An Optimal FFT-Based Algorithm for Mosaicking Images, With Applications to Satellite Imaging and Web Search

Stephen Gibson[1,2], Olga Kosheleva[3], Luc Longpré[1],
Brian Penn[2], and Scott A. Starks[2,3]

[1]Department of Computer Science,
[2]NASA Pan-American Center for Earth
and Environmental Studies (PACES)
[3]Department of Electrical and Computer Engineering
University of Texas at El Paso, 500 W. University
El Paso, TX 79968, USA, email olga@ece.utep.edu

Abstract. *Digital data storage is becoming ever more abundant and cheap. This, along with other technological advances, has brought about an age of mass storage of information, much of it in the form of images. In order to be able to process these stockpiles of image data, new and faster computer algorithms are needed.*

One area of interest is that of image mosaicking, i.e., comparing two overlapping images and finding the proper scaling, angle of rotation, and translation needed to fit one with the other. Early methods for mosaicking images included visual inspection or exhaustive, pixel by pixel, search for the best match. With such large quantities of images as we have today, and the increasing need for accuracy, these methods are no longer feasible. Several new mosaicking methods have been proposed based on Fast Fourier Transform (FFT).

The existing FFT-based algorithms do not always lead to reasonable mosaicking. In this paper, we formalize the corresponding expert rules and, as a result, design an optimal FFT-based mosaicking algorithm.

Introduction. Today, more data is stored in the form of images than ever before. In particular, satellite photos provide a good description of the Earth areas. Often, we are interested in the area which is covered by several satellite photos, so we need to combine (*mosaic*) these photos into a single image. The problem is that we do not know the exact orientation of the satellite-based camera, so the photos may be shifted and rotated with respect to each other, and we do not know the exact values of these shift and rotation. Therefore, to mosaic two images, we must find how shifted and rotated these images are relative to one another.

A similar problem occurs when we search images stored on the web. We may want to find all images which contain a certain pattern (e.g., a certain text), but this pattern may be scaled differently in different web images. So, we must be able to mosaic two images: the image which contains the desired pattern, and the image which is stored on the web. We must be able to find the shift, rotation, and scaling after which these two images match in the best possible way.

At present, mosaicking of satellite images is mainly done manually, by trial and error. This trial-and-error procedure is difficult to automate: for $n \times n$ images, where n can be from 1,000 to 6,000, we have n^2 possible shifts, which, together with $\approx n$ possible rotations and $\approx n$ possible scalings, make for an impossible number of $\approx n^4$ ($\geq 10^{12}$) possible image comparisons. It is therefore necessary to come up with time-saving mosaicking algorithms.

The existing FFT-based mosaicking algorithms. To decrease the mosaicking time, researchers have proposed methods based on Fast Fourier Transform (FFT). The best of known FFT-based mosaicking algorithms is presented in [5]. The main idea of FFT-based mosaicking is as follows.

It is known that if two images $I_1(\vec{x})$ and $I_2(\vec{x})$ differ only by shift, i.e., if $I_2(\vec{x}) = I_1(\vec{x}+\vec{a})$ for some (unknown) shift $\vec{a}$, then their Fourier transforms $F_i(\vec{\omega})$ are related by the following formula:

$$F_2(\vec{\omega}) = e^{\mathrm{i}\cdot(\vec{\omega}\cdot\vec{a})} \cdot F_1(\vec{\omega}). \tag{1}$$

Therefore, if the images are indeed obtained from each other by shift, then we have $M_2(\vec{\omega}) = M_1(\vec{\omega})$, where we denoted $M_i(\vec{\omega}) = |F_i(\vec{\omega})|$. The actual value of the shift can be obtained if we use the formula (1) to compute the value of the following ratio:

$$R(\vec{\omega}) = \frac{F_1^*(\vec{\omega}) \cdot F_2(\vec{\omega})}{|F_1^*(\vec{\omega}) \cdot F_2(\vec{\omega})|}.$$

We get $R(\vec{\omega}) = e^{\mathrm{i}\cdot(\vec{\omega}\cdot\vec{a})}$. Therefore, the inverse Fourier transform $P(\vec{x})$ of this ratio is equal to the delta-function $\delta(\vec{x}-\vec{a})$. In other words, in the ideal no-noise situation, this inverse Fourier transform $P(\vec{x})$ is equal to 0 everywhere except for the point $\vec{x} = \vec{a}$; so, from $P(\vec{x})$, we can easily determine the desired shift.

In the presence of noise, we expect the values of $P(\vec{x})$ to be slightly different from the delta-function, but still, the value $|P(\vec{a})|$ should be much larger than all the other values of this function. So, we can find the shift $\vec{a}$ as the point for which $|P(\vec{x})|$ takes the largest possible value.

