Unnecessary Image Pair Detection for a Large Scale Reconstruction

Jaekwang Lee[1], Chang-Joon Park[2]

[1] Smart Game Research Team, Contents Research Division, Electronics and Telecommunications Research Institute(University of Science and Technology), Daejeon, Korea
[2] Smart Game Research Team, Contents Research Division, Electronics and Telecommunications Research Institute, Daejeon, Korea
ljk3815@naver.com, chjpark@etri.re.kr

Abstract. This paper proposes an algorithm to detect unnecessary image pairs for efficient structure from motion. Since image pair with small baseline is considered as a poor condition for reconstruction, we focus on computing cameras closely located. We address a term, "*remoteness*" which indicates the distance between two images in this paper. The *remoteness* is not affected by image's intrinsic parameters because camera intrinsic matrix is applied to put the extracted features in the normalized coordinate. The *remoteness* is computed using feature disparity in normalized coordinate. Therefore, we can detect redundant image pair captured at the near position without reconstruction. The proposed algorithm is proved by experimental results with Notre Dame images.

Keywords: web image collection, reconstruction, unnecessary image pair, remoteness

1 Introduction

As digital cameras are generalized, we can easily get thousands of pictures of a certain tourist spot from internet. Many researchers have developed algorithms dealing with large number of images downloaded from internet. Snavely et al.[1], [2] propose a structure from motion algorithm using internet photo collections and Chum et al.[3], [4] propose a large scale web image clustering method by computing similarity. In order to compute similarity, a near duplicate image detection(NDID) approach[3]-[5] is used and it is applied to detect duplicated images for 3D reconstruction[1], [2] as well.

3D reconstruction means a construction of a three-dimensional model of an object from several two-dimensional views of it. In other words, the location of cameras is very important factor for reconstructing a scene. It is a common agreement that if two images are captured with wide baseline and many correspondences, then the images are good pair for reconstruction. However the near duplicate image detection method computes superficial similarity for every matching pair. Even though the algorithm works well for detecting image pairs which have similar image content, it is not a proper method to remove duplicate images for 3D reconstruction using NDID.

Snavely et al.[6] propose a novel method for a large scale reconstruction using skeletal graph. They reduce a computational complexity for the reconstruction by making uncertainty based skeletal graphs. They compute reconstruction and covariance for each matching pair and then remove near duplicate images. While removing the near duplicate images, they use a NDID method but it is not good enough to remove redundant images for 3D reconstruction because it detects similar images instead of the closely located images. However our method(we'd like to call it Nearly Located Image Detection method, NLID) not only is able to detect similar images but also is able to detect images with small baseline. It is helpful to reduce parameters and reconstruction cost. In this paper, an algorithm for detecting the nearest neighboring image by their locations are proposed. In order to extract the nearest camera, comparative distances are computed between one and the others. We define a term, "*remoteness*" as a measurement of how far away between two cameras are.

To make the algorithm be robust to these problems, the feature extraction method, SIFT[8] and the feature matching method, RANSAC[9] are employed in this paper. The main requirements for good results are that the extracted correspondences are accurate and the number of features should be enough to guarantee the two images containing common scene or object. We only consider the pair with more than 30 correspondences. Without loss of generality, we assume that every image has EXIF information as mentioned in [2]. From the EXIF tag, we can obtain the focal length, so the features for each image can be easily normalized by multiplying the inverse of calibration matrix to each feature[10]. We show that the proposed *remoteness* measures can be efficiently computed and used successfully to find the nearest camera in this paper.

The remainder of this paper is structured as follows. In section 2 we review the related works and a nearly located image detection method is described in section 3. In section 4 some primary results are shown to prove that the proposed algorithms can detect the nearest cameras successfully. We conclude with future work in section 5.

