University of Texas at El Paso
ScholarWorks@UTEP

Departmental Technical Reports (CS) Computer Science

4-2000

Towards Optimal Mosaicking of Multi-Spectral Images

Francisco Garcia

Roberto Araiza
The University of Texas at El Paso, raraiza@miners.utep.edu

Brain Rzycki

Follow this and additional works at: https://scholarworks.utep.edu/cs_techrep

Part of the Computer Engineering Commons

Comments:
UTEP-CS-00-19.
Published in the *Proceedings of the Second National NASA Student Conference on Earth Sciences, Aero-Space Technology, Human Exploration and Development of Space, Space Science, and Crosscutting Technologies*, Nashville, Tennessee, April 7-10, 2000, pp. 261-266.

Recommended Citation

Garcia, Francisco; Araiza, Roberto; and Rzycki, Brain, "Towards Optimal Mosaicking of Multi-Spectral Images" (2000). *Departmental Technical Reports (CS)*. 475.
https://scholarworks.utep.edu/cs_techrep/475

This Article is brought to you for free and open access by the Computer Science at ScholarWorks@UTEP. It has been accepted for inclusion in Departmental Technical Reports (CS) by an authorized administrator of ScholarWorks@UTEP. For more information, please contact lweber@utep.edu.

Towards Optimal Mosaicking of Multi-Spectral Images

Francisco Garcia, Roberto Araiza, and Brain Rzycki

NASA Pan-American Center for Earth
and Environmental Sciences (PACES)
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
email fgarcia@cs.utep.edu

Abstract

To cover a certain area, it is often necessary to combine several satellite photos. To get a proper combination, we need to appropriately position and orient these photos relative to one another, i.e., *mosaic* these photos. With the new generation of multi-spectral satellites, for each area, we have several hundred images which correspond to different wavelengths. At present, when we mosaic two images, we only use one of the wavelengths and ignore the information from the other wavelengths. It is reasonable to decrease the mosaicking error by using images corresponding to all possible wavelengths in mosaicking. In this paper, we present an algorithm for such optimal mosaicking.

1 Mosaicking satellite images: the problem and the existing methods of solving this problem

1.1 Main problem: Mosaicking satellite images

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.

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.

1.2 Auxiliary problem: Searching for a pattern in a web image

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.

One particular case of this problem is searching for text in web images. The gaining popularity of the World Wide Web also means increasing security risks. As the World Wide Web has become an affordable way for different political groups to reach a broad audience it is becoming harder to monitor all these web sites for their content. While numerous web search tools can be used to automatically monitor plain text in web pages, search for text in graphical images is still a considerable challenge. This fact is used by designers of such web pages who "hide" their text by placing it inside of graphical images, avoiding detection from regular search engines. At present, the only known way to find all occurrences of suspicious words like "terror" in images is to use character recognition to find and read all the texts in all the images. Performing such a character recognition is still a very computational intensive task which takes a long time, and has to be done for every image, because the only known way to check whether the image contains text is to apply a character recognition to this image. It is therefore desirable to develop faster algorithms for detecting text in web pages.

1.3 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 [2]. The main ideas behind FFT-based mosaicking in general and this algorithm in particular are as follows.

1.4 The simplest case: shift detection in the absence of noise

Let us first consider the case when two images differ only by shift. It is known that if two images $I(\vec{x})$ and $I'(\vec{x})$ differ only by shift, i.e., if $I'(\vec{x}) = I(\vec{x}+\vec{a})$ for some (unknown) shift $\vec{a}$, then their Fourier transforms

$$F(\vec{\omega}) = \frac{1}{2\pi} \cdot \int\int I(\vec{x}) \cdot e^{-2\pi \cdot \mathrm{i} \cdot (\vec{x}\cdot\vec{\omega})}\, \mathrm{d}x \mathrm{d}y,$$

$$F'(\vec{\omega}) = \frac{1}{2\pi} \cdot \int\int I'(\vec{x}) \cdot e^{-2\pi \cdot \mathrm{i} \cdot (\vec{x}\cdot\vec{\omega})}\, \mathrm{d}x \mathrm{d}y,$$

are related by the following formula:

$$F'(\vec{\omega}) = e^{2\pi \cdot \mathrm{i} \cdot (\vec{\omega}\cdot\vec{a})} \cdot F(\vec{\omega}). \tag{1}$$

