# Optimal Point Correspondence for Image Registration in 2D Images 

Kitae $\mathrm{Bae}^{1}$ and Hyoungjin $\mathrm{Kim}^{2 *}$<br>${ }^{l}$ Department of New Media, Korean German Institute of Technology<br>661, Deungchon-Dong, Gangseo-Gu, Seoul, 157-033, Republic of Korea<br>${ }^{2}$ Department of IT Applied System Engineering, Chonbuk National University<br>567, baekje-daero, Jeonju-si, Jeollabuk-do, 561-756, Republic of Korea<br>ktbae@kgit.ac.kr, kim@jbnu.ac.kr<br>*Corresponding Author


#### Abstract

The conventional search methods have computational complexity problem and imprecision problem in correspondence matching process. To resolve these problems, we propose how to effectively make feature space (distance map) and how to rapidly search the optimal point correspondence. The proposed distance map named Voronoi distance map is a 2-Dimensional surface that contains the distance information between each element(x-y coordinates) of image and the nearest feature point. The proposed distance map is efficiently created based on the priority-based calculation algorithm. The general distance calculation algorithm has a time complexity of $O\left(w^{*} h^{*} n\right) .(w=$ width, $h=$ height, $n=$ the number of feature points) whereas the priority-based distance calculation algorithm is a effective method with a computational cost of $O\left(w^{*} h^{*} \log n\right)$. Also, The partition search algorithm is a efficient method that can detect corresponding points very rapidly because this method can reduce the search range by a quarter at a time. Experimental results show that the proposed method outperforms conventional methods in reducing computation time and detecting the optimal correspondence.


Keywords: Feature space, Point correspondence, Priority-based calculation, Partition search, Image Matching, Image registration, Image mosaic

## 1. Introduction

Image registration is one of the basic techniques for image-based rendering and a general method for obtaining a wide field of view image with high resolution. Image mosaic techniques have attracted much interest in the vision community in the last few years because of their broad range of applications in various fields, including image compression, object detection, virtual reality, and video conferencing [1]. In a simple image registration approach, images are captured using a moving camera, and then overlapping areas are detected and stitched together to obtain a higher resolution or larger image.

In order to perform image registration successfully, it is first necessary to detect a correct registration area between two images and then combine them smoothly without creating discontinuity or distortion during the registration process. However, it is very difficult to find correct registration areas and compose perfectly images without seams.

Therefore, many image registration techniques have been developed using processes such as global registration techniques [2, 3], local registration techniques, geometrical transformation methods for warping, and blending techniques for composition [8].

Finding corresponding points in two images is the key issue in the image registration process and has been studied intensively for many years. However, there is still no unified method for solving the corresponding point problem because none of them are robust and complete. The methods for finding corresponding points are mainly divided into two types of correspondence: feature-based correspondence [4, 5, 9, 11]; area-based correspondence [6, 7, 10]. In area-based correspondence, every pixel is taken into consideration, while in feature-based correspondence, only a small set of feature points is considered. Because the feature-based correspondence method can obtain feature points with good correspondence using less computation, it is more popular than the region-based method.

With these issues in mind, our paper has the following purposes. We present a feature space that can determine whether or not the matching result is correct. We present a fast image composition algorithm using a Voronoi map. We present the feature space efficiently. We present a method for automating model selection using statistical information without human interaction. We minimize the matching error using statistical and image color information.

In this paper, we present a new method for detecting the correspondence between two images by using the Voronoi distance and partition search algorithm for high speed image registration. The Voronoi distance calculation algorithm creates a Voronoi surface that contains the distance information for the feature points in the reference image, while the partition search algorithm searches the Voronoi surface for the corresponding points overlapping with the model patch. The Voronoi distance method is easy to implement and run in a reasonable amount of time, and the result is very robust.

This paper is organized as follows. In Section 2, we present an overview of the algorithm and describe the Voronoi distance calculation algorithm used to create the Voronoi surface. We also explain the partition search algorithm that uses queues. Section 3 includes the experiments and results and, finally, Section 4 presents our conclusions and several issues for future research.

## 2. Overview of Algorithm

Figure 1 shows the overall process used by our algorithm. Throughout the entire process, we assume that every image is overlapped by the adjacent image by at least one fourth of the view area. This constraint is used for the purpose of reducing several effects coming from the abrupt rotation and scale change of the input images and computing the robust correspondence between images.

