Theses and Dissertations

2004-03-24

Fast Registration of Tabular Document Images Using the Fourier-Mellin Transform

Luke Alexander Daysh Hutchison
Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>

Part of the [Computer Sciences Commons](#)

BYU ScholarsArchive Citation

Hutchison, Luke Alexander Daysh, "Fast Registration of Tabular Document Images Using the Fourier-Mellin Transform" (2004). *Theses and Dissertations*. 4.
<https://scholarsarchive.byu.edu/etd/4>

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

FAST REGISTRATION OF TABULAR DOCUMENT IMAGES
USING THE FOURIER-MELLIN TRANSFORM

by

Luke A. D. Hutchison

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Brigham Young University

December 2003

Copyright © 2003 Luke A. D. Hutchison
All Rights Reserved

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Luke A. D. Hutchison

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

March 19, 2004
Date

William A. Barrett
Dr. William A. Barrett, Chair

19 March 2004
Date

Tom Sederberg
Dr. Tom Sederberg

March 19, 2004
Date

Dennis Ng
Dr. Dennis Ng

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Luke A. D. Hutchison in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

March 19, 2004
Date

William A. Barrett
Dr. William A. Barrett
Chair, Graduate Committee

Accepted for the Department

David W. Embley
Dr. David W. Embley
Graduate Coordinator

Accepted for the College

G. Rex Bryce
G. Rex Bryce
Associate Dean, College of Physical
and Mathematical Sciences

ABSTRACT

FAST REGISTRATION OF TABULAR DOCUMENT IMAGES USING THE FOURIER-MELLIN TRANSFORM

Luke A. D. Hutchison

Department of Computer Science

Master of Science

Image registration, the process of finding the transformation that best maps one image to another, is an important tool in document image processing. Having properly-aligned microfilm images can help in manual and automated content extraction, zoning, and batch compression of images. An image registration algorithm is presented that quickly identifies the global affine transformation (rotation, scale, translation and/or shear) that maps one tabular document image to another, using the Fourier-Mellin Transform. Each component of the affine transform is recovered independantly from the others, dramatically reducing the parameter space of the problem, and improving upon standard Fourier-Mellin Image Registration (FMIR), which only directly separates translation from the other components. FMIR is also extended to handle shear, as well as different scale factors for each document axis. This registration method deals with all transform components in a uniform way, by working in the frequency domain. Registration is limited to foreground pixels (the

document form and printed text) through the introduction of a novel, locally adaptive foreground-background segmentation algorithm, based on the median filter. The background removal algorithm is also demonstrated as a useful tool to remove ambient signal noise during correlation. Common problems with FMIR are eliminated by background removal, meaning that apodization (tapering down to zero at the edge of the image) is not needed for accurate recovery of the rotation parameter, allowing the entire image to be used for registration. An effective new optimization to the median filter is presented. Rotation and scale parameter detection is less susceptible to problems arising from the non-commutativity of rotation and “tiling” (periodicity) than for standard FMIR, because only the regions of the frequency domain directly corresponding to tabular features are used in registration. An original method is also presented for automatically obtaining blank document templates from a set of registered document images, by computing the “pointwise median” of a set of registered documents. Finally, registration is demonstrated as an effective tool for predictive image compression. The presented registration algorithm is reliable and robust, and handles a wider range of transformation types than most document image registration systems (which typically only perform deskewing).

ACKNOWLEDGEMENTS

With thanks to my thesis advisor, Dr. Bill Barrett, for his guidance and feedback, to the other members of my thesis committee, Dr. Sederberg and Dr. Ng, and to my parents for their support and encouragement while I have attended BYU.

Contents

List of Tables	xii
List of Figures	xv
1 Introduction	1
2 Motivation	3
2.1 Assistance of manual extraction	3
2.2 Compression of document images	5
2.2.1 Image compression by predictive coding	6
2.3 Unification of framework	7
2.4 Need for document image registration	7
3 Related Work	9
3.1 Skew detection	9
3.1.1 Projection histogram methods	10
3.1.2 Hough transform methods	10
3.1.3 Vectorization methods	11
3.1.4 Other methods	12
3.2 Full image registration methods	12

4	Fourier-Mellin Image Registration	15
4.1	The Discrete Fourier Transform	15
4.1.1	Background	15
4.1.2	Continuous and Discrete Fourier Transforms	17
4.1.3	The Shift Theorem	18
4.1.4	The Convolution Theorem	19
4.2	The Mellin transform	20
4.3	The Fourier-Mellin transform	20
4.4	Fourier-Mellin image registration	21
4.5	Problems with Fourier-Mellin registration	23
5	Methods	30
5.1	Preprocessing	30
5.1.1	Automatic background removal	31
5.1.2	Histogram-based median filtering	40
5.1.3	Padding image prior to taking DFT	41
5.2	Choice of base image	42
5.3	Recovery of transform parameters	42
5.3.1	Recovery of rotation angle	42
5.3.2	Recovery of shear factor	52
5.3.3	Recovery of scale factor	53
5.3.4	Recovery of translation offset	59
5.4	Pointwise mean of registered images	60
5.5	Pointwise median for blank templates	64
5.6	Postprocessing	64
5.7	Summary	66

6	Results	67
6.1	Qualitative analysis	67
6.2	Quantitative analysis	80
6.3	Limitations	81
6.4	Non-affine transforms	84
6.5	Run time	84
7	Discussion	89
7.1	Robustness	89
7.2	Quadratic parameter fitting	90
7.3	Considerations in median filtering	91
7.3.1	Properties of the median filter	91
7.3.2	Median filter kernel size	92
7.3.3	Noise thresholding	93
7.4	Speed of DFT	93
7.5	Considerations in recovering rotation	96
7.6	Considerations in recovering scale	96
7.7	Considerations in recovering translation	97
7.8	Resolution enhancement	98
7.9	Predictive image compression	98
7.10	Optimization	100
8	Future Work	103
8.1	Perspective transform correction	103
8.2	Consensus-based registration	103
8.3	Median filter parameter selection	103
8.4	Improvement of translational registration	104

<i>CONTENTS</i>	xi
8.5 Image compression via registration	104
8.6 Image compression via background removal	104
8.7 Open question	105
9 Conclusions	109
Appendix	110
A.1 Source code	110
A.1.1 C source code for 1D quadratic parameter fitting	110
A.1.2 C source code for 2D quadratic parameter fitting	111
A.1.3 C source code for fast median filter	112
A.1.4 C source code for production of radial projection histograms .	114
A.1.5 C source code for log-polar histogram computation	115
A.2 Perspective distortion test datasets	116
A.2.1 Perspective Dataset 1	116
A.2.2 Perspective Dataset 2	117
A.3 MPEG Animation of Registration	118
Bibliography	128

List of Tables

6.1	Measured error in transformation parameter recovery	81
6.2	Run time of document image registration system	88

List of Figures

1.1	The document image registration process	2
1.2	Pointwise average of registered census document images	2
2.1	Example of manual extraction from genealogical records	4
4.1	The 1D Discrete Fourier Transform (DFT)	16
4.2	The 2D Discrete Fourier Transform (DFT)	18
4.3	The Fourier Shift Theorem	19
4.4	The 2D Fourier Mellin Transform	21
4.5	The non-commutativity of tiling and rotation	25
4.6	“Wrapping” of features in frequency space	26
4.7	Effect of shear transform on frequency domain	26
4.8	Apodization to reduce Fourier “edge effects”	29
5.1	Automatic background removal: Example 1	32
5.2	Automatic background removal: Example 2	33
5.3	Automatic background removal: Example 3	34
5.4	Automatic background removal: Detail	38
5.5	Automatic background removal: Comparison with Niblack	39
5.6	Additional edge effects created by padding image	43

5.7	Parallel lines in tabular document images act as “wavefronts”	44
5.8	1D background removal	46
5.9	Automatic background removal—effect on Fourier “edge effects”	48
5.10	“Edge effects” in the radial projection histogram	49
5.11	Elimination of “edge effects” from radial projection histogram	50
5.12	Final radial projection histogram	51
5.13	Effect of perspective distortion on frequency domain	54
5.14	Log- r histogram for x -Axis	56
5.15	Log- r histogram for y -Axis	57
5.16	Detail of pointwise mean and pointwise median images	61
5.17	Pointwise quantile images	62
5.18	Pointwise quantile images, closeup	63
5.19	Background removal applied to blank document form	65
6.1	Dataset 1, Pointwise mean of 445 unregistered images	68
6.2	Dataset 1, Pointwise mean of 445 registered images	69
6.3	Dataset 1, Pointwise median of 445 registered images	70
6.4	Dataset 2, Pointwise mean of 27 unregistered images	71
6.5	Dataset 2, Pointwise mean of 27 registered images	72
6.6	Dataset 2, Pointwise median of 27 registered images	73
6.7	Dataset 3, Pointwise mean of 23 unregistered images	74
6.8	Dataset 3, Pointwise mean of 23 registered images	75
6.9	Dataset 3, Pointwise median of 23 registered images	76
6.10	Dataset 4, Pointwise mean of 125 unregistered images	77
6.11	Dataset 4, Pointwise mean of 125 registered images	78
6.12	Dataset 4, Pointwise median of 125 registered images	79

6.13	Example of ineffective registration	83
6.14	Results of image registration under non-affine transform	85
7.1	Comparison of the Gaussian blur and median filters	91
7.2	Median filtering with different kernel sizes	94
7.3	Effect of kernel size on document background removal	95
7.4	Detail of document before and after subtraction of template	99
8.1	Mystery frequency-domain features	107
8.2	Mystery frequency-domain features (more examples)	108
	Appendix: Datasets	110
	MPEG animation of registration process	118

Chapter 1

Introduction

In many image processing and document archival fields, the ability to register, or align, related images to one another (Figure 1.1) in the presence of various image transformations is important [BHQ⁺04, Nag00]. In image registration, one image is stretched to align to another in its salient features, effectively undoing the relative transformation between the two images [Bro92, ZF03].

The need for image registration arises when a set of images share features which need to be correlated to make sense of the relationship between the images. Image registration is therefore important when dealing with a set of images that share a common template (Figure 1.2), e.g. in the case of batches of document images written on tabular forms [NB03]. By first registering the images, more information is available for reliably determining which pixels in the image are part of the original document form (Figure 1.2), making table structure recognition [CK93, XL98, TLLX97] and content extraction [CBG⁺96] potentially more reliable.

Figure 1.1: The document image registration process—These six photographs of a document are aligned by detecting the transformations mapping them to one another

[illegible]

Figure 1.2: Pointwise average of 27 registered census document images. Note the clarity of the table form and printed text, and that the handwritten content averages out to gray

Chapter 2

Motivation

Many image collections in digital libraries contain batches of documents sharing an identical tabular form, such as census records. The information in these forms is often handwritten. Unfortunately, handwritten information is currently hard to automatically extract in a reliable way [Vin02], so most extraction work for genealogy is done manually by volunteers. For example, in October 2002, The Church of Jesus Christ of Latter-day Saints released for free access on their *www.familysearch.org* web site [LDS02b] the complete, fully-indexed US 1880 and Canadian 1881 censuses. A total of 55 million names were extracted from these two censuses, after a total contribution by volunteers of 15 million hours of time [LDS02a]. The resulting indexed resource is invaluable for genealogists, because a list of records matching a given query may be returned almost instantaneously. However, there is still much work to be done to make the huge collection of genealogical records in the world accessible online.

2.1 Assistance of manual extraction

The US/Canadian census extraction project was accomplished by distributing batches of images to volunteer extractors on CD, paper, or sometimes microfilm,

(a) Image well-aligned to extraction field

(b) Image not well aligned to extraction field

Figure 2.1: Example of manual extraction of date field from hospital death records, for genealogical indexing. Screenshots are from the UDE (Universal Data Extractor) environment. Although the highlighted extraction field (in the middle of the window) aligns well with the “Date of Death” field in Figure (a), not all images will necessarily align this well. Figure (b) shows an example of an image that does not align with the extraction template: the date field falls above the template region, due to differences in rotation, scale and/or translation between the documents. (Names have been changed)

depending on the extractor's preference. A program called UDE (Universal Data Extractor) was built for those extracting from CD. UDE allows users to create a grid overlaying the image that specifies where the table cells are expected to lie on the page (Figure 2.1(a)). The program then highlights each cell in turn during data entry. The user also has a way to modify the grid by moving it or skewing it, because sometimes documents don't line up with the template (Figure 2.1(b)). All this has to be done by hand currently. Automatic document image registration would streamline and simplify this process, by aligning each document with the template grid, allowing extractors to get on with the work of extraction without worrying about creating and aligning extraction templates. It would also enable templates to be created just once for an entire class of documents and shared amongst extractors, or for algorithms to automatically create an extraction template in a reliable way, by taking a consensus of many document images [NB03].

Even if the table cells in the document image are to be read by computer rather than by a human (in the case of automatic indexing by OCR [Doe98] or handwriting recognition [SRI99, PS00, Vin02]), it is useful to have the image straightened out so that the table cells become axis-aligned rectangles, and so that the scale of all the images is exactly the same. This means that the text recognition algorithms do not need to be able to cope with potentially significant sources of global image distortion between different images.

2.2 Compression of document images

The Church of Jesus Christ of Latter-day Saints currently has archived 2.3 million rolls of genealogical microfilm, with each roll containing an average of 1300 images. Around 60,000 new films are added each year. In order to better preserve these films, and to make them readily accessible over the Internet, one million of the rolls which

are likely to be most useful to genealogists are currently being selected, and digitally scanned. The films are scanned at approximately 200 dpi (relative to the original page size) in 8-bit grayscale, and with page sizes ranging from 8.5"×11" to 17"×20". Images are typically 2000×3000 to 4000×5000 pixels in size. Uncompressed TIFF images therefore range in size from 6-20MB, averaging around 13MB. The million selected rolls of microfilm alone would require approximately 17 Petabytes to store in an uncompressed format. This would fill approximately 140,000 120GB hard drives. Obviously, efficient image compression mechanisms are important to a digital library project of this scale.

2.2.1 Image compression by predictive coding

Between one third and one half of the images in the 2.3 million roll collection were written on some sort of tabular form (parish registers, census records, death records etc.). Tabular records are often the most valuable to genealogists, because of their information-rich nature, thus they are more likely to be selected for the million-roll digitized collection. We can exploit the repetitive nature of form templates in a series of tabular documents, improving image compression ratios by predictive encoding. In this approach to compression, the blank document template is stored once, and then each image is stored as a residual difference to the blank template, i.e. the template is subtracted from each image before storage, which first necessitates registration of the image to the template. This technique has been successfully used before, at a smaller scale, for compression of machine-printed text: commonly-occurring glyphs on the page are identified by clustering, and then the page is stored symbolically, with each symbol as an index into the library of common glyphs [KD96, KD97], and with an associated residual (difference) to the library version of the glyph.

2.3 Unification of framework

Previous methods for document image registration typically only addressed document skew, and where scale and/or translational correction was also required, separate, usually unrelated methods were used to correct for those components of the transform. This can result in a somewhat ad-hoc registration system.

Shear, as well as non-affine distortions such as perspective and lens warp, is not typically addressed at all by document deskew systems. Full camera image registration mechanisms exist for generic registration problems, however these do not make use of special properties of tabular document images that make the registration task easier and less error-prone.

The presented technique allows for all transformation parameters to be recovered in the frequency domain, making the overall approach much more uniform than previous methods. The technique also handles shear distortion, which can correct for a small amount of perspective warp, and different scale factors in both document axes, neither of which is handled by standard Fourier-Mellin image registration. We also make special use of frequency-domain features of our problem space: specifically, that tabular document images produce strong radial frequency-domain features. By adapting to the specific problem domain, the present system is made more robust than a general image registration system, for this specific problem domain.

2.4 Increasing need for document image registration

With the recent availability of consumer-level digital cameras, direct digitization of document images has also become a much simpler process. Digital cameras provide great benefits, such as the ability to instantly verify image quality and re-take the image if needed. Microfilm will no doubt one day be seen as a backup mechanism rather than a primary storage mechanism, and more and more images will be

captured directly in digital form. This should translate to an even greater number of document images photographed per year. Document photography is also less intrusive than scanning. Photography can result in some image distortion, however, including changes in scale (which is not the case with document scanning). An image registration technique is needed that is able to correct for more than just document skew.

Chapter 3

Related Work

The brute-force approach to image registration under arbitrary affine transformation involves a search (local or global) through up to six dimensions of transform space (scale: two dimensions; translation: two dimensions; rotation: one dimension; and shear: a possible further one dimension, although these transform parameters are not all independent). For images of any significant resolution, this search space is unreasonably large. The brute-force approach is thus never used for image registration in practice, and various algorithms which exploit properties of specific image registration problem domains have been developed to reduce the total parameter space that needs to be searched.

3.1 Skew Detection: Detection of Angle of Rotation

The majority of previous work related to document image registration has been in skew detection and correction, e.g. correcting for rotational variation when a document is not axis-aligned on a scanner. Most skew detection algorithms are specifically designed to detect skew angle, so they do not easily extend to handle scale, translation or other transformations. However, as most of the body of document image

registration literature deals with skew detection, it is described below. A description of full image registration techniques then follows in Section 3.2.

3.1.1 Skew Detection: Projection Histogram Methods

A common strategy for finding the skew (rotation) angle of a document is to find the angle at which a projection histogram through the document has the strongest and the best-separated peaks, as determined by some cost function.

Many algorithms for skew angle detection by projection histogram have been described in the literature. Postl [Pos86, Pos88] describes a “simulated skew scan” method, where a cost function (the “premium”) is maximized while scanning through the document at a range of angles. Baird [Bai87] reduces each connected component to a single point and builds projection profiles from the set of these points—this algorithm is designed to work well with machine-printed text where each character is separate, as opposed to large tables filled with handwriting, where almost the entire image may be a single connected component.

3.1.2 Skew Detection: Hough Transform Methods

The Hough transform [DH72] is another method for determining document rotation [LTW94]. The standard Hough transform detects straight lines by mapping points in the (x, y) domain to sinusoidal curves in the (ρ, θ) domain. After accumulation of these curves, and smoothing of the accumulator, local maxima in the (ρ, θ) domain correspond to straight lines in the original image.

Amin [AFPS96] describes a method where the image is thresholded, and then groups of connected components are iteratively built, one scanline at a time. The Hough transform is applied to one connected component in each group to determine the skew angle. The algorithm is somewhat similar to the algorithm of Hinds et al. [HFA90], who use run-lengths rather than connected components to choose points

for Hough analysis. Using larger connected features such as this can reduce the computational complexity of the Hough transform [PGP99], but such block-based data reduction methods often don't work well for tabular documents with handwritten content, because so much of the document ends up being connected as one component.

In general, the Hough transform can be computationally expensive and is sensitive to noise. There is always a trade-off in accuracy of the recovered parameters and the size of the accumulator array (and thus the computational complexity). Issues such as accumulator smoothing and accumulator granularity have to be addressed during implementation.

3.1.3 Skew Detection: Vectorization Methods

Cao et al. [CWL03] have developed a method for skew detection by fitting a straight line to the set of baseline midpoints of the connected components in the image. Again, this is designed for machine print and is not suited to tabular documents containing handwriting.