If, in addition to shift, we also have rotation and scaling, then the absolute values $M_i(\vec{\omega})$ of the corresponding Fourier transforms are not equal, but differ from each by the corresponding rotation and scaling. If we go from Cartesian to polar coordinates (r, θ) in the $\vec{\omega}$-plane, then rotation by an angle θ_0 is described by a simple shift-like formula $\theta \to \theta + \theta_0$. In these same coordinates, scaling is also simple, but not shift-like: $r \to \lambda \cdot r$. If we go to *log-polar* coordinates (ρ, θ), where $\rho = \log(r)$, then scaling also becomes shift-like: $\rho \to \rho + b$, where $b = \log(\lambda)$. So, in log-polar coordinates, both rotation and scaling are described by a shift.

Therefore, in order to detect the rotation and scaling between M_1 and M_2, we move both images to log-polar coordinates and then use the above FFT-based algorithm to determine the corresponding shift $(\theta_0, \log(\lambda))$. From the corresponding "shift" values, we can reconstruct the rotation angle θ_0 and the scaling coefficient λ. Now, we can apply the corresponding rotation and scaling to one of the original images. As a result, we get two images which are already aligned in terms of rotation and scaling, and the only difference between them is in an (unknown) shift. So, we can again apply the above described FFT-based algorithm: this time, actaully to determine shift. Hence, we get the desired mosaicking.

In many real life situations, this algorithm works well, but in many others, it fails to determine the shift correctly. In this paper, we describe an improvement of this algorithm. The details are given in [1].

Main idea. In the existing algorithm, we determine the shift as the point $\vec{x}$ on a grid for which $|P(\vec{x})|$ attains the largest possible value. The actual values of rotation angle θ_0 and log-scaling $\log(\lambda)$ may not be exactly on the grid; as a result, when we use the FFT-based shift-detection algorithm to determine rotation and scaling, we do not determine them exactly. Hence, the alignment made by these approximate values of rotation angle and scaling is not exact. For noisy images, the additional distortion produced by this mis-alignment often prevents the shift-detecting algorithm from finding the shift between the images I_1 and I_2.

To decrease this distortion, in shift detection, we would like to be able to find a more accurate estimate of the shift, even when its actual value is not from the grid. In 1-D case, if the function $|P(x)|$ has a large maximum at a point a and is equal to 0 for all $x \neq a$, then, of course, the actual value of the shift is a. However, if the value $|P(x)|$ is large for two sequential points x_1 and x_2, then probably the actual shift is somewhere between x_1 and x_2. In other words, the actual shift should be equal to $x = w_1 \cdot x_1 + w_2 \cdot x_2$ for some weights $w_1 + w_2 = 1$.

The larger $|P(x_i)|$, the closer the actual shift point to x_i; so, the larger the weight w_i should be. It therefore seems reasonable to take $w_i = f(|P(x_i)|)$ for some monotonically increasing function $f(z)$. For this choice, however, we cannot guarantee that $w_1 + w_2 = 1$, so we must *normalize* these weights, and take

$$x = \frac{f(|P(x_1)|) \cdot x_1 + f(|P(x_2)|) \cdot x_2}{f(|P(x_1)|) + f(|P(x_2)|).} \quad (1)$$

In a 2-D case, we can similarly take two points x_1, x_2, y_1, y_2 in each of the grid's directions, and use the sums of the corresponding values $f(|P))$ as the weights:

$$x = \frac{w_{x1} \cdot x_1 + w_{x2} \cdot x_2}{w_{x1} + w_{x2}}; \quad (2)$$

$$y = \frac{w_{y1} \cdot y_1 + w_{y2} \cdot y_2}{w_{y1} + w_{y2}}, \quad (3)$$

where

$$w_{xi} = f(|P(x_i, y_1)|) + f(|P(x_i, y_2)|); \quad (4)$$

$$w_{yi} = f(|P(x_1, y_i)|) + f(|P(x_2, y_i)|). \quad (5)$$

The question is: which function $f(z)$ is the best?

We want to determine a family of functions $f(z)$. All we want from the function $f(z)$ is the weights. These probabilities are computed according to the formulas (1–5). From the expressions (1–5), one can easily see that if we multiply all the values of this function $f(z)$ by an arbitrary constant C, i.e., if we consider a new function $\widetilde{f}(z) = C \cdot f(z)$, then this new function will lead (after the normalization involved in (1–3)), to exactly the same values of the weights. Thus, whether we choose $f(z)$ or $\widetilde{f}(z) = C \cdot f(z)$, does not matter. So, what we are really choosing is not a *single* function $f(z)$, but a *family* of functions $\{C \cdot f(z)\}$ (characterized by a parameter $C > 0$).

We want to choose the best function $f(z)$. There can be several different optimality criteria: reconstruction accuracy, computational simplicity, etc. We would like to get our result in the most general form, i.e., we would like to have a result which would be applicable to all possible reasonable optimality criteria.

One thing which is reasonable to require is that if we change the units for measuring P, i.e., if we replace P by $\lambda \cdot P$, the relative quality of different weight-generating functions $f(z)$ should not change.

