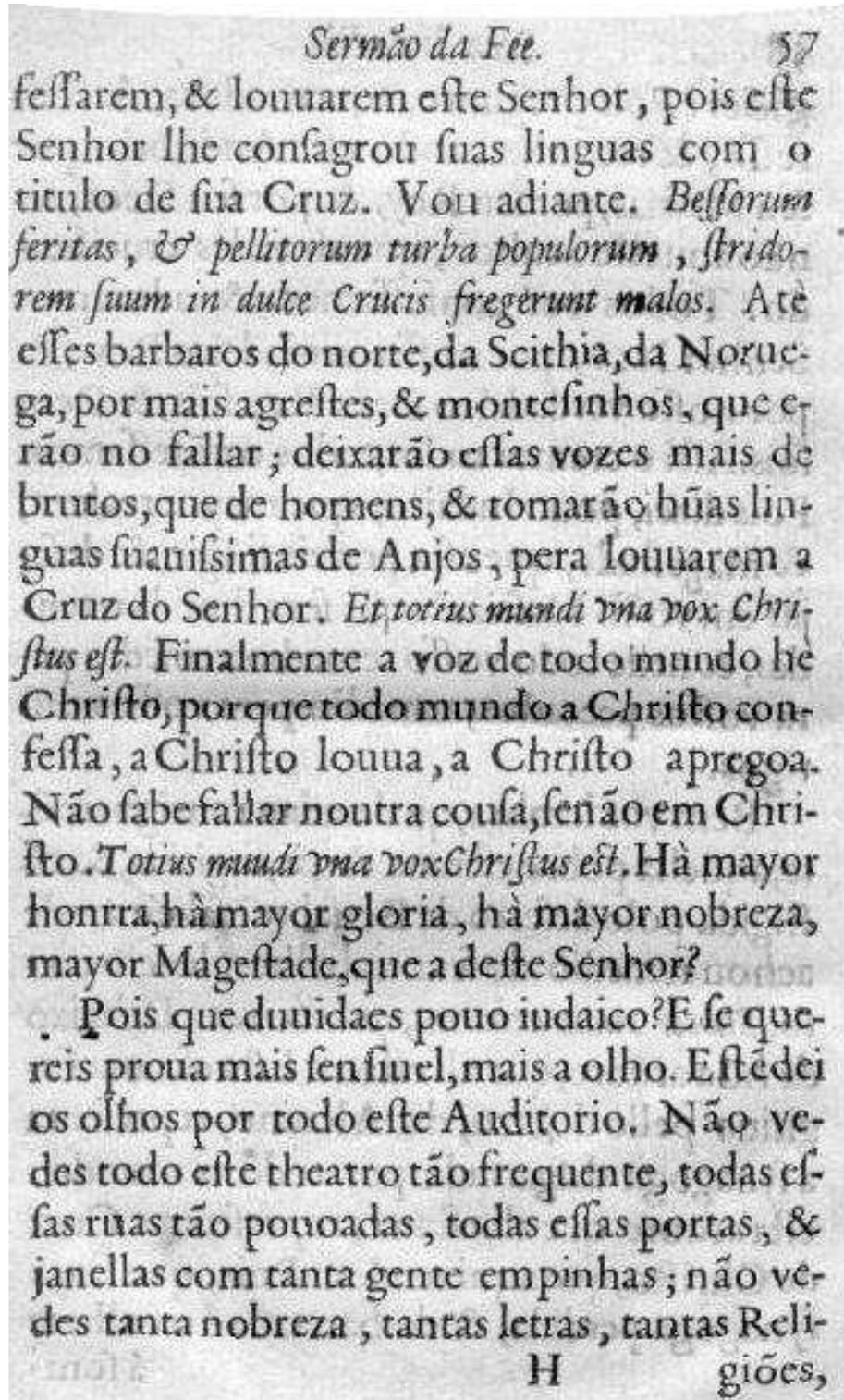


Figura 3.25: Registro da imagem destino.

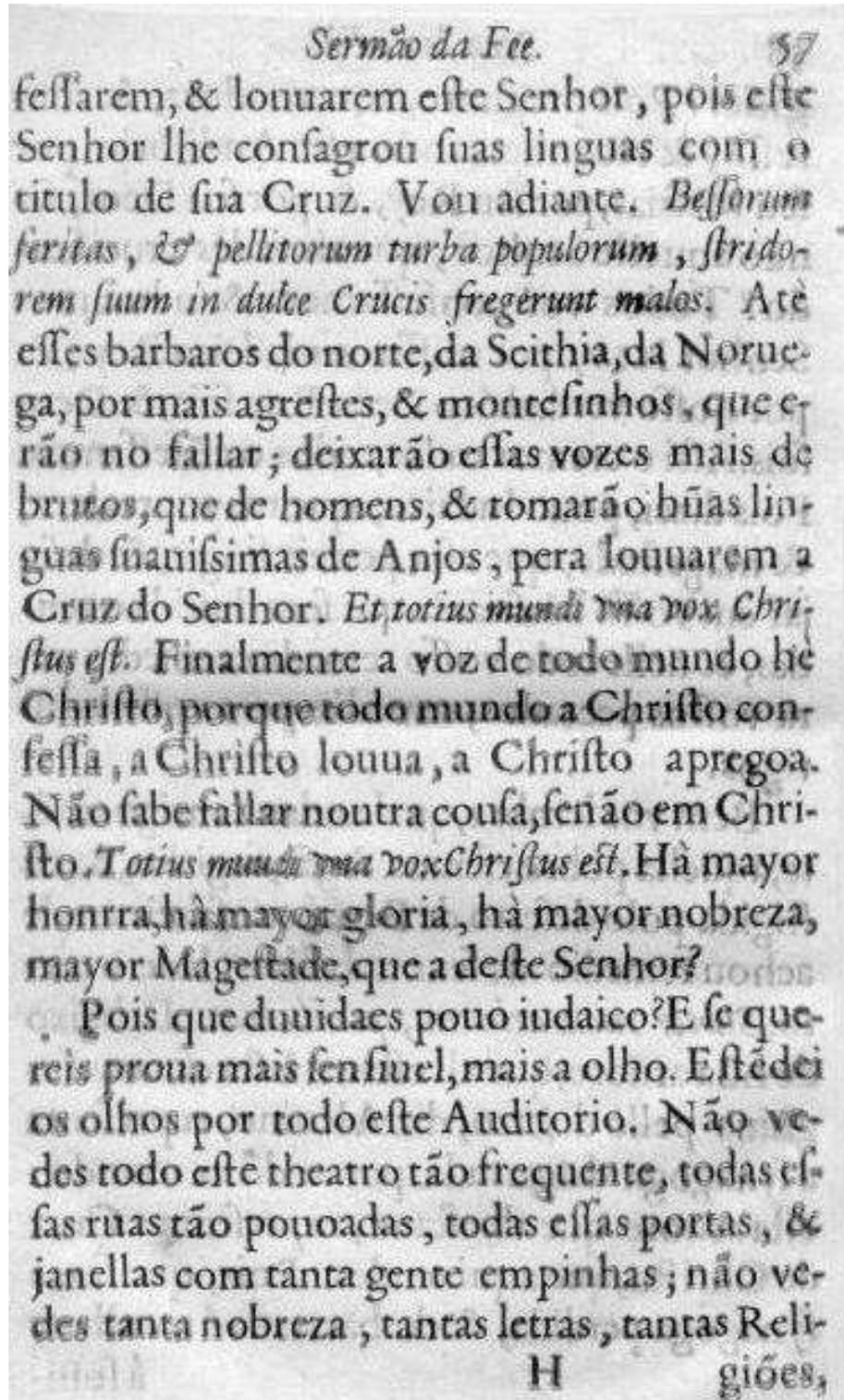


Figura 3.26: Soma da imagem registrada com imagem de referência.