The procedure is as follows: We first extract feature points from the two images using the SUSAN corner detector, and then construct a Voronoi surface, which includes distance information for the feature points. Next, we select the portion of the input image (model patch area) in which the feature points have the maximum variance of position to estimate the correspondence area of the input image. Shifting the model patch over the input image and calculating the minimum Voronoi distance allow us to obtain the corresponding feature set. Next, we perform the partiion serch algorithm to detect the optimal correspondence. In the last step, those images are stitched into one large scale image by applying the transformation matrix. Finally, we can acquire high-quality stitching image.


Figure 1. The Overall process of our system

### 2.1 Extracting feature point and Selecting Model patch area

The correspondence between images can be detected by comparing the brightness change around peculiar feature points. We use the SUSAN corner detector to detect the feature points. This is a rapid method based on a brightness comparison. It has excellent reliability and noise tolerance.

First, we assume that every image is registered in a panoramic style and two images will be located side-by-side.

We detect feature points in three quarters of the right side of the base image, the left image, and one quarter of the left side of the model image, the right image. In particular, the feature points of the reference image are defined as base points, where the distance values are all zero on the Voronoi surface. The model patch area is defined as the area $(38 \times 38)$ that has the largest variance in the coordinates of the feature points in the input image. The variance ( $\sigma^{2}$ ) in the coordinates of the feature points is denoted as follows.

$$
\begin{equation*}
\sigma^{2}=\sum_{i=1}^{n}\left(x_{i}-\mu_{x}\right)^{2}+\sum_{i=1}^{n}\left(y_{i}-\mu_{y}\right)^{2} \tag{1}
\end{equation*}
$$

where $x_{i}$ is the x coordinate of the i-th feature point, $y_{i}$ is the y coordinate of the i-th feature point, and $\mu_{x}$ and $\mu_{y}$ are the averages of $x_{i}$ and $y_{i}$.

Figure 2(a) shows an example of the feature points in the base image for a Voronoi surface, and Figure 2(b) shows an example of the model area evaluated as the area that has
the largest variance in the coordinates of the feature points in the input image. The model patch area is the rectangular area in Figure 2(b).


Figure 2. Example of feature points detected by SUSAN corner detector

### 2.2 Extracting feature point and Selecting Model patch area

A Voronoi surface is an abstract surface matched with the model area in order to detect similar areas in the base image. More specifically, a Voronoi surface is a 2D surface (or 2D array) that contains the distance information between each element ( $\mathrm{x}, \mathrm{y}$ ) of the base image and the nearest feature point. The distance information for a Voronoi surface is called Voronoi distance. Each element coordinate of a Voronoi surface is equal to that of the reference image. Thus, we must calculate the Voronoi distance in order to create a Voronoi surface. Figure 3 shows an example of a Voronoi surface containing Voronoi distance information. The point where the distance value is 0 on the Voronoi surface symbolizes a feature point. In Figure 3(b), know that the feature points in the reference image are represented by zero. we know that the feature points in the reference image are represented by zero. Figure 3(a) shows that each element of a reference image becomes brighter as its distance from a feature point increases.


Figure 3. Example of Voronoi surface whti Voronoi distance information

The simplest method for calculating the Voronoi distance is to detect the smallest distance information value when comparing each element of a reference image with the adjacent feature points. By using this method, we can see that the distance value of a diagonal coordinate around a feature point is equal to that of the x or y coordinate, as in Figure 3(b).

This algorithm is easy to understand. However, the algorithm is not very efficient because its time complexity is $O\left(w^{*} h^{*} n\right)$ ( $\mathrm{w}=$ width, $\mathrm{h}=$ height, $\mathrm{n}=$ the number of feature points). To improve the efficiency of the algorithm, we propose a priority order-based Voronoi distance calculation algorithm, which has a time complexity of $O\left(w^{*} h^{*} \log n\right)$.


Figure 4. Priority order-based Voronoi distance calculation algorithm

Figure 4 shows the priority order-based Voronoi distance calculation algorithm, which is an improvement over the existing method. The process used by the algorithm is as follows, First, calculate the row, column, and diagonal priorities for the feature points: (1) Searching for the feature point with the Y coordinate in the row priority order, (2) Searching for the feature point with the X coordinate in the column priority order, (3) Searching for the feature point with the $\mathrm{X}+\mathrm{Y}$ coordinates, (4) Calculating each difference value, (5) Selecting the maximum value among the difference values, (6) Setting the selected minimum value as the Voronoi distance value.