Arguing like in [2, 4], we get the following results:

Definition 1. *Let $f(z)$ be a differentiable strictly increasing function from real numbers to non-negative real numbers. By a family that corresponds to this function $f(z)$, we mean a family of all functions of the type $\widetilde{f}(z) = C \cdot f(z)$, where $C > 0$ is an arbitrary positive real number. (Two families are considered equal if they coincide, i.e., consist of the same functions.)*

In the following text, we will denote the set of all possible families by Φ.

Definition 2. *By an optimality criterion, we mean a consistent pair $(\prec, \sim)$ of relations on the set Φ of all alternatives which satisfies the following conditions, for every $F, G, H \in \Phi$:*

(1) *if $F \prec G$ and $G \prec H$ then $F \prec H$;*
(2) *$F \sim F$;*
(3) *if $F \sim G$ then $G \sim F$;*
(4) *if $F \sim G$ and $G \sim H$ then $F \sim H$;*
(5) *if $F \prec G$ and $G \sim H$ then $F \prec H$;*
(6) *if $F \sim G$ and $G \prec H$ then $F \prec H$;*
(7) *if $F \prec G$ then $G \not\prec F$ and $F \not\sim G$.*

Comment. The intended meaning of these relations is as follows:

- $F \prec G$ means that with respect to a given criterion, G is better than F;
- $F \sim G$ means that with respect to a given criterion, F and G are of the same quality.

Under this interpretation, conditions (1)–(7) have simple intuitive meaning; e.g., (1) means that if G is better than F, and H is better than G, then H is better than F.

Definition 3.

- *We say that an alternative F is optimal (or best) with respect to a criterion $(\prec, \sim)$ if for every other alternative G either $F \succ G$ or $F \sim G$.*

- *We say that a criterion is final if there exists an optimal alternative, and this optimal alternative is unique.*

Definition 4. *Let $\lambda > 0$ be a positive real number.*

- *By a λ-rescaling of a function $f(x)$ we mean a function $\widetilde{f}(x) = f(\lambda \cdot x)$.*

- *By a λ-rescaling $R_\lambda(F)$ of a family of functions F we mean the family consisting of λ-rescalings of all functions from F.*

Definition 5. *We say that an optimality criterion on Φ is unit-invariant if for every two families F and G and for every number $\lambda > 0$, the following two conditions are true:*

i) if F is better than G in the sense of this criterion (i.e., $F \succ G$), then $R_\lambda(F) \succ R_\lambda(G)$;

ii) if F is equivalent to G in the sense of this criterion (i.e., $F \sim G$), then $R_\lambda(F) \sim R_\lambda(G)$.

Theorem. *If a family F is optimal in the sense of some optimality criterion that is final and unit-invariant, then every function $f(z)$ from this family F has the form $C \cdot z^\alpha$ for some real numbers C and α.*

This theorem was, in effect, proven in [3, 4].

Experiments show that for image mosaicking, the best choice of α is as follows:

- on the first stage, when we determine rotation and scaling, the optimal value is $\alpha_r \approx 1.55$;

- on the second stage, on which we determine the shift, the optimal value is $\alpha_s \approx 0.65$.

For these values, we indeed get a pretty good mosaicking.

Acknowledgments. This work was supported in part by NASA under cooperative agreement NCC5-209, by NSF grants No. DUE-9750858 and CDA-9522207, by United Space Alliance, grant No. NAS 9-20000 (PWO C0C67713A6), by the Future Aerospace Science and Technology Program (FAST) Center for Structural Integrity of Aerospace Systems, effort sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-95-1-0518, and by the National Security Agency under Grant No. MDA904-98-1-0561.

References

[1] S. Gibson, *An optimal FFT-based algorithm for mosaicking images*, Master Thesis, Department of Computer Science, University of Texas at El Paso, December 1999.

[2] L. T. Koczy, V. Kreinovich, Y. Mendoza, H. T. Nguyen, and H. Schulte, "Towards Mathematical Foundations of Information Retrieval: Dependence of Website's Relevance on the Number of Occurrences of a Queried Word", *These Proceedings*.

[3] O. Kosheleva, L. Longpré, and R. Osegueda, "Detecting Known Non-Smooth Structures in Images: Fuzzy and Probabilistic Methods, with Applications to Medical Imaging, Non-Destructive Testing, and Detecting Text on Web Pages", *Proceedings of The Eighth International Fuzzy Systems Association World Congress IFSA'99*, Taipei, Taiwan, August 17–20, 1999, pp. 269–273.

[4] H. T. Nguyen and V. Kreinovich, *Applications of continuous mathematics to computer science*, Kluwer, Dordrecht, 1997.

[5] B. S. Reddy and B. N. Chatterji, "An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration," *IEEE Transactions on Image Processing*, 1996, Vol. 5, No. 8, pp. 1266–1271.