Complexity Analysis of Featured-Based Image Matching

Hassan Hajjdiab
College of Engineering and Computer Science
Abu Dhabi University
Abu Dhabi, UAE, P.O.Box 59911
Email: hassan.hajjdiab@adu.ac.ae

Robert Laganière
VIVA Research Laboratory
School of Information Technology and Engineering
University of Ottawa, Ottawa, ON, Canada, K1N 6N5
Email: laganier@site.uottawa.ca

***Abstract*—**A fundamental problem in computer vision is to map a pixel in one image to a pixel in another image of the same scene. This is called the image correspondence problem. Many algorithms have been proposed in literature to solve the problem, however no rigorous analysis has been conducted to study the parameters that contribute to the complexity. The main objective of this paper is to investigate the complexity of matching feature points (*pixels*) between multiple views of a given scene. The advantage of this work is that a formal analysis is introduced to explore the relationship between the minimum Euclidean distance between the feature points detected on the image and the area of the search region on the overall computational time complexity of the problem.

***Index Terms*—**Correspondence problem, Complexity analysis, Image matching

I. Introduction

The image matching problem, also known as the correspondence problem[1], [2], [3], is one of the most challenging research tasks in the computer vision research. Image matching is a basic and fundamental step and not the final goal of a computer vision system. Many problems such as camera calibration [4], [5], [6], [7], 3D object reconstruction [8], [9], [10], obstacle detection [11], [12], motion estimation [13] and object tracking [14], [15] require solving the correspondence problem as an initial step in the processing sequence of steps. The matching problem can be defined as the establishment of the correspondence between features extracted from two or more images of the same scene. However, the matching problem is a well known ill-posed problem [9]. The solution may not exist, a point in one image may not have a corresponding match in the other image due to occlusion or the solution may not be unique, there may be more than one match due to scenes with repetitive patterns. To solve such problems a wider search space may be required that may dramatically impact the numerical computation cost and thus the overall complexity of the algorithm. The main contribution of this paper is that it describes the relationship between the search space, the features extracted and the total complexity of the matching process.

The paper is organized as follows: Section II presents an overview of Feature-Based Image Matching algorithms, Section III describes the complexity of the problem and finally Section IV is the conclusion

II. Feature Based Image Matching

Feature based matching starts by detecting some features on each of the input images. Features may a global property of the image like the average gray value or a local property like points, edges or circles within the image. The detected features should be invariant to geometric transformation and radiometric changes. If a feature point is detected on one image the corresponding feature point should be detected on the other image. The feature should be distinct with respect to other features in the same image. Feature extraction algorithms [16], [17], [18], [10] usually start by using a local operator, also known as an interest operator, to extract a point feature.

A circular or rectangular windows are often used in all directions to calculate certain attributes like local change in grey level intensity. The feature point attribute is then compared with a certain threshold to decide whether to keep it or not. To eliminate tight clustering of detected feature points, a window to suppress the local non-maxima is used. This will ensure that the detected features are at least certain number of pixels from each other. For more thorough introduction on feature extraction the reader is referred to [18], [10].

The threshold and the inter-corner Euclidean distance in pixels are parameters that can be set by the user, this may impact the number of features being extracted. As an example the images in Fig. 1 shows features extracted using Harris corners detector [16] with minimum inter-corner Euclidean distances of 6 pixels, the number of corners detected is 1136. Obviously, the number of corner points detected is inversely proportional to the inter-corner distance. The number of corner points detected on the same image (Fig. 1) for inter-corner distances of 9 and 12 pixels is 698 and 491 respectively.

III. Algorithm Analysis

After the feature extraction step comes the matching step. In this paper the focus is on studying the time complexity of such step. The problem is modeled as shown in Fig. 4, a feature point on Image 1 is to be matched with all feature points within a circle of radius R in Image 2. In this analysis we assume following a RANSAC-like scheme [19] in the matching process. A feature point (call it $\mathbf{p}$) is randomly selected from the first image at pixel location $(x_\mathbf{p}, y_\mathbf{p})$, the candidate matches on the second image (Image 2) are located

Fig. 1. Feature Extraction using an image of 640×480 pixels: 1136 features were detected with inter-corner distance of at least 6 pixels. For inter-corner distances of 9 and 12 pixels, the detected feature points are 698 and 491 respectively.