Nielsen [NB03] has developed a system for alignment of tabular microfilm images, wherein a vector-based template is extracted from the document, representing the lines of the document template. A “snapping algorithm” is used to match one document's vector template to another. A consensus template is formed by combining several snapped grids together, making the algorithm more robust to noise. The production of the consensus template also allows for the transformation mapping one image to another to be found. This algorithm works well, but the author assumes that there is no major rotation of the document image away from axis alignment, because the algorithm relies on axis-aligned projection histograms to find table lines. The algorithm does allow for some variation in scale, by a brute-force search through a small neighborhood of the scale parameter space.

3.1.4 Skew Detection: Other Methods

Some other various methods and features used for skew angle detection include gradient direction [Si97], eigenvectors of the covariance matrix [OPS99, SIR99], texture directional evidence analysis [SP95] and mathematical morphology [Naj04]. A particularly interesting (and generalized) approach makes use of the Wigner-Ville distribution [KFK02].

3.2 Full Image Registration: Scale, Rotation, Translation

It should be noted again that document skew detection methods such as those described above are rarely designed to detect all affine components of the affine transform (scale, translation and shear) that best maps one image to another. If such non-rotational transformations are present in a set of images, a separate algorithm is usually needed to correct for these other components of the transform once skew angle has been detected and corrected for. This can lead to an unwieldy mix of fundamentally different algorithms that correct for each separate component of the transform; in other cases, the solution is not very general or does not handle some transform parameter at all. Solutions for the general image registration problem do exist, however.

Garris and Grother [GG96] introduce a method for both rotational and translational registration, based on detecting linear features in document forms. They demonstrate the efficacy of their method for a set of scanned forms. However, scale is not detected because it is not a consideration for scanned documents. Digital microfilm images are usually produced with a camera, and thus scale correction is a necessary capability when working with microfilm images.

Wolberg and Zokai [WZ00b] present a method for robustly registering images under rotation and scale variations, by warping an image in the image-space into

log-polar form: in this form, changes of rotation and scale become simple shifts in x and y . The system does not handle translation however, and the center for the log-polar warp has to be chosen reliably for registration to work at all.

Algorithms also exist for registration of images in a hierarchical “coarse-to-fine” fashion [ZB01, WZ00a]. Wolberg and Zokai’s method in [WZ00a] is particularly interesting, as it recovers the parameters of a perspective distortion by breaking down the images into tiles, and then finding the best affine fit between pairs of tiles in the two images to be registered. The perspective parameters are inferred by the piecewise-affine match between the images.

Other algorithms work hierarchically by starting with a low-resolution version of the image. Note that, in general, such hierarchical systems won’t work for tabular document images, because scaling down the document image causes the thin lines of the table to be easily lost.

Image registration for general (non-affine) image warp correction may be performed by identifying common features between two images, and estimating the transform model that best maps the features in one image to the features in the other [ZF03, Bro92]. These methods can be complex (and invariant feature selection strategies must be chosen), but highly-generalized techniques like these might work well for specific subsets of the registration problem, such as tabular document image registration, although a specialized technique is likely to do better by exploiting specific properties of the problem domain.

Other common approaches to full image registration have been devised which work in the frequency domain, and these are particularly interesting because, by working with spectral magnitudes alone, translation can be removed from consideration during recovery of scale and rotation parameters, simplifying the total parameter space of the problem. Frequency-based methods are covered in detail in the next chapter

(see Section 4.4 for a history of these techniques), and form the basis of the method presented in this thesis.

Chapter 4

Fourier-Mellin Image Registration

A technique that has been used successfully for image registration under rotation, scale and translation employs the “Fourier-Mellin transform” [STM03], a log-polar sampling of the frequency-space representation of an image.

4.1 The Discrete Fourier Transform

4.1.1 Background

The context in which most people are familiar with the Fourier transform is in their CD player: the “spectroscope” or moving bar graph which depicts power output at a range of sound frequencies (from bass on the left to treble on the right) is actually the Discrete Fourier Transform (DFT) of a small moving window at the current point in the output sound signal (Figure 4.1). The DFT is the conversion of a discretely-sampled signal from representation in the time domain to representation in the frequency domain. Discrete Fourier transforms are extremely useful because they reveal strong periodic components in a signal.

A *Fourier series*, named for Jean Baptiste Joseph Fourier (1768-1830), is an expansion of an arbitrary periodic function $f(x)$ in terms of an infinite sum of sinusoidal

Figure 4.1: The 1D Discrete Fourier Transform (DFT) converts from the time domain (top) to the frequency domain (bottom). The values in the frequency domain are actually complex numbers, and are usually depicted by showing only the magnitude of each frequency component.

basis functions. The *continuous Fourier transform* $F(t)$ of the function $f(x)$ is a generalization of the complex Fourier series, as the period of $f(x)$ tends to infinity. The *Discrete Fourier Transform* (DFT) is the Fourier transform of a function which is defined only at domain values separated by discrete intervals, and which is periodic over a finite number of these sampling intervals. The DFT of a signal of length N samples results in a series of N complex terms, which are the *dual* representation of the signal in the frequency domain, and thus may be used to uniquely and completely reconstruct the original signal in the signal domain. The *Fast Fourier Transform* (FFT) is a divide-and-conquer algorithm for quickly computing the Discrete Fourier Transform. Because of its divide-and-conquer design, the standard FFT can only operate on signals whose period is a power of two, but it is fast enough compared to the brute-force DFT for large signals that this limitation is usually accepted and worked around in practice (for example, by padding the signal with zeroes to round

its size up to the next power of two (Section 5.1.3)).

4.1.2 Continuous and Discrete Fourier Transforms

The Fourier Transform is a type of “integral transform” [Weic]. Integral transforms take the general form

$$g(\alpha) = \int_a^b f(t)K(\alpha, t) dt, \quad (4.1)$$

where $K(\alpha, t)$ is called the *integral kernel* of the transform.

The continuous Fourier transform and its inverse are defined as, respectively,

$$F(s) = \mathcal{F}(f(t)) = \int_{-\infty}^{\infty} f(t)e^{-i2\pi st} dt \quad (4.2)$$

$$f(t) = \mathcal{F}^{-1}(F(s)) = \int_{-\infty}^{\infty} F(s)e^{i2\pi st} ds, \quad (4.3)$$

and the discrete Fourier transform (DFT) and its inverse are then defined over a discretely-sampled and finitely-periodic domain as, respectively,

$$F(n) = \sum_{k=0}^{N-1} f_k e^{-i2\pi nk/N} \quad (4.4)$$

$$f(k) = \frac{1}{N} \sum_{n=0}^{N-1} F_n e^{i2\pi kn/N}. \quad (4.5)$$

While the Fourier transform has been described as being applied to a 1D signal, the technique extends simply to an arbitrary number of dimensions, by repeated application of the transform, one dimension at a time. A 2D DFT has several interesting properties, such as the fact that the rotation of an image in 2D results in a rotation by the same angle of the DFT. (While this is strictly true for the continuous Fourier transform, it is somewhat of an oversimplification for the discrete case, because of the

Figure 4.2: The 2D Discrete Fourier Transform (DFT)

non-commutativity of rotation and “wrapping” or tiling, and because of rotationally-dependent aliasing; see Section 4.5 for details.) The fact that rotation affects both the image and the frequency domains in similar ways is one of the properties of the Fourier transform that is made use of in the document image registration approach presented in this thesis.

A 2D DFT of an image is shown in Figure 4.2. Note that since the result of a DFT is complex, in order to visualize the result, the complex modulus (magnitude) of each point in the resulting DFT image is displayed. The log of each magnitude is also taken to enable more detail to be shown.

4.1.3 The Shift Theorem

A useful property of the Fourier transform is illustrated by the “Fourier shift theorem” (Figure 4.3): a translation (shift) of the signal by an offset of x_0 in the signal domain results in multiplication of the Fourier transform by $e^{-i2\pi x_0 t}$, which

$$\begin{aligned}
F(k) = \mathcal{F}(f(x - x_0)) &= \int_{-\infty}^{\infty} f(x - x_0) e^{-i2\pi x k} dx \\
&= \int_{-\infty}^{\infty} f(x - x_0) e^{-i2\pi(x-x_0)t} e^{-i2\pi x_0 k} dx \\
&= e^{-i2\pi x_0 t} \int_{-\infty}^{\infty} f(x - x_0) e^{-i2\pi(x-x_0)k} dx \\
&= e^{-i2\pi x_0 t} \int_{-\infty}^{\infty} f(x - x_0) e^{-i2\pi(x-x_0)k} d(x - x_0) \\
&= e^{-i2\pi x_0 t} \mathcal{F}(f(x)).
\end{aligned}$$

Figure 4.3: The Fourier Shift Theorem: a shift of the signal by an offset of x_0 in the time domain results in a pure phase shift by an angle of $2\pi x_0$ radians in the frequency domain

produces a phase shift in the Fourier domain by an angle of $2\pi x_0$ radians. The spectral magnitudes of the transform do not change, since

$$|e^{-i2\pi x_0 t} \mathcal{F}(f(x))| = |\mathcal{F}(f(x))|. \quad (4.6)$$

Thus, magnitudes in the frequency domain are invariant under translation in the signal domain, and when working only with magnitudes in the frequency domain (i.e. ignoring phase), signal translation can be completely removed from consideration. This simplifies the task of 2D image registration dramatically, by reducing the total parameter space for the registration task.

4.1.4 The Convolution Theorem: convolution and correlation in the frequency domain

Note also that the DFT may be used for correlation (also known as “cross-correlation”) of two signals: the “convolution theorem” in signal processing shows that convolution in the signal domain is the same as pointwise multiplication in the frequency domain, and vice versa. Taking the complex conjugate merely flips the

kernel, thus pointwise multiplying by the complex conjugate merely changes the operation from convolution to correlation.

4.2 The Mellin transform

The Mellin transform is another integral transform, defined as follows [Doe50, Weid]:

$$\phi(z) = \int_0^\infty t^{z-1} f(t) dt. \quad (4.7)$$

The continuous Mellin transform may be approximated for a discrete function by sampling the function at exponentially-increasing intervals, and then weighting each sample by a function of the size of the interval. (Care should be taken to sample with an interval of no more than 1, to avoid aliasing problems.)

In the case of a 2D DFT, sampling at exponential intervals results in a log-polar mapping of the frequency space, $(\log \rho, \theta)$. (See Figure 4.4.) Because of the logarithmic mapping, a scale-invariant system [LV96] is shift-invariant under the Mellin transform, and this is useful for image registration because the detection of shift in a signal is much easier than the detection of changes in scale (Section 5.3.3).

The Mellin transform is sometimes called the “scale transform” in signal processing, as it transforms the scaling operation into a simple time offset.

4.3 The Fourier-Mellin transform

By taking the Mellin transform of the Fourier transform of a 2D signal or image, we end up with a log-polar mapping of the frequency domain, $(\log \rho, \theta)$, in which rotation and scale in the image domain correspond to simple linear shifts (Figure 4.4). Thus detection of rotation and scale becomes a simple 2D correlation problem (Sections 5.3.1, 5.3.3).

Figure 4.4: The 2D Fourier Mellin Transform

There are many issues that arise with trying to compute the Fourier-Mellin transform in discrete space. A good summary of the analytical issues, how to avoid some of the problems of accuracy, and various discrete approximations to the continuous Fourier-Mellin Transform, is given in [DG01].

4.4 Fourier-Mellin image registration

Frequency-domain techniques have long been used for image registration. Kuglin and Hines [KH75] originally proposed that “phase correlation” be used for translational registration in the frequency domain. Postl [Pos86] described a Fourier-based method for skew detection, using a radial line integral through a 2D power spectrum to determine the rotation angle. De Castro and Morandi [CM87] presented a two-step process, first identifying the angle of rotation and then determining the translational shift. In the late 1970’s, the Fourier-Mellin transform was introduced by the optical research community [CP76b, CP76a, TY81], and it was later used in digital signal and image processing [Pra78, SA86, GS91], e.g. for ship classification from radar data [ZK83, IR95]. More recently, the Fourier-Mellin transform has been shown to

be particularly useful as a registration technique when image scale transformation is present [CDD94, RC96, Sto02, STM03]. Other areas in which this technique has been useful include image watermarking [LWB⁺01] and image classification [LM00].

The Fourier-Mellin transform has proved to be a powerful technique for image registration, because translation can be removed from consideration until rotation and scale have been corrected for, and by working with the Fourier-Mellin log-polar mapping, full RST (rotation-scale-translation) registration can ideally be performed as a sequence of two simple 2D correlations (one using the log-polar spectrum to recover rotation and scale, and then one in image space to recover translation). As will be discussed in more detail later, the problem is actually made slightly more complex than this due to aliasing artifacts and “edge effects.” In this thesis, it is shown that for tabular document images, the 2D correlation involved in recovering rotation and scale is completely separable into two 1D correlations (which are significantly faster to compute, and suffer from fewer aliasing problems). We also perform the final translational correlation in the frequency domain, meaning all parameters are recovered in a consistent manner, by working in frequency space. We also present methods for overcoming the problem with “edge effects” experienced with standard Fourier-Mellin registration.

It should be noted that removing translation from consideration by discarding phase information is a fairly drastic step. Phase information is a very important in forming a signal’s spatial shape. To see this, for example, we can associate the phases of each point in the DFT D_A of an image A with the magnitudes of each corresponding point in the DFT D_B of a second image B , and then take the inverse DFT of the result to get C . This will cause C to look much more like A (whose phase information was used) than B (whose magnitude information was used), because the phases in an image define the translational positions of the edges in the image. However, when

only the period and wavefront-angle are being investigated (as is the case in the first two steps of recovery of rotation angle and scale factor), the actual position of linear features in the document is unimportant.

It does not appear that the Fourier-Mellin transform has previously been applied specifically to the registration of tabular document images, however it becomes a particularly powerful approach in this specific problem domain, because of the frequency-domain properties of tabular documents (e.g. the existence of strong radial features which lie at right angles to the document axes, allowing for easy detection of rotation angle—see Figure 5.7), and the image-domain properties of tabular documents (e.g. that all useful information in a typical document is darker than its background, and that foreground information is of low spatial density, allowing for automatic background removal). These properties are exploited heavily in the presented specialization of Fourier-Mellin techniques to tabular document image registration.

4.5 Problems with the Fourier-Mellin Transform as a tool for image registration

The Fourier-Mellin transform is a powerful tool for registering images under simple combinations of rotation, scale and translation (RST) [Sto02, McG98, McG01]. It also has several weaknesses, which are analyzed in excellent detail by Stone and McGuire in [STM03].

One limitation in using the Fourier-Mellin transform for document image registration is that in its usual form, this technique doesn't handle shear distortion or anisotropic scaling (meaning it is not a solution for registration under arbitrary affine transformation), because shearing and anisotropic scaling of the image both distort the Fourier transform in a way that is nonlinear with respect to angle θ from the origin (Figure 4.7). Shear correction, together with linear anisotropic scaling, is useful for

approximating a small amount of perspective distortion in photographed documents, so it is desirable to handle shear for document images that are photographed by hand.

In practice, the 2D log-polar correlation step for the recovery of rotation and scale is actually collapsed down by projection into two separate 1D correlation problems. Working directly with the 2D $(\log \rho, \theta)$ space brings up sampling problems due to “rotationally dependent aliasing” [STM03], which is where radially sampling at an arbitrary angle in an axis-aligned discrete domain (such as the DFT) results in aliasing artifacts that differ at each sampling angle. Performing full 2D correlation can result in instability and inaccuracy of registration; using the correlation of two orthogonal 1D projection histograms to estimate the full 2D correlation helps these problems by effectively averaging the sampling process over all angles or all scale offsets, but can unfortunately lead to spurious registrations due to loss of dimensionality.

A further limitation of the Fourier-Mellin transform is that there can potentially be problems associated with recovering rotation, because the operation of “wrapping” or “tiling” (induced by the periodicity of the DFT) doesn’t commute with rotation (Figure 4.5). Thus, high-magnitude “wrapped” features in the frequency space can interfere with the log-polar transform, because rotation in the image domain does not cause a pure rotation in the frequency domain. Thus, *the DFT of a rotated image is not exactly the rotated DFT of the image*. (Note that in the continuous Fourier transform, the period of the function tends to infinity, and so the continuous Fourier transform of a rotated 2D signal *is* the rotated continuous Fourier transform of the signal, but that is not true for the DFT.) This complicates Fourier-Mellin image registration, since the 2D correlation problem may now be “thrown off” by wrapped, higher-order harmonics.

To get around problems of the non-commutativity of rotation and tiling, and more especially problems with rotationally-dependent aliasing, it would be possible

(a) Image is infinitely tiled due to assumption in DFT that domain is periodic, i.e. that it wraps at edges

(b) Rotation followed by wrapping (as is the case when we rotate in image space)

(c) Wrapping followed by rotation (ideally required for Fourier-Mellin image registration, but not what happens in reality)

(d) The DFT of a rotated image is not the rotated DFT of the image: note the overlapping radial features from adjacent tiles in the DFT (indicated with arrows)

Figure 4.5: The non-commutativity of tiling (i.e. “wrapping”) and rotation

Figure 4.6: “Edge effects” are the result of the assumption made by the DFT that the input domain wraps infinitely at the edges. Sharp black-white transitions at wrapped edges cause axis-aligned “cross artifacts” in the DFT.

Figure 4.7: The shear transform linearly distorts the DFT (by shearing the DFT at right angles to the image shear direction), but this results in a *nonlinear* distortion in the log-polar mapping of the frequency domain with respect to θ : some quadrants are “squashed”, others are “stretched” in polar coordinate representation.

to use a more sophisticated approach to produce the log-polar Fourier transform, such as computing the transform directly without using the Cartesian DFT as an intermediate step. However the computational complexity of this would be high, and so generally a log-polar sampling of the Cartesian DFT is employed, with some form of interpolation to reduce aliasing artifacts.

Perhaps most severe, however, are “edge effects” resulting from sharp, high-frequency transitions in image intensities where edges wrap (Figure 4.6) or where an image is padded in size up to the next power of two (Figure 5.6). When this happens, axis-aligned radial features, or “cross artifacts,” are created in the frequency domain, which can throw off rotation angle detection.

For Fourier-Mellin image registration, an “apodization function,” e.g. a Blackman window [STM03, Weib, BT59], is usually applied to the Fourier transform to try to curb some of these effects, the effect of which may be defined as follows:

“An apodization function (also called a ‘tapering function’) is used to bring an interferogram smoothly down to zero at the edges of the sampled region. This suppresses sidelobes which would otherwise be produced, but at the expense of widening the lines and therefore decreasing the resolution” [Weia].

In 2D apodization, a circular subregion of the image is chosen before applying the Fourier transform, and pixels outside the circular region are set to zero. Pixels within the circular subregion are weighted so that they transition smoothly to zero at the edge of the circle (Figure 4.8), as a function of radius from the center, in order to remove the high-frequency transitions at the edge of the image. Unfortunately, such windowing requires that we somehow choose a circular subregion within the image to apodize, and we throw away the rest, meaning the image cannot be used in its entirety for registration.

Thus, image registration based on the Fourier-Mellin transform, as it is typically

performed with apodization as a preprocessing step, is potentially very sensitive to the choice of apodization subregion if the images are not square. And yet without apodization, “cross artifacts” produced by edge effects can interfere with proper registration. An alternative to apodization which eliminates Fourier edge-effects, automatic background removal using the median filter, is proposed in Section 5.1.1.

