Abstract

We test three algorithms which can be used to register and mosaic panoramic photographs taken with a digital camera. For testing purposes, our sets of sample photographs have been taken using hand-held camera and sample images that we scaled and rotated for test purposes. The three algorithms we implemented were in papers [5,8,9] published on the Internet. All three algorithms use the Discrete Fourier Transform of the two images to register a pair of overlapping images. The first [5] uses the cross-power spectrum to find the rotation and translation of the second photo relative to the first. Log-polar coordinate transformation is used in the second and third algorithm. The second [8] uses the cross-power spectrum in conjunction with the log-magnitude spectra, transformed into log-polar coordinates, of the two images to find the angle of rotation and scale of the second photo, then after rotating and scaling it, uses the cross-power spectrum again to find the translation coordinates relative to the first photo. The third [9] transforms the two photos to log-polar coordinates, then cross-correlates the two photos in an iterative algorithm. This last algorithm [9] uses low-resolution samples of each photo and iteratively increases the resolution until it is restored to the original resolution.

1. Introduction

We began our search for photo-mosaic algorithms with a paper [1] published on the Internet. A link to it was provided by the class website set up by our teacher, Dr. Sarang Joshi, for the course, Image Processing. This paper has a large reference list from which we selected a number of papers to examine. These led us to a selection of other papers which we have included in our list of References.

Because of time constraints, we chose to limit our research to registration and mosaicing of panoramic photographs. We are assuming the photos in our sample set were taken without the camera's zoom level being altered between photos. This means that any scaling would come from the photographer moving closer or further from the scene during panning. Since these photos are of distant landscapes, movement of this type will produce little or no scaling. Thus, two of the algorithms [5,8] were chosen, since they either did not calculate scale [5] or could calculate only small scale values [8]. Because these photos are taken from far away, we were able to eliminate the need to compensate for projective deformation. Also, we did not need to compensate for mechanical deformation which occurs in microscopic mosaicing. This reduced the set of calculations which we had to make to get three parameters: scaling, rotation, and translation.

Two [8,9] of the three algorithms which we chose to implement, claimed to calculate all three transformations. The first algorithm [5] only calculates rotation and translation. In spite of this limitation, we decided to use it to study the use of the cross-power spectrum formula. This formula is also used in the second algorithm [8]. We chose the third
algorithm [9], because it claimed to calculate a larger range of scale, rotation, and translation transformations than the second algorithm [8].

During the implementation of the second two algorithms [8,9], we needed to research transformations from Cartesian coordinates to log-polar coordinates. We found two websites [6,7] and one paper [10] describing the transformation. One of the website articles [7] gave us the two formulas needed to implement log-polar transformation. The paper [10] described in detail a set of formulas for calculating the transformation coordinates, but, because of time constraints, we did not implement them. We also researched about feature based mosaic algorithm [11].

2. An algorithm to calculate rotation and translation for image registration [5]

This paper describes an algorithm that uses Fourier domain approach to match images that are translated and rotates with respect to one another.

The algorithm is as under:

1. Down sample the 2 images by 2 levels. Let the two sampled images be $I_1'$ and $I_2'$.
2. For $I = 1: \text{step} : 360$
   
   Rotate $I_2'$ by $I$ degrees. Let the rotated images be $I_2' \text{rot}$.
   
   Compute the Fourier transforms $FI_1'$ and $FI_2' \text{rot}$ of images $I_1'$ and $I_2' \text{rot}$ respectively.
   
   Let $Q(u,v)$ be the Phase correlation value of $I_1'$ and $I_2' \text{rot}$, based on $FI_1'$ and $FI_2' \text{rot}$

   $$Q(u,v) = \frac{FI_1'(u,v)FI_2' \text{rot}^*(u,v)}{\text{abs}(FI_1'(u,v)FI_2' \text{rot}^*(u,v))}$$

   Compute the inverse Fourier transform $q(x,y)$ of $Q(u,v)$

   Locate the peak of $q(x,y)$

   Store the peak value in a vector at position $i$.

   End for

3. Find the index of maximum peak from the values store in the vector in step 2.6. It gives the angle of rotation. Let it be $\theta'$.
4. Repeat sub-steps of 2 for $i = \theta' - \text{step} : \theta' + \text{step}$.
5. Find the angle of maximum peak from step 4. It becomes the angle of rotation.
   Let it be $\theta$. 