Therefore, if the images are indeed obtained from each other by shift, then we have

$$M'(\vec{\omega}) = M(\vec{\omega}), \tag{2}$$

where we denoted

$$M(\vec{\omega}) = |F(\vec{\omega})|, \quad M'(\vec{\omega}) = |F'(\vec{\omega})|. \tag{3}$$

The actual value of the shift $\vec{a}$ can be obtained if we use the formula (1) to compute the value of the following ratio:

$$R(\vec{\omega}) = \frac{F'(\vec{\omega})}{F(\vec{\omega})}. \tag{4}$$

Substituting (1) into (4), we get

$$R(\vec{\omega}) = e^{2\pi \cdot \mathrm{i} \cdot (\vec{\omega}\cdot\vec{a})}. \tag{5}$$

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 by using the following *algorithm*:

- *first, we apply FFT to the original images $I(\vec{x})$ and $I'(\vec{x})$ and compute their Fourier transforms $F(\omega)$ and $F'(\omega)$;*
- *on the second step, we compute the ratio (4);*
- *on the third step, we apply the inverse FFT to the ratio $R(\vec{\omega})$ and compute its inverse Fourier transform $P(\vec{x})$;*
- *finally, on the fourth step, we determine the desired shift $\vec{a}$ as the only value $\vec{a}$ for which $P(\vec{a}) \neq 0$.*

1.4.1 Shift detection in the presence of noise

In the ideal case, the absolute value of the ratio (4) is equal to 1. In real life, the measured intensity values have some noise in them. For example, the conditions may slightly change from one overflight to another, which can be represented as the fact that a "noise" was added to the actual image.

In the presence of noise, the observed values of the intensities may differ from the actual values; as a result, their Fourier transforms also differ from the values and hence, the absolute value of the ratio (4) may be different from 1.

We can somewhat improve the accuracy of this method if, instead of simply processing the measurement results, we take into consideration the additional knowledge that the absolute value of the actual ratio (4) is exactly equal to 1. Let us see how this can be done.

Let us denote the actual (unknown) value of the value $e^{2\pi \cdot \mathrm{i} \cdot (\vec{\omega}\cdot\vec{a})}$ by r. Then, in the absence of noise, the equation (1) takes the form

$$F'(\vec{\omega}) = r \cdot F(\vec{\omega}). \tag{5}$$

In the presence of noise, the computed values $F(\vec{\omega})$ and $F'(\vec{\omega})$ of the Fourier transforms can be slightly different from the actual values, and therefore, the equality (5) is only approximately true:

$$F'(\vec{\omega}) \approx r \cdot F(\vec{\omega}). \tag{6}$$

In addition to the equation (6), we know that the absolute value of r is equal to 1, i.e., that

$$|r|^2 = r \cdot r^* = 1, \tag{7}$$

where r^* denotes a complex conjugate to r.

As a result, we know two things about the unknown value r:

- that r satisfies the approximate equation (6), and

- that r satisfies the additional constraint (7).

We would like to get the best estimate for r among all estimates which satisfy the condition (7). To get the optimal estimate, we can use the *Least Squares Method* (LSM). According to this method, for each estimate r, we define the error

$$E = F'(\vec{\omega}) - r \cdot F(\vec{\omega}) \tag{8}$$

with which the condition (6) is satisfied. Then, we find among all estimates which satisfy the additional condition (7), a value r for which the square $|E|^2 = E \cdot E^*$ of this error is the smallest possible.