2 Previous Work

Camera localization is well studied in computer vision and robotics. Several algorithms have been proposed attempting to give camera locations by image search or structure from motion. Typically sparse features such as interest points and straight edges are detected and described. Restrictive assumptions such as static scenes[11] or the existence of 3D models[12] are used to match those visual features between query images and exemplar database images of the same scene viewed under different viewpoints or illumination conditions. In Simultaneous Localization and Mapping (SLAM), both camera motion and the 3D points are recovered[13].

Self-localization can be categorically divided into global and find localization[14]. The first one considered in this paper determines the position and orientation of the robot pose within a world coordinate system. The second one deals with the dynamic-pose of the robot. They conceptualized as the extraction and processing of image features, which by means of recursive state estimations provide the continuous location of the cameras. However, partially significant visual landmarks and assessed

poses provide insufficient useful information for real applications. Also the height of the camera should be the same.

Johansson and Cipolla[12] have developed a system to automatically determine the 3D viewpoint position related to model buildings using an image. For each building, they create a template and map assuming linearity of the object in high contrast regions. Matching is then performed by measuring the fit of each 3D model to the query image. Sivic and Zisserman present video google[15] where key frames from movies are used to query and find similarities between them using a text retrieval approach. The descriptors extracted from the images are quantized into visual words. The collection of the visual words are then used in Term Frequency Inverse Document Frequency(TF-IDF) scoring. The scoring is accomplished using a list of references to the images. Although location of features may be of importance on small vocabularies, it becomes insignificant when the vocabulary grows. All these approaches tend to work well for recognizing specific locations but we should have reference image data.

3 Nearly Located Image Detection

In this section, a method finding *remoteness* between two cameras is described As mentioned above, we assume that the cameras are calibrated(i.e., focal length is given for each image) and features are matched completely.

The *remoteness* is obtained by the process as follows:

- Locate the feature points in normalized coordinate
- Center the converted feature points
- Select a point randomly
- Rotate all features to put the selected point on x axis
- Compute *remoteness* by differencing the points

3.1 Feature points in normalized coordinate

The resolution of each image collection can be vary. To deal with features regardless of its resolution, the features should be in normalized coordinate. By applying inverse of the calibration matrix to the feature points, we can express the points with the points in normalized coordinate.

As described in [10], the calibration matrix K can be written as

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where, a_x, a_y are the scale factor in the x and y coordinate direction, respectively, s is the skew, and $(x_0, y_0)^T$ are the coordinates of the principal point. Normally, we assume that the camera has zero skew, the pixels are square, and also the principal point is at the image centre. Then, the calibration matrix can be

$$K = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where, f is the focal length of the image and x_0 and y_0 are the half of the image resolution.

By multiplying the inverse of calibration matrix, K to the features, the features are normalized and we can deal with the normalized features with regardless of the image's zooming effect(focal length) and size.

3.2 Center the features in normalized coordinate

Let the normalized feature points for first and second camera are $\hat{p}$ and $\hat{q}$, respectively, then a centre point is computed by averaging the points.

$$\overline{p} = \frac{1}{n}\sum_{i=1}^{n} \hat{p}_i, \qquad \overline{q} = \frac{1}{n}\sum_{i=1}^{n} \hat{q}_i \tag{1}$$

and the centered points by

$$p' = \hat{p}_i - \overline{p}, \qquad q' = \hat{q}_i - \overline{q}$$

Note that the sum of the centroid points is zero.

3.3 Compute *remoteness*

In order to compute *remoteness* between two cameras, we should consider the fact that the images can be rotated at the random angle. By rotating the selected point to be on the x axis, two coordinate systems of features can be the same. By doing this, we can overcome the rotation problem of images.