A further problem with using standard Fourier-Mellin image registration for document images is that the final step relies upon standard 2D cross-correlation (sometimes implemented or expressed as “phase correlation”) to correct for translation. This may work well for images in general, but does not work well for document images, particularly when the document photograph was taken under bad lighting conditions (e.g. if there is a bright region in the middle of the image, and the intensity of the document background fades out to gray at the edges of the image). In translational correlation, we are not interested at all in registering the backgrounds (either the white or gray paper, or the black borders around the page) of two images to each other. All that we are interested in registering is the foreground of the document image, i.e. fine dark lines on a lighter background. However, due to the relatively large spatial density of the document background when compared to the foreground, the background coloring can easily and adversely affect the chosen correlation offset, by producing a huge upward trend in the correlation output in the region of the correlation where the background intensities best line up. Thus, the fact that background pixels form the largest proportion of document image area can mean that backgrounds dominate the translational correlation process. Subtracting the mean value before correlation does not generally work either, specifically because background intensity often changes across the document. A locally-adaptive thresholding or background removal algorithm is needed to solve this problem. Significantly, the same automatic background removal algorithm proposed in Section 5.1.1

for elimination of Fourier edge-effects also solves the problem of incorrect translational registration due to changes in background intensity: by subtracting the median background intensity across a neighborhood, the background is altogether removed from consideration during 2D correlation.

The uncolored Houses are situated within the Boundaries of the										Page 1
Ord. Panel (or Township) of	City or Corporate Borough of	Municipal Ward of	Parish or Precinct of	Ward or Village of	Urban Sanitary District of	Rural Sanitary District of	Population of	Total or Partial		
of	of	of	of	of	of	of	of	of	of	
No. of	NAME of HOUSE	NAME and Address of each Person	RELATION	AGE last Birthday of	Rank, Profession, or OCCUPATION	WEEKS WORK	☐ Dead and Dead of War ☐ Deceased in this Census			
1	Edwards St.	William J. Edwards	Head of Family	40	Blacksmith	10				
1	Edwards St.	Mary J. Edwards	Wife	30	Housewife	10				
1	Edwards St.	John J. Edwards	Son	10	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	8	Student	10				
1	Edwards St.	William J. Edwards	Son	5	Student	10				
1	Edwards St.	John J. Edwards	Son	3	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				
1	Edwards St.	John J. Edwards	Son	1	Student	10				
1	Edwards St.	Elizabeth J. Edwards	Daughter	1	Student	10				
1	Edwards St.	William J. Edwards	Son	1	Student	10				

DATE	NO. OF SEEDS	NO. OF PLANTS	NAME OF PLANT
1900	100	100	Red Pine
1901	100	100	Red Pine
1902	100	100	Red Pine
1903	100	100	Red Pine
1904	100	100	Red Pine
1905	100	100	Red Pine
1906	100	100	Red Pine
1907	100	100	Red Pine
1908	100	100	Red Pine
1909	100	100	Red Pine
1910	100	100	Red Pine
1911	100	100	Red Pine
1912	100	100	Red Pine
1913	100	100	Red Pine
1914	100	100	Red Pine
1915	100	100	Red Pine
1916	100	100	Red Pine
1917	100	100	Red Pine
1918	100	100	Red Pine
1919	100	100	Red Pine
1920	100	100	Red Pine
1921	100	100	Red Pine
1922	100	100	Red Pine
1923	100	100	Red Pine
1924	100	100	Red Pine
1925	100	100	Red Pine
1926	100	100	Red Pine
1927	100	100	Red Pine
1928	100	100	Red Pine
1929	100	100	Red Pine
1930	100	100	Red Pine
1931	100	100	Red Pine
1932	100	100	Red Pine
1933	100	100	Red Pine
1934	100	100	Red Pine
1935	100	100	Red Pine
1936	100	100	Red Pine
1937	100	100	Red Pine
1938	100	100	Red Pine
1939	100	100	Red Pine
1940	100	100	Red Pine
1941	100	100	Red Pine
1942	100	100	Red Pine
1943	100	100	Red Pine
1944	100	100	Red Pine
1945	100	100	Red Pine
1946	100	100	Red Pine
1947	100	100	Red Pine
1948	100	100	Red Pine
1949	100	100	Red Pine
1950	100	100	Red Pine
1951	100	100	Red Pine
1952	100	100	Red Pine
1953	100	100	Red Pine
1954	100	100	Red Pine
1955	100	100	Red Pine
1956	100	100	Red Pine
1957	100	100	Red Pine
1958	100	100	Red Pine
1959	100	100	Red Pine
1960	100	100	Red Pine
1961	100	100	Red Pine
1962	100	100	Red Pine
1963	100	100	Red Pine
1964	100	100	Red Pine
1965	100	100	Red Pine
1966	100	100	Red Pine
1967	100	100	Red Pine
1968	100	100	Red Pine
1969	100	100	Red Pine
1970	100	100	Red Pine
1971	100	100	Red Pine
1972	100	100	Red Pine
1973	100	100	Red Pine
1974	100	100	Red Pine
1975	100	100	Red Pine
1976	100	100	Red Pine
1977	100	100	Red Pine
1978	100	100	Red Pine
1979	100	100	Red Pine
1980	100	100	Red Pine
1981	100	100	Red Pine
1982	100	100	Red Pine
1983	100	100	Red Pine
1984	100	100	Red Pine
1985	100	100	Red Pine
1986	100	100	Red Pine
1987	100	100	Red Pine
1988	100	100	Red Pine
1989	100	100	Red Pine
1990	100	100	Red Pine
1991	100	100	Red Pine
1992	100	100	Red Pine
1993	100	100	Red Pine

(a) Original image

(b) Image after processing with a Gaussian apodization function (size is largest power of 2 that fits within image)

Figure 4.8: Apodization to reduce “cross artifacts” caused by Fourier “edge effects.” This process unfortunately prevents the entire image from being used for registration, which makes standard Fourier-Mellin image registration sensitive to the choice of apodized subregion. With median filter-based background removal, the entire image may be used for registration, since there are no sharp intensity transitions at the image edges after the background has been removed.

Chapter 5

Fourier-Mellin Document Image Registration: Methods

We present here a specialization of the Fourier-Mellin transform which exploits the properties of tabular document images to overcome most of the standard problems with using the Fourier-Mellin transform for image registration.

5.1 Preprocessing

Many of the problems experienced with applying the Fourier-Mellin transform to document image registration, e.g. Fourier “edge effects” and incorrect translational registration, occur because of the commonly-uneven intensity of document backgrounds. The background is comprised of the parts of the image we are not interested in registering (the gray paper color, the black border that frequently occurs around a page, etc.), and makes up the bulk of the pixels in the image (far more pixels in a standard document image are background pixels than foreground pixels, dramatically decreasing the signal-to-noise ratio for translational correlation when the background intensity varies). In order to accurately register document images, this

document background must be removed completely from consideration. We introduce here a novel automatic background removal algorithm for document images. Specific details of the problems solved by background removal are presented in Sections 5.3.1 and 5.3.4.

5.1.1 Automatic “locally-adaptive” background removal using the median filter

Typically, a document is written with dark ink on light paper. To separate the document content from the paper, many thresholding algorithms attempt to find a threshold based on the global histogram of graylevel intensities [Ots79]. Better algorithms are locally-adaptive, choosing the segmentation threshold according to statistical properties of the histogram of a local neighborhood of each pixel, e.g. the Niblack algorithm [Nib86] (Figure 5.5(b)).

The presented foreground-background segmentation algorithm is not really a thresholding algorithm at all, it is more a “background removal algorithm.” Simply described, we perform median filtering of a document image to remove the smaller foreground features (writing, lines etc.). We then subtract this “background image” from the original image, leaving just the foreground (Figures 5.1, 5.2). This background removal method works because foreground pixels are not usually spatially dense, and thus comprise substantially fewer than half the pixels in any reasonably-sized neighborhood of a document image.

Significantly, this background removal method in its presented form does not produce a binary image mask, but leaves all of the subtle foreground variations in intensity intact (Figure 5.4(b)). A grayscale foreground image can provide a great deal more useful information than a thresholded (bitonal) image, and thus may preserve information that is useful for later processing. Also, the black border around the

(a) Original document image (800×630)

(b) “Background image” (median filter of original image; kernel size 17)

(c) “Foregound image” (result of subtracting background image from original)

(d) Manually-thresholded foreground image (for comparison)

Figure 5.1: New algorithm for automatic background removal based on the median filter, used as a preprocessing stage to the Fourier-Mellin transform. This algorithm can eliminate Fourier “edge-effects,” shown in Figure 4.5.

(a) Original document image (1600×1100)

(b) “Background image” (median filter of original image; kernel size 22)

(c) “Foreground image” (result of subtracting background image from original)

(d) Manually-thresholded foreground image (for comparison)

Figure 5.2: New algorithm for automatic background removal based on the median filter, used as a preprocessing stage to the Fourier-Mellin transform. This algorithm can eliminate Fourier “edge-effects”, shown in Figure 4.5.

This is a high-resolution scan of a historical document. The title at the top reads "SCHEDULE of the whole number of Persons within the Division allotted to *Abolition*". Below the title is a large table with multiple columns and rows. The left side of the table lists names, and the right side contains numerical data. The document is aged and shows some wear.

(a) Original document image (1200×980)

(b) "Background image" (median filter of original image; kernel size 17)

This image shows the foreground of the original document, which has been isolated by subtracting the background image from the original. The text and table structure are clearly visible, but the background is white, and there is some noise and artifacts from the subtraction process.

(c) "Foreground image" (result of subtracting background image from original)

This image shows a manually-thresholded version of the foreground image. It is a binary image where the text and table structure are represented by black pixels on a white background. This image is provided for comparison with the result of the automatic background removal algorithm.

(d) Manually-thresholded foreground image (for comparison)

Figure 5.3: New algorithm for automatic background removal based on the median filter, used as a preprocessing stage to the Fourier-Mellin transform. This algorithm can eliminate Fourier "edge-effects," shown in Figure 4.5.

page is removed as background, even though it is the same color as the document foreground pixels.

Figure 5.5(b) shows that the Niblack algorithm does not remove the black border from around the page, and can even have trouble with local changes in background intensity. This was approximately the best global result that could be obtained with the Niblack algorithm using a manual search of the parameter space. As shown, adjusting parameters to give a globally-optimal thresholding doesn't always give locally-optimal results, even with a locally-adaptive algorithm. (The image is cropped, but the background intensity level varied fairly significantly across the page.)

Median filter background removal is similar to the “top hat transform”, derived from morphological closing. The top hat transform is a subtraction of the original image from the result of the morphological closing operator. The transform is very sensitive to the size of the closing element used (perhaps more sensitive than median filter-based background removal is to the size of the median filtering kernel, although the comparison of these two algorithms is left as future work). This method is used for “layered coding” in image compression developed by Huang et. al. [HWW98], which uses the top hat transform to accomplish a similar removal of fine foreground details in producing a background image.

If a bitonal image is required, a simple thresholding operation can be performed after background removal (Figure 5.5(a)). Visual comparison with the Niblack locally-adaptive thresholding algorithm [Nib86] can be made by thresholding after background removal (Figures 5.5(a), 5.5(b)), and it can be seen that the thresholded version of the image after background removal is cleaner than the Niblack-thresholded version, since sometimes no global parameter can be found to completely segment the foreground from the background. The quality of the non-thresholded output of median filter background removal (Figure 5.4(b)) is far superior to either thresholded

version, because of the extra grayscale information available. It is almost certainly better not to threshold after background removal unless needed, because of the additional information available to image processing algorithms, including the image registration algorithm presented in this thesis. For example, a grayscale image can provide a much more accurate estimate of rotation angle than a bitonal image. The median filter may be useful as a “conditioning algorithm” for binary thresholding however.

Another feature of this background removal algorithm is that it is able to distinguish between black foreground pixels (such as penstrokes) and black background pixels (such as the black border that frequently occurs around microfilm images), because of difference in spatial density of each. Thus black background components are removed as well as light-gray background components, a feat that is difficult with most locally-adaptive thresholding algorithms.

Importantly for the problem of Fourier-Mellin image registration, this form of background removal is able to distinguish between the black of the foreground (the handwriting or document form), and the black (or even gray) of the background (the border around the page in a microfilm document image). Thus black borders are stripped out as easily as gray paper backgrounds, and different background grayscale levels at opposite edges of an image are normalized to zero, eliminating edge transitions, i.e. performing “image-leveling”. This is difficult with most locally-adaptive thresholding algorithms. Removal of grayscale transitions at the edge of the page and the edge of the image aids in accurate recovery of rotation angles (Section 5.3.1), by eliminating “cross artifacts” that otherwise can throw off rotational registration (Figure 5.9). Traditionally, an apodization function has been used to accomplish this (Section 4.5). Preprocessing with median filter-based background removal, however, allows the entire image to be used for registration (not just a circular window taken

from somewhere within the image), and allows all foreground pixels in the image to be weighted equally (i.e. they are not faded out to zero at the edges). Thus, in theory, median filter background removal (which is only applicable to document images) should allow us to register document images more accurately and robustly than for arbitrary images with standard Fourier-Mellin registration.

Like the Niblack algorithm, the median filter background removal algorithm has parameters that need to be manually set (or heuristically determined). Most importantly, the kernel size has to be appropriately chosen for each image or batch of images. The choice of this parameter is discussed in Section 7.3.

5.1.1.1 Efficiency considerations in median filtering

Speed is also an issue with median filtering: a straight quicksort on the neighborhood of each point can result in long processing times. However, fast algorithms for median filtering exist, and a significant new optimization to one of the current fastest algorithms is presented below.

Fast algorithms for finding the median (or, generally, the k -th largest element) mostly work by only sorting as much of the kernel array as is needed to find the median, or by only performing as many comparisons as are needed to determine which element is the median value [Dev98, GW93]. (The code from [Dev98], used in the presented document image registration system, is given in Appendix A.1.3.) These algorithms are typically linear in the number of items being partially sorted, i.e. in the case of 2D median filtering, they scale with the square of the kernel radius.

Median filtering for background removal can be sped up by working on a much lower-resolution version of the image. We also present here an optimization to the fastest algorithm presented in [Dev98], which can double the processing speed for the median filter on typical microfilm images. Median filtering speed is discussed again

(a) Original document image

(b) Foreground image, after median filter background removal

Figure 5.4: Detail of automatic document background removal algorithm, based on the median filter. Significantly, The median filter algorithm does not force the image to be bitonal, and is able to distinguish between the black of penstrokes (foreground) and the black border surrounding the page (background).

(a) Foreground image, thresholded

(b) Niblack-thresholded version of original

Figure 5.5: Image after background removal and manual thresholding, compared to original image thresholded with the Niblack locally-adaptive thresholding algorithm [Nib86].

in Section 7.10.

5.1.1.2 Significant new optimization of median filtering algorithm, by using a layer of indirection

The fastest algorithm for finding the median of those analyzed in [Dev98] can be sped up by exploiting spatial coherence to avoid re-sorting most of kernel array at adjacent kernel positions. A layer of indirection is simply added above the kernel array, so that pointers to the kernel values are swapped during the filtering operation, rather than swapping the values themselves. Because the only values changing position in the array between adjacent kernel positions are those entering the leading edge of the kernel (which take the place of those leaving the trailing edge of the kernel), most of the array can remain in the partially-sorted state that it was in after the previous median operation. This dramatically reduces the work that must be done in determining the median at each successive kernel position. Appendix A.1.3 describes the specific changes that must be made to the median filtering code to implement this optimization.

This simple optimization doubles the processing speed for the median filter, when applied to microfilm images of typical resolution (e.g. 1600x1200), with typical density of contents, and employing a typical median filter kernel size (e.g. radius = 17). Gains could be more or less for different parameterizations, and in particular for different document contents (the optimization reduces the information-theoretic complexity of the median filtering algorithm, and not the computational complexity).

5.1.2 Histogram-based median filtering

An median-filtering algorithm exists [TGG79] that is potentially even faster, as long as the values in the input are quantized over a small range of values (it scale approximately linearly with the kernel radius). This algorithm does not store the

kernel contents at all, but instead uses a histogram of the number of pixels within the kernel area with each graylevel. When the kernel moves, the number of pixels of each graylevel entering the leading edge of the kernel are added to the histogram, and the number of each graylevel leaving the trailing edge are subtracted. The median is moved from its previous position to balance the histogram areas on either side of the median.

The barrier to general application of histogram-based median filtering is that it only works when the data is quantized to a small range of values. Grayscale document images with 256-levels of intensity may be successfully processed with this algorithm.

This should in theory be the fastest algorithm, especially for large kernels and small numbers of input values (it has a lower computational complexity than [Dev98], which was used to implement median filtering), although it has not yet been tested for this registration system—the comparison of these two techniques is left as future work.

5.1.3 Padding image prior to taking DFT

At several stages in the presented document image registration system, the DFT of the document images is computed. The FFT is used to compute the DFT quickly (the complexity of the FFT is $\mathcal{O}(N \log_2 N)$, compared to the direct DFT computation's complexity of $\mathcal{O}(N^2)$ for the standard DFT). The FFT requires the input domain size to be a power of two, because it is defined as a divide-and-conquer algorithm. The image size is therefore rounded up to the next power of two before computing the FFT, and the image is padded with zeroes. The larger of the two image dimensions is used when rounding up, in order to make the image square, which greatly simplifies further processing.

Note that padding the size of the image up to the next power of two can actually

introduce intensity transitions above and beyond those already present when tiling the image at its original size (Figure 5.6). Full automatic background removal is needed to remove these edge transitions.

5.2 Choice of base image for registration

The recovery of rotation and shear angles for a document image allows the image to be axis-aligned without comparing it to other documents. The task of image registration, however, requires two or more images to be registered to one another. We arbitrarily choose the first image, de-rotate it / de-shear it, and then use this as our “reference image” to which the scale and translation of the other images are matched. The first image may not be the optimal choice for a scale and translation reference point, due to nonaffine distortion, low image quality, or non-ideal size. A better system would use a consensus-based approach to ensure the reference image was not badly chosen.

5.3 Recovery of transform parameters

5.3.1 Recovery of rotation angle

5.3.1.1 Finding document axis angles in the DFT

As shown in Figure 5.7, groups of parallel lines in an image act as “wavefronts,” and thus produce a pattern of “delta functions,” or peaks in magnitude in the frequency domain, corresponding to the harmonics of the wavefronts. (For comparison, see Figure 4.2, which depicts the Fourier transform of an image without strong linear features, except for slight intensity transitions where the image wraps at the edges.) This pattern of peaks radiates out in both directions in a straight line through the origin of the frequency domain, at right angles to each wavefront. Rotation of the image produces a corresponding rotation of these radial patterns.

(a) Original image

(b) Padding image size to next power of 2 produces additional intensity transitions

(c) Just inverting the image won't prevent intensity transitions at all original edges

(d) Background removal and grayscale inversion eliminates edge transitions

Figure 5.6: Padding images to the next power of 2 in size, prior to taking the FFT, can introduce intensity transitions at the edges; background removal fixes the problem.

Figure 5.7: Parallel lines in tabular document images act as “wavefronts,” whose harmonics produce radial ridges of strong magnitude in the DFT. The ridges in the DFT radiate in the direction of motion of the wavefronts, and thus lie at right angles to the table lines.