Para avaliar os resultados do registro obtidos nas Seções 3.3.4 e 3.4.3 utiliza-se a soma dos quadrados das diferenças de pares de imagens. Para as imagens de referência e destino dadas pelas Figuras 3.11 e 3.13 respectivamente, obteve-se o valor $E_{1,2}^2 = 5.5650 \times 10^6$.

A Figura 3.17 mostra o teste de registro com pontos de controle selecionados dos extremos da linhas de texto. A soma dos quadrados das diferenças para a imagem de referência e a imagem registrada é $E_{1,r_1}^2 = 5.0865 \times 10^6$. Já Figura 3.25 mostra o teste de registro com pontos de controle selecionados do início das palavras. A soma dos quadrados das diferenças para a imagem de referência e a imagem registrada é $E_{1,r_2}^2 = 1.5063 \times 10^6$.

Ao dividir $E_{1,2}^2$ por E_{1,r_1}^2 obteve-se 0.9186 e ao dividir $E_{1,2}^2$ por E_{1,r_2}^2 obteve-se 0.2719. Com base nesses resultados considera-se que o registro utilizando pontos de controle do início das palavras é bem melhor que o registro utilizando pontos dos extremos das linhas de texto. Mais que isso, o resultado do registro com pontos de controle do início de palavras apresentou uma melhora significativa na diferença entre a imagem de referência e a imagem registrada, melhora essa que torna o resultado do registro satisfatório.

Capítulo 4

Remoção do Efeito da Transparência

Um dos principais problemas da digitalização de livros antigos é o tipo de papel utilizado naquela época. Esse papel era mais transparente que aquele utilizado hoje e, por isso, deixava visível a impressão do verso da folha. Tal problema pode não parecer grave para leitores do documento original, uma vez que o olho humano consegue distinguir entre as marcas da folha da frente e as marcas do verso da folha. Entretanto, quando a imagem é digitalizada e exibida em um monitor, as marcas do verso da folha aparecem como ruído e dificultam bastante a leitura, tornando necessário realizar um tratamento da imagem para eliminar marcas do verso da folha.

Na tarefa de remoção do efeito da transparência do papel deseja-se obter uma imagem digital de um lado do documento, sem marca alguma do lado oposto. Durante o desenvolvimento deste trabalho foram encontrados na literatura poucos trabalhos relacionados com o problema da remoção do efeito da transparência.

O Projeto Nabuco [15] tem como um dos seus objetivos a obtenção de imagens monocromáticas de boa definição de documentos históricos do acervo de Joaquim Nabuco. No caso de documentos escritos na frente e no verso do papel nota-se o efeito da transparência. Para remover tal efeito de uma imagem da página com 256 níveis de cinza é aplicado um método de binarização por entropia. Nesse trabalho está descrito que os resultados foram bastante satisfatórios, sendo que de cinquenta testes realizados, apenas quatro não apresentaram bons resultados.

No entanto, a técnica descrita não é indicada para o problema de remoção do efeito de transparência do Projeto Isidoro da Fonseca – PIF uma vez que produz como resultado imagens monocromáticas e deseja-se imagens em tons de cinza. O projeto PIF objetiva digitalizar textos de autores brasileiros ou de livros sobre o Brasil disponíveis em domínio público, em especial os do período colonial e do império.

Sharma [20] apresentou recentemente um trabalho de remoção do efeito de transparência que produz imagens em tons de cinza que, como no trabalho do Stolfi, utiliza um modelo em camadas da imagem, sendo uma camada de papel entre duas camadas de tinta. No entanto, diferentemente do trabalho de Stolfi, necessita de apenas uma imagem de cada lado da página. Uma das considerações de Sharma é que a transmitância do papel é muito menor que a fração de luz espalhada pelo mesmo. Essa hipótese funciona bem para papéis modernos, mas não pode ser adotada para papéis antigos, pois eram mais transparentes que os atuais. Desta forma, essa técnica também não pode ser utilizada diretamente para o problema dos documentos antigos.

4.1 Remoção do Efeito da Transparência para Imagens Binárias

Uma técnica muito simples para remoção do efeito da transparência pode ser aplicada, se as imagens são binárias. A remoção é feita calculando a média B dos valores dos pixels em uma região da página que não tenha impressão em ambos os lados e calculando a média V dos valores dos pixels em uma região da página que contenha impressão apenas no verso da folha. Então, percorre-se a imagem e para cada pixel verifica-se se o valor nesse ponto é menor que V e maior que o limiar. Em caso afirmativo, o valor do pixel da imagem recebe B . O limiar é obtido através da observação do histograma da imagem.

A Figura 4.1 mostra a imagem de uma página da qual se quer remover o efeito da transparência. A Figura 4.2 ilustra a imagem da Figura 4.1 como o efeito da transparência removido utilizando essa idéia. O efeito da transparência foi totalmente removido utilizando a técnica descrita acima. No entanto, além de remover a transparência, é possível perceber, ao observar a Figura 4.2, que alguns pixels dos caracteres foram substituídos por B , o que resultou em caracteres com menos pixels em preto, isto é, com menos informações.

4.2 Remoção do Efeito da Transparência por Stolfi

Para resolver o problema da transparência do papel, um modelo em camadas do documento foi proposto por Stolfi em [22]. O modelo consiste de uma camada de papel entre duas camadas de tinta e denomina-se **modelo sanduíche**. Segundo esse modelo, cada página do documento deve ser digitalizada por um *scanner* duas vezes, uma com uma folha de fundo branca e outra com uma folha de fundo preta.