## 3. Partition Search Algorithm

We must detect candidate registration areas in the reference image using the model patch area and Voronoi surface. This is the same method used to detect the coordinate $(t)$ that has the sum of the Voronoi distances within threshold ( $\tau$ ), as in equation (2).

$$
\begin{equation*}
f(t)=\sum_{b_{i} \in B}^{N} d\left(a_{i}\right)=\sum_{\left(x_{v_{i}}, y_{b_{i}}\right) \in B}^{N} d\left(x_{t}+x_{b_{i}}, y_{t}+y_{b_{i}}\right) \leq \tau \tag{2}
\end{equation*}
$$

In equation (2), $t$ is the left-top base coordinate of the Voronoi surface when the Voronoi surface is matched with the model area in Figure 4. Thus, $f(t)$ is the sum of the Voronoi distances of the coordinates corresponding to the feature points of the model area based on $t$. Here, $b_{i}$ represents the relative coordinates of the feature points based on the left-top coordinate $(0,0)$ of the model area, $a_{i}$ is the coordinate of the Voronoi surface corresponding to $b_{i}$, and $d\left(a_{i}\right)$ is the Voronoi distance of $a_{i} . N$ is the number of feature points in the model area. $x_{t}$ and $y_{t}$ are the $x$ and $y$ coordinates of $t$, respectively, while $x_{b_{i}}$ and $y_{b_{i}}$ are the $x$ and $y$ coordinates of $b_{i}$, respectively. In equation (2), if $\tau$ is zero, $f(t)$ must be zero. This means that the feature points of the model area correctly corresponding to those of the reference image, as in Figure 5.


Figure 5. Sum of Voronoi distances on Voronoi surface correctly corresponding to feature points of model patch area

As $f(t)$ is a linear equation, if $f(t)$ is greater than zero, then it is necessary to move $t$ and find $t^{\prime}$ where $f\left(t^{\prime}\right)$ is 0 . When $t$ is moved to $t^{\prime}$, the absolute value of $f(t)-f\left(t^{\prime}\right)$ is
forced to be a value that is less than or equal to the Euclidean distance between $t$ and $t^{\prime}$ multiplied by $N$ (the number of feature points in the model area). This is denoted as follow:

$$
\begin{equation*}
f(t)=\sum_{b_{i} \in B}^{N} d\left(a_{i}\right)=\sum_{\left(x_{b_{i}}, y_{b_{i}}\right) \in B}^{N} d\left(x_{t}+x_{b_{i}}, y_{t}+y_{b_{i}}\right) \leq \tau, \tag{3}
\end{equation*}
$$

In equation (3), if $t^{\prime}$ is the goal coordinate, then $f\left(t^{\prime}\right)$ becomes zero. Therefore, we can rewrite (3) as follows.

$$
\begin{equation*}
f(t) \leq s \times N=\tau \tag{4}
\end{equation*}
$$

In equation (4), $s$ is the transfer distance, namely the Euclidean distance $\left(\sqrt{\left(x_{t}-x_{t^{\prime}}\right)^{2}+\left(y_{t}-y_{t}\right)^{2}}\right)$. This means that if $f(t)$ is less than or equal to $s \times N$, then there is a goal coordinate within $s$ around $t$. In other words, if $f(t)$ is greater than $s \times N$, then there is no goal coordinate within $s$ around $t$. Using this property of equation (4), we can develop a partition search algorithm that uses queues, as in Figure 6.

In the algorithm of Figure 6, the shape of the initial search area is adjusted to a square, and this square area is divided into 4 squares. We calculate $f(t)$ in the center point coordinate $t$ of each square area. If $t$ is greater than the threshold, $\tau$, then the square area is excluded from the search area. Otherwise, this square area is divided into 4 squares again, the new center point coordinate $t_{c}$ is calculated, and the search work is repeated until the width of the search area become too small. The new center point coordinate is the left-top coordinate corresponding to the feature points of the model area. It is denoted as follows,

$$
\begin{equation*}
t_{c}=\left(\frac{x_{t}^{\text {low }}+x_{t}^{h i g h}}{2}, \frac{y_{t}^{\text {low }}+y_{t}^{h i g h}}{2}\right), \tag{5}
\end{equation*}
$$

where $\left(x_{t}^{\text {low }}, y_{t}^{\text {low }}\right)$ is the left-top coordinate of the square area, and $\left(x_{t}^{\text {high }}, y_{t}^{\text {high }}\right)$ is the right-bottom coordinate of the square area. Figure 6 is an example of the square search area used in the repeated search process.


Figure 6. Example of search area in search process
In the repeated search process, threshold $\tau$ is a very important factor that determines whether or not the search area is included, and the search time of our algorithm depends on it. In equation (4), as $\tau$ is $s \times N$, where $N$ is a constant that represents the number of feature points in the model, the variable $s$ is the determining factor. In the search area, $s$ is the distance from the center point coordinate to each side or vertex. Figure 7 shows an example of the distance from the center point of the search area, the square area.