Because of conservation of power under the Fourier transform (known as Rayleigh’s Theorem), the most significant and/or periodic linear components of an image produce the strongest radial line integral in the power spectrum. These radial features pass through the origin of the frequency domain, and lie at right angles to the linear components in the image domain that produce them. The most strongly periodic components of a tabular document are the lines forming the table itself. This makes it easy to find the angle of rotation and shear of the table—we just look for the two largest peaks in the radial projection histogram of the power spectrum (Figure 5.12).

5.3.1.2 The radial projection histogram

The radial projection histogram is the histogram produced by taking a line integral through the origin of the power spectrum at small angular increments, at angles between 0 and 180°. It is produced as follows. (For implementation details, see the source code contained in Appendix A.1.4.)

The polar coordinates (ρ, θ) of each point (x, y) in the frequency domain are determined, relative to the (0,0) frequency origin. (Care must be taken when working with the frequency domain, as the FFT places (0,0) at the top left, meaning the frequency domain quadrants are all diagonally transposed.) For all pixels in the incircle of the 2D FFT (i.e. where $\rho < N/2$, and N is the size of each dimension of the FFT), the magnitude of point (x, y) is accumulated in a histogram in bin number $b = (\theta \bmod 180)/180 \times N_B$, where N_B is the total number of bins in the histogram. ($N_B = 4096$ bins were used to represent 180 degrees in this implementation, giving an internal resolution of 0.044°, i.e. an expected error of $\pm 0.022^\circ$ (Table 6.1).) Linear interpolation can be used in this “forward mapping” approach to distribute the magnitude to the two neighboring bins of the fractional bin position b , which somewhat reduces the effect of “rotationally-dependent aliasing,” and empirically decreases the

error rate for recovery of the rotation parameter by a significant factor.

Additionally, a small amount of smoothing is applied to the histogram (by 1D convolution with a Gaussian (with $\sigma = N_B * 0.015$), performed by multiplication in the frequency domain) before searching for the peaks representing the table axes, in order to remove small, meaningless spikes in the histogram. Note however that too much smoothing can cause ambient “background noise” in the radial projection histogram to end up being greater in magnitude than the true peaks, because the true peaks are so narrow that their total magnitude is comparatively small. Thus, a 1D version of median filter background removal is applied to the radial projection histogram before smoothing, to remove the ambient signal noise, by subtraction of the 1D median-filtered version of the signal (Figures 5.8, 5.12).

5.3.1.3 Detection of peaks corresponding to document table axes

Finally, the two largest peaks in the histogram that are separated by approximately 90° are found. This is accomplished by finding the angle θ_1 corresponding to the globally maximum value in the radial projection histogram, then finding the local maximum θ_2 within the angular range of $(\theta_1 + 90^\circ) \pm 45^\circ$ (modulo 180°). θ_1 and θ_2 are both at right angles to one of the document axes.

5.3.1.4 Exploitation of properties of tabular document images

This is a more robust approach to finding the angle of rotation than in general Fourier-Mellin image registration, where the 2D log-polar space is collapsed down into two 1D correlation problems, because are only looking for discrete peaks with tabular document images, and not trying to perform a full rotational correlation, so no significant information is lost in producing the radial projection histogram. In contrast, with traditional Fourier-Mellin image registration, spurious registrations can result from projecting the 2D rotation/scale problem down into two 1D problems. Thus, by

Figure 5.8: 1D analog to 2D automatic background removal—subtraction of median-filtered version of signal removes ambient noise, facilitating cross-correlation

making use of properties of the specific problem domain (strong radial frequency-space features exhibited by tabular document images), rotation is completely separable from scale for the special case of tabular document images, which is not the case with the standard Fourier-Mellin approach to image registration. We also get separate peaks in the 1D radial projection histogram for each axis, enabling the recovery of shear factor (described in Section 5.3.2), which is not possible with standard Fourier-Mellin registration.

In addition, the intransitivity of wrapping (tiling) and rotation (Figure 4.5) does not adversely affect the correct selection of rotation angle, since radial features, when wrapped at the edges of the frequency domain due to higher-order harmonics, do not significantly affect the radial projection histogram at any specific angle. Rotated-then-wrapped features get spread over a range of angles (from a viewpoint at the center of the frequency domain), meaning the true maxima in the histograms (corresponding to the table axes) are not likely to get disturbed. This is not true for general Fourier-Mellin image registration, because wrapped harmonics change the entire surface of the log-polar transform, potentially making registration more difficult, as a full correlation is performed (rather than just looking for the two largest peaks in the histogram).

As discussed in Section 5.1.1, by applying median filter background removal to tabular document images, apodization (usually performed using Blackman windowing [STM03]) is therefore not needed to eliminate Fourier “edge effects,” which can otherwise throw off rotational registration (Figure 4.6, 5.10), especially, for document image registration, when the true document axis angles are close to 0 or 90°. Thus, the entire image (not just an arbitrarily-chosen, windowed portion of it) can be used in the registration process. In this way, preprocessing with a median filter can dramatically improve accuracy and robustness of recovery of rotation and other

(a) Original document image

(b) FFT of original document image

(c) Original with background removed, then inverted

(d) FFT of original with background removed, then inverted

Figure 5.9: Automatic background removal eliminates axis-aligned “cross artifacts” caused by Fourier “edge effects.” See also Figure 4.6, 5.10.

(a) Input image (with sharp intensity transitions at the edges), and its DFT

(b) The radial projection histogram—note the spurious spikes at $0/180$ and 90° (the “cross artifact”)

Figure 5.10: “Edge Effects” in the DFT can cause false spikes in the radial projection histogram (compare with Figure 5.11)

(a) Image with background removed and then inverted, and its DFT

(b) The radial projection histogram of the DFT (the spurious spikes at 0/180 and 90° have mostly disappeared, making it easy to find the maxima corresponding to the document axes)

Figure 5.11: Automatic background removal eliminates “edge effects,” enabling detection of rotation and shear parameters (compare with Figure 5.10)

Figure 5.12: Radial projection histogram produced by applying 2D automatic background removal to the image, taking the DFT, computing the radial projection histogram, filtering the histogram using the 1D analog of background removal to remove ambient signal noise, and then smoothing the histogram with a standard Gaussian blur (c.f. Figure 5.11(b))

parameters over standard Fourier-Mellin registration. This is only possible because of specific properties of tabular document images.

5.3.2 Recovery of shear factor

Interestingly, by looking for the two most significant maxima in a radial projection histogram, the two image axes are decoupled, since an angle is returned for each axis. Thus we can separately determine the rotation angle (for the horizontal axis) and the shear angle (deviation of the vertical axis from the horizontal angle plus 90 degrees). Recovery of shear is something that is not possible with standard Fourier-Mellin image registration, because the log-polar space is nonlinearly distorted in a shear operation (Figure 4.7). It is made possible here by exploiting properties of this particular problem domain (specifically, that document tables produce two strong radial components in the frequency domain). Some care does need to be taken with sheared images, however, because if there is any non-affine perspective distortion, the radial pattern of maxima tends to spread (Figure 5.13), so the recovery of scale (Section 5.3.3) may not be able to be performed accurately by sampling in a single straight line. A way of approaching this problem is discussed in Section 7.6.

It should be noted too that a shear of the image results in a shear (at 90 degrees to the original shear direction) in the frequency domain (i.e. not just a rotation of one axis relative to the other), so the scale factor for the sheared axis should be corrected for by multiplying by the secant of the shear angle.

5.3.3 Recovery of scale factor

Scale recovery happens in log-polar coordinates, as in standard Fourier-Mellin image registration. However, because we are dealing with tabular document images, only the strongly linear components of the image are important for registration. Thus determination of scale need only involve conversion to log-polar form of the Fourier

Page 6

BURIALS in the Parish of *St. Peter, under Barmston* in the County of *West Yorkshire* in the year One thousand eight hundred and *Eighty*

Name	Abode	When buried	Age	By whom interred
<i>Isaac Bels</i>	<i>Remusworth</i>	<i>Feb 12th 37</i>		<i>W. G. Gatt</i>
<i>William Stappan</i>	<i>Hopel The Home</i>	<i>April 12th 94</i>		<i>W. G. Gatt</i>
<i>Edward Gough</i>	<i>Remusworth</i>	<i>April 12th 76</i>		<i>W. G. Gatt</i>
<i>Thomas Smith</i>	<i>Mill Hill</i>	<i>April 25th 100</i>		<i>W. G. Gatt</i>
<i>Thomas Gough</i>	<i>Newton</i>	<i>May 1st 67</i>		<i>W. G. Gatt</i>
<i>James Gough</i>	<i>Remusworth</i>	<i>June 18th 64</i>		<i>W. G. Gatt</i>
<i>John Melling</i>	<i>The Gate House Apper Hill</i>	<i>June 19th 10</i>		<i>W. G. Gatt</i>
<i>Elizabeth Bels</i>	<i>The Gate House Apper Hill</i>	<i>June 23rd 60</i>		<i>W. G. Gatt</i>

(a) An image and its DFT

Page 6

BURIALS in the Parish of *St. Peter, under Barmston* in the County of *West Yorkshire* in the year One thousand eight hundred and *Eighty*

Name	Abode	When buried	Age	By whom interred
<i>Isaac Bels</i>	<i>Remusworth</i>	<i>Feb 12th 37</i>		<i>W. G. Gatt</i>
<i>William Stappan</i>	<i>Hopel The Home</i>	<i>April 12th 94</i>		<i>W. G. Gatt</i>
<i>Edward Gough</i>	<i>Remusworth</i>	<i>April 12th 76</i>		<i>W. G. Gatt</i>
<i>Thomas Smith</i>	<i>Mill Hill</i>	<i>April 25th 100</i>		<i>W. G. Gatt</i>
<i>Thomas Gough</i>	<i>Newton</i>	<i>May 1st 67</i>		<i>W. G. Gatt</i>
<i>James Gough</i>	<i>Remusworth</i>	<i>June 18th 64</i>		<i>W. G. Gatt</i>
<i>John Melling</i>	<i>The Gate House Apper Hill</i>	<i>June 19th 10</i>		<i>W. G. Gatt</i>
<i>Elizabeth Bels</i>	<i>The Gate House Apper Hill</i>	<i>June 23rd 60</i>		<i>W. G. Gatt</i>

(b) The image after perspective distortion, with its DFT

Figure 5.13: The effect of perspective distortion on the frequency domain. The radial feature that extends vertically from the origin of the DFT (and which corresponds to the horizontal lines in the document) spreads when the horizontal lines converge to a perspective vanishing point. Each individual line in the spread radial feature is at right angles to one of the horizontal lines in the table.

magnitudes along the direction of radial projection maxima detected during determination of rotation and shear. While a radial projection is performed to detect these angles, we do not produce potentially spurious correlation offsets by collapsing to one dimension in this manner, as happens with the projection of log-polar space in the standard Fourier-Mellin transform. With tabular document images, the specific features that are being correlated are linear radial features with maximal total line-integral magnitude.

During the Mellin transform, i.e. while sampling at exponential intervals along the radial features in the frequency-domain that correspond to the document axes, the positive- and negative-frequency magnitudes (the magnitudes at 180° to each other, at the same radius from the origin) are summed. The resulting value is weighted by the size of the current exponential interval. The result is two log-radius histograms, one for each document axis (Figures 5.14(a), 5.15(a)). The logarithmic base b can be chosen so as to yield any desired accuracy in scale determination. A base of $b = 1.001$ was found to give ample accuracy for a range of images ($\pm 0.5\%$ without quadratic parameter fitting). Appendix A.1.5 gives source code for generation of the log-polar histogram.

The lowest frequency regions of the DFT, i.e. those closest to the origin, are subject to the greatest amount of aliasing during the Mellin transform, because they are sampled the most densely (possibly hundreds of times (with different weightings), depending on the logarithmic base used), and thus are the least spatially informative in the log- r histogram. This can create significant problems in determination of scale [DG01], particularly as the innermost pixels produce the majority of the log-radial histogram because of the exponentially-increasing radius step during sampling. Thus, the innermost 10% of the radius is skipped during sampling (i.e. the lowest 10% of frequencies are ignored; this is not a problem because the same information

(a) Log- r histograms for two images, obtained by sampling at exponential intervals along the radial feature in the DFT that corresponds to the x -axis of the document

(b) The correlation of the two log- r histograms—the offset from zero of the global maximum gives the log of the x -axis scale factor

Figure 5.14: Log- r histogram for x -axis for two images from the dataset given in Appendix A.2.1, and the corresponding correlation graph—used to determine the scale S_x mapping the x -axis of the second image to the first

(a) Log- r histograms for two images, obtained by sampling at exponential intervals along the radial feature in the DFT that corresponds to the y -axis of the document

(b) The correlation of the two log- r histograms—the offset from zero of the global maximum gives the log of the y -axis scale factor

Figure 5.15: Log- r histogram for y -axis for two images from the dataset given in Appendix A.2.1, and the corresponding correlation graph—used to determine the scale S_y mapping the y -axis of the second image to the first

is available in higher-order harmonics anyway). Because of high-magnitude/high-frequency noise, the outermost 30% of the radius is also not sampled. (Note that Stone and McGuire use an “ H -filter” to improve the SNR of the signal to get rid of high-frequency noise [STM03].) Thus the values r_{MIN} and r_{MAX} in Figures 5.14 and 5.15 are $N \times 0.1$ and $N \times 0.7$ respectively.

In order to reduce effects during later correlation of the log-radius histograms by having “clipped” the outer portions of the histograms in this way, 1D apodization is applied to the innermost and outermost log-radius values, by ramping the log-radius histogram down to zero at the edges. Note that this is not the same thing as apodization of the 2D image prior to taking the 2D DFT, to eliminate edge effects. In the case of the 1D log- r histogram, apodization is necessary and unavoidable. The log- r histogram doesn’t wrap at the edges, but the behavior of cross-correlation is undefined whenever the kernel and the signal don’t fully overlap (which almost always the case). We perform the correlation operation in the frequency domain for speed, which treats the signals as wrapping during the correlation operation. One-dimensional apodization is thus used as a preprocessing step to correlation, to reduce the effect of the assumption of the DFT that the domain is supposed to wrap at the edges.

A different problem also arises in this log-radial histogram which can throw off correlation: there can be a significant amount of background, “ambient” noise in the signal (particularly in higher frequencies), much the same as for the radial projection histogram used in the determination of rotation angle. The same method as was used in the rotation-recovery step is used to eliminate this, i.e. 1D background removal via the median filter is applied to eliminate the background noise (Figure 5.8).

We perform two scale detections, one for each document axis. This is done by performing simple 1D correlations of each corresponding axis of the two images being

registered (Figures 5.14(b), 5.15(b)). This yields a log-scale factor for both horizontal and vertical document axes, i.e. the log of the x - and y -scale needed to stretch one image to another. An image is chosen as the “base image” for registration for this purpose, and this is the image all others are scaled to (Section 5.2). (In this implementation, the closest document axis to horizontal was used as the horizontal axis; rotations of $\pm 45^\circ$ are handled by this approach. In practice, this does not present a problem for document image registration. See Section 7.5 for information as to how to get around this limitation.)

In $\log$ - r space with logarithmic base b , a translation by $\log_b(S)$ corresponds to a change of scale by a factor of S :

$$r' = r \cdot S \tag{5.1}$$

$$\Rightarrow \log_b(r') = \log_b(r) + \log_b(S). \tag{5.2}$$

Thus determination of scale reduces to a 1D correlation problem. This works much better than trying to dilate one histogram to match the other in non-log space, because resampling a histogram in the discrete frequency domain creates many aliasing and point-spread problems (e.g. “ δ -function” peaks get sub- or super-sampled, but should remain unchanged through the sampling operation).

Because separate scale factors are determined for both the x and y axes, full affine anisotropic scaling is supported during registration. This can correct for some document distortion due to scanning or camera angle artifacts, and is not possible with standard Fourier-Mellin registration, which only yields a single global scale factor.

5.3.4 Recovery of translation offset

Now that rotation, shear and scale have been recovered, the transformation matrix which inverts these transforms can be easily determined, and the transformed image is rendered into a temporary image buffer. The image in this buffer now needs

only to be aligned with the base image (Section 5.2) using 2-D cross-correlation. Plain 2D (cross-)correlation is also performed in the frequency domain, by pointwise multiplication with the complex conjugate. Thus the DFT of the temporary image buffer is computed, and the correlation is performed, followed by the inverse DFT. The correlation offset (the offset from the origin of the point in the correlation image which has the greatest value) is used as the translation offset. (Correlation offsets greater than $N/2$, where N is the dimension of the DFT, are interpreted as being negative, by subtracting N .) The correlation offset is only accurate down to 1 pixel, thus, to enable sub-pixel translational registration, a multidimensional application of quadratic parameter fitting needs to be applied to a neighborhood of the correlation offset to increase accuracy (Section 7.7).

Although this final correlation step is the simplest operation of the whole process, straight 2D correlation is not guaranteed to find the correct correlation offset without removal of the background and the black borders commonly found around a page (Section 5.1.1). Changes in intensity across the background of the image can adversely affect the choice of a correlation offset, because background pixels constitute the bulk of the pixels being correlated. We are not interested in registering relative to the the intensity changes in the image of the page the form was printed on; also, the table form may not always be printed in the same position relative to the edge of the page, and we are not interested in finding the best registration offset of the black around each page either. By applying median filter background removal, only foreground pixels (specifically, the document form) are involved in recovery of the translation offset.

If the document form contains writing or marks added after the form was printed, this acts as noise during correlation. In practice, document content does not affect the magnitudes in the 2D correlation by any significant amount, because so much

more of the actual document form is likely to line up between the documents than between the form in one image and the non-form content in the other image.

5.4 Pointwise mean of registered images

Once a batch of images are all registered together, they can be overlaid on top of one another (as if printed on transparent slides), and the mean value across all registered images that corresponds to each pixel in the output image can be used to give a “consensus” view of the entire batch of images (Figure 5.16(b)). This provides a quick window into the qualitative performance of the registration algorithm. Several examples of “pointwise mean” images, before and after registration, are shown in the results chapter (Chapter 6).

5.5 Pointwise median of registered images, for obtaining blank templates

Rather than taking the mean at each pixel across all registered images (i.e. the pointwise mean), the median value at each pixel can be used instead (i.e. giving the “pointwise median”). This gives an unbiased estimate of the average graylevel at each pixel in the registered set. By using the median rather than the mean, we are able to recover a fairly clean blank document template, based on a consensus of the images. Figure 5.16(c) shows a closeup of a pointwise median image.

Some handwriting is still present in the pointwise median image, in locations where more than half the images contained handwriting rather than background. By computing a pointwise *quantile* value, at a higher percentile than the median (i.e. higher than the 50th percentile), more of this handwriting can be excluded, at the expense of template line strength (Figures 5.17, 5.18).

NAME and Surname of each Person	RELATION to Head of Family	CON-DITION as to Marriage	AGE last Birthday of		Rank, Profession, or OCCUPATION
			Males	Females	
William Lane	Head	Mar.	47		Sea Dealer & Grocer
Fanny Lane	Wife	Mar.	1	50	Grocer's Wife
Joseph P. Pearce	Apprentice	Unm.	16		do apprentice
Eliza Case	Serv.	Unm.		28	do do

(a) A single registered image

NAME and Surname of each Person	RELATION to Head of Family	CON-DITION as to Marriage	AGE last Birthday of		Rank, Profession, or OCCUPATION
			Males	Females	

(b) Pointwise mean of 27 registered images

NAME and Surname of each Person	RELATION to Head of Family	CON-DITION as to Marriage	AGE last Birthday of		Rank, Profession, or OCCUPATION
			Males	Females	

(c) Pointwise median of 27 registered images

Figure 5.16: Detail of pointwise mean and pointwise median images, for comparison. The pointwise mean is useful in qualitative analysis of registration; the pointwise median produces a mostly-blank document form.