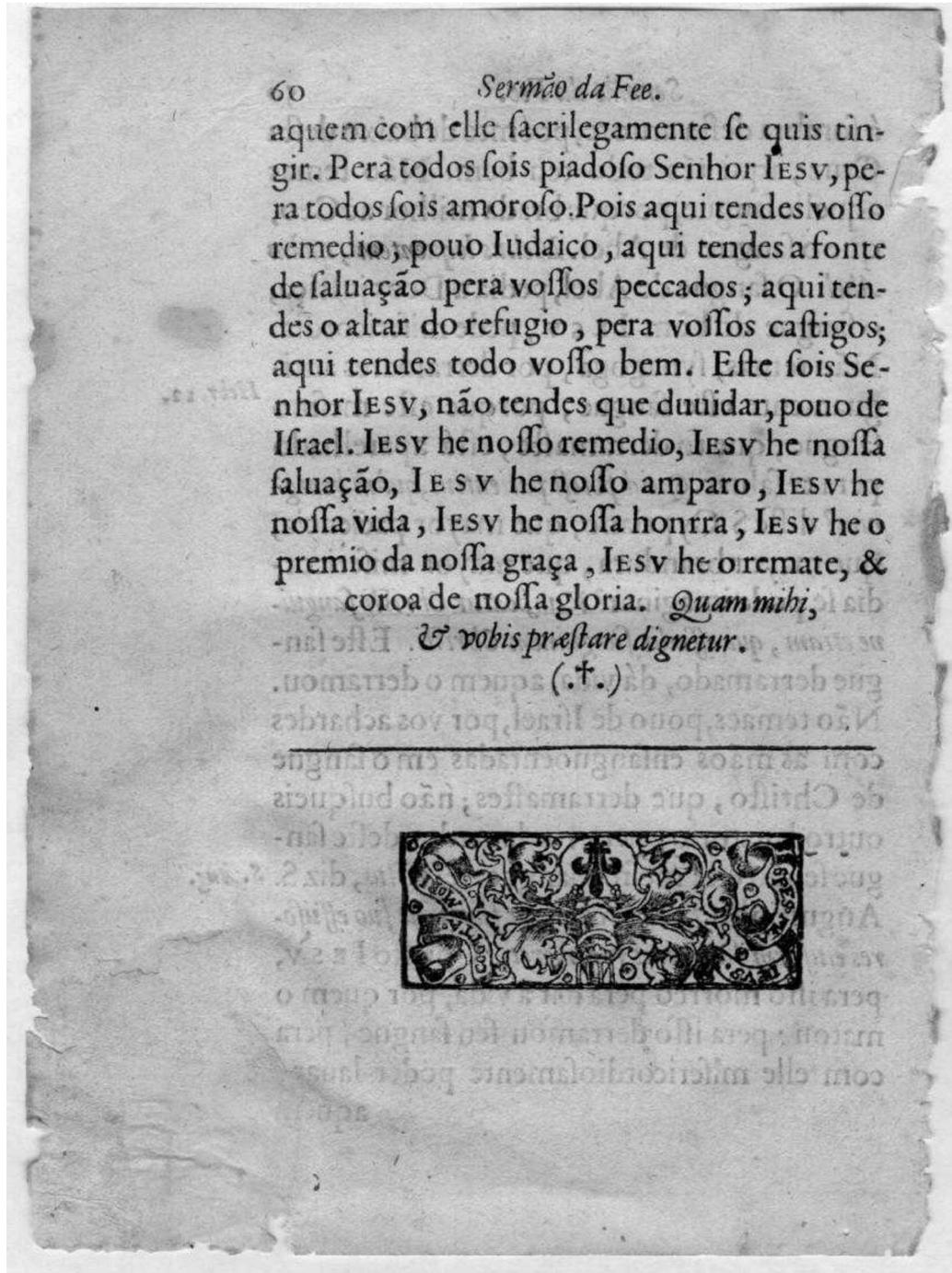


Figura 4.1: Imagem com efeito da transparência.

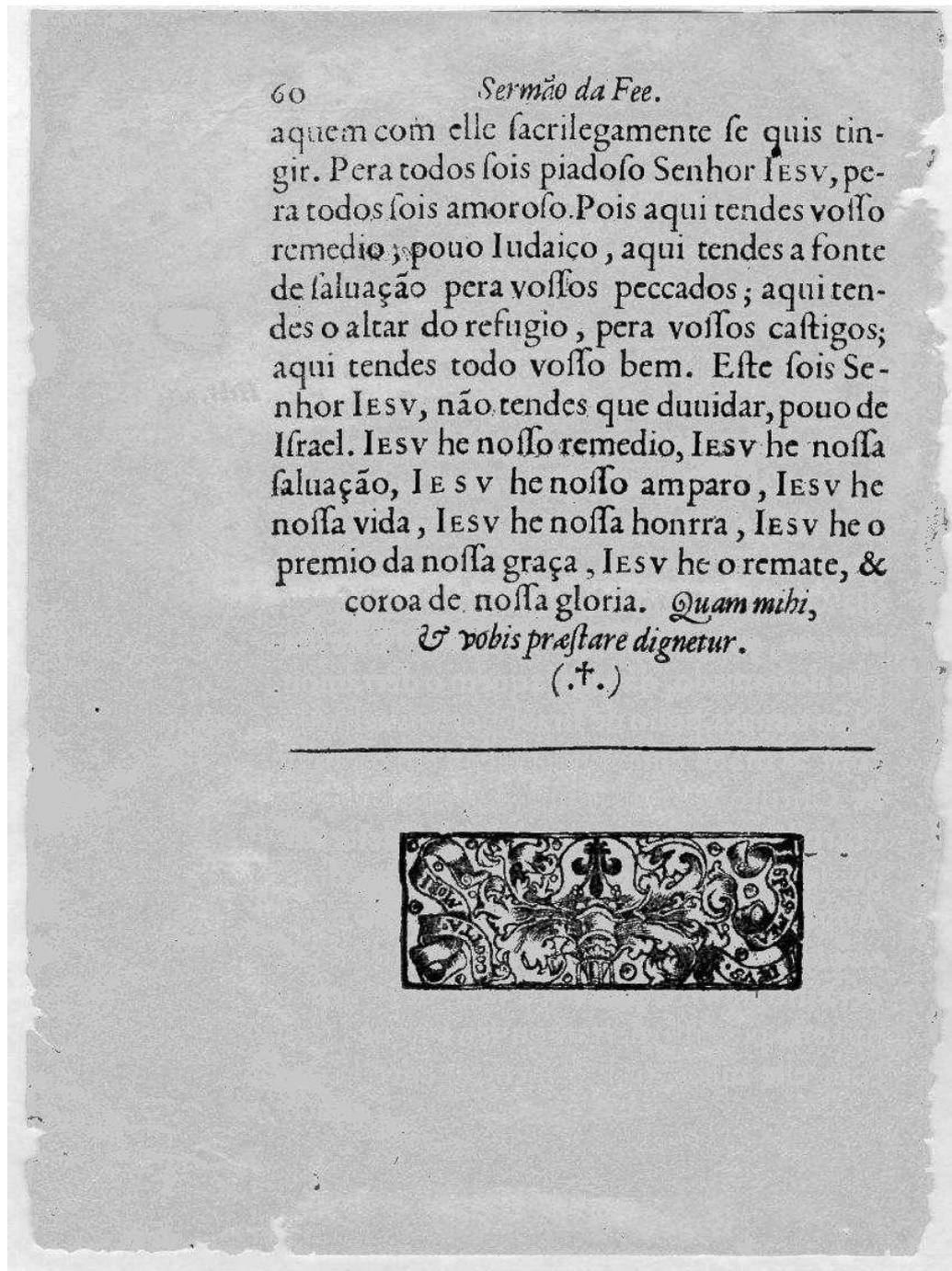


Figura 4.2: Imagem com efeito da transparência removido.