The square $|E|^2$ of the error E can be reformulated as follows:

$$E \cdot E^* = \left(F'(\vec{\omega}) - r \cdot F(\vec{\omega})\right) \cdot \left(F'^*(\vec{\omega}) - r^* \cdot F^*(\vec{\omega})\right) =$$

$$F'(\vec{\omega}) \cdot F'^*(\vec{\omega}) - r^* \cdot F^*(\vec{\omega}) \cdot F'(\vec{\omega}) - r \cdot F(\vec{\omega}) \cdot F'^*(\vec{\omega}) + r \cdot r^* \cdot F(\vec{\omega}) \cdot F^*(\vec{\omega}). \tag{9}$$

We need to minimize this expression under the condition (7).

For conditional minimization, there is a known technique of Lagrange multipliers, according to which the minimum of a function $f(x)$ under the condition $g(x) = 0$ is attained when for some real number λ, the auxiliary function $f(x) + \lambda \cdot g(x)$ attains its unconditional minimum; this value λ is called a *Lagrange multiplier.*

For our problem, the Lagrange multiplier technique leads to the following unconditional minimization problem:

$$F'(\vec{\omega}) \cdot F'^*(\vec{\omega}) - r^* \cdot F^*(\vec{\omega}) \cdot F'(\vec{\omega}) - r \cdot F(\vec{\omega}) \cdot F'^*(\vec{\omega}) + r \cdot r^* \cdot F(\vec{\omega}) \cdot F^*(\vec{\omega}) + \lambda \cdot (r \cdot r^* - 1) \to \min. \tag{10}$$

We want to find the value of the complex variable r for which this expression takes the smallest possible value. A complex variable is, in effect, a pair of two real variables, so the minimum can be found as a point at which the partial derivatives with respect to each of these variables are both equal to 0. Alternatively, we can represent this equality by computing the partial derivative of the expression (10) relative to r and r^*. If we differentiate (10) relative to r^*, we get the following linear equation:

$$-F^*(\vec{\omega}) \cdot F'(\vec{\omega}) + r \cdot F(\vec{\omega}) \cdot F^*(\vec{\omega}) + \lambda \cdot r = 0. \tag{11}$$

From this equation, we conclude that

$$r = \frac{F^*(\vec{\omega}) \cdot F'(\vec{\omega})}{F(\vec{\omega}) \cdot F^*(\vec{\omega}) + \lambda}. \tag{12}$$

The coefficient λ can be now determined from the condition that the resulting value r should satisfy the equation (7). The denominator $F(\vec{\omega}) \cdot F^*(\vec{\omega}) + \lambda$ of the equation (12) is a real number, so instead of finding λ, it is sufficient to find a value of this denominator for which $|r|^2 = 1$. One can easily see that to achieve this goal, we should take, as this denominator, the absolute value of the numerator, i.e., the value

$$|F^*(\vec{\omega}) \cdot F'(\vec{\omega})| = |F^*(\vec{\omega})| \cdot |F'(\vec{\omega})|. \tag{13}$$

For this choice of a denominator, the formula (11) takes the following final form:

$$r = \frac{F^*(\vec{\omega}) \cdot F'(\vec{\omega})}{|F^*(\vec{\omega})| \cdot |F'(\vec{\omega})|}. \tag{14}$$

So, in the presence of noise, instead of using the exact ratio (4), we should compute, for every $\vec{\omega}$, the optimal approximation

$$R(\vec{\omega}) = \frac{F^*(\vec{\omega}) \cdot F'(\vec{\omega})}{|F^*(\vec{\omega})| \cdot |F'(\vec{\omega})|}. \tag{15}$$

In the ideal non-noise case, the inverse Fourier transform $P(\vec{x})$ of this ratio is equal to the delta-function $\delta(\vec{x} - \vec{a})$, i.e., equal to 0 everywhere except for the point $\vec{x} = \vec{a}$. 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 arrive at the following *algorithm* for determining the shift $\vec{a}$:

- *first, we apply FFT to the original images $I(\vec{x})$ and $I'(\vec{x})$ and compute their Fourier transforms $F(\omega)$ and $F'(\omega)$;*
- *on the second step, we compute the ratio (15);*
- *on the third step, we apply the inverse FFT to the ratio $R(\vec{\omega})$ and compute its inverse Fourier transform $P(\vec{x})$;*
- *finally, on the fourth step, we determine the desired shift $\vec{a}$ as the point for which $|P(\vec{x})|$ takes the largest possible value.*

1.5 Reducing rotation and scaling to shift

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.