Figure 5.17: Pointwise quantile images for the 10th to the 90th percentile (the 50th percentile image is the pointwise median image). The n -th quantile is found for each pixel by sorting the intensity values for the set of N_I registered images at each pixel, and then taking the pixel value at position $N_I \times n/100$ in the sorted array. Thus $n\%$ of registered images are brighter at each point than the corresponding pixel in the n -th quantile image. Using a different quantile than the median can eliminate more of the handwriting, to give a cleaner blank form.

[illegible][illegible][illegible]

Figure 5.18: Closeup of 10th, 50th (median) and 90th percentile images

[illegible]

Figure 5.19: Result of applying background removal as a post-processing step to blank form obtained by pointwise median (compare to Figure 6.6)

5.6 Postprocessing

Median filter background removal can, of course, also be used as an additional postprocessing step, to remove the background from the blank document template obtained by computing the pointwise median (e.g. compare Figure 5.19 to Figure 6.6). This can significantly clean up the document form that is obtained, and may be useful if document templates are required for further processing.

One example of further processing that may be performed using the obtained blank document template is to subtract the template from each page of a set of documents, providing a simple way of finding hand-written or manually typed content in tabular documents. Subtraction of the template can also be used for predictive or residual

coding in image compression (discussed in Section 7.9).

5.7 Summary

In this chapter, a new registration algorithm for tabular document images was presented, which is based on the Fourier-Mellin transform. Preprocessing with the median filter to remove the background gets around some common and serious problems with Fourier-Mellin image registration. As with standard Fourier-Mellin image registration, we can remove translation from consideration while considering rotation and scale, but by exploiting properties specific to the problem domain of tabular document image registration, we can completely separate rotation and scale from one another too, allowing us to “peel off” one translation component at a time (even more so than is possible with standard Fourier-Mellin registration). We are also able to separate scale factors for horizontal and vertical axes, and handle shear transformations, neither of which is possible with standard Fourier-Mellin registration. A method of obtaining blank, “consensus” document template images was also presented.

Chapter 6

Results

6.1 Qualitative analysis

Figures 6.1–6.12 show the results of applying the registration process to several datasets of tabular images. The first and second image for each dataset show the result of pointwise-averaging all the images in the dataset together before and after registration respectively; the third image for each dataset shows a blank form obtained by computing the pointwise median of the registered images. The pointwise mean images show us that the document images are well-aligned in all datasets, as evidenced by the sharpness of the document template when performing a pointwise-average of all registered images. (A counterexample, where registration did not fare so well, is shown in Figure 6.13.) Figure 5.16(b) shows a closeup of the pointwise mean of a set of registered images.

In the PDF version of this thesis, Appendix A.3 contains an MPEG-2 file attachment which illustrates the registration process in action.

Figure 6.1: Dataset 1, Pointwise mean of 445 unregistered images

THIS IS A LEGAL RECORD AND WILL BE PERMANENTLY FILED

SEE OTHER SIDE

FILL IN WITH A TYPEWRITER OR WRITE FAIRLY WITH DARK INK. DO NOT USE GREEN NOR RED INK. LEGAL COPIES CANNOT BE MADE IF ENTRIES ARE DIM

ALL ITEMS MUST BE COMPLETE AND ACCURATE

IF NO DOCTOR WAS IN ATTENDANCE MEDICAL CERTIFICATION SHOULD BE COMPLETED BY THE LOCAL HEALTH OFFICER, OR CORONER IF HE IS A PHYSICIAN OR IF INQUEST WAS HELD

CERTIFICATE OF DEATH
STATE OF ALABAMA

1. NAME OF DEATH a. County		10. Mar. No.		3. USUAL RESIDENCE (Where deceased lived, if institution; residence before admission).	
4. City (If outside city or town limits, write RURAL) b. Town		11. Length of Stay c. (In this house)		5. State d. City (If outside city or town limits, write RURAL) e. Town	
6. Full Name of (If not in hospital or institution, give street address or location)		7. Street Address (If rural, give location)		8. Date of Death a. Year b. Month c. Day	
9. Name of DECEASED (Type or Print)		12. Sex		13. Color of Race	
14. Usual Occupation (Give kind of work done during most of working life, even if retired)		15. Kind of Business or Industry		16. Marital Status (Specify) Married, Never Married, Widowed, Divorced	
17. Father's Name		18. Social Security No.		19. Age in years (If Under 1 Year, last birthday) Months Days Hours Mins.	
20. Was Deceased Ever in U. S. Armed Forces? (If yes, give service and dates of service)		21. Social Security No.		22. CITIZENSHIP: NAME, ADDRESS	
23. Cause of Death Enter only one cause per line for (a), (b), and (c)		24. Disease or Condition Directly Leading to Death* (a)		25. Antecedent Causes Morbid conditions, if any, giving rise to the above cause (a) stating the underlying cause last.	
26. Date of Operation		27. Major Findings of Operation		28. Autopsy? Yes ☐ No ☐	
29. Accident Suicide, Homicide		30. Place of Injury (Home, Store, Factory, Street, Office, Road, etc.)		31. City, Town, or Rural (Country) (State)	
32. Time (Month) (Day) (Year) (Hour)		33. Injury Occurred While at Work ☐ Not While at Work ☐		34. How Did Injury Occur?	
35. I hereby certify that I attended the deceased from _____, 19____, to _____, 19____, and that death occurred at _____, from the cause and on the date stated above.					
36. SIGNATURE		37. Address		38. Date Signed	
39. Burial, Crema- tion, Removal (Specify)		40. Date		41. Name of Cemetery or Cremation	
42. Date Rec'd by Local Registrar's Signature		43. Address		44. Location (City, town, or county) (State)	

ALABAMA DEPARTMENT OF HEALTH AND PUBLIC AFFAIRS

Figure 6.2: Dataset 1, Pointwise mean of 445 registered images

THIS IS A
LEGAL
RECORD AND
WILL BE PER-
MANENTLY
FILED
IN THE
OTHER
SIDE
FILL IN
WITH A
TYPEWRITER
OR WHITE
PENCIL
WITH DARK
INK. DO NOT
USE GREEN
NOR RED INK.
LEGAL COPIES
CANNOT BE
MADE IF
ENTRIES
ARE DIM
ALL ITEMS
MUST BE
COMPLETE
AND
ACCURATE
IF NO DOCTOR
WAS IN
ATTENDANCE
MEDICAL CER-
TIFICATION
SHOULD BE
COMPLETED
BY THE LOCAL
HEALTH
OFFICER, OR
CORONER IF
HE IS A
PHYSICIAN OR
IF INQUEST
WAS HELD

CERTIFICATE OF DEATH

STATE OF ALABAMA

1. Name of Deceased a. Given		b. Last Name		2. USUAL RESIDENCE (Where deceased lived, if institution; resi- dence before admission).	
3. City (If outside city or town limits, write RURAL) or Town		d. Length of Stay (In this home)		5. State	
6. Full Name of (If not in hospital or institution, give street address or location)		7. Street Address		8. City (If outside city or town limits, write RURAL) or Town	
9. Name of (Type or Print)		10. (Alphabet)		11. Last or Death	
12. Sex		13. Color or Race		14. Married, Never Married, Widowed, Divorced (Specify)	
15a. Usual Occupation (Give kind of work done during most of working life, even if retired)		15b. Kind of Business or Industry		16. Birthplace (State and county or foreign country)	
17. Father's Name		18. Mother's Name		19. Age (In years, if Under 1 Year, if Under 24 Hrs. last birthday) Months Days Hours (Min.)	
20. Was Deceased Ever in U. S. Armed Forces? (If yes, give war or date(s) served)		21. Social Security No.		22. HYPOGLYCEMIA: NAME, AGE, ADDRESS	
23. Cause of Death Enter only one cause per line for (a), (b), and (c)		MEDICAL CERTIFICATION			
24. Disease or Condition Directly Leading to Death* (a)		Interval Between Onset and Death			
25. Antecedent Causes Morbid conditions, if any, giving rise to the above cause (a) stating the underlying cause last.		Due To (b)			
26. Other Significant Conditions Conditions contributing to death but not related to the disease or condition causing death.		Due To (c)			
27a. Date of Operation		27b. Major Findings of Operation		28. Autopsy? Yes () No ()	
29a. Accident Suicide, Homicide		29b. Place of Injury (Home, farm, school, street, office, shop, etc.)		29c. (City, town, or rural) (County) (State)	
30a. Time (Month) (Day) (Year) (Hour)		30b. Injury Occurred While at Not While at Work () at Work ()		30c. How Did Injury Occur?	
31. I hereby certify that I attended the deceased from _____, 19____, to _____, 19____ and that death occurred at _____, from the causes and on the date stated above.					
32a. SIGNATURE		32b. Address		32c. Date Signed	
33a. Burial, Crema- tion, Removal (Specify)		33b. Date		33c. Name of Cemetery or Crematorium	
33d. Location (City, town, or county) (State)		33e. Name of Doctor		33f. Address	
34. Date Rec'd by Local Registrar's Signature					
35. Name of Doctor					
36. Address					

ALABAMA DEPARTMENT OF HEALTH AND PUBLIC HEALTH SERVICE

Figure 6.3: Dataset 1, Pointwise median of 445 registered images

Figure 6.4: Dataset 2, Pointwise mean of 27 unregistered images

[illegible]

Figure 6.5: Dataset 2, Pointwise mean of 27 registered images

The undermentioned Houses are situate within the Boundaries of the									
Civil Parish (or Township) of	City or Municipal Borough of	Municipal Ward of	Parliamentary Borough of	Town or Village or Hamlet of	Urban Sanitary District of	Rural Sanitary District of	Ecclésiastical Parish or District of		
No. of Schedule	ROAD, STREET, &c. and No. or NAME of HOUSE	HOUSES No. of Tenements (B)	NAME and Surname of each Person	RELATION to Head of Family	CON- DITION as to Marriage	AGE last Birthday of Males Females	Rank, Profession, or OCCUPATION	WHERE BORN	(1) Deaf and Dumb (2) Blind (3) Lame or Stupid (4) Lunatic
1	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100		John Smith	Head	Married	45	Labourer	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100	
2	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100		John Smith	Head	Married	45	Labourer	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100	
3	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100		John Smith	Head	Married	45	Labourer	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100	
4	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100		John Smith	Head	Married	45	Labourer	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100	
5	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100		John Smith	Head	Married	45	Labourer	11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 5	

Figure 6.6: Dataset 2, Pointwise median of 27 registered images

Figure 6.7: Dataset 3, Pointwise mean of 23 unregistered images

[illegible]

Figure 6.8: Dataset 3, Pointwise mean of 23 registered images

[illegible]

Figure 6.9: Dataset 3, Pointwise median of 23 registered images

Figure 6.10: Dataset 4, Pointwise mean of 125 unregistered images

WRITE PLAINLY, WITH UNFADING INK—THIS IS A PERMANENT RECORD

Every item of information should be carefully supplied. AGE should be stated EXACTLY. PREVIOUS RESIDENCE should state CAUSE OF DEATH in plain terms, so that it may be properly classified. Exact statement of OCCUPATION is very important.

1 PLACE OF DEATH (Dist. No. _____) <small>(To be inserted by Registrar)</small>		Section No. _____ West Virginia State Department of Health DIVISION OF VITAL STATISTICS	
County _____		Register No. _____	
District _____ or _____		No. _____	
Town or City _____		(If death occurred in a hospital or institution give its NAME, location of street and number)	
2 FULL NAME			
PERSONAL AND STATISTICAL PARTICULARS			
3 SEX _____	4 COLOR OR RACE _____	5 SINGLE, MARRIED, WIDOWED, OR DIVORCED <small>(Write the word)</small>	
6 DATE OF BIRTH _____ <div style="display: flex; justify-content: space-between;">(Month)(Day)(Year)</div>			
7 AGE _____ yrs _____ mos _____ ds		IF LESS than 1 day, _____ hrs	
8 OCCUPATION (a) Trade, profession or particular kind of work _____ (b) General nature of industry, business, or establishment in which employed (or employer) _____			
9 BIRTHPLACE (State or country) _____			
PARENTS	10 NAME OF FATHER _____		
	11 BIRTHPLACE OF FATHER (State or country) _____		
	12 MAIDEN NAME OF MOTHER _____		
	13 BIRTHPLACE OF MOTHER (State or country) _____		
14 THE ABOVE IS TRUE TO THE BEST OF MY KNOWLEDGE. (Informant) _____ <div style="text-align: right;">(Address) _____</div>			
15 Filed _____, 192____ REGISTRAR _____			
MEDICAL CERTIFICATE OF DEATH			
16 DATE OF DEATH _____, 192____ <div style="display: flex; justify-content: space-between;">(Month)(Day)(Year)</div>			
17 I HEREBY CERTIFY, That I attended deceased _____ _____ 192____ to _____ 192____ that I last saw him alive on _____, 192____ and that death occurred, on the date stated above, at _____ m. The CAUSE OF DEATH was as follows: (Primary) _____ _____ (Duration) _____ yrs _____ mos _____ ds			
CONTRIBUTORY (Secondary) _____ (Duration) _____ yrs _____ mos _____ ds (Signed) _____ M. D. _____, 192____ (Address) _____			
NOTE: State the DISEASE CAUSING DEATH. Is death from VIOLENT CAUSE, State MANNER, or INJURY, and whether ACCIDENTAL, SUICIDE, or HOMICIDE.			
18 LENGTH OF RESIDENCE (FOR HOSPITALS, INSTITUTIONS, TRANSIENTS OR RECENT RESIDENTS) At place of death _____ yrs _____ mos _____ ds State _____ yrs _____ mos _____ ds Where was disease contracted, if not at place of death? _____ Former or usual residence _____			
19 PLACE OF BURIAL OR CREMATION _____		DATE OF BURIAL _____, 192____	
20 SIGNATURE _____			
ADDRESS _____			

Figure 6.11: Dataset 4, Pointwise mean of 125 registered images

WRITE PLAINLY, WITH UNFADING INK—THIS IS A PERMANENT RECORD

N.B.—Every item of information should be carefully supplied. AGE should be stated EXACTLY. PHYSICIAN'S RECORD STATE CAUSE OF DEATH in plain terms, so that it may be properly classified. Exact statement of OCCUPATION is very important.

1 PLACE OF DEATH (Dist. No.)
To be inserted by Registrar
 County.....
 District.....
 or
 Town or City..... No.

West Virginia State Department of Health
 DIVISION OF VITAL STATISTICS
CERTIFICATE OF DEATH
 Register No.

2 FULL NAME.....

PERSONAL AND STATISTICAL PARTICULARS

3 SEX	4 COLOR OR RACE	5 SINGLE, MARRIED, WIDOWED, OR DIVORCED (Write the word)
-----------------------	---------------------------------	---

6 DATE OF BIRTH

 (Month) (Day) (Year)

7 AGE
 yrs. mos. ds. or min?
 IF LESS than 1 day, hrs.

8 OCCUPATION
 (a) Trade, profession or particular kind of work
 (b) General nature of industry, business, or establishment in which employed (or employer)

9 BIRTHPLACE
 (State or country)

PARENTS

10 NAME OF FATHER
11 BIRTHPLACE OF FATHER (State or country)
12 MAIDEN NAME OF MOTHER
13 BIRTHPLACE OF MOTHER (State or country)

14 THE ABOVE IS TRUE TO THE BEST OF MY KNOWLEDGE
 (Informant).....
 (Address).....

15
 Filed....., 192.....
 REGISTRAR

MEDICAL CERTIFICATE

16 DATE OF DEATH
 192.....
 (month) (year) (date)

17 I HEREBY CERTIFY That I attended deceased from....., 192....., to....., 192.....
 that I last saw him alive on....., 192.....
 and that death occurred, on the date stated above, at.....
 The CAUSE OF DEATH was as follows:
 (Primary)

 (Secondary)

 (Signed)..... M. D.
 192..... (Address).....

NOTE: State the DISEASE CAUSING DEATH. In deaths from VIOLENT CAUSES, State MEANS OF INJURY; and whether ACCIDENTAL, SUICIDAL, or HOMICIDAL.

18 LENGTH OF RESIDENCE FOR HOSPITALS, INSTITUTIONS, TRANSIENTS OR RECENT RESIDENTS:
 At place of death..... yrs. mos. ds. State..... yrs. mos. ds.
 Where was disease contracted, if not at place of death?
 Former or usual residence.....

19 PLACE OF BURIAL.....
 (Address).....

20 SIGNATURES
 ADDRESS.....

Figure 6.12: Dataset 4, Pointwise median of 125 registered images

6.2 Quantitative analysis: Accuracy of algorithm

In order to quantitatively measure the accuracy of these methods, a representative tabular document image was selected, and subjected in turn to 100 instances of each transform distortion: rotation through various random (fractional) angles (between -45° and $+45^\circ$), scaling by random fractional amounts (between 0.5 and 2.0), and translation by random sub-pixel distances in both x and y ; this resulted in a total of 300 transformed images. Each transformed image was then registered to the original, and then the detected transformation parameter was compared to the randomly-chosen parameter that was used to distort the image in the first place. The RMS error over 100 trials was measured.

Results are shown in Table 6.1. The “Internal Resolution” column gives the internal expected error in resolution of each parameter, due to an internal accuracy of twice the same value (e.g. the internal resolution of translation recovery is 1 pixel in both axes, making the internal resolution accurate down to ± 0.5 pixel). This is the error rate to be expected before quadratic parameter fitting (see Section 7.2). “Measured Error” shows the error rate that was measured after quadratic parameter fitting. The decrease in error in every case shows that the simple addition of quadratic parameter fitting was able to improve parameter resolution by a significant factor for this dataset (approximately $20\times$ for rotation, $2\times$ for scale, and $3\times$ for translation—accuracy of recovery of rotation parameter is increased by a more substantial factor than for the other two, probably because the neighborhood of the true parameter value is more naturally quadratic in nature).

The results show that parameter recovery is possible at a very high resolution (tenths/hundredths/thousandths of a pixel/percent/degree). The measured accuracy calculation after quadratic parameter fitting, for example, implies that an image

Transformation	Internal Resolution	Measured Error
Rotation/Shear	$\pm 0.022^\circ$	$\pm 0.001^\circ$
Scale	$\pm 0.05\%$	$\pm 0.03\%$
Translation	± 0.5 pixels	± 0.16 pixels

Table 6.1: Measured error in transformation parameter recovery, based on application of 100 random transform parameters to a single selected image, for each transform component. Internal resolution of parameter-space search (before quadratic parameter fitting) is given, followed by measured mean absolute error of transformation parameter recovery (with quadratic parameter fitting)

would have to be 115,000 pixels wide to be misaligned by one pixel at the outside edges due to error in recovering rotation angle; a 3000-pixel wide image would have an average error in width of one pixel after scale correction; and a 3000-pixel wide image would have an average error during translational recovery of only .005% of its total width. The error rate measured is actually significantly smaller than the amount of warp that is likely to be observed (due to lens distortion, printing distortion, or scanner lag) in most document images.