Intuitivamente, ao digitalizar uma página do livro sob uma folha preta, o efeito da transparência é praticamente eliminado, e ao digitalizar a mesma página sob uma folha branca, o efeito da transparência é realçado. Devido a diferenças na aquisição, essas duas imagens devem ser alinhadas e, por fim, processadas para eliminar o efeito da transparência.

Segundo Stolfi, os documentos consistem de marcas de tinta preta, correspondendo a texto ou figuras, depositados sobre ambas as superfícies de uma folha de papel claro. A imagem do documento obtida com uma digitalização é a soma de muitos termos, correspondendo a diferentes caminhos da luz através das camadas de tinta e papel.

Considera-se que a tinta é colorida, mas transparente, o que significa que ela pode apenas absorver a luz incidente, ou transmiti-la. Considera-se também que o papel é feito de material translúcido de cor uniforme, mas de espessura desigual, que pode tanto absorver, refletir ou transmitir luz, sempre com grande quantidade de dispersão.

A Figura 4.3 ilustra um pedaço de uma página de um livro antigo com o efeito da transparência e a Figura 4.4 mostra a mesma imagem com o efeito da transparência removido utilizando a técnica proposta por Stolfi. Essa imagem foi processada na UNICAMP, antes da técnica de registro ser implementada, sendo que as imagens de entrada foram alinhadas à mão. Um filtro foi aplicado na Figura 4.4 e o resultado pode ser visto na Figura 4.5.

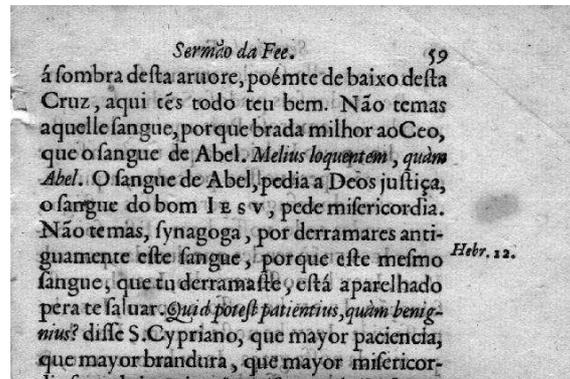


Figura 4.3: Imagem com o efeito da transparência.

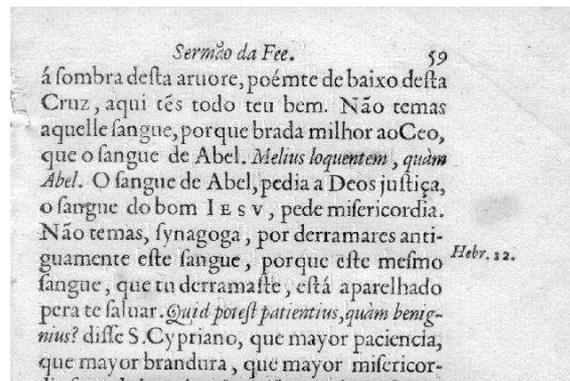
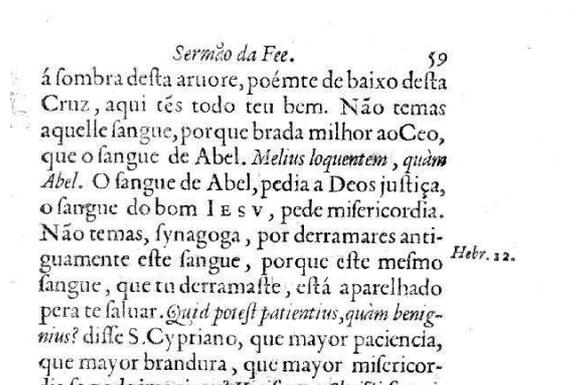


Figura 4.4: Imagem com o efeito da transparência removido utilizando-se a técnica desenvolvida por Stolfi [22].

Figura 4.5: Imagem obtida da imagem 4.4 com o filtro $y = x/0.6 - 1/3$.

Capítulo 5

Conclusão

Um dos principais problemas da digitalização de livros antigos é o tipo de papel utilizado naquela época. Esse papel era mais transparente que aquele utilizado hoje e, por isso deixava visível a impressão do verso da folha. Para solucionar esse problema, existe um método proposto por Stolfi [22] que utiliza duas imagens de cada página do documento, uma digitalizada sob um fundo branco e outra sob fundo preto.

Tendo como motivação a remoção do efeito de transparência de imagens de documentos, foram estudadas técnicas de registro de imagens que pudessem solucionar o problema do alinhamento de documentos. O estudo de tais técnicas permitiu que fossem conhecidos os vários métodos utilizados para que tornasse possível a escolha de um apropriado para o problema. Não foi encontrado na literatura nenhum trabalho que fizesse registro de imagens de documentos.

Ao analisar as diferenças entre as imagens digitalizadas obtidas, diferenças de rotação e translação foram encontradas, além de distorções. Para minimizar essas diferenças, foram escolhidas funções TPS como método de registro de imagens. As funções TPS apresentam os resultados mais precisos em registro de imagens com distorções geométricas.

A seleção de pontos de controle foi um problema crucial, uma vez que sem uma escolha apropriada dos pontos não é possível obter um registro satisfatório das imagens. Para encontrar pontos de controle foram estudadas técnicas de processamento de documentos, em especial, a segmentação do documento em linhas de texto e depois em palavras. Vale ressaltar que foram encontrados na literatura poucos trabalhos relacionados.

Devido à forma de seleção de pontos de controle adotada, foi necessário realizar um pré-processamento das imagens, antes de selecionar os pontos de controle, ou seja, corrigir a inclinação da página. Para efetuar tal tarefa foi proposto um método simples baseado na observação das páginas inclinadas e

da contagem de número de pixels.

A utilização da segmentação da imagem do documento em palavras e a extração de um ponto de controle de cada palavra foi uma escolha adequada como solução para o problema. Sabendo que a seleção dos pontos de controle é feita linha por linha, foi proposto um método para correspondência dos pontos de controle que aproveita essa informação. A melhor complexidade de tempo dos algoritmos que implementam a correspondência de pontos é $O(n^2 \log n)$, onde n é o número de pontos. Já o método proposto tem desempenho bastante satisfatório, linear no número de pontos. Embora o método de correspondência proposto seja específico para a seleção de pontos empregada, não sendo útil para outras aplicações, foi a melhor solução encontrada para o problema estudado neste trabalho.

A seleção de pontos de controle restringiu-se a páginas de documentos que possuam somente texto. Assim, uma sugestão de trabalho futuro é a segmentação da imagem do documento em uma parte textual e outra de figuras. Isso feito, pode-se aplicar a seleção de pontos de controle descrita neste trabalho na parte textual e aplicar uma outra técnica de seleção na parte de figuras.