1.6 How to determine rotation and scaling

In view of the above reduction, in order to determine the rotation and scaling between M and M', we can do the following:

- transform both images from the original Cartesian coordinates to log-polar coordinates;
- use the above FFT-based algorithm to determine the corresponding shift $(\theta_0, \log(\lambda))$;
- from the corresponding "shift" values, reconstruct the rotation angle θ_0 and the scaling coefficient λ.

Comment. The main computational problem with the transformation to log-polar coordinates is that we need values $M(\xi, \eta)$ on a rectangular grid in log-polar space $(\log(\rho), \theta)$, but computing $(\log(\rho), \theta)$ for the original grid points leads to points outside that grid. So, we need interpolation to find the values $M(\xi, \eta)$ on the desired grid. One possibility is to use *bilinear* interpolation. Let (x, y) be a rectangular point corresponding to the desired grid point $(\log(\rho), \theta)$, i.e.,

$$x = e^{\log(\rho)} \cdot \cos(\theta), \quad y = e^{\log(\rho)} \cdot \sin(\theta).$$

To find the value $M(x, y)$, we look at the intensities M_{jk}, $M_{j+1,k}$, $M_{j,k+1}$, and $M_{j+1,k+1}$ of the four grid points (j, k), $(j+1, k)$, $(j, k+1)$, and $(j+1, k+1)$ surrounding (x, y). Then, we can interpolate $M(x, y)$ as follows:

$$M(x, y) = (1-t) \cdot (1-u) \cdot M_{jk} + t \cdot (1-u) \cdot M_{j+1,k} + (1-t) \cdot u \cdot M_{j,k+1} + t \cdot u \cdot M_{j+1,k+1},$$

where t is a fractional part of x and u is a fractional part of y.

1.7 Final algorithm: determining shift, rotation, and scaling

- *First, we apply FFT to the original images $I(\vec{x})$ and $I'(\vec{x})$ and compute their Fourier transforms $F(\omega)$ and $F'(\omega)$.*
- *Then, we compute the absolute values $M(\vec{\omega}) = |F(\vec{\omega})|$ and $M'(\vec{\omega}) = |F'(\vec{\omega})|$ of these Fourier transforms.*
- *By applying the above algorithm and scaling detection algorithm to the functions $M(\omega)$ and $M'(\omega)$, we can determine 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, e.g., to the first image $I(\vec{x})$. As a result, we get a new image $\widetilde{I}(\vec{x})$.*
- *Since we rotated and re-scaled one of the images, the images $\widetilde{I}(\vec{x})$ and $I'(\vec{x})$ 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 for determining shift: this time, actually to determine shift.*

As a result, we get the desired values of shift, rotation, and scaling; hence, we get the desired mosaicking.

2 Mosaicking multi-spectral satellite images

2.1 Formulation of the problem

With the new generation of multi-spectral satellites, for each area, we have several hundred images which correspond to different wavelengths. At present, when we mosaic two images, we only use one of the wavelengths and ignore the information from the other wavelengths. It is reasonable to decrease the mosaicking error by using images corresponding to all possible wavelengths in mosaicking.

Similarly, in detecting the known text in colored web images, we would like to take into consideration all color components.

In this paper, we present an algorithm for such optimal mosaicking.

2.2 Derivation of the new algorithm

For multi-spectral imaging, instead of a single image $I(\vec{\omega})$, we get several images $I_i(\vec{\omega})$, $1 \le i \le n$, which correspond to different wavelengths. So, we have two groups of images:

- the images $I_i(\vec{\omega})$ which correspond to one area, and
- the images $I'_i(\vec{\omega})$ which correspond to an overlapping area.