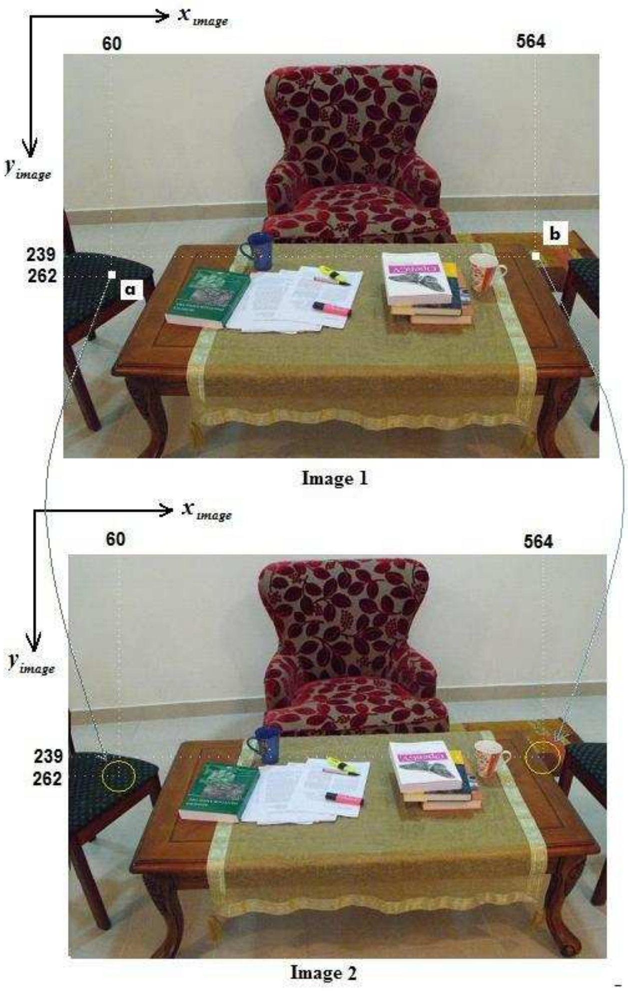


Fig. 2. Narrow baseline feature matching: Feature point **a** in Image 1 may have more than one match in Image 2 due to repetative pattern. Image point **b** in Image 1 has no corresponding match in Image 2 due to occlusion.

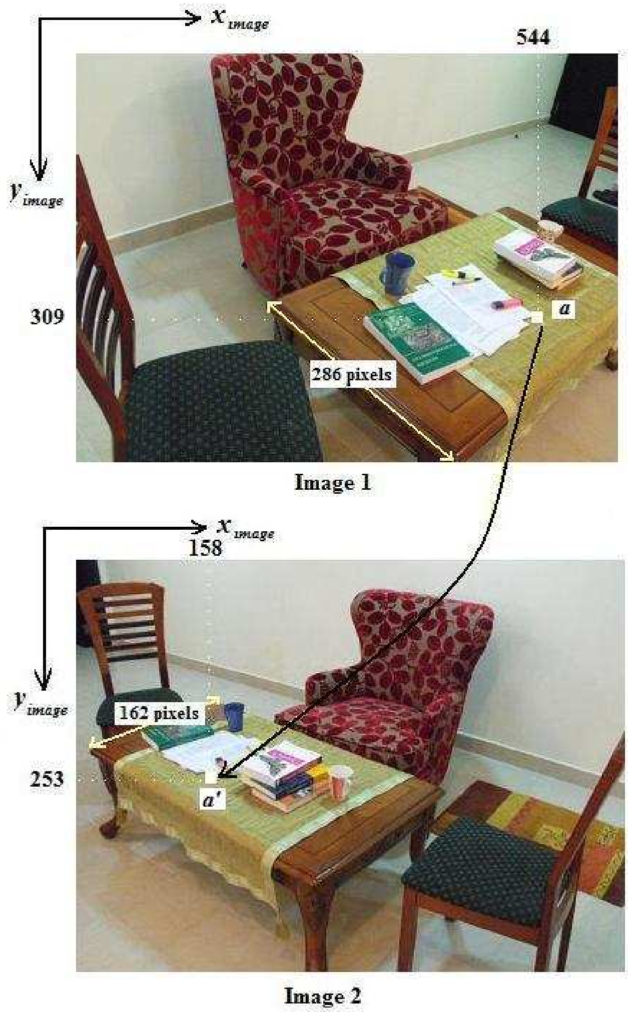
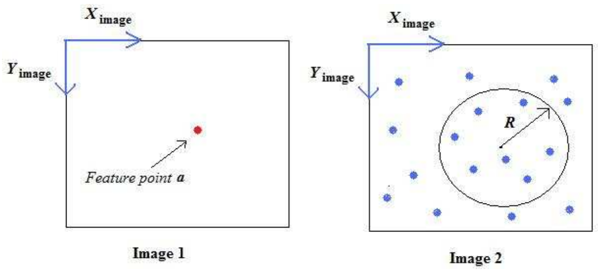


Fig. 3. Wide baseline feature matching: To match feature point **a** in Image 1 a wide search region is needed. The Euclidean space in not preserved, table width in 286 pixels in Image 1, however it is 162 pixels in Image 2.