In Figure 7, we can understand why the maximum $s$ is $\frac{w}{2} \sqrt{2}$.


Figure 7. Distance from center point to each side or vertex.
Therefore, the threshold is defined as follows:

$$
\begin{equation*}
\tau=N \times \frac{w}{2} \sqrt{2} \tag{6}
\end{equation*}
$$

As the algorithm of Figure 8 is performed until the search width $(w)$ is less than or equal to 2 , some candidate goal coordinates of the registration area appear.

| A B |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 4 | 3 | 2 | 2 | 2 | 2,2 | 3 | 45 | 55 | 6 | 0 |
| 4 | 3 | 2 | 1 | 1 | 1:2 | 3 |  | 45 | 6 | 0 : Feature Point: 2 |
| 4 | 3 | 2 | 1 | 0 | 1:2 | 3 | 34 | 45 | 6 | Search rectangle width : 6 |
| 4 | 3 | 2 | 1 | 1 | 1.2 | 2 | 34 | 45 | 6 | Search retangle widh : |
| 4 | 3 | 2 | 2 | 1 | 1,1 | 2 | 34 | 45 | 6 | $(\mathrm{l})=3 \times \sqrt{2} \times 2=8.484$ |
| 3 | 3 | 3 | 2 | 1 | $0 \cdot 1$ | 2 | 34 | 45 | 6 | $A: f(t)=1+1=2<\tau$ |
| 3 | 2 | 2 | 2 | 1 | 1:1 | 2 | 34 | 45 | 6 | $B: f(t)=4+5=9\rangle$ |
| 2 | 1 | 1 | 1 | 1 | 2.2 | 2 | 34 | 45 | 6 | $1+2=3<\tau$ |
| 2 | 1 | 0 | 1 | 2 | 3,3 | 3 | 34 | 45 | 6 | $4+5=9\rangle$ ¢ |
| 2 | 1 | 1 | 1 | 2 | $3: 4$ | 4 | $4{ }^{4} 4$ | 45 | 6 | (First test) |
| 2 | 1 | 2 | 22 | 2 | 3.4 | 5 | 55 | 55 | 6 | Search Area : A, C |
| 3 | 3 | 3 | 3 | 3 | 3.4 | 6 | 66 | 6.6 | 6 | Omitted Area: B, D |
|  |  |  |  |  |  |  | D |  |  |  |

Figure 8. Example of partition search algorithm test.

## 4. Experimental Results

Our experiment is implemented on a PC (Pentium Core $2.4 \mathrm{G}, 2 \mathrm{G}$ ). In this experiment, still image registration is tested, mainly by using outdoor images.

Table 1 shows a comparison of the Voronoi distance calculations. The results in Table 1 show that the new Voronoi distance calculation algorithm is excellent. Figure 9 shows the results of Table 1 .

Table 1. The comparision result(simple voronoi calculation \& priority orderbased voronoi)

|  | test <br> image <br> 1 | test <br> image <br> 2 | test <br> image <br> 3 | test <br> image <br> 4 | test <br> image <br> 5 | test <br> image <br> 6 | test <br> image <br> 7 | test <br> image <br> 8 | test <br> image <br> 9 | test <br> image <br> 10 |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| feature points | 2177 | 1430 | 872 | 1698 | 1396 | 2128 | 457 | 171 | 1857 | 3390 |
| Voronoi distance <br> calculation (simple) | 4.66 | 3.05 | 1.75 | 3.63 | 3.14 | 4.27 | 1.03 | 0.38 | 3.83 | 6.51 |
| Voronoi distance <br> calculation (new) | 0.34 | 0.27 | 0.22 | 0.25 | 0.23 | 0.28 | 0.14 | 0.14 | 0.28 | 0.30 |



Figure 9. The Comparison graph of Voronoi distance calculations

The results in Figure 10 show that most check fields receive the effects of the number of feature points, except the Voronoi distance and partition search fields.


Figure 10. Experimental results for image composing tested with 20 pairs of sample images

Figure 11 shows the experimental results in comparison with other methods using 15 pairs of sample images. The results in Figure 11 show that the Voronoi distance calculation algorithm is better than other methods.


Fig. 11 Results of comparison experiment with other methods (processing time (sec))

Figure 12 shows an example of successfully stitching together two outdoor still images. The results of the experiment also show that the processing time for our algorithm is comparatively efficient.