Let us first consider the case when two images differ only by some (unknown) shift $\vec{a}$. For every wavelength i, the corresponding two images $I_i(\vec{x})$ and $I'_i(\vec{x})$ differ only by shift, i.e., $I'_i(\vec{x}) = I_i(\vec{x} + \vec{a})$. Therefore, for every wavelength i, their Fourier transforms

$$F_i(\vec{\omega}) = \frac{1}{2\pi} \cdot \int\int I_i(\vec{x}) \cdot e^{-2\pi \cdot \mathrm{i} \cdot (\vec{x} \cdot \vec{\omega})} \, \mathrm{d}x\mathrm{d}y,$$

$$F'_i(\vec{\omega}) = \frac{1}{2\pi} \cdot \int\int I'_i(\vec{x}) \cdot e^{-2\pi \cdot \mathrm{i} \cdot (\vec{x} \cdot \vec{\omega})} \, \mathrm{d}x\mathrm{d}y,$$

are related by the formula:

$$F'_i(\vec{\omega}) = e^{2\pi\cdot\mathrm{i}\cdot(\vec{\omega}\cdot\vec{a})}\cdot F_i(\vec{\omega}). \tag{16}$$

In the ideal no-noise situation, all these equations are true, and we can determine the value $r = e^{2\pi\cdot\mathrm{i}\cdot(\vec{\omega}\cdot\vec{a})}$ from any of these equations. In the real-life situations, where noise is present, these equations (16) are only approximately true, so we have the following problem instead: find r for which, for all i,

$$F'_i(\vec{\omega}) \approx r\cdot F_i(\vec{\omega}). \tag{17}$$

and which satisfies the condition (7).

We would like to get the best estimate for r among all estimates which satisfy the condition (7). To get the optimal estimate, we can use the Least Squares Method, according to which, for each estimate r and for each i, we define the error

$$E_i = F'_i(\vec{\omega}) - r\cdot F_i(\vec{\omega}) \tag{18}$$

with which the condition (17) is satisfied. Then, we find among all estimates which satisfy the additional condition (7), a value r for which the sum of the squares

$$|E_1|^2 + \ldots + |E_n|^2 = E_1\cdot E_1^* + \ldots + E_n\cdot E_n^*$$

of these errors is the smallest possible.

The square $|E_i|^2$ of each error E_i can be reformulated as follows:

$$E_i\cdot E_i^* = \left(F'_i(\vec{\omega}) - r\cdot F_i(\vec{\omega})\right)\cdot\left(F'^*_i(\vec{\omega}) - r^*\cdot F_i^*(\vec{\omega})\right) =$$

$$F'_i(\vec{\omega})\cdot F'^*_i(\vec{\omega}) - r^*\cdot F_i^*(\vec{\omega})\cdot F'_i(\vec{\omega}) - r\cdot F_i(\vec{\omega})\cdot F'^*_i(\vec{\omega}) + r\cdot r^*\cdot F_i(\vec{\omega})\cdot F_i^*(\vec{\omega}). \tag{19}$$

We need to minimize the sum of these expressions under the condition (7).

For this conditional minimization, we will use the Lagrange multipliers technique, which leads to the following unconditional minimization problem:

$$\sum_{i=1}^{n}\left(F'_i(\vec{\omega})\cdot F'^*_i(\vec{\omega}) - r^*\cdot F_i^*(\vec{\omega})\cdot F'_i(\vec{\omega}) - r\cdot F_i(\vec{\omega})\cdot F'^*_i(\vec{\omega}) + r\cdot r^*\cdot F_i(\vec{\omega})\cdot F_i^*(\vec{\omega})\right) + \lambda\cdot(r\cdot r^* - 1)\to\min. \tag{20}$$

Differentiating (20) relative to r^*, we get the following linear equation:

$$-\sum_{i=1}^{n} F_i^*(\vec{\omega})\cdot F'_i(\vec{\omega}) + r\cdot\sum_{i=1}^{n} F_i(\vec{\omega})\cdot F_i^*(\vec{\omega}) + \lambda\cdot r = 0. \tag{21}$$

From this equation, we conclude that

$$r = \frac{\sum_{i=1}^{n} F_i^*(\vec{\omega})\cdot F'_i(\vec{\omega})}{\sum_{i=1}^{n} F_i(\vec{\omega})\cdot F_i^*(\vec{\omega}) + \lambda}. \tag{22}$$