within circle of radius R and center $\mathbf{C}(x_{\mathbf{C}}, y_{\mathbf{C}})$. For narrow baseline image matching [1] the location of the center $\mathbf{C}$ may be selected to be at $(x_{\mathbf{p}}, y_{\mathbf{p}})$. Feature point **p** is then compared with all feature points within the circle in Image 2. If the difference is sufficiently low, then the points are considered to be good candidate matches. There may be no match or more than one matches due to the nature of the problem being ill-posed as mentioned in Section I. As an example, consider point $\mathbf{a}(60, 262)$ in Fig. 2 is matched using a circle $\mathbf{C}(60, 262)$ of radius $10pixels$, in this case there may be more than one match due to the repetitive pattern in the search area. Also point $\mathbf{b}(564, 239)$ has no match in Image 2 due to occlusion. For wide-baseline marching the problem is more challenging, a wider search region is needed. Image point $\mathbf{a}(544, 309)$ in Fig. 3 has a match $\mathbf{a}^{'}(158, 253)$ in Image



(a) (b)

Fig. 4. Image matching: (a) A feature point on the first image (Image 1) to be matched with feature points in the second image (Image 2), (b) The candidate features on the second image are within a circle of radius R.

2, obviously a search circle of radius 10 pixel around the point $(544, 309)$ is not enough and the whole feature points in Image 2 has to be matched. This will add significantly to the computational complexity. The perspective deformation due to significant view point change lead to matching criteria that is invariant to such deformation. For example the side of the table in Image 1 is 286 pixels, however it is 162 pixels in Image 2 and the Euclidean space is not preserved under such motion[20]. As mentioned before the problem is modeled as shown in Fig. 4 regardless of the motion between the images, and the radius R of the search region is a parameter that can be set by the user.

The matching algorithm under study can be summarized into the following steps:

IMAGE MATCHING ALGORITHM

- **Step 1**: Select a random feature point ($\mathbf{p}$) from the first image (Image 1).
- **Step 2**: On the second image (Image 2) search within a circle of radius R and center $(x_{\mathbf{C}}, x_{\mathbf{C}})$ for the best candidate matches.
- **Step 3**: Select the matches that are below certain threshold.

A. Complexity Analysis

In this section a time complexity analysis is presented. Before starting the analysis let:

- N_1 : The total number of detected features in Image 1
- R : Radius of the search region in Image 2.
- d_{cor} : Feature isolation number. Any two detected features in one image are at least d_{cor} pixels apart.

In **Step** 1 a random feature point $\mathbf{p}$ is selected out of the N_1 features from Image 1. Obviously this may need constant time. In **Step** 3 only a simple comparison operator is needed to check if a certain match below certain threshold and this can be embedded in **Step** 2. The bulk of the complexity comes from **Step** 2 where we study in details. Fig. 5 shows the search region in Image 2 for the randomly selected feature point $\mathbf{p}$ within a distance R from $(x_{\mathbf{C}}, y_{\mathbf{C}})$. In the worst case, the maximum number of matches for the corner point $\mathbf{p}$ is achieved by triangulating the search region. The triangles are equilateral where each side equals d_{cor}. The area of each triangle is $\frac{\sqrt{3}}{4}d_{cor}^2$ and the area of the search region (the circle) is πR^2. Thus the total number of triangles in the search region is:

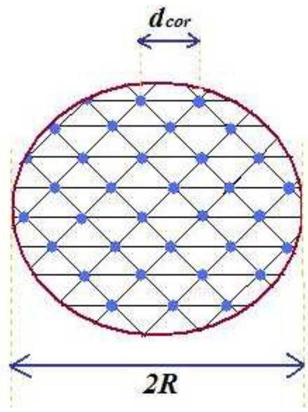


Fig. 5. Triangulated search region in Image 2 to match a feature point from Image 1. The triangles are equilateral with each side = d_{cor}

$$n_\triangle = \frac{4\pi R^2}{\sqrt{3}d_{cor}^2} \tag{1}$$

The triangulated region represents a planar graph $G(V, E)$ with vertex set V and edge set E. The number of vertices v, edges e and faces f are related by the classical *Euler's formula* [21], [22] as follows:

$$v - e + f = 2 \tag{2}$$

In our case each triangle has at least two edges in common with the neighboring triangles, then each vertex in the graph has degree ≥ 3. By applying this property to Equation(2), it is simple to prove that:

$$v \leq 2f - 4 \tag{3}$$

The number of faces equals $n_\triangle$ plus the face outside of the triangulated region, then $f = n_\triangle + 1$. The number of vertices of graph G can be obtained by substituting f in Equation (3) to obtain:

$$v \leq 2n_\triangle = \frac{8\pi R^2}{\sqrt{3}d_{cor}^2} \tag{4}$$

Then, $v \leq k\frac{R^2}{d_{cor}^2}$ where k is constant and equals $\frac{8\pi}{\sqrt{3}}$. The time complexity of matching one feature point $\mathbf{p}$ from image 1 is:

$$C_p \leq O\left(\frac{kR^2}{d_{cor}^2}\right) \tag{5}$$

Finally, the complexity of matching all the N_1 feature points in image 1 with points from image 2 is:

$$C_{Total} \leq O\left(\frac{kN_1R^2}{d_{cor}^2}\right) \approx O\left(\frac{N_1R^2}{d_{cor}^2}\right) \tag{6}$$