We can rotate the feature points by multiplying the rotation matrix

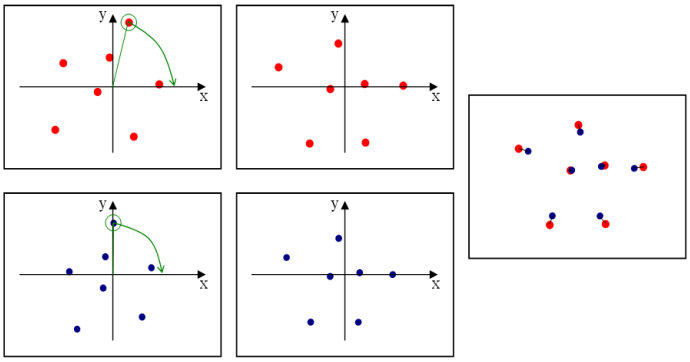
$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Where, $\cos\theta = \dfrac{x}{\sqrt{x^2 + y^2}}$, $\sin\theta = \dfrac{-y}{\sqrt{x^2 + y^2}}$ and x, y are the coordinates of the selected point. Then, rotated points can be denoted by

$$u_i = R \cdot p_i', \qquad v_i = R \cdot q_i' \tag{2}$$

Note that we ignore a scale factor to discriminate the camera position from the object. The *remoteness* can be represented by differencing the features

$$R = \frac{1}{n}\sum_{i=1}^{n} e_i = \frac{1}{n}\sum_{i=1}^{n} |u_i - v_i| \tag{3}$$



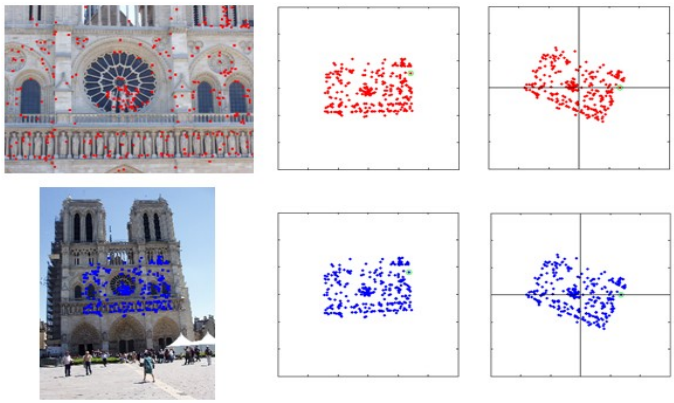
(a) Features in norm. coord. (b) Rotated features (c) Pixel difference

Fig. 1. Process to find the remoteness

Fig. 1 depicts the process for computing the *remoteness*. The first row of Fig. 1 is for the first camera features, the second row is for the second camera and the axes in each image indicate image coordinate axes with origin (0, 0). Fig. 1(a) shows points normalized and centered, and randomly selected point and its correspondence to take rotation for each image feature. Fig. 1(b) is a result of rotated features to make the selected point lie on the x axis by eq(2). Fig. 1(c) is a disparity between normalized and rotated features of two images. To make the R be general, R should be divided by the number of correspondences.

4 Experimental Results

In this section, we apply our algorithm to various cases which can be happened in large scale images including zoom case.