Apêndice A

Implementação

A implementação da técnica de registro de imagens necessita de rotinas para seleção e correspondência de pontos de controle, assim como de rotinas de mapeamento da imagem destino para a imagem registrada. Antes de selecionar os pontos de controle é necessário efetuar a correção da inclinação da imagem e rotinas capazes de realizar essa tarefa foram desenvolvidas.

A tarefa de seleção de pontos de controle inclui rotinas de segmentação da imagem em linhas de texto e, em seguida, a segmentação de cada linha de texto em palavras. A rotina de correspondência dos pontos de controle das duas imagens de entrada é efetuada em conjunto com a seleção de pontos, aplicada a cada linha de texto. Por fim, para aplicar as funções mapeamento é necessária uma rotina que calcule os coeficientes das funções TPS e outra que efetue o mapeamento propriamente dito.

Uma implementação inicial desse conjunto de rotinas foi realizada utilizando o aplicativo MATLAB 5.3, que apresenta muitas facilidades. A maior facilidade encontrada no MATLAB é um conjunto de funções para processamento de imagens que abrangem grande parte das técnicas básicas. Uma dessas funções manipula arquivos de imagens: dado o nome do arquivo, a função realiza a leitura do mesmo e o armazena em uma matriz. Além das ferramentas para o processamento de imagens, existem outras ferramentas com funções matemáticas muito úteis. As rotinas de registro de imagens implementadas em MATLAB são apresentadas na Seção A.1 e estão disponíveis em

<http://www.pcs.usp.br/~valguima/mestrado/codigomatlab.html>.

A.1 Implementação em Matlab

```

%Registro de Imagens

%Leitura dos arquivos de imagem e armazenamento em matriz
arq1=input('Nome do arquivo da imagem de referência:', 's')
I1=imread(arq1);
figure, imshow(I1)
arq2=input('Nome do arquivo de imagem destino:', 's')
I2=imread(arq2);
figure, imshow(I2)

S12=soma(I1,I2);
figure, imshow(S12)

figure, imhist(I1,256) %histograma de I1 para 256 níveis de cinza
figure, imhist(I2,256) %histograma da I2 para 256 níveis de cinza

limiar=input('Limiar para as imagens:');

%Verificacao de inclinacao da imagem
I1=rotacao(I1,limiar);
I2=rotacao(I2,limiar);

%Selecao dos conjuntos de pontos de controle
[pc1,pc2]=selecaopontos(I1,I2,limiar);

%Registro da imagem destino
Ir=TPS(pc1,pc2,I2);
figure, imshow(Ir)

S1r=soma(I1,Ir);
figure, imshow(S1r)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function C=soma(A,B)
%C=soma(A,B), soma das intensidades das imagens A e B.
%Entrada: A e B = imagens a serem somadas.
%Saida: C = imagem soma de A e B.

```

```
[lin,col]=size(A);

A=im2double(A);
B=im2double(B);

for i=1:lin
    for j=1:col
        C(i,j)=(A(i,j)+B(i,j))/2;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Ir=rotacao(I,limiar)
%Ir=rotacao(I,limiar), verifica se ha inclinacao da pagina, se
%houver gira a imagem ate que imagem nao possua mais inclinacao.
%Entrada: I = imagem e limiar = limiar para limiarizacao.
%Saida: Ir = imagem I girada

ang=0.02;

[lin,col]=size(I);

flag=1;

Ir=I;

while flag
    lin1=linha(I,limiar);
    meanl=sum(lin1)/lin;

    l=0;
    cont=0;
    for i=1:lin
        if (lin1(i)>= meanl)
            cont=cont+1;
        else
            if cont > 0
                l=l+1;
            end
        end
    end
    if l==lin
        flag=0;
    end
end
```

```

        x_line(1)=i-cont; %indice do inicio da linha
        tam(1)=cont; %tamanho da linha
        cont=0;
    end
end
end

desvio_padrao=std(tam);
tam_linha=sum(tam)/l;
tam_linha1=floor(tam_linha)-ceil(desvio_padrao);
tam_linha2=ceil(tam_linha)+ceil(desvio_padrao);

%verifica se ha inclinacao na pagina para primeira
% linha de texto
i=1;
if (tam(i) >= tam_linha1) & (tam(i) <= tam_linha2)
    top=round(x_line(i)-tam(i)/0.75);
    baseline=x_line(i)+tam(i);
    bottom=round(baseline+tam(i)/0.75);

    %Verifica sentido da inclinacao da pagina.
    s=sentidorotacao(I,top,x_line(i),limiar);
    if s ~= 2 %inclinada
        Ir=rot(I,ang,s);
        input('Possui inclinação. ');
    else %sem inclinacao
        input('Não possui inclinação. ');
        flag=0;
    end
end
end
I=Ir;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Ir=rot(I,t,s)
%Ir=rot(I,t,s), gira a imagem I de um angulo t, no sentido s.
%Entrada: I = imagem a ser girada, t = angulo de rotacao e
%         s = sentido da rotacao.

```

```
%Saida: Ir = imagem I girada.
```

```
[lin,col]=size(I);
if s==1
    t=-t;
end

for i=1:lin
    for j=1:col
        x=round(cos(t)*i-sin(t)*j);
        y=round(sin(t)*i+cos(t)*j);
        if x>0 & y>0 & x<=lin & y<=col
            Ir(i,j)=I(x,y);
        else
            Ir(i,j)=255; %branco
        end
    end
end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function s=sentidorotacao(I,ini,fim,limiar)
%s=sentidorotacao(I,ini,fim,limiar), verifica o sentido da rotacao,
%0= sentido horario, 1=sentido anti-horario e %2=sem inclinacao.
%Entrada: I = imagem, ini e fim = linha ascendente e inicio da
%          media da primeira linha de texto,
%          limiar = limiar para limiarizacao.
%Saida: s = sentido da rotacao.
```

```
%Tolerancia = numero de pixels em preto aceitos na regioao
%entre a linha ascendente e inicio da media
tolerancia=180;
```

```
s=2;
[lin,col]=size(I);
conte=0; contd=0;
for i=ini:fim
    for j=1:round(col/2)
        if I(i,j)<limiar
            conte=conte+1;
```

```

        end
    end
    for j=col:-1:round(col/2)
        if I(i,j)<limiar
            contd=contd+1;
        end
    end
end

if conte>tolerancia
    s=1;
end
if contd>tolerancia
    s=0;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [pc1,pc2]=selecaoPontos(I1,I2,limiar)
%[pc1,pc2]=selecaoPontos(I1,I2,limiar), seleciona pontos
%de controle das imagens usando segmentacao em palavras.
%Entrada: I1 e I2 = imagem de referencia e destino e
%         limiar = limiar para limiarizacao.
%Saida: pc1 e pc2 = conjuntos de pontos de controle.

linha1=pixelslinha(I1,limiar);
linha2=pixelslinha(I2,limiar);

linhas1=linhastexto(I1,linha1);
linhas2=linhastexto(I2,linha2);

[lin1,col1]=size(linhas1);
[lin2,col2]=size(linhas2);

if size(linhas1)==size(linhas2)
    [lin,col]=size(linhas1);
end

for i=1:lin
    %encontra palavras para linha da imagem 1