It is worth noting here that for computational efficiency only a constant number of feature points from image 1 need to be matching rather than matching all the N_1 feature points. By finding at least 8 corresponding matched points between the two images, the epipolar geometry can be recovered and this reduces the two dimensional search space to a one dimensional search along the epipolar line.

IV. Conclusion

This paper shows the relationship between the time complexity of featured based matching and the parameters selected for feature extraction (the corner isolation distance d_{cor}) and the search area (circle radius R). This relation is expressed in Equation (6). The equation shows that increasing the corner isolation distance (d_{cor}) or decreasing the search region (R) will improve the overall complexity of the problem.

References

[1] Z. Zhang, “Determining the epipolar geometry and its uncertainty: a review,” Tech. Rep. 2927, INRIA, July 1996.

[2] R. Hartley and A. Zisserman, *Multiple View Geometry in computer Vision*, Cambridge University Press, 2000.

[3] J. Weng, N. Ahuja, and T. Huang, “Two-view matching,” in *Int. Conf. on Computer Vision*, 1988, pp. 65–73.

[4] J.F. Vigueras M.O. Berger and G. Simon, “Iterative multi-planar camera calibration: improving stability using model selection,” *Vision, Video and Graphics*, pp. 1–8, 2003.

[5] C. Matsunaga anf K. Kanatani, “Calibration of a moving camera using a planar pattern: optimal computation, reliability evaluation and stabilization by model selection,” *Proc. 6th Euro. Conf. Computer Vision, Dublin, Ireland*, vol. 2, pp. 595–609, 2000.

[6] B. Triggs, “Autocalibration from planar scenes,” in *In Proc. European Conference on Computer Vision*, 1998, pp. 89–105.

[7] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” in *In Proc. 7th International Conference on Computer Vision*, 1999, vol. 1, pp. 666–673.

[8] H.C. Longuet-Higgins, “A computer algorithm for reconstructing a scene from projections,” *Nature*, vol. 293, pp. 133–135, 1981.

[9] O. Faugeras, *Three-Dimensional Computer Vision, A geometric Viewpoint*, MIT Press, Cambridge, MA, 1996.

[10] E. Trucco and A. Verri, *Introductory Techniques for 3D Computer Vision*, Prentice-Hall, New Jersey, 1998.

[11] Z. Zhang, R. Weiss, and A.R. Hanson, “Obstacle detection based on qualitative and quantitative 3d reconstruction,” *IEEE trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 15–26, Jan 1997.

[12] R.Elias H.Hajjdiab and R.Laganière, “Wide baseline obstacle detection and localization,” in *in Proceedings of the IEEE Seventh International Symposium on Signal Processing and its Applications*, Paris-France, July 2003, vol. 1, pp. 21–24.

[13] P. Fornland, “Direct obstacle detection and motion from spatio-temporal derivatives,” in *CAIP*, Prague,Czech Republic, Sept 1995.

[14] T. Williamson and C. Thorpe, “A specialized multibaseline stereo technique for obstacle detection,” in *IEEE conf. on Computer Vision and Pattern Recognition*, 1998, pp. 238–244.

[15] S. Carlsson, “Recognizing walking people,” in *European conf. on Computer Vision*, 2000.

[16] C. Harris and M. Stephens, “A combined corner and edge detector,” *Alvey vision Conf*, pp. 147–151, 1988.

[17] H. Moravec, “*Obstacle avoidance and navigation in the real world by a seeing robot rover*,” Technical Report CMU-RI-TR-3, Carnegie-Mellon University, Robotics Institute, 1980.

[18] C. Schmid, R. Mohr, and C. Bauckhage, “Evaluation of interest point detectors,” *International Journal of Computer Vision*, vol. 37, no. 2, pp. 151–172, June 2000.

[19] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography,” *Commun. Assoc. Comp.*, vol. 24, pp. 381–395, 1981.

[20] P. Montesinos, V. Gouet, R. Deriche, and D. Pel, “Matching color uncalibrated images using differential invariants,” *Image and Vision Computing*, vol. 18, no. 9, pp. 659–672, June 2000.

[21] B. Bollobas, *Graphy Theory, An Introductory Course*, Springer-Verlag, New York, 1979.

[22] O. Overmars M. deBerg, M.Van Kreveld and O. Schwarzkopf, *Coumputational Geometry Algorithms and Applications*, Springer, 1997.