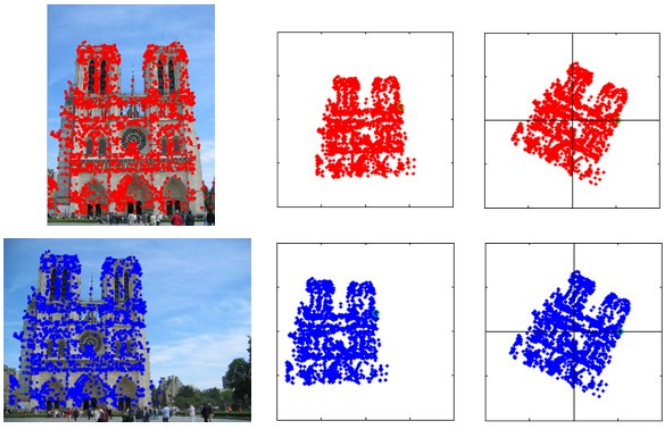


(a) original image+features (b) features in norm. coord. (c) rotated features

Fig. 2. *Remoteness* computation for a zoom in case

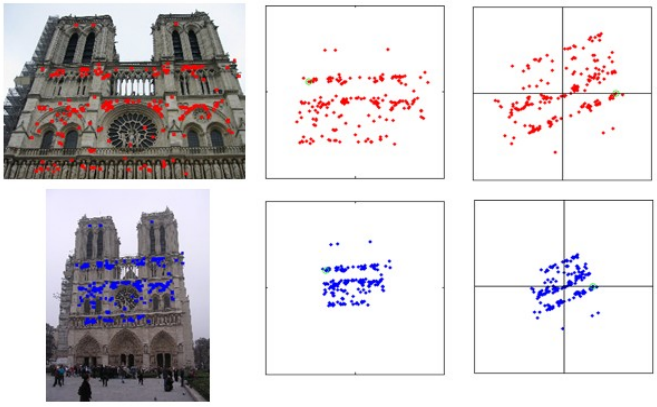
We will not show the result for the cropped, resized and duplicated cases because we can easily see the *remoteness* is nearly zero for these cases.

Fig. 2 is a case to detect that the two images are taken at the same position. The *remoteness* of our method is 0.000304 with 316 features. Fig. 3 and Fig. 4 are the cases that the two images are taken at the near position and at the distant position. The *remoteness* of our method is 0.000561 with 2481 features and 0.1229 with 123 features. The small value of *remoteness* means they are captured at the near position and the pair is not good for reconstruction because of the small baseline.



(a) original image+features (b) features in norm. coord. (c) rotated features

Fig. 3. *Remoteness* computation for a close case



(a) original image+features (b) features in norm. coord. (c) rotated features

Fig. 4. *Remoteness* computation for a distant case

Fig. 5 shows that the relationship between *remoteness* and distances with a selected image. The distance is computed after reconstructing the scene. The reason we reconstruct the scene is to demonstrate the relationship between them.

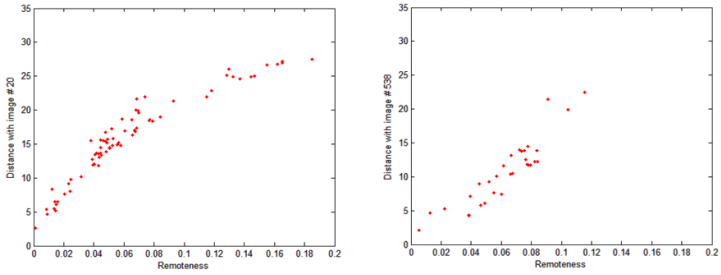


Fig. 5. *Remoteness* vs. distance with selected image

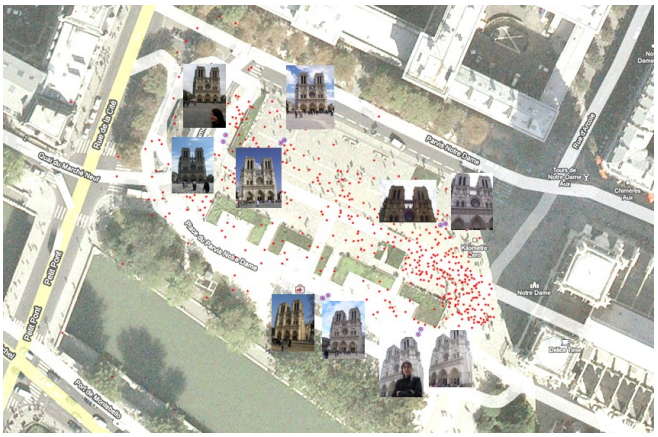


Fig. 6. Superimposed image with registration of cameras to an overhead map

Fig. 6 is a superimposed image of reconstructed camera locations to an overhead map. The reconstruction is performed to show whether our algorithm is good for detecting nearest neighboring camera or not. The figure depicts that the results of nearest cameras for 5 sample cameras. It shows that we can easily detect nearest camera without reconstruction, hence redundant image pairs for reconstruction are extracted simply.