6. Rotate the original image $I_2$ by $\theta$. Let the rotated image be $I_{2,\text{rot}}$.

7. Phase correlate $I_2$ and $I_{2,\text{rot}}$. Let the result be $P(u,v)$.

8. Compute the inverse Fourier transform $p(u,v)$ of $P(u,v)$.

9. Locate the position $(tx,ty)$ of the peak of $p(u,v)$ which become the translation parameters.

10. Output the parameters $(tx,ty,\theta)$.

We implemented this algorithm. However, we had to rotate the second image 360 degree while computing the peak value at each step angle. This took lot of time and was computationally very inefficient.

Also, this algorithm only worked for images with translation and rotational shifts. It would not work for the scaled images.

3. An algorithm to calculate the Cartesian to log-polar transformation [7]

Given an image using Cartesian coordinates, $f(x,y)$, where $(x,y) \in [0,m-1] \times [0,n-1]$, calculate the log-polar transformation as follows:

1. Calculate the base of the logarithm:
   \[
   b = 10^{\log_{10}(m)/m}
   \]

2. Calculate the mapping of the log-polar coordinates in the original $m \times n$ image to an $m \times n$ Cartesian space where the log-radius $\rho$ is the domain and the angle $\theta$ is the range:
   \[
   \rho \in [\left\lfloor \frac{\pi}{2} \right\rfloor - 1] \quad \text{and} \quad \theta \in \left[0, \frac{2\pi}{n}\right]
   \]

3. Transform the image using some form of interpolation (we chose bilinear):
   \[
   f(\rho,\theta) = f\left(\rho \cos \theta + \frac{m}{2}, \rho \sin \theta + \frac{n}{2}\right)
   \]
4. An algorithm to calculate limited scale, rotation, and translation for image registration [8]

This algorithm is based on the description in the paper by Reddy and Chatterji [8]. In their paper, they describe how to combine the cross-power spectrum, log-magnitude, and log-polar coordinate transformation techniques to first calculate scale and rotation values. These values are then used to transform the second of two images to align with the first. Once the second image is transformed using these two values, the algorithm finally calculates the translation required to mosaic the two images together. Scale is limited to 1±0.8. Any scaling beyond this range of values cannot be accurately calculated. Since there is no iteration involved in this algorithm, its run time is proportional to the resolution (number of pixels) of the two images.

The algorithm in outline form

1. Given the two images, \( f_1(x,y) \) and \( f_2(x,y) \) where \((x,y)\in[0,m-1]\times[0,n-1]\), shift the center of each to the upper-left corner, then calculate their Fourier transforms:
   \[
   F_1(\xi,\eta) = \mathcal{F}[f_1(x,y)] \quad \text{and} \quad F_2(\xi,\eta) = \mathcal{F}[f_2(x,y)]
   \]

2. Transform the frequency spectra to their magnitude spectra:
   \[
   M_1(\xi,\eta) = \|F_1(\xi,\eta)\| \quad \text{and} \quad M_2(\xi,\eta) = \|F_2(\xi,\eta)\|
   \]
3. Transform the magnitude spectra to their log-magnitude spectra:
\[ LM_1(\xi, \eta) = \log_{10}(M_1(\xi, \eta)) \quad \text{and} \quad LM_2(\xi, \eta) = \log_{10}(M_2(\xi, \eta)) \]

4. Create a high-pass filter of the same dimensions as the two images using element-wise multiplication:
\[ H(\xi, \eta) = (1 - X(\xi, \eta))(2 - X(\xi, \eta)) \]
where:
\[ X(\xi, \eta) = [\cos(\xi \pi) \cos(\eta \pi)] \quad \text{and} \quad (\xi, \eta) \in [-0.5,0.5] \times [-0.5,0.5] \]

5. Shift the center of the log-magnitude spectra of the two images to the upper-left corner, then element-wise multiply them by the high-pass filter:
\[ LMH_1(\xi, \eta) = LM_1(\xi, \eta)H(\xi, \eta) \quad \text{and} \quad LMH_2(\xi, \eta) = LM_2(\xi, \eta)H(\xi, \eta) \]

6. Transform the filtered log-magnitude spectra of the two images to log-polar coordinates:
\[ LMH_1(\rho, \theta) = \text{cart2logpol}[LMH_1(\xi, \eta)] \quad \text{and} \quad LMH_2(\rho, \theta) = \text{cart2logpol}[LMH_2(\xi, \eta)] \]

7. Calculate the Fourier transforms of the log-polar forms of the two filtered log-magnitude spectra:
\[ FLMH_1(\mu, \nu) = \mathcal{F}[LMH_1(\rho, \theta)] \quad \text{and} \quad FLMH_2(\mu, \nu) = \mathcal{F}[LMH_2(\rho, \theta)]. \]
This step gives us the frequency spectra of the log-polar form of the log-magnitude spectra. These frequency spectra are required for the next step of this algorithm.