Further accuracy tests should be performed, but it is likely that other document images would fare similarly, as long as they do not possess a significant amount of non-affine distortion, due to the robust nature of the parameter recovery steps.

6.3 Limitations of the presented registration algorithm

Several assumptions are made by the image registration algorithm. For registration to be effective, the document must be written on a tabular form (i.e. it must have strong, axis-aligned rectilinear features), and each page must be written on the same form (although the manually-added content of the form may differ).

Not all batches of images satisfy these criteria: specific problems that commonly occur are:

- Batches of seemingly-related images actually differ slightly, due to the form having been re-made at some point, and a mix of old and new forms having been used until the old ones all ran out
- Left and right pages sometimes differ in layout
- Lens distortion can warp the image in a nonaffine manner, particularly at short focal lengths
- Nonlinear stretches of small bands of the document can occur during printing or scanning.

For examples of the first of these problems, where the document form differs across different images, see Figures 6.2 and 6.11, where the printed text in some fields on the form is obviously not always in the same position. Having the document lines change between forms is worse than having the printed text change, because most transform parameter recovery steps (with the exception of translation) are based only on linear features in the document pages.

The last of the listed problems, nonlinear stretch in document images, is related to the first. This problem is illustrated in Figure 6.13. Rotational registration succeeded for this image, as well as horizontal registration (with respect to the original page orientation), but scale and vertical registration were not accurate, because the line spacing on some of the document forms actually changed partway down the form. The algorithm found a single affine transform that registers the images fairly well, but was not able to cope with the nonaffine distortion, so the overall registration is less than perfect.

Figure 6.13: Pointwise mean of 31 registered images; registration was ineffective due to inconsistent line spacing on some pages

6.4 Non-affine transforms

Often, document images have undergone non-affine transformations, such as perspective distortion or spherical distortion due to lens warping. Some, but not all, of this type of distortion may be corrected for using a shear and a separate scale factor for horizontal and vertical document axes, which is handled by Fourier-Mellin document image registration as described.

In order to test how well the presented document image registration system performed under perspective distortion, digital photographs were taken of a document from various angles and distances (Appendix A.2). The camera was not brought closer than a meter from the document, so that significant lens distortion was not produced. Registration was largely successful in finding an affine transformation that approximated the perspective distortion in each case. Notice the ghosting at the edges of the registered image, particularly in the second dataset, because significant perspective distortion was present in some of the images, and was not correctable by affine transformation alone.

6.5 Run time

In Table 6.2, time taken to run the document image registration system on images of several different sizes is shown. Times are in seconds per image, e.g. to register 100 images, the total time needs to be multiplied by 100. Times were measured on a P4-2.2GHz-HT system. Two concurrent threads were used for background removal and FFT operations, giving a 30% speedup on the single-processor Hyperthreading architecture. On a true multiprocessing system, the speedup would be closer to 100% when multithreading these tasks.

In general, background removal and translation recovery took the largest proportion of time for images of lower resolution. As the resolution increased, rotation and

(a) Dataset 1, Pointwise mean before and after document image registration

(b) Dataset 2, Pointwise mean before and after document image registration

Figure 6.14: Results of image registration of two datasets shown in Appendix A.2, which contain a non-affine transform component (perspective warp)

translation became about equally prominent (because the FFT portion of those steps began to dominate the calculation). In particular, notice how time taken to recover rotation and translation suddenly jumped between the 2040×1383 and 2144×1779 images. This is because the largest image dimension crossed a power-of-two boundary (i.e. 2048), and so the image had to be padded up to the next power of two in size (i.e. 4096) before taking the FFT (see Section 5.1.3), resulting in an increase in processing time by a factor of around 4.5. This can actually happen for translation but *not* rotation for image sizes slightly smaller than a power of two, since when the temporary image buffer (representing the image with scale and rotation corrected for) is rendered prior to translation recovery, the rotated/scaled version can be bigger than the original, and so the FFT for translation might have to use the next size of two in power (even though it was not needed during the recovery of the rotation parameter). Making sure images are scaled to around 20% less than the next power of two in their largest dimension provides a window for reasonable differences in rotation and scale between images in the dataset. Note though that a good image scaling algorithm must be used, otherwise aliasing artifacts can cause problems with correct recovery of image parameters from the DFT.

The radius for median filtering used during background removal was set to 1.5% of the largest image dimension, which typically works well for document images. However it is obvious from Table 6.2 that once an image gets to a certain size, the time taken to remove the background becomes exorbitant, and dominates the registration process. This is because as the image dimensions scale up linearly, the image area and the median filter kernel both scale up quadratically, so the median filtering process scales as the fourth power of the image dimensions, i.e. $\mathcal{O}(d^4)$ (because the kernel radius is proportional to the image dimension). Ways of optimizing median filtering, or optionally filtering lower-resolution versions of the image to extract the background,

are suggested in Section 7.10.

In general, different transform parameters all tend to scale differently to one another with the size of the image, due to differences in underlying algorithmic complexity. In the case of median filtering, time taken can also depend on the information-theoretical complexity of the image, due to the nature of the partial-sorting algorithm used.

It may seem that the presented algorithm is extremely slow compared to common deskew algorithms. However, this algorithm is doing substantially more than plain deskewing, by performing full and accurate registration of tabular document images under arbitrary affine transforms.

Image size	Processing Step	Run time (seconds/image)
701×475 (332,975 pixels)	Background removal (r=11)	0.48
	Rotation recovery	0.83
	Scale recovery	0.04
	Translation recovery	0.77
	Final transformation	0.05
	TOTAL	2.17
1020×846 (862,920 pixels)	Background removal (r=16)	2.44
	Rotation recovery	0.83
	Scale recovery	0.05
	Translation recovery	3.21
	Final transformation	0.12
	TOTAL	6.65
2040×1383 (2,821,320 pixels)	Background removal (r=31)	28.01
	Rotation recovery	3.04
	Scale recovery	0.05
	Translation recovery	3.26
	Final transformation	0.39
	TOTAL	34.74
2144×1779 (3,814,176 pixels)	Background removal (r=33)	49.11
	Rotation recovery	13.62
	Scale recovery	0.08
	Translation recovery	15.35
	Final transformation	0.53
	TOTAL	78.70

Table 6.2: Run time of document registration system on a P4-2.2GHz-HT system

Chapter 7

Discussion

7.1 Robustness

Most steps are very robust. One of the less-robust parts of the process is in fact the simplest part: the final step to recover the translational offset. As noted previously, it is very important that a good thresholding algorithm be used to separate foreground from background, and that large areas of black (foreground color) surrounding the page are removed prior to this correlation step. Failure to do so can result in badly registered images. With background removal in place, translational registration can still be thrown off if the table form is warped (Figure 6.13). Under purely-affine distortion, rotation and shear angles are virtually always recovered correctly. If there is any perspective distortion in the image, however, then the radial ridge of maxima in the frequency domain begins to spread, and although the skew angles will be chosen somewhere in the correct range, the linear $\log-r$ histogram used for scale calculation may be determined incorrectly. Thus significant perspective transforms are currently not as well handled as affine distortions. This can be improved by sampling for scale over an angular window of the chosen rotation angle, but this decreases accuracy in the case of images with purely affine distortion.

Scale recovery is also a difficult problem – the algorithm described works reasonably well, but there is room for improvement in the area of robustness, as it is not too difficult to throw off the recovery of the scale parameter. For example, most tabular documents have significant numbers of strong horizontal lines (for western scripts, which are written horizontally), but a few do not have strong vertical lines. Thus the vertical scale can probably be trusted more than the horizontal scale. If the value of the horizontal scale is unrealistic based on the value of the vertical scale, the vertical scale could be used for both. (In practice, just a couple of vertical lines in the document are usually sufficient to have the horizontal scale parameter recovered correctly, but the correlation result is not as clean, so the wrong correlation offset could more easily be obtained.)

This principle should be applied in general – if the transform parameters are wildly out of range, an error should be reported, or other candidate parameter values should be sought instead. (In particular, this would prevent huge amounts of memory being allocated in the case of a huge scale parameter being recovered, for example.)

7.2 Quadratic parameter fitting for improved registration accuracy

Quadratic parameter fitting was used in the neighborhood of all recovered parameters to better achieve sub-pixel accuracy in the results. A parabola is fit to each recovered parameter x_2 (with parameter certainty value y_2) and two neighboring points in parameter space x_1 and x_3 (with certainty values y_1 and y_3), and then the x -coordinate of the turning point of the parabola is used as a better estimate of the actual parameter value. Pseudocode for this is given in Appendix [A.1.1](#). This technique significantly improves the accuracy of recovery of a parameter in general, as long as the y -values in the immediate neighborhood of x_2 are approximately quadratic. (This is usually a fairly safe assumption, since only the immediate neighbors are used.)

Figure 7.1: Comparison of the Gaussian blur and median filters

Table 6.1 shows that quadratic parameter fitting significantly improved parameter recovery accuracy in registration of a representative image.

7.3 Considerations in median filtering and background removal

7.3.1 Properties of the median filter

7.3.1.1 General properties

The median filter has some very interesting nonlinear filtering properties. It preserves the low-frequency components of a signal, and also sharp edges in the signal, but not high-frequency “spikes.” In 2D, it preserves sharp grayscale transitions, but removes salt-and-pepper noise. This contrasts with the Gaussian blur filter, which is able to remove high-frequency salt-and-pepper noise, but removes the high-frequency components of grayscale transitions along edges in the process (Figure 7.1).

7.3.1.2 Median filter kernel shape

In general, 2D median filtering is performed using a square kernel. Different kernel shapes may be used to subtly alter the behavior of the median filter. For example, a “+”-shaped kernel can be used to preserve the corners of axis-aligned rectangles during median filtering.

In the presented work, a circular kernel of the specified radius was used in order to produce a filter with the least bias towards orientation-dependent properties.

7.3.2 Median filter kernel size

The median filter is particularly good at lifting handwritten or printed text off a page, because of the typical low spatial density of the foreground color comprising textual features. However the size of the kernel must be chosen appropriately.

Setting the median filter radius too small can cause problems: the median filter has to be big enough to fully remove the foreground features from the page, or foreground pixels will be marked as background, because they dominate the kernel area (c.f. Figure 7.2, 7.3(b)).

Setting the kernel size too big can cause additional problems however, because while the median filter preserves sharp grayscale transitions, it “rounds off” corners in 2D (Figure 7.1(c)). This can change the shape of features in the resulting image, meaning that when it is used for background removal, if the kernel is too big, the background will not be removed properly (Figure 7.3(c)). Also, median filtering with a large kernel size can potentially take a long time to compute.

It is important therefore to find the kernel size that maximizes foreground preservation, maximizes background removal, and minimizes running time. In practice, this parameter found to be fairly insensitive to change of image size and/or change in scale of the document contents. In general a median filter kernel of radius 17 pixels was found to suffice for full-page document images of up to 1200×1000 , and a radius of 22 was sufficient for images of up to 1600×1200 , i.e. the kernel radius needed to be around 1.5% of the size of the largest dimension of the image (although this obviously depends somewhat on the scale of features in the image). The kernel size can scale linearly with the image size without change in relative effect. Setting the kernel ra-

dius as a function of image size makes an assumption about pen stroke thickness and density relative to page size, but is a good first estimate. The optimal kernel radius could also be determined by looking at RMS error rates of the background-removed version of the image compared to the original, for various kernel sizes.

Median filtering is time-consuming for large kernels. The complexity of standard median filtering scales as $\mathcal{O}(r^2)$, where r is the kernel radius, because partial sorting to find the median (known as “Quick Select” or “K-Selection”, as given in Appendix A.1.3) is an $\mathcal{O}(N)$ operation. Some of the information theoretical complexity (in contrast with the algorithmic complexity) of partial sorting can be reduced by exploiting coherency between adjacent kernel positions, by leaving the kernel array in its partially-sorted state as far as possible (see Section 5.1.1.2). If a large kernel is needed, even with this optimization, median filtering can take a large proportion of the total processing time for document image registration. Image resolution may be reduced for the background removal step, or histogram-based filtering (which scales as $\mathcal{O}(r)$) may be used if very high-resolution images must be filtered.

7.3.3 Noise thresholding during background removal

When the median image is subtracted from the original image, if any pixel ends up negative, it is assumed to be light-colored noise (“salt noise”), and the difference is set to be zero rather than negative, because it is assumed that anything lighter than the background color has to be noise. The amount of “salt noise” in a neighborhood may be used as an estimate of the amount of dark-colored “pepper noise” in a neighborhood, so that everything within a noise tolerance is set to pure white.

7.4 Speed of Discrete Fourier Transform Calculation

The Discrete Fourier Transform (DFT) can be calculated the fastest using its divide-and-conquer implementation, the Fast Fourier Transform (FFT). For this reg-

But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
 But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
 But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
 But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.

(a) Original image, 800×177 pixels

But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
 But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
 But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
 But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.

(b) After median filtering, with kernel radius 7

But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
 But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
 But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
 But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.
But, soft! What light through yonder window breaks? It is the east, and Juliet is the sun.

(c) After median filtering, with kernel radius 15

Figure 7.2: Median filtering with different kernel sizes

DEPARTMENT OF COMMERCE AND LABOR—BUREAU OF THE CENSUS
THIRTEENTH CENSUS OF THE UNITED STATES: 1910—POPULATION

NAME OF INCORPORATED PLACE _____

ENUMERATED BY ME ON THE 16 DAY OF April, 1910.

RELATION.	PERSONAL DESCRIPTION.				NATIVITY.			OCCUPATION.
	Sex.	Color or race.	Age or last birthday.	Whether married, single, or divorced.	Place of birth of each person and parents of each person enumerated. If born in the United States, give the state or territory. If of foreign birth, give the country.	Place of birth of this person.	Place of birth of Father of this person.	
4	M.	W.	40		Virginia	Virginia	Virginia	22

(a) Original image, 1344×351 pixels

DEPARTMENT OF COMMERCE AND LABOR—BUREAU OF THE CENSUS
THIRTEENTH CENSUS OF THE UNITED STATES: 1910—POPULATION

NAME OF INCORPORATED PLACE _____

ENUMERATED BY ME ON THE 16 DAY OF April, 1910.

RELATION.	PERSONAL DESCRIPTION.				NATIVITY.			OCCUPATION.
	Sex.	Color or race.	Age or last birthday.	Whether married, single, or divorced.	Place of birth of each person and parents of each person enumerated. If born in the United States, give the state or territory. If of foreign birth, give the country.	Place of birth of this person.	Place of birth of Father of this person.	
4	M.	W.	40		Virginia	Virginia	Virginia	22

(b) After median filter background removal, with kernel radius 5 (took 0.42s)

DEPARTMENT OF COMMERCE AND LABOR—BUREAU OF THE CENSUS
THIRTEENTH CENSUS OF THE UNITED STATES: 1910—POPULATION

NAME OF INCORPORATED PLACE _____

ENUMERATED BY ME ON THE 16 DAY OF April, 1910.

RELATION.	PERSONAL DESCRIPTION.				NATIVITY.			OCCUPATION.
	Sex.	Color or race.	Age or last birthday.	Whether married, single, or divorced.	Place of birth of each person and parents of each person enumerated. If born in the United States, give the state or territory. If of foreign birth, give the country.	Place of birth of this person.	Place of birth of Father of this person.	
4	M.	W.	40		Virginia	Virginia	Virginia	22

(c) After median filter background removal, with kernel radius 100 (took 163.79s)

Figure 7.3: Effect of kernel size on document background removal: foreground pixels may be lost if kernel is too small; background may not be removed properly if kernel is too big. Time required also increases sharply with kernel radius (time measured on a P4-2.2GHz-HT system, with 2 threads).

istration system, the FFT was calculated using the freely-available library known as FFTW, “The Fastest Fourier Transform in the West” [FJ03]. This currently appears to be the fastest general-purpose implementation of the FFT algorithm available, and was measured to be several orders of magnitude faster than unoptimized FFT code (which was implemented directly from the algorithmic definition of the FFT algorithm).

7.5 Considerations in recovering rotation

It is assumed that images within a single dataset to be registered are within $\pm 45^\circ$ of one another in orientation, because the closest axis of the input image to horizontal is used as the horizontal axis of the document. If arbitrary rotations of documents are to be allowed by the registration system, all four possible axis-alignments (i.e. 90° rotations) must be performed to see which gives the best correlation with the base image.

7.6 Considerations in recovering scale

Also, the presented registration algorithm relies on the frequency-domain features corresponding to a document axis lying in a single straight line. This is not the case with perspectively-distorted document images (Figure 5.13). To get around this, the $\log-r$ histogram used in scale recovery needs to sample over a small angular range at each radius value to produce the correct histogram for scale parameter recovery. This unfortunately reduces accuracy of non-perspectively warped images due to the introduction of non-axis aligned noise, so it may be enabled if handling severe perspective distortion is a requirement. This particular feature was not needed to align the moderately warped images shown in Figure 6.14, because the perspective distortion was not bad enough to throw off registration. Producing the scale histograms by searching over a small angle range in this way makes the image registration behave a

little more like the standard Fourier-Mellin registration method, so is more generic, but it loses some of its selectivity in dealing with document images, hence the increase in range of distortions that may be handled, with a corresponding drop in accuracy.

7.7 Considerations in recovering translation

The correlation offset used in determination of translational offset is only accurate down to 1 pixel, and so sub-pixel registration is not possible for translational registration using this method. Thus, quadratic parameter fitting (Section 7.2) is applied to a neighborhood of the correlation offset to increase accuracy.

Quadratic parameter fitting is generally only used in 1 dimension, because a point and its two immediate neighbors in 1D fully specify a quadratic curve that passes through the three points. A higher-dimension curve would have to be used in 2 dimensions, since fitting a paraboloid to the neighborhood of the correlation offset is an overspecified problem (only four points are needed in total, but the correlation offset has eight 8-connected neighbors, for a total of nine points, or five even when just using 4-connected neighbors, leaving the problem still overspecified by one variable.)

To get around this problem, four 1-dimensional quadratic parameter fit operations are performed, using the vertical neighbors of the correlation offset, then the horizontal neighbors, then the neighbors on the leading diagonal, then the trailing diagonal. The weighted centroid of these four quadratic fit operations is the accepted correlation offset (see Appendix A.1.2 for source code). The vertical and horizontal operations weighted by a factor of 1, and the diagonal fit operations weighted by $1/\sqrt{2}$ (i.e. weighted by the inverse of their distance from the central point, because their greater distance increases the likelihood that the neighborhood is not quadratic). Using this weighted sum of quadratic parameter fit operations, significant subpixel accuracy is achieved for translation (Table 6.1). Comparison could be performed with

other techniques, such as a 2D least-squares approach (left as future work).

7.8 Resolution enhancement through frame-averaging

Note that it is possible to produce a pointwise mean (or median) of the set of registered images at a higher resolution than that of any of the input images, because of subpixel registration. This will have the effect of producing a much higher-resolution image of the document template than was originally available. This technique is known as “frame averaging” or “super resolution” [BS98], and is used in several image processing fields (e.g. by the FBI to read low-resolution license plates in a sequence of video frames, or in advertising to recover high-resolution posters from a video sequence).