```

```
    histo1=histpalavra(I1,linhas1(i,1),linhas1(i,2),limiar);
    brancos1=branco(I1,histo1);
    espaco1(i)=espaco(brancos1);

    %encontra palavras para linha da imagem 2
    histo2=histpalavra(I2,linhas2(i,1),linhas2(i,2),limiar);
    brancos2=branco(I2,histo2);
    espaco2(i)=espaco(brancos2);
end

%esp1 e eps2 = tamanho do espaco entre palavras
esp1=sum(espaco1);
esp1=round(esp1/lin);

esp2=sum(espaco2);
esp2=round(esp2/lin);

m=0; %indice dos conjuntos de pontos de controle

for i=1:lin
    histo1=histpalavra(I1,linhas1(i,1),linhas1(i,2),limiar);
    histo2=histpalavra(I2,linhas2(i,1),linhas2(i,2),limiar);
    ini1=separapalavras(histo1,esp1);
    ini2=separapalavras(histo2,esp2);

    [inicio1, inicio2]=casapalavra(ini1, ini2);

    %inicio1 e inicio2 possuem mesmas dimensoes
    [ind1, ind]=size(inicio1);

    %pc1: coord x = inicio da linha de texto e
    %coord y = inicio da palavra
    for j=1:ind
        m=m+1;
        pc1(m,1)=linhas1(i,1);
        pc1(m,2)=inicio1(j);

        pc2(m,1)=linhas2(i,1);
        pc2(m,2)=inicio2(j);
    end
end
end
```

```

%Inclui pontos das bordas das imagens
[pc1,pc2]=ptosborda(pc1,pc2,I1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function linha=pixelslinha(I,limiar)
%linha=pixelslinha(I,limiar), calcula vetor de numero de pixels
%preto por linha.
%Entrada: I = imagem e limiar = limiar para limiarizacao.
%Saida: linha = vetor de numero de pixels em preto por linha.

[lin,col]=size(I);

for i=1:lin
    soma=0;
    linha(i)=soma;
    for j=1:col
        if I(i,j)<limiar
            soma=soma+1;
            linha(i)=soma;
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function linhas=linhastexto(I,lin1)
%linhas=linhastexto(I,lin1), separa a imagem em linhas de texto.
%Entrada: I = imagem e lin1 = vetor de numero de pixels em
%        preto por linha.
%Saida: linhas = armazena as linhas de inicio e o fim das
%        linhas de texto,
%        linhas(k,1)=inicio da linha (x_line) e linhas(k,2)=fim
%        da linha (baseline).

[lin,col]=size(I);

meanl=sum(lin1)/lin;

```

```

l=0;
cont=0;
for i=1:lin
    if (lin1(i)>= mean1)
        cont=cont+1;
    else
        if cont > 0
            l=l+1;
            ind(l)=i-cont; %indice do inicio da linha
            aux(l)=cont; %tamanho da linha
            cont=0;
        end
    end
end

desvio_padrao=std(aux);
tam_linha=sum(aux)/l;
tam_linha1=floor(tam_linha)-ceil(desvio_padrao)-1;
tam_linha2=ceil(tam_linha)+ceil(desvio_padrao)+1;

%para cada linha computa inicios das palavras
k=0; %numero de linhas
for i=1:l
    %percorre as linhas
    if (aux(i) >= tam_linha1 & aux(i) <= tam_linha2)
        k=k+1;
        linhas(k,1)=ind(i); %linhas(k,1)=ponto do inicio da linha
        linhas(k,2)=ind(i)+aux(i); %linhas(k,2)=pto do fim da linha
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function histo=histpalavra(I,inilin,fimlin,limiar)
%histo=histpalavra(I,inilin,fimlin,limiar), calcula numero de
%pixels em preto por coluna.
%Entrada: I = imagem, inilin e fimlin= coordenadas de inicio e
%         fim de uma linha de texto e
%         limiar = limiar para limiarizacao.
%Saida: histo = vetor com numero de pixels em preto por coluna,
%        histograma de palavras.

```

```

[lin,col]=size(I);

for j=1:col
    cont=0;
for i=1:lin
    if I(i,j)<limiar %I(i,j)==preto
        cont=cont+1;
    end
end
    histo(j)=cont;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function brancos=branco(I,histo)
%brancos=branco(I,histo), calcula vetor de tamanho de espacos
%em brancos da linha de texto.
%Entrada: I =imagem e histo: vetor de numero de pixels
%          em preto por coluna
%Saida: brancos = vetor em que cada posicao k contem o numero
%       de espacos em branco de tamanho k.

[lin,col]=size(I);

%espaco inicial (paragrafo)
ini=1;
while histo(ini)==0
    ini=ini+1;
end

fim=col;
while histo(fim)==0
    fim=fim-1;
end

l=0;
cont=0;
for i=ini:fim
    if histo(i)==0

```

```

        cont=cont+1;
    else
        if cont > 0
            l=l+1;
            aux(l)=cont;    %tamanho do espaco em branco
            cont=0;
        end
    end
end

maxi=max(aux);

for i=1:maxi
    brancos(i)=0;
end

for i=1:l
    brancos(aux(i))=brancos(aux(i))+1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function esp=espaco(brancos)
%esp=espaco(brancos), calcula espaco entre palavras,
%analise do grafico.
%Entrada: brancos = vetor em que cada posicao k contem o
%          numero de espacos em branco de tamanho k.
%Saida: esp = tamanho do espaco entre palavras.

[lin,col]=size(brancos);

flag=1;
cont=0;

esp=0;

for i=2:col
    if brancos(i)>brancos(i-1) & cont==1
        if flag
            esp=i-1;

```

```

        end
        break;
    elseif brancos(i)==brancos(i-1)
        if flag
            esp=i;
        end
        flag=0;
    else
        cont=1;
        flag=1;
    end
end
end

if esp==0
    esp=round(col/2);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [inicio]=separapalavras(histo,espaco)
%[inicio]=separapalavras(histo,espaco), separa linha de texto
%em palavras.
%Entrada: histo = vetor de numero de pixels em preto por
%         coluna e espaco = tamanho do espaco entre palavras.
%Saida: inicio = vetor com pontos do inicio das palavras.

[lin,col]=size(histo);

k=0;
flag=1;
cont=0;

for j=1:col
    if histo(j)>0
        if flag==1
            ini=j;
            flag=0;
        else
            cont=0;
        end
    end
end