Figure 12 The Composite image implemented by the proposed algorithm. (Two outdoor images)

## 5. Conclusion

In conclusion, we would like to state the following three points. 1) We proposed a new method that can moderately reduce the processing time of the image registration procedure and exactly detect the corresponding points. 2) We proposed a method that can compose a feature space efficiently for easily computing the distance. 3) We automated the model selection using statistical and image information. An experiment showed that the algorithm proposed in the paper is effective. The priority-based Voronoi distance algorithm is a new method with a computational cost of $O\left(w^{*} h^{*} \log n\right)$, rather than $O\left(w^{*} h^{*} n\right)$. The partition search algorithm is also a new method that can detect corresponding points very quickly because the search area is divided into four squares every time. In future work, we will develop a 3D feature space to solve the problem of 3D correspondence.

## References

[1] D. Burschka, D. Cobzas, Z. Dodds, G. Hager, M. Jagersand and K. Yerex, "Recent Methods for Image-based Modeling and Rendering", IEEE Virtual Reality tutorial 1, (2003) March.
[2] B. S. Reddy and B. N. Chatterji, "A FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration", IEEE Trans. Image Processing, vol. 5, no. 8, (1996).
[3] M. Y. Chong, S. H. Choi, K. T. Bae and C. W. Lee, "Real-time Image Mosaic Using DirectX", Journal of Korea Information Processing Society, vol. 10-B, no. 7, (2003) December, pp. 803-810.
[4] N. Chiba, H. Kano, M. Higashihara, M. Yasuda and Osumi, "Feature-Based Image mosaic", MVA'98 IAPR Workshop on Machine Vision Applications, (1998), pp. 5-10.
[5] T. Hu and M. Huang, "A New Stereo Matching Algorithm for Binocular Vision", International Journal of Hybrid Information Technology, vol. 3, no. 1, (2010) January, pp. 1-8.
[6] G. Borgefors, "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm", IEEE Trans, Pattern Analysis and Machine Intelligence, vol. 10, no. 6, (1988), pp. 849-865.
[7] M. Dubuisson and A. K. Jain, "A Modified Hausdorff Distance for Object Matching", Proc. 12th Int'l Conf. Pattern Recognition, (1994) October, pp. 566-568.
[8] D. Huttenlocher and P. Torr, "Efficient Algorithms for Matching", Int'l Conf ICCV, (2003).
[9] P. Premaratne and F. Safaei, "Feature based Stereo Correspondence using Moment Invariant", Proc. 4th Int'l Conf. ICIAFS 2008, (2008) December, pp. 104-108.
[10] L. Guo, Y. Xia and J. Yang, "Adaptive Search Region Fast Area-Based Stereo Correspondence", Proc. 2008 Congress on Image and Signal Processing, vol. 2, (2008), pp. 310-313.
[11] Y. Wang and C. -S. Chua, "Robust face recognition from 2D and 3D images using structural Hausdorff distance", Image and Vision Computing, vol. 24, no. 2, (2006) February, pp. 176-185.
[12] M. Keyvanpour and R. Tavoli, "Document Image Retrieval: Algorithms, Analysis and Promising Directions", International Journal of Software Engineering and Its Applications, vol. 7, no.1, (2013) January, pp. 93-106.
[13] P. P. Shete, P. P. K. Venkat, D. M. Sarode, M. Laghate and S. K. Bose, "Applying Object Oriented Design Patterns to CUDA based Pyramidal Image Blending - An Experience", International Journal of Software Engineering and Its Applications, vol. 6, no. 2, (2012) April, pp. 47-62.
[14] J. Lee and H. W. Byun, "Image Deformation using Moving Least Squares based on Unequal Grids", International Journal of Software Engineering and Its Applications, vol. 6, no. 3, (2012) July, pp. 43-50.

## Authors



## Kitae Bae

He received M.S. degree in Department of Computer Engineering and Ph.D. degrees in Department of Computer Information Engineering, Chonnam National University respectively. He currently an associate professor in Department of New Media in Korean German Institute of Technology

Research interests: Computer Vision, Image Processing, Smart Media, Affective Computing, HCI, Natural Interface


## Hyoungjin Kim

He received M.S. and ph.D. degrees in Department of Information and Communication Engineering from Kunsan National University. He currently an associate professor in Department of IT Applied System Engineering, Chonbuk National University.

Research interests: Multimedia Systems, Ubiquitous, Sensor Networks, Computer Vision, Image Processing, Smart Media

International Journal of Multimedia and Ubiquitous Engineering Vol.8, No. 6 (2013)