7.9 Image registration for predictive image compression

Image registration may yield advantages in document image compression. We obtained a document template using the median filter, as in Figure 6.3, and subtracted this template from each registered image (Figure 7.4). The resulting image, lacking the template, was approximately 14% smaller when compressed using bzip2 than the original. In this way, image information which is shared by many pages of a document may be stored once and removed from each page which shares the information. This is an example of “predictive encoding” or “residual coding” for image compression. Further gains can be found by improving the registration of the template to each image. This can be performed by local snapping, by performing a second registration of the consensus template to each image, or by removal after template subtraction of small connected components.

Further compression may be possible by exploiting background removal to eliminate parts of the document with no semantic meaning, or to compress those parts at a much lower spatial resolution (Chapter 8). Some promising research has been

City or Municipal Borough of		Municipal Ward of	Parliamentary
Salisbury		St Edmund	Salisbury
HOUSES		NAME and Surname of each Person	RELATION to Head of Family
Inhabited	Uninhabited (U), or Building (B)		
1		James Parsons	Head
		Annie	Wife
		Alfred	Son
		Edith	Daughter
		Ernest	Son
		Elizabeth Ann	Daughter

(a) Closeup of a single registered image

Salisbury		St Edmund	Salisbury
1		James Parsons	Head
		Annie	Wife
		Alfred	Son
		Edith	Daughter
		Ernest	Son
		Elizabeth Ann	Daughter

(b) Closeup of same image after subtraction of document template

Figure 7.4: Detail of registered document before and after subtraction of document template. The image shown is 14% smaller after template subtraction, using a bit-stream compressor (illustrates “predictive encoding” by image registration)

completed in this field by Huang et. al., who separate a document image into foreground and background layers using the “top hat transform” (i.e. morphological closing) [HWW97, HWW98].

It should be noted that resampling a document image under an affine transform (as is done during registration) actually raises the entropy of the image, making it less compressible. Subtraction of the median image during background removal can also increase the entropy of the foreground image. Adjustment of contrast, thresholding, adaptive quantization [Pop01, KB03] etc. can be used to reduce the entropy again after resampling or background removal.

7.10 Optimization

All but the smallest correlation operations can be performed faster by pointwise multiplication of the Fourier transform of one image with the complex conjugate of the Fourier transform of the other image, followed by finding the largest maximum in the inverse Fourier transform of the result. Fourier correlation was used in the implementation of this algorithm (see Section 4.1.4).

In addition, the 2D correlation operation used to recover translation in the final step may be performed as two separate 1-D correlations of the projection histograms of the images (which is much faster and requires much less memory, but has a greater chance of finding the wrong correlation offset, because it “collapses” the problem dimensionality).

Both median filtering and the Fourier transform algorithm are highly parallelizable, allowing for significant speedups in hardware or on multiprocessing architectures. Both algorithms were multithreaded in the implementation of this system, providing significant speedups on symmetric multiprocessing and hyperthreading architectures. Multiprocessing support that is even more scalable may be implemented

by processing different images on different machines, since the registration process has very few processing dependencies between images. MPI [MPI97] (or some similar clustering system) would provide the necessary framework for scaling the algorithm for a clustered or supercomputing environment.

Many operations may be performed at a lower resolution for speed. For example, the median filter background removal process can be accomplished by working on a low-resolution version of the background with a smaller median kernel, and the result can then be scaled up using bilinear or bicubic interpolation before it is subtracted from the full-resolution image. The drawback of this approach is that parameter recovery will drop in accuracy, and background removal will leave more of the background behind when the low-res version is scaled up before subtraction from the original document image. It may be possible to use low-resolution images as a “first cut” at registration, however, and successively refine the registration within a small neighborhood of the recovered parameters, or it may be that extremely high precision is not required (as with manual extraction of fields from document forms), as long as the new registration is better than the original alignment. Note that a good image scaling algorithm must be used for parameter recovery (otherwise aliasing artifacts can cause problems with correct recovery of image parameters from the DFT), although background removal will not suffer as much from this problem at lower resolutions.

The speed of median filtering may also be improved by using a level of indirection, so that only a few items need sorting again when the median filter kernel moves. (Those pixels which move onto the right side of the kernel replace the values that were lost from the left side of the kernel.) As long as the pivot element is chosen well in the fast median algorithm to be close to the true median (i.e. using the previous median as an estimate), then the algorithm will run more quickly than if the kernel elements

start unsorted each time. This was implemented, extending Devillard’s fastest technique [Dev98], and the optimization provided a speedup of approximately a factor of two for typical microfilm images and kernel sizes, as discussed in Section 5.1.1.2 and Appendix A.1.3.

For particularly large median filter kernels (e.g. if filtering high-resolution images at full resolution is required), histogram-based median filters would be by far the fastest approach (Section 5.1.2). For images of typical resolution and content found on microfilm, kernel sizes of radius 15-25 pixels were generally needed for background removal, which can result in fairly long preprocessing times, and it is probable that histogram-based median filtering will provide a significant speedup for this kernel size, even though the most significant differences are likely to be seen for even larger kernel radius values (e.g. 30 or more). Implementing this optimization is left as future work.

Chapter 8

Future Work

8.1 Perspective transform correction

The current algorithm does a reasonable job of registering documents under some degree of non-affine transformation, but since many microfilm documents contain slight perspective or lens distortion, extending this technique to correct for these distortions would render it useful in a wider variety of situations.

8.2 Consensus-based registration

Currently, registration is a pairwise process, with one image chosen as the base image for registration. (This is, arbitrarily, the first image of the set.) A consensus-based approach could be investigated to ensure the base image was not badly chosen.

8.3 Median filter parameter selection

Methods for quickly and automatically determining optimal median filter parameters for a specific image or set of images would be immensely useful, to ensure high image qualities and low running times. Histogram-based median filtering (Section 5.1.2) should also be implemented, as it would theoretically provide significant speedup for large kernel sizes (e.g. for a kernel radius of 30 or more).

8.4 Improvement of translational registration

Subpixel translational correlation is slightly more difficult than the other transform components to get right, because the problem is overspecified (Section 7.7). In Appendix A.1.2, a method is presented of performing 2D quadratic parameter fitting, by finding the weighted centroid of four 1D parameter fit operations. The presented technique works well (it was shown earlier that the RMS error in translational registration over 100 different translation offsets was 0.16 pixels), but For completeness, comparison could be performed with other subpixel registration techniques, such as a 2D least-squares approach.

8.5 Document image compression, exploiting image registration

Further research should also be performed into the use of image registration for document image compression, including improving locally-adaptive registration (and cleanup of small artifacts left after subtraction of the template, which were right next to the template before subtraction, and so are probably noise).

8.6 Document image compression, exploiting background removal and/or layered coding

An interesting area for future research is the possibility for image compression afforded by the presented automatic background removal algorithm. This has been investigated to some degree by Huang et. al. using the “top hat transform” (involving morphological closing operations) [HWW97, HWW98]. The median filter should be compared to the top hat transform for background removal, and it is likely either method will work well, but one or the other may have properties that make it more amenable to processing document images.

The background of an image makes up the majority of its uncompressed size, and

a significant part of its compressed size, yet it is of little informational value. The median filter-based automatic background removal algorithm presented in this paper allows the background to be removed reliably without discreet foreground/background segmentation (as is employed in DjVu/JBIG2), and the “holes” in the resulting background image (where there were foreground pixels) are filled in automatically by the median filter, creating a continuous background image and a non-binary (but much cleaner) foreground image. This could enable a whole new class of image compression algorithms to be developed, which store the background at a much lower resolution than the foreground, load it progressively/selectively, or even discard it entirely. (Note that subtraction of the median background image can actually increase the entropy of the foreground image, by creating an image with grayscales that never existed in the original image. Adaptive quantization to a small number of grayscales [Pop01, KB03] could eliminate this problem, while still preserving most of the useful information contained in the grayscales of the foreground image.)

8.7 Open question

An open question is also posed here, which will hopefully be answered through future investigation. Figures 8.1 and 8.2 show examples of strong frequency-space features which radiate from the edges and/or corners of the DFT, and not from the origin. They run parallel to features radiating from the origin, but are not the result of “wrapped” features which pass through the origin. The exact reason for these features is not known at time of writing, but it is possible that they could be exploited to make parameter recovery even more robust, by building composite histograms using the magnitudes that radiate from the center of the DFT, as well as from the center of each edge and the corners.

(Note that the bottom example in Figure 8.2 also illustrates perspective distortion,

and has a DFT that looks like a “grid of dots”, because the Periodic Table image actually has non-axis-aligned periodic properties, e.g. when viewed at 45° .)

The author would be interested to hear from anyone who can provide a theoretical explanation for these non-origin features. A check of value US\$1.62 (approximately the value of the *Golden Ratio*) will be sent to the first person to contact the author with a plausible explanation.

Figure 8.1: Examples of mystery frequency-domain features, which do not radiate from the origin. The interpretation of these features is posed as an open question in Section 8.7. (The source image for this DFT is given in the top example in Figure 8.2.)

Figure 8.2: Examples of images with mystery frequency-domain features (c.f. Figure 8.1)

Chapter 9

Conclusions

We have presented an algorithm for fast registration of tabular document images, based on the Fourier-Mellin transform. This algorithm is robust and accurate in aligning tabular documents under affine transformations, and does a reasonable job of aligning tabular documents under non-affine transformations commonly found in document photographs. A novel automatic background-removal algorithm was introduced, based on the median filter, that enables many limitations of the Fourier-Mellin transform to be overcome for document image registration (such as the need for apodization, or “Blackman windowing”), and allowing for more accurate entire-document registration. Fourier-Mellin image registration was adapted to exploit properties specific to the problem domain of tabular image registration, and to deal with all components of the affine transform in a uniform way, by working in the frequency domain. The Fourier-Mellin transform was also extended to handle shear, and anisotropic scaling (with a separate scale factor for horizontal and vertical document axes). A significant new optimization of the median filter was presented. Using a set of registered images, a novel method for generation of blank forms was presented, and application of this technique to document image compression were discussed.

Appendix

A.1 Source code

Source code is given here for several core algorithms employed in the presented document image registration system.

A.1.1 C source code for 1D quadratic parameter fitting

The following code illustrates “quadratic parameter fitting” (or “parabolic neighborhood fitting”) for improvement of parameter estimates, as described in Section 7.2.

```
// Fit a parabola to three points, and return the x coord of the turning  
// point (point 2 is the central point, points 1 and 3 are its neighbors)  
  
double parabolic_fit(double x1, double y1,  
                    double x2, double y2,  
                    double x3, double y3) {  
  
    double a = (y3 - y2) / ((x3 - x2) * (x3 - x1)) -  
              (y1 - y2) / ((x1 - x2) * (x3 - x1));  
  
    double b = (y1 - y2 + a * (x2 * x2 - x1 * x1)) / (x1 - x2);  
  
    double xmin = x2;      // Just use central point if parabola is flat  
    if (fabs(a) > EPSILON)  
        xmin = -b / (2 * a); // [avoids division by zero]  
  
    // Use the following to calculate the y-value at the turning point  
    // of the parabola:  
    //  
    // double c = y1 - a * x1 * x1 - b * x1;  
    // double ymin = a * xmin * xmin + b * xmin + c;  
  
    return xmin;  
}
```

A.1.2 C source code for 2D quadratic parameter fitting

The following code illustrates how 1D quadratic parameter fitting was extended to handle the over-specified problem that exists in a 2D neighborhood, as used to recover the translation offset at sub-pixel resolution (see Section 7.7).

```
// Do a 2D parabolic fit to give sub-pixel accuracy for translational
// registration, by performing four 1D parabolic fits through the closest
// integer correlation offset: horizontally, vertically, and on the leading
// and trailing diagonals. Take the weighted centroid of all four to get
// the true correlation offset.

// off_x and off_y are the one-pixel-accurate correlation offsets recovered
// by correlation.

// x1d and y1d are x and y values for the 1D quadratic fit function
double y1d[9] = { // Get magnitudes at (off_x, off_y) and 8 neighbors
    dft_mag(dft, off_x - 1, off_y - 1),
    dft_mag(dft, off_x, off_y - 1),
    dft_mag(dft, off_x + 1, off_y - 1),
    dft_mag(dft, off_x - 1, off_y),
    dft_mag(dft, off_x, off_y),
    dft_mag(dft, off_x + 1, off_y),
    dft_mag(dft, off_x - 1, off_y + 1),
    dft_mag(dft, off_x, off_y + 1),
    dft_mag(dft, off_x + 1, off_y + 1)
}

// Sum contributions to centroid of each quadratic fit
double x1d_tot = 0.0, y1d_tot = 0.0, x1d;

// Parabolic fit in horiz direction through correlation maximum
x1d = parabolic_fit(-1, y1d[3], 0, y1d[4], 1, y1d[5]);
x1d_tot += x1d;

// Parabolic fit in horiz direction through correlation maximum
x1d = parabolic_fit(-1, y1d[1], 0, y1d[4], 1, y1d[7]);
y1d_tot += x1d; // [x1d is x in parabola space, but y in correlation space]

// Weight contributions of diagonal by the inverse of their distance
#define RT2_OV_2 0.7071067811865475244 // sqrt(2)/2 (= 1/sqrt(2))

// Parabolic fit in leading diagonal direction through correlation maximum
x1d = parabolic_fit(-1, y1d[0], 0, y1d[4], 1, y1d[8]);
x1d_tot += x1d * RT2_OV_2;
y1d_tot += x1d * RT2_OV_2;

// Parabolic fit in leading diagonal direction through correlation maximum
x1d = parabolic_fit(-1, y1d[2], 0, y1d[4], 1, y1d[6]);
x1d_tot -= x1d * RT2_OV_2;
y1d_tot += x1d * RT2_OV_2;

// Take centroid of all parabolic fits, weighting diagonals by RT2_OV_2;
// make relative to correlation coords by adding off_x, off_y
double subpix_off_x = off_x + x1d_tot / (2.0 + 2.0 * RT2_OV_2);
double subpix_off_y = off_y + y1d_tot / (2.0 + 2.0 * RT2_OV_2);
```

A.1.3 C source code for fast median filter

The code on the following page is the original median algorithm described in [Dev98] for quickly finding the median of a dataset, and forms the basis of median filtering for automatic background removal (Section 5.1.1).

The median algorithm was extended beyond what is shown in the source code shown here (as described in Section 5.1.1.2), by adding a layer of indirection to preserve previous partial sorts of the kernel, exploiting spatial coherency to increase filtering speed by decreasing the information-theoretic complexity of the problem. The optimization involves passing in a pointer array `p_arr[]` with elements of type `(elem_type *)`, which point to each element of the kernel array in turn, rather than passing the actual kernel array `arr[]` as in the original source code. References to `ELEM_SWAP(arr[i],arr[j])` are then replaced with `P_ELEM_SWAP(p_arr[i],p_arr[b])`, defined as

```
#define P_ELEM_SWAP(pa,pb) { register elem_type *t=(pa);\n                             (pa)=(pb);(pb)=t; }
```

This is the same as `ELEM_SWAP(a,b)`, but swaps pointers to elements rather than elements themselves. The other change that is required is to add a dereference operation to element access operations which are not involved in swapping, e.g. replacing

```
if ( arr[low] > arr[high] )
```

with

```
if ( *p_arr[low] > *p_arr[high] )
```

(since we need to compare the kernel element values, and not the pointer values).

The routine that calls `find_median()` needs to only replace the kernel elements that fall off the trailing edge of the kernel with the elements that enter the leading

edge of the kernel, but leaves all `p_arr[]` pointers in their previous sort order between subsequent calls to `find_median()` across a row of the image.

```
// This Quickselect routine is based on the algorithm described in
// "Numerical recipes in C", Second Edition,
// Cambridge University Press, 1992, Section 8.5, ISBN 0-521-43108-5
// This code by Nicolas Devillard - 1998. Public domain.

typedef int elem_type;
#define ELEMSWAP(a,b) { register elem_type t=(a);(a)=(b);(b)=t; }

elem_type find_median(elem_type arr[], int n) {
    int low, high;
    int median;
    int middle, ll, hh;

    low = 0; high = n-1; median = (low + high) / 2;
    for (;;) {
        if (high <= low) // One element only
            return arr[median];

        if (high == low + 1) { // Two elements only
            if (arr[low] > arr[high])
                ELEMSWAP(arr[low], arr[high]);
            return arr[median];
        }

        // Find median of low, middle and high items; swap into position low
        middle = (low + high) / 2;
        if (arr[middle] > arr[high]) ELEMSWAP(arr[middle], arr[high]);
        if (arr[low] > arr[high]) ELEMSWAP(arr[low], arr[high]);
        if (arr[middle] > arr[low]) ELEMSWAP(arr[middle], arr[low]);

        // Swap low item (now in position middle) into position (low+1)
        ELEMSWAP(arr[middle], arr[low+1]);

        // Nibble from each end towards middle, swapping items when stuck
        ll = low + 1;
        hh = high;
        for (;;) {
            do ll++; while (arr[low] > arr[ll]);
            do hh--; while (arr[hh] > arr[low]);

            if (hh < ll)
                break;

            ELEMSWAP(arr[ll], arr[hh]);
        }

        // Swap middle item (in position low) back into correct position
        ELEMSWAP(arr[low], arr[hh]);

        // Re-set active partition
        if (hh <= median)
            low = ll;
        if (hh >= median)
            high = hh - 1;
    }
}
```

A.1.4 C source code for production of radial projection histograms

Source code for production of radial projection histograms (Section 5.3.1) is given below. For speed, “Forward mapping” is used rather than “backward mapping”, so the histogram must be smoothed after it is produced to reduce aliasing.

```
double bin[NUM_BINS];
for (int i = 0; i < NUM_BINS; i++)
    bin[i] = 0.0;

// Add magnitudes of each point in DFT into histogram
for (int y = -radius; y <= radius; y++) { // Radius == N/2
    int y2 = (y < 0 ? y + dimy : y); // Re-center DFT
    for (int x = -radius; x <= radius; x++) {
        int x2 = (x < 0 ? x + dimx : x);
        double r = hypot((double) y, (double) x);

        // Ignore lowest frequencies, and frequencies outside the incircle of radius
        // N/2 that just fits into the DFT
        if (r > min_radius && r < max_radius) {
            double ang = atan2((double) -y, (double) x);
            double bin_num_f = (ang / M_PI) * (double) NUM_BINS;
            int bin_num_i = (int) bin_num_f;
            double wt2 = bin_num_f - (double) bin_num_i;

            // Wrap at 0 and 180 deg
            int bin_num1 = (bin_num_i + NUM_BINS * 2) % NUM_BINS;
            int bin_num2 = (bin_num1 + 1) % NUM_BINS;

            // Add the magnitude at (x2,y2) into the two bins neighboring the desired
            // fractional bin position, using linear interpolation to weight the
            // contributions in order to reduce rotationally-dependent aliasing
            double mag = cabs(cimage[y2 * dimx + x2]);
            bin[bin_num1] += mag * (1.0 - wt2);
            bin[bin_num2] += mag * wt2;
        }
    }
}
// Bin values are subsequently smoothed using convolution
```