```

```

else
    if flag==0
        cont=cont+1;
        if cont>=espaco
            flag=1;
            k=k+1;
            inicio(k)=ini;
            cont=0;
        end
    end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [inicio1, inicio2]=casapalavra(ini1, ini2)
%[inicio1, inicio2]=casapalavra(ini1, ini2), corresponde os dois
%conjuntos de pontos ini1 e ini2.
%Entrada: ini1 e ini2= vetores com pontos dos inicios das
%          palavras.
%Saida: inicio1 e inicio2 = vetores com pontos dos inicios das
%       palavras correspondidos.

[l1, c1]=size(ini1);
[l2, c2]=size(ini2);

k=0;
i=1; j=1;
while i<=c1 & j<=c2
    dist=abs(ini1(i)-ini2(j));
    if dist<8
        k=k+1;
        inicio1(k)=ini1(i);
        inicio2(k)=ini2(j);
        i=i+1;
        j=j+1;
    else
        dist=ini1(1, i)-ini2(1, j);
        if dist<0
            i=i+1;
        end
    end
end

```



```

function Ir=TPS(pc1,pc2,Idest)
%Ir=TPS(pc1,pc2,Iref), aplica funcoes TPS para registrar imagens.
%Entrada: pc1 e pc2 = conjuntos de pontos de controle e
%          Iref = Imagem destino.
%Saida: Ir = imagem destino registrada.

%Matriz A
A=A_TPS(pc1);

%Vetores b_x e b_y, das coordenadas x e y dos pontos de controle
%da imagem destino
[b_x,b_y]=B_TPS(pc2);

%Calculo dos coeficientes das funcoes TPS, utilizando o
%sistema  $A \cdot X_x = b_x$  e  $A \cdot X_y = b_y$ 
[X_x,X_y]=solucao_svd(A,b_x,b_y);

%Mapeamento das coordenadas da imagem destino para
%a image registrada
Ir=mapeamento(Idest,X_x,X_y,pc1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function A=A_TPS(pc1)
%A=A_TPS(pc1), monta matriz A.
%Entrada: pc1 = conjunto de pontos de controle.
%Saida: A=Matriz de coeficientes.

[lin,col]=size(pc1);

n=lin+3;

for i=1:3
    for j=1:3
        A(i,j)=0;
    end
end
end

```

```

for i=4:n
    %3 linhas iniciais
    A(1,i)=1;
    A(2,i)=pc1(i-3,1);
    A(3,i)=pc1(i-3,2);
    %3 colunas iniciais
    A(i,1)=1;
    A(i,2)=pc1(i-3,1);
    A(i,3)=pc1(i-3,2);
end

for i=1:lin
    for j=1:lin
        dist(i,j)=power(pc1(j,1)-pc1(i,1),2)+
            power(pc1(j,2)-pc1(i,2),2);
    end
end

l=1;
for i=4:n
    k=1;
    for j=4:n
        if (i==j)
            A(i,j)=0;
        else
            A(i,j)=dist(k,l)*log(dist(k,l));
        end
        k=k+1;
    end
    l=l+1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [b_x,b_y]=B_TPS(pc2)
%[b_x,b_y]=B_TPS(pc2), monta vetores b_x e b_y.
%Entrada: pc2 = conjunto de pontos de controle da imagem destino.
%Saida: b_x e b_y = vetores que armazenam as coordenadas
%      x e y dos pontos de controle da imagem destino.

```

```

[lin,col]=size(pc2);
n=lin+3;

for i=1:3
    b_x(i,1)=0;
    b_y(i,1)=0;
end

for i=4:n
    b_x(i,1)=pc2(i-3,1);
    b_y(i,1)=pc2(i-3,2);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [X_x,X_y]=solucao_svd(A,b_x,b_y)
%[X_x,X_y]=solucao_svd(A,b_x,b_y). Solucao do sistema
%A.X_x=b_x e A.X_y=b_y, para A singular. Por svd A=U.W.Vt,
%nesse caso, X_x=V*Diag(W)*UT*b_x e X_y=V*Diagonal*UT*b_y.
%Ver: "Numerical Recipes in C: The Art of Scientific Computing"
%Entrada: A = matriz do sistema e b_x e b_y = vetores.
%Saida: X_x e X_y = coeficientes das funcoes TPS.

[U,W,V]=svd(A);

[lin,col]=size(A);

for i=1:lin
    for j=1:col
        if i==j
            if W(i,j)==0
                D(i)=0;
            else
                D(i)=1/W(i,j);
            end
        end
    end
end

Diagonal=diag(D);

```

```

UT=transpose(U);

X_x=V*Diagonal*UT*b_x;

X_y=V*Diagonal*UT*b_y;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function I_transf=mapeamento(I,X_x,X_y,pc1)
%I_transf=mapeamento(I, X_x,X_y,pc1,pc2), aplica funcoes de
%mapeamento nos pontos da imagem destino.
%Entrada: I = imagem destino, X_x e X_y= coeficientes das funcoes
%          TPS e pc1 = conjunto pontos de controle da imagem de
%          referencia.
%Saida: I_transf = imagem destino registrada usando funcoes TPS.

[lin1,col1]=size(X_x);
n=lin1-3;
[lin,col]=size(I);

for i=1:lin
    for j=1:col
        somax=0;
        somay=0;
        for l=1:n
            dist=power(i-pc1(1,1),2)+power(j-pc1(1,2),2);
            if dist ~= 0
                somax=somax+X_x(l+3,1)*dist*log(dist);
                somay=somay+X_y(l+3,1)*dist*log(dist);
            end
        end
        x=round(X_x(1,1)+X_x(2,1)*i+X_x(3,1)*j+somax);
        y=round(X_y(1,1)+X_y(2,1)*i+X_y(3,1)*j+somay);
        if x>0 & y>0 & x<=lin & y<=col
            I_transf(i,j)=I(x,y);
        else
            I_transf(i,j)=255; %branco
        end
    end
end
end
end

```

Referências Bibliográficas

- [1] D. I. BARNEA AND F. H. SILVERMAN, *A class of algorithms for fast digital image registration*, IEEE Transactions on Computers, C-21 (1972), pp. 179–186.
- [2] I. BARRODALE, D. SKEA, M. BERKLEY, R. KUWAHARA, AND R. POECKERT, *Warping digital images using thin plate splines*, Pattern Recognition, 26 (1992), pp. 375–376.
- [3] F. L. BOOKSTEIN, *Principal warps: thin-plate splines and decomposition of deformation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (1989).
- [4] L. G. BROWN, *A survey of image registration techniques*, ACM Computing Surveys, 24 (1992), pp. 325–376.
- [5] R. L. BURDEN AND J. D. FAIRES, *Numerical Analysis*, Brooks/Cole Publishing Company, 1997.
- [6] F. H. CHENG, *Point pattern matching algorithm invariant to geometrical transformation and distortion*, Pattern Recognition Letters, 17 (1996), pp. 1429–1435.
- [7] L. F. COSTA AND R. M. C. JUNIOR, *Shape Analysis and Classification. Theory and Practice*, CRC Press, 2000.
- [8] P. A. V. DEN ELSEN, E. D. POL, AND M. A. VIERGEVER, *Medical image matching – a review with classification*, IEEE Engineering in Medicine and Biology, 12 (1993), pp. 26–39.
- [9] A. GOSHTASBY, *Registration of images with geometric distortions*, IEEE Transactions on Geoscience and Remote Sensing, 25 (1988), pp. 60–64.
- [10] A. GOSHTASBY, G. STOCKMAN, AND C. V. PAGE, *A region-based approach to image registration with subpixel accuracy*, IEEE Transactions on Geoscience and Remote Sensing, GE-24 (1986), pp. 390–399.