8. Calculate the cross-power spectrum from the two log-polar form, log-magnitude spectra:
\[ PSLM(\mu, \nu) = \frac{FLMH_1(\mu, \nu)FLMH_2(\mu, \nu)}{\|FLMH_1(\mu, \nu)FLMH_2(\mu, \nu)\|} \]

9. Calculate the inverse Fourier transform of the cross-power spectrum to get an impulse image:
\[ pslm(\rho, \theta) = \mathcal{F}^{-1}[PSLM(\mu, \nu)] \]

10. Locate the highest peak of the impulse image:
\[ (\rho_0, \theta_0) = \max(pslm(\rho, \theta)) \]
11. Convert these "coordinates" to the scale and rotation values to apply to the second image:

\[ b^\rho_0 \] is the scale, where \( b = 10^{\log_{10}(m)/m} \), and \( \frac{180\theta_0}{n} \) is the angle of rotation.

12. Rotate and scale the second image:

\[
\mathbf{frs}_2(x, y) = \begin{bmatrix}
    s & 0 \\
    0 & s
\end{bmatrix}
\begin{bmatrix}
    \cos \theta_0 & \sin \theta_0 \\
    -\sin \theta_0 & \cos \theta_0
\end{bmatrix}
\mathbf{f}_2(x, y)
\]

13. Calculate the Fourier transform of the scaled/rotated second image:

\[
\mathbf{FRS}_2(\xi, \eta) = \mathcal{F}[\mathbf{frs}_2(x, y)]
\]

14. Calculate the cross-power spectrum from the frequency spectra of the first image and the scaled/rotated second image:

\[
\mathbf{PS}(\xi, \eta) = \frac{\mathbf{F}_1(\xi, \eta)\mathbf{FRS}_2^*(\xi, \eta)}{\|\mathbf{F}_1(\xi, \eta)\mathbf{FRS}_2(\xi, \eta)\|}
\]

15. Calculate the inverse Fourier transform of the cross-power spectrum which is an impulse image:

\[
\mathbf{ps}(x, y) = \mathcal{F}^{-1}[\mathbf{PS}(\xi, \eta)]
\]
16. Locate the highest peak of this second impulse image to get the translation coordinates of the displacement of the second image where it overlaps the first:

\[(x_0, y_0) = \max(p(x, y))\]

Sample runs:
1. water lilies, 100% overlap
2. water lilies, 25% overlap

5. An algorithm to calculate any scale, rotation, and translation for image registration [9]

This algorithm couples the log-polar transform with a non-linear least squares algorithm to estimate the affine transformation parameters.
Any point \((x, y)\) can be represented in polar co-ordinates:

\[
r = \sqrt{(x - x_c)^2 + (y - y_c)^2}
\]

\[
a = \tan^{-1}\left(\frac{y - y_c}{x - x_c}\right)
\]

where \(r\) denotes radial distance from the center \((x_c, y_c)\), \(a\) denotes angle, and \((r, a)\) constitute the polar coordinate system.

The rotational shift in the Cartesian space of an Image would simply be the translational shift in \(y\) - direction in the log-polar form. Similarly, the scale in the Cartesian form would be the translational shift in the \(x\) - direction in the Log-polar form. Hence to depict the affine transform parameters of the image, it would suffice to know the translation parameters in the log-polar plane. The following algorithm is able to find the parameters that relate the two images in the Cartesian plane:

1. Crop central region \(I_1'\) from \(I_1\)
2. Compute \(I_{1p}'\), the log-polar transformation of \(I_1'\)
3. For all positions \((x, y)\) in \(I_2\):
   - Crop region \(I_2'\)
   - Compute \(I_{2p}'\)
   - Cross-correlate \(I_{1p}'\) and \(I_{2p}'\) \(\rightarrow (dx, dy)\)
   - If maximum correlation, save \((x, y)\) and \((dx, dy)\)
4. Scale \(\leftarrow dx\)
5. Rotation \(\leftarrow dy\)
6. Translation \(\leftarrow (x, y)\)

We implemented the above algorithm. However, since this algorithm searches exhaustively for the parameters in the spatial space, it was not computationally efficient. The For loop in step 3 iterates over each pixel in the second image which we would like to mosaic with the first image. Hence, it took more than 8 hours to come up with the results, that is, the three parameters.