A.1.5 C source code for log-polar histogram computation

The following source code produces the log-polar histogram discussed in Section 5.3.3.

```
// Create a log-polar histogram (log rho, theta) by sampling at exponential
// intervals at a given angle from the origin of Fourier space.
// Assumes FFT origin is at center of fft array.
double get_log_polar_histogram(complex *fft, int fft_dimxy, double theta,
                               double **p_log_hist, int *p_n_log_hist) {
    double ctr = (double) fft_dimxy * 0.5, c = cos(theta), s = -sin(theta);
    double r = (double) (fft_dimxy - 1) * 0.5 - 1.0;
    // Rounding up hist size to next power of two, and adjust log base
    double log_max = log(r) / log(SCALE_LOG_BASE);
    int n_log_hist = *p_n_log_hist = next_power_of_two(log_max);
    double actual_log_base = exp(log(r) / (double) (n_log_hist + 1));
    double *log_hist = *p_log_hist = (double *) malloc(n_log_hist * sizeof(double));
    // Ignore lowest frequencies
    int min_log_radius = (int) ceil(log(MIN_SCALE_FREQ_RADIUS) / log(actual_log_base));

    // Convert histogram along line at angle theta to "log rho" histogram
    double prev_radius = 0.0;
    for (int i = 0; i < n_log_hist; i++) {
        if (i < min_log_radius) {
            log_hist[i] = 0.0; // Ignore lowest frequencies
            if (i == min_log_radius - 1)
                prev_radius = pow(actual_log_base, (double) i);
        } else {
            double radius = pow(actual_log_base, (double) i); // Exponential intervals
            double dx = radius * c, dy = radius * s;
            double x1 = ctr + dx, y1 = ctr + dy, x2 = ctr - dx, y2 = ctr - dy;
            int ix1 = (int) x1, iy1 = (int) y1, ix2 = (int) x2, iy2 = (int) y2;
            double pix00, pix01, pix10, pix11, pix02, pix20, pix;
            double wt_x = x1 - (double) ix1, wt_y = y1 - (double) iy1;

            // Get bilinear interpolation for positive frequency magnitude at (+dx,+dy)
            pix00 = cabs(fft[iy1 * fft_dimxy + ix1]);
            pix10 = cabs(fft[iy1 * fft_dimxy + (ix1 + 1)]);
            pix01 = cabs(fft[(iy1 + 1) * fft_dimxy + ix1]);
            pix11 = cabs(fft[(iy1 + 1) * fft_dimxy + (ix1 + 1)]);
            pix02 = pix00 * (1.0 - wt_x) + pix10 * wt_x;
            pix20 = pix01 * (1.0 - wt_x) + pix11 * wt_x;
            pix = pix02 * (1.0 - wt_y) + pix20 * wt_y;

            // Get bilinear interpolation for negative frequency magnitude at (-dx,-dy)
            pix00 = cabs(fft[iy2 * fft_dimxy + ix2]);
            pix10 = cabs(fft[iy2 * fft_dimxy + (ix2 + 1)]);
            pix01 = cabs(fft[(iy2 + 1) * fft_dimxy + ix2]);
            pix11 = cabs(fft[(iy2 + 1) * fft_dimxy + (ix2 + 1)]);
            pix02 = pix00 * wt_x + pix10 * (1.0 - wt_x);
            pix20 = pix01 * wt_x + pix11 * (1.0 - wt_x);
            pix += pix02 * wt_y + pix20 * (1.0 - wt_y); // Add +ve and -ve frequency mags

            pix *= (radius - prev_radius); // Weight magnitudes by interval size
            log_hist[i] = pix; // Store in log histogram
            prev_radius = radius; // Remember interval size
        }
    }
    normalize_hist_area(log_hist, min_log_radius, n_log_hist);
    return actual_log_base; // Return the log base used (hist is returned in params)
}
```


A.2 Perspective distortion test datasets

The following images were used in Section 6.4 to test registration of document images under perspective distortion. (This document was particularly effective for frequency-domain analysis because of its highly periodic nature!)

A.2.1 Perspective Dataset 1 – camera up to 15° from the surface normal

A.2.2 Perspective Dataset 2 – camera up to 30° from the surface normal

A.3 MPEG Animation of Registration

In the PDF version of this document, the image below represents an attached MPEG-2 movie which shows document image registration in action. Transform parameters (rotation, scale, translation and shear) are interpolated gradually to move each of the images to its registered position. To view the movie, right-click on the image from within Adobe Acrobat Reader v6.0 or above, and choose “Save Embedded File to Disk”. You will need a media player capable of MPEG-2 playback (e.g. Xine, MPlayer, or a recent version of Windows Media Player) to play the movie. Other free PDF viewers may be capable of extracting the file, as long as they support extraction of file attachments as text annotations (this is part of the PDF-1.3 specification).

Bibliography

- [AFPS96] Adnan Amin, Stephen Fischer, Tony Parkinson, and Ricky Shiu. [Fast algorithm for skew detection](#). In *Proceedings, Symposium on Electronic Imaging*. IS&T/SPIE (The International Society for Optical Engineering), 1996. 10
- [Bai87] H. S. Baird. The skew angle of printed documents. In *Proceedings, Society of Photographic Scientists and Engineers*, volume 40, pages 21–24, 1987. 10
- [BHQ⁺04] William Barrett, Luke Hutchison, Dallan Quass, Heath Nielson, and Douglas Kennard. [Digital Mountain: From Granite Archive to Global Access](#). In *Proceedings, Document Image Analysis for Libraries*. IEEE, 2004. 1
- [Bro92] Lisa G. Brown. [A survey of image registration techniques](#). *ACM Computing Surveys*, 24(4):325–376, December 1992. 1, 13
- [BS98] Sean Borman and Robert Stevenson. [Spatial Resolution Enhancement of Low-Resolution Image Sequences—A Comprehensive Review with Directions for Future Research](#). Research Report, University of Notre Dame, 1998. 98

- [BT59] R. B. Blackman and J. W. Tukey. *Particular Pairs of Windows*. New York: Dover, 1959. 27
- [CBG⁺96] S. Chandran, S. Balasubramanian, T. Gandhi, A. Prasad, and R. Kasturi. Structure recognition and information extraction from tabular documents. *International Journal of Imaging Systems and Technology*, 7:289–303, 1996. 1
- [CDD94] Q. Chen, M. Defrise, and F. Deconinck. [Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition](#). *Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1156–1168, December 1994. 21
- [CK93] S. Chandran and R. Kasturi. [Structural recognition of tabulated data](#). In *Proceedings, International Conference on Document Analysis and Recognition*, pages 516–519, 1993. 1
- [CM87] E. D. Castro and C. Morandi. Registration of translated and rotated images using finite Fourier transforms. *Transactions on Pattern Analysis and Machine Intelligence*, 9(5):700–703, September 1987. 21
- [CP76a] D. Casasent and D. Psaltis. Position, rotation, and scale-invariant optical correlation. *Applied Optics*, 15:1793–1799, 1976. 21
- [CP76b] D. Casasent and D. Psaltis. Scale-invariant optical transform. *Optical Engineering*, 15(3):258–261, May/June 1976. 21
- [CWL03] Y. Cao, S. Wang, and H. Li. [Skew detection and correction in document images based on straight-line fitting](#). *Pattern Recognition Letters*, 24(12):1871–1879, August 2003. 11

- [Dev98] Nicolas Devillard. Fast median search: an ANSI C implementation.
<http://ndevilla.free.fr/median/>, July 1998. 37, 40, 41, 102, 112
- [DG01] Stéphane Derrode and Faouzi Ghorbel. Robust and Efficient Fourier-Mellin Transform Approximations for Gray-Level Image Reconstruction and Complete Invariant Description. *Computer Vision and Image Understanding*, 83(1):57–78, 2001. 21, 55
- [DH72] R. Duda and P. Hart. Use of the Hough Transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972. 10
- [Doe50] G. Doetsch. *Handbuch der Laplace Transformation*, volume I. Birkhauser Verlag, Basel-Stuttgart, 1950. 20
- [Doe98] David Doermann. The Indexing and Retrieval of Document Images: A Survey. *Computer Vision and Image Understanding*, 70(3):287–298, 1998. 5
- [FJ03] Matteo Frigo and Steven G. Johnson. *The Fastest Fourier Transform in the West, Version 3*. Massachusetts Institute of Technology, <http://www.fftw.org/>, April 2003. 96
- [GG96] M. D. Garriss and P. J. Grother. Generalized Form Registration Using Structure-Based Techniques. In *Proceedings, Fifth Annual Symposium on Document Analysis and Information Retrieval*, pages 321–334, April 1996. 12
- [GS91] A. E. Grace and M. Spann. A comparison between Fourier-Mellin descriptors and moment based features for invariant object recognition using

- neural networks. *Pattern Recognition Letters*, 12:635–643, October 1991. 21
- [GW93] J. Gil and M. Werman. [Computing 2-D Min, Median, and Max Filters](#). *Transactions on Pattern Analysis and Machine Intelligence*, 15(5):504–507, May 1993. 37
- [HFA90] S. C. Hinds, J. L. Fisher, and D. P. D. Amato. [A Document Skew Detection Method Using Run-length Encoding and the Hough Transform](#). In *Proceedings, International Conference on Pattern Recognition*, pages 464–468, 1990. 10
- [HWW97] J. Huang, Y. Wang, and E. K. Wong. [Check Image Compression: A Comparision of JPEG, Wavelet and Layered Coding Methods](#). In *Proceedings, International Conference on Image Processing*, pages 694–697. IEEE, October 1997. 100, 104
- [HWW98] J. Huang, Y. Wang, and E. K. Wong. [Check Image Compression using a layered coding method](#). *Journal of Electronic Imaging, Special issue on Image/Video Processing and Compression for Visual Comunication*, 7(3):426–442, July 1998. 35, 100, 104
- [IR95] M. R. Inggs and A. R. Robinson. [Neural approaches to ship target recognition](#). In *Proceedings, International Radar Conference*, pages 386–391. IEEE, May 1995. 21
- [KB03] Douglas J. Kennard and William A. Barrett. [Just-In-Time Browsing for Digitized Microfilm and Other Similar Image Collections](#). In *Proceedings, International Conference on Document Analysis and Recognition*, August 2003. 100, 105

- [KD96] Omid Kia and David Doermann. [Structural compression for document analysis](#). In *Proceedings, International Conference on Pattern Recognition*, 1996. 6
- [KD97] Omid Kia and David Doermann. [Integrated Segmentation and Clustering for Enhanced Compression of Document Images](#). In *Proceedings, International Conference on Document Analysis and Recognition*, volume 1, 1997. 6
- [KFK02] E. Kavallieratou, N. Fakotakis, and G. Kokkinakis. [Skew angle estimation for printed and handwritten documents using the Wigner-Ville distribution](#). *Image and Vision Computing*, 20:813–824, 2002. 12
- [KH75] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. In *Proceedings, Conference on Cybernetics and Society*, pages 163–165. IEEE, 1975. 21
- [LDS02a] LDS.org. *News release: “Facts About the 1880 U.S. Census”*. The Church of Jesus Christ of Latter-day Saints, October 2002. <http://www.lds.org/newsroom/showpackage/0,15367,3881-1---4-645,00.html>. 3
- [LDS02b] LDS.org. *Online access to fully-indexed census records, currently including the 1880 United States, 1881 British Isles, and the 1881 Canadian Census*. The Church of Jesus Christ of Latter-day Saints, October 2002. http://www.familysearch.org/Eng/Search/frameset_search.asp?PAGE=census/search_census.asp. 3
- [LM00] X. Luo and G. Mirchandani. [An integrated framework for image classification](#). In *Proceedings, International Conference on Acoustics, Speech and Signal Processing*, 2000. 21

- [LTW94] D. X. Lee, G. Thoma, and H. Weschler. [Automated Page Orientation and Skew Angle Detection for Binary Document Images](#). *Pattern Recognition*, 27(10):1325–1344, October 1994. 10
- [LV96] Jacques Lévy-Vehel. Utilisation de la transformée de Mellin en traitement de signaux fractals—[Some applications of the Mellin transform in signal processing](#). *INRIA Research Report No. 2992*, 1995–1996. 20
- [LWB⁺01] C-Y. Lin, M. Wu, J. A. Bloom, M. L. Miller, I. J. Cox, and Y-M. Lui. [Rotation, Scale, and Translation Resilient Public Watermarking for Images](#). *Transactions on Image Processing*, 10(5), May 2001. 21
- [McG98] Morgan McGuire. [An Image Registration Technique for Recovering Rotation, Scale and Translation Parameters](#). *NEC Tech Report 98-018*, February 1998. 23
- [McG01] Morgan McGuire. [Image Registration Method](#). U.S. Patent number 6,266,452, U.S. Patent and Trademarks Office, July 2001. 23
- [MPI97] MPI Forum. *MPI-2: Extensions to the Message-Passing Interface*. Message-Passing Interface Forum, <http://www.mpi-forum.org/>, 1997. 101
- [Nag00] G. Nagy. [Twenty years of document image analysis in PAMI](#). *Transactions on Pattern Analysis and Machine Intelligence*, 22(1):38–62, January 2000. 1
- [Naj04] L. Najman. [Using mathematical morphology for document skew estimation](#). In *Proceedings, Symposium on Electronic Imaging: Document Recognition and Retrieval XI*. IS&T/SPIE (The International Society for Optical Engineering), January 2004. 12

- [NB03] Heath E. Nielson and William A. Barrett. [Consensus-Based Table Form Recognition](#). *Proceedings, Seventh International Conference on Document Analysis and Recognition*, II:906–910, August 2003. 1, 5, 11
- [Nib86] W. Niblack. *An Introduction to Digital Image Processing*. Englewood Cliffs, N.J.: Prentice Hall, 1986. 31, 35, 39
- [OPS99] Oleg Okun, Matti Pietikainen, and Jaakko J. Sauvola. [Document skew estimation without angle range restriction](#). *International Journal of Document Analysis and Recognition*, 2(2–3):132–144, 1999. 12
- [Ots79] N. Otsu. A threshold selection method from grey-level histograms. *Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. 31
- [PGP99] S. Perantonis, B. Gatos, and N. Papamarkos. [Block decomposition and segmentation for fast Hough transform evaluation](#). *Pattern Recognition*, 32(5):811–824, 1999. 11
- [Pop01] Kris Popat. [Lossy Compression of Grayscale Document Images by Adaptive-Offset Quantization](#). In *Proceedings, Electronic Imaging: Document Recognition and Retrieval VIII*. IS&T/SPIE (The International Society for Optical Engineering), January 2001. 100, 105
- [Pos86] W. Postl. Detection of linear oblique structures and skew scan in digitized documents. In *Proceedings, International Conference on Pattern Recognition*, pages 687–689, 1986. 10, 21
- [Pos88] W. Postl. [Method for automatic correction of character skew in the acquisition of a text original in the form of digital scan results](#). U.S. Patent number 4,723,297, U.S. Patent and Trademarks Office, 1988. 10

- [Pra78] W. K. Pratt. *Digital Image Processing*. John Wiley & Sons, New York, 1978. 21
- [PS00] R. Plamondon and S. Srihari. [On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey](#). *Transactions on Pattern Analysis and Machine Intelligence*, 22(1), January 2000. 5
- [RC96] B. S. Reddy and B. N. Chatterji. [An FFT-based technique for translation, rotation, and scale-invariant image registration](#). *Transactions on Pattern Analysis and Machine Intelligence*, 5(8):1266–1271, August 1996. 21
- [SA86] Y. Sheng and H. H. Arsenault. [Experiments on pattern recognition using invariant Fourier-Mellin descriptors](#). *Journal of the Optical Society of America A*, 3(6):771–776, June 1986. 21
- [Si97] Deyi Si. [Skew and Slant Correction for Document Images Using Gradient Direction](#). In *Proceedings, International Conference on Document Analysis and Recognition*, pages 142–146, August 1997. 12
- [SIR99] Tal Steinherz, Nathan Intrator, and Ehud Rivlin. [Skew Detection via Principal Components Analysis](#). In *Proceedings, International Conference on Document Analysis and Recognition*, January 1999. 12
- [SP95] J. Sauvola and M. Pietikäinen. [Skew Angle Detection Using Texture Direction Analysis](#). In *Proceedings, Scandinavian Conference on Image Analysis*, 1995. 12
- [SRI99] Tal Steinherz, Ehud Rivlin, and Nathan Intrator. [Offline cursive script word recognition—a survey](#). *International Journal of Document Analysis and Recognition*, 2(2–3):90–110, 1999. 5

- [STM03] H. S. Stone, B. Tao, and M. McGuire. [Analysis of image registration noise due to rotationally dependent aliasing](#). *Journal of Visual Communication and Image Representation*, 14(2), June 2003. [15](#), [21](#), [23](#), [24](#), [27](#), [52](#), [55](#)
- [Sto02] Harold S. Stone. *NEC Tech Report: Fourier-Based Image Registration Techniques*. NEC Research,
<http://www.censsis.neu.edu/hstone.fourier.pdf>, January 2002. [21](#), [23](#)
- [TGG79] T.S.Huang, G.J.Yang, and G.Y.Tang. [A fast two-dimensional median filtering algorithm](#). *Transactions on Acoustics, Speech and Signal Processing*, 27(1), February 1979. [40](#)
- [TLLX97] Y. Tang, J. Liu, B. F. Li, and D. Xi. Multiresolution analysis in extraction of reference lines from documents with gray level background. *Transactions on Pattern Analysis and Machine Intelligence*, 19(8), August 1997. [1](#)
- [TY81] H. Saito T. Yatagay, K.Choji. Pattern classification using optical mellin transform and circular photodiode array. *Optical Communication*, 38(3):162–165, August 1981. [21](#)
- [Vin02] A. Vinciarelli. [A Survey on Off-Line Cursive Script Recognition](#). *Pattern Recognition*, 35(7):1433–1446, 2002. [3](#), [5](#)
- [Weia] Eric W. Weisstein. Apodization function. From *Mathworld*—A Wolfram Web Resource.
<http://mathworld.wolfram.com/ApodizationFunction.html>. [27](#)

- [Weib] Eric W. Weisstein. Blackman function. From *Mathworld*—A Wolfram Web Resource. <http://mathworld.wolfram.com/BlackmanFunction.html>. 27
- [Weic] Eric W. Weisstein. Integral transform. From *Mathworld*—A Wolfram Web Resource. <http://mathworld.wolfram.com/IntegralTransform.html>. 17
- [Weid] Eric W. Weisstein. Mellin transform. From *Mathworld*—A Wolfram Web Resource. <http://mathworld.wolfram.com/MellinTransform.html>. 20
- [WZ00a] G. Wolberg and S. Zokai. [Image Registration For Perspective Deformation Recovery](#). In *Proceedings, Conference on Automatic Target Recognition X. IS&T/SPIE (The International Society for Optical Engineering)*, April 2000. 13
- [WZ00b] G. Wolberg and S. Zokai. [Robust Image Registration Using Log-Polar Transform](#). In *Proceedings, International Conference on Image Processing*. IEEE, September 2000. 12
- [XL98] D. Xi and S. Lee. [Table structure extraction from form documents based on gradient-wavelet scheme](#). In *Proceedings, Document Analysis Systems: Theory and Practice: Third IAPR Workshop*. International Association for Pattern Recognition, 1998. 1
- [ZB01] Zhong Zhang and Rick S. Blum. [A hybrid image registration technique for a digital camera image fusion application](#). *Information Fusion*, 2(2):135–149, 2001. 13
- [ZF03] B. Zitova and J. Flusser. [Image registration methods: a survey](#). *Image and Vision Computing*, 21(11):977–1000, October 2003. 1, 13

- [ZK83] P. E. Zwicke and I. Kiss, Jr. A new implementation of the Mellin transform and its application to radar classification of ships. *Transactions on Pattern Analysis and Machine Intelligence*, 5(2):191–199, March 1983. [21](#)