-
- [11] A. GOSHTASBY AND G. C. STOCKMAN, *Point pattern matching using convex hull edges*, IEEE Transactions on Systems, Mans, and Cybernetics, SMC-15 (1985), pp. 631–637.
- [12] B. HILL, *Mapping gene expression using thin plate spline warping*. Disponível em: www.hgu.mrc.ac.uk/Users/Bill.Hill/tps.160497.seminar/tps.160497.0.html. Acesso em: 23/03/2001.
- [13] D. LAVINE, B. LAMBIRD, AND L. KANAL, *Recognition of spatial point patterns*, Pattern Recognition, 16 (1983), pp. 289–295.
- [14] C. H. LEUNG, W. C. TAM, AND Y. S. CHEUNG, *Recognition of hand-written chinese characters by elastic matching*, Image and Vision Computing, 16 (1998), pp. 979–988.
- [15] C. A. B. MELLO AND R. D. LINS, *Segmentação de imagens de documentos históricos*, in XVII Simpósio Brasileiro de Telecomunicações, 1999.
- [16] F. MENDONÇA, *Sermão no Auto da Fé em Coimbra*, Na oficina de Diogo Gomez de Loureiro, 1619.
- [17] F. MUGE, I. GRANADO, M. MENGUCCI, P. PINA, V. RAMOS, N. SIRAKOV, J. R. C. PINTO, A. MARCOLINO, M. RAMALHO, P. VIEIRA, AND A. M. AMARAL, *Automatic feature extraction and recognition for digital access of books of the renaissance*, in Research and Advanced Technology for Digital Libraries, J. Borbinha and T. Baker, eds., vol. 1923 of Lecture Notes in Computer Science, Springer-Verlag, 2000, pp. 24–34.
- [18] L. O’GORMAN AND R. KASTURI, eds., *Document Image Analysis*, IEEE Computer Society Press, 1995.
- [19] W. H. PRESS, S. A. TEUKOLSKY, W. T. VETTERLING, AND B. P. FLANNERY, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1994.
- [20] G. SHARMA, *Show-through cancellation in scans of duplex printed documents*, IEEE Transactions on Image Processing, 10 (2001), pp. 736–754.
- [21] G. STOCKMAN, S. KOPSTEIN, AND S. BENETT, *Matching images to models for registration and object detection via clustering*, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-4 (1982), pp. 390–399.

-
- [22] J. STOLFI, *Removing optical bleedthrough from imaged documents*. A ser publicado.
- [23] Y. TAKEUCHI AND M. HERBERT, *Finding images of landmarks in video sequences*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98), 1998.
- [24] Y. Y. TANG, S. LEE, AND C. Y. SUEN, *Automatic document processing: a survey*, Pattern Recognition, 29 (1996), pp. 1931–1952.
- [25] A. K. C. WONG AND R. SALAY, *An algorithm for constellation matching*, in Eighth International Conference on Pattern Recognition, 1986, pp. 546–554.
- [26] A. ZAPPALÁ, A. GEE, AND M. TAYLOR, *Document mosaicing*, Image and Vision Computing, 17 (1999), pp. 589–595.

Índice Remissivo

- afinamento de caracter
 - seleção de pontos de controle por, 18
- alinhamento, 4
- alinhamento de imagens, *veja* alinhamento
- binarização por entropia, 49
- deformação, 11, 12
 - energia de, 12, 14, 15
 - superfície de, 13
- distorção, 4, 9
 - geométrica, 8
- divisão em palavras, 40
- documentos antigos, *veja* livros antigos
- escala, 7
- espaçamento, 40
 - do texto, 38
 - irregular, 40
- função
 - de mapeamento, 2, 6, 8, 13
 - de transformação, 6
 - spline*, *veja spline*, função
 - thin plate spline*, *veja thin plate spline*, função
- imagem
 - de referência, 6, 8, 30, 43
 - destino, 6, 8, 13, 30, 43
 - registrada, 30, 43
 - segmentação da, 19, 22
- imagens
 - soma das, 30
 - soma dos quadrados das diferenças, 43
- implementação
 - correção da inclinação, 57
 - correspondência de pontos de controle, 57
 - em MATLAB, 57
 - função de mapeamento, 57
 - registro de imagens, 57
 - segmentação da imagem, 57
 - seleção de pontos de controle, 57
- inclinação, 23, 25, 30
 - correção da, 24, 25
 - da página, 25
 - sentido da, 25–26
- janela, 7
- limiar, 19, 22, 25, 40, 42, 50
- linha
 - ascendente, 20, 25, 26
 - descendente, 20
 - fim, 20, 24
 - identificação de, 19
 - início, 20, 25, 26, 40
 - média
 - fim, *veja* linha fim
 - início, *veja* linha início
- livros antigos, 1
- modelo
 - elástico, 11
 - em camadas, 50
 - sanduíche, 4, 50
 - thin plate spline*, *veja thin plate spline*, modelo

- pontos de controle, 6, 40
 - correspondência de, 2, 6, 19, 41
 - técnicas para, 7–8, 18–43
 - dos extremos das linhas, 30
 - dos inícios das palavras, 38
 - extrínsecos, 6
 - intrínsecos, 6
 - automáticos, 6
 - manuais, 6
 - seleção de, 2, 6, 17, 40
 - seleção por pontos extremos, 19, 30
- Projeto
 - Isidoro da Fonseca, 49
 - Nabuco, 49
- registro de imagens, 2, 5, 6, 8, 13, 18
 - aplicação
 - mosaico de documentos, 18
 - reconhecimento de caracteres, 18
- rotação, 7
- segmentação
 - de linhas, 24, 25, 30
 - em palavras, 38
- singular value decomposition*, 15
- spline*
 - função, 11
 - função de grau p , 11
 - superfície, 11
- superfície, 13
- SVD, *veja singular value decomposition*
- thin plate spline*
 - complexidade de tempo, 15–16
 - função, 12, 13, 30
 - coeficientes da, 13
 - modelo, 11
- TPS, *veja thin plate spline*
- transformação, 6, 9, 13
 - afim, 10, 13
 - elástica, 10
 - global, 9
 - local, 9
 - projetiva, 10
 - rígida, 9
- transformação espacial, *veja transformação*
- translação, 7
- transparência
 - óptica, 4
 - física, 4
 - remoção do efeito da, 2, 3, 49, 50
 - para imagens binárias, 50
 - técnica de Stolfi, 53
- transparência do papel, *veja transparência*
- vetor do número de pixels, 22
 - em preto, 23, 38