To make this algorithm computationally efficient, we first tried to narrow down our search space by first finding out the region where the two images overlapped maximum. We used least squares algorithms as in Assignment 2 to find this initial guess of maximum overlap. Then, we applied the above algorithm in this initial guess. This technique made the algorithm more efficient.

One of the ways to do image mosaic is through feature detection algorithms. Due to time constraints we didn’t implement these kinds of algorithms but we selected a paper and analyzed it. The paper uses geometric corners to build a 2D mosaic from a set of images.

A point is represented using homogeneous coordinates, a point \(m\) in an image will be represented by the vector \((x\ y\ z)^t\), with Cartesian coordinates \((x/z\ y/z)^t\). The images of a plane seen from two points of view are related by a homography \(H\). Thus a point \(m_1\) has a corresponding point \(m_2\) defined by:

\[
\begin{bmatrix}
    x_2 \\
    y_2 \\
    z_2
\end{bmatrix} =
\begin{bmatrix}
    h_{00} & h_{01} & h_{02} \\
    h_{10} & h_{11} & h_{12} \\
    h_{20} & h_{21} & h_{22}
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    y_1 \\
    z_1
\end{bmatrix}
\]

The algorithm is described in the following three steps:

1. Extract the points using Harris’ detector. Keep the points with large corner response function (greater than threshold).
2. Compute all possible homographies defined by pairs of four-tuples of points of interest.
3. Find among the homographies, the best one.

To check the \(H_k\) correlate the intensity values at all points of interest \(p_{1i}\) of the first image with those of their corresponding points \(H_k p_{1i}\), the best one will be the maximum of the zero mean cross-correlation. See formula:

\[
H_{opt} = \max_{H_k} \left( \frac{1}{n} \sum ZNCC(p_{1i}, H_k p_{1i}) \right)
\]

The results obtained with an overlap of 70%, are satisfactory they selected a small quantity of points, 12 points and 4 in the overlap area. The computation time is fairly long, 12 minutes. This is because there are a lot of homographies to check. For a smaller overlap of 50%, 20 points were extracted. This represented 100 millions of homographies, to check this it required a few hours of computation. We can see that this is not an efficient method especially for pictures whose area of overlap is very small.

7. Literature Review and Work Division among Our Group’s Members

Group Members:

- Maria Lorena Carlo
- Kathlea Quebbeman
- Reet Dipti Dhakal
We started out by studying the AN INTRODUCTION TO IMAGE MOSAICING [1] paper as a first step for this project. We then went over to the reference papers it pointed out. We selected nine relevant papers from the reference section and then each of us studied three papers. This was a good starting point.

We studied additional papers and found that there were a lot of ways of accomplishing the task of image registration. Some papers were very good on theoretical points whereas some had presented more detailed algorithms.

The first algorithm we implemented was described in section 3. Reet Dipti Dhakal implemented this algorithm. It was computationally very inefficient because we had to rotate the second image one degree at a time. Apart from this, this algorithm was able to calculate the translational and rotational offset parameters.

We needed an algorithm that could convert the image in Cartesian plane to the log-polar plane. The function that performed this task "cart2logpol" as described in section 4 was implemented by Maria Lorena Carlo.

We then moved on to implementing the algorithm described in section 5. Kathy Quebbeman implemented this algorithm. This algorithm was able to get the translation, rotation and scale parameters. This algorithm was more computationally efficient than the first algorithm that we implemented and was able to produce the outputs in less than a minute.

We then again thought of implementing the algorithm described in section 6. This algorithm also used the same "cart2logpol" function to convert the image (rotated and scaled) in the Cartesian plane into the log-polar plane. Reet Dipti Dhakal and Maria Lorena Carlo implemented this algorithm. However, this algorithm took very long time as it had iterative approach and did exhaustive searching in the log-polar space as described in section 6.

We also went further and researched the Feature based Image Mosaicing Technique. This was described in section 7 which was written by Maria Lorena Carlo.

8. Conclusion

As explain in section 7, the most efficient algorithm was the phase correlation with log-polar mapping which is described in section 5. It requires only two phase correlation calculations to get the three parameters: scale factor, angle of rotation and translation coordinates.

9. Acknowledgements

We owe a debt of gratitude to our professor, Dr. Sarang Joshi for his guidance in this project.

10. References