5 Conclusions

In this paper we proposed algorithms to extract nearest cameras by their 3D location without reconstruction. By extracting nearest cameras we can easily remove unnecessary image pairs for 3D reconstruction. Since images contain EXIF information, the feature points can be located in normalized coordinate by applying inverse of calibration matrix. By differencing the points in normalized coordinate, *remoteness* is computed. Of course every pair has different number of correspondences, so we have to consider this factor for comparing *remoteness* with other image pair.

Recently most digital cameras have zoom lenses so practically many pictures downloaded from the internet are zoomed images even they are taken at the same location. In this paper, we showed that we can easily remove many redundant pairs including zooming, cropping and resizing cases using proposed algorithm. In addition we can partition the cameras by their location using Hierarchical clustering[7] which attempts to partition n images into nested clusters. In fact the *remoteness* is not a metric distance since it doesn't satisfy the triangle inequality. However the nearest camera pair can be uniquely determined so we can partition the cameras maintaining the closest pair.

By clustering the cameras we can make efficient graph like skeletal sets and eventually the reconstruction is simply performed. Hence efficient structure from motion can be achieved by using proposed algorithm and it can be applied for 3D scene reconstruction for various content applications such as virtual reality game, etc.

Acknowledgement

This work was supported by the Industrial Strategic Technology Development Program (KI002095, Online Game Quality Assurance Technology Development using Scenario Control of Massive Virtual User) funded by the Ministry of Knowledge Economy (MKE, Korea)

References

1. Snavely N., Seits S., Szeliski R.: Photo Tourism: Exploring Photo Collections in 3D. In ACM SIGGRAPH, pp. 835-846 (2006)
2. Snavely N., Seits S., Szeliski R.: Modeling the World from Internet Photo Collections. Intl. Journal of Computer Vision, vol. 80, no. 2, pp. 189-210 (2008)
3. Chum O., Philbin J., Zisserman A.: Near Duplicate Image Detection: min-Hash and tf-idf Weighting. BMVC (2008)
4. Chum O., Matas J.: Web Scale Image Clustering. Technical Report CMP, CTU in Prague (2008)
5. Ke Y., Sukthankar R., Huston, L.: Efficient Near-duplicate Detection and Sub-image Retrieval. In ACM Intl. Conf. on Multimedia (2004)
6. Snavely N., Seits S., Szeliski R.: Skeletal Graphs for Efficient Structure from Motion. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1-8 (2008)
7. Eppstein D.: Fast Hierarchical Clustering and Other Applications of Dynamic Closest Pairs. 9th Symp. Discrete Algorithms, pp. 619-628 (1998)
8. Lowe D.: Distinctive Image Features from Scale-invariant Keypoints. Intl. Journal of Computer Vision, vol. 60, no. 2, pp. 91-110 (2004)
9. Fischler M.A., Bolles R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartograph. Comm. of the ACM, vol. 24, no. 6, pp. 381-395 (1981)
10. Hartley R., Zisserman A., *Multiple View Geometry in computer vision*. Cambridge university press, (2003)

11. Wang J., Zha H., Cipolla R.: Coarse-to-fine Vision-based Localization by Indexing Scale-invariant features. IEEE Trans. On Systems, Man and Cybernetics, vol. 36, no. 2, pp. 413-422 (2006)
12. Johansson B., Cippola R.,:A System for Automatic Pose-estimation from a Single Image in a city scene. Intl. Conf. Signal Processing, Pattern Recognition and Applications (2002)
13. Williams B., Klein G., Reid I.: Real-time Slam Relocalisation. Intl. Conf. on Computer Vision (2007)
14. Se S., Lowe D., Little J.: Vision-based Global Localization and Mapping for Mobile Robots. IEEE Trans. On Robotics, vol. 21, no. 3, pp.364-375 (2005)
15. Sivic J., Zisserman A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. Intl. Conf. on Computer Vision, pp. 1470-1477 (2003)