The coefficient λ can be now determined from the condition that the resulting value r should satisfy the equation (7). The denominator

$$\sum_{i=1}^{n} F_i(\vec{\omega})\cdot F_i^*(\vec{\omega}) + \lambda$$

of the equation (22) is a real number, so instead of finding λ, it is sufficient to find a value of this denominator for which $|r|^2 = 1$. One can easily see that to achieve this goal, we should take, as this denominator, the absolute value of the numerator, i.e., the value

$$\left|\sum_{i=1}^{n} F_i^*(\vec{\omega})\cdot F'_i(\vec{\omega})\right|. \tag{23}$$

For this choice of a denominator, the formula (21) takes the following final form:

$$r = \frac{\sum_{i=1}^{n} F_i^*(\vec{\omega})\cdot F'_i(\vec{\omega})}{\left|\sum_{i=1}^{n} F_i^*(\vec{\omega})\cdot F'_i(\vec{\omega})\right|}. \tag{24}$$

So, for multi-spectral images, in the presence of noise, instead of using the exact ratio (4), we should compute, for every $\vec{\omega}$, the optimal approximation

$$R(\vec{\omega}) = \frac{\sum_{i=1}^{n} F_i^*(\vec{\omega}) \cdot F'_i(\vec{\omega})}{\left|\sum_{i=1}^{n} F_i^*(\vec{\omega}) \cdot F'_i(\vec{\omega})\right|}. \tag{25}$$

Hence, we arrive at the following *algorithm*:

2.3 A new algorithm for determining the shift between two multi-spectral images

If we have images $I_i(\vec{\omega})$ and $I'_i(\vec{\omega})$ which correspond to different wavelengths, then, to determine the shift $\vec{a}$ between these two multi-spectral images, we do the following:

- *first, we apply FFT to the original images $I_i(\vec{x})$ and $I'_i(\vec{x})$ and compute their Fourier transforms $F_i(\omega)$ and $F'_i(\omega)$;*
- *on the second step, we compute the ratio (25);*
- *on the third step, we apply the inverse FFT to the ratio $R(\vec{\omega})$ and compute its inverse Fourier transform $P(\vec{x})$;*
- *finally, on the fourth step, we determine the desired shift $\vec{a}$ as the point for which $|P(\vec{x})|$ takes the largest possible value.*

2.4 A new algorithm for mosaicking two multi-spectral images

For rotation and scaling, we can use the same reduction to shift as for mono-spectral images:

- *First, we apply FFT to the original images $I_i(\vec{x})$ and $I'_i(\vec{x})$ and compute their Fourier transforms $F_i(\omega)$ and $F'_i(\omega)$.*
- *Then, we compute the absolute values $M_i(\vec{\omega}) = |F_i(\vec{\omega})|$ and $M'_i(\vec{\omega}) = |F'_i(\vec{\omega})|$ of these Fourier transforms.*
- *After that, we transform all the images $M_i(\vec{\omega})$ and $M'_i(\vec{\omega})$ from the original Cartesian coordinates to log-polar coordinates.*
- *Then, we use the above FFT-based algorithm to determine the corresponding shift $(\theta_0, \log(\lambda))$;*
- *From the corresponding "shift" values, 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, e.g., to images $I_i(\vec{x})$. As a result, we get new images $\widetilde{I}_i(\vec{x})$.*
- *Since we rotated and re-scaled one of the images, the images $\widetilde{I}_i(\vec{x})$ and $I'_i(\vec{x})$ 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 for determining shift: this time, actually to determine shift.*

As a result, we get the desired values of shift, rotation, and scaling; hence, we get the desired mosaicking.

Acknowledgments

This work was supported in part by NASA under cooperative agreement NCC5-209, by NSF grant No. CDA-9522207, by United Space Alliance, grant No. NAS 9-20000 (PWO C0C67713A6), and by the National Security Agency under Grant No. MDA904-98-1-0561.

The authors are thankful to Ann Q. Gates, Randy Keller, Vladik Kreinovich, Luc Longpré, Brian Penn, and Scott A. Starks for their help.

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] 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.