K-007

Efficient Scale Database Construction for Scale Filtering in SURF-based AR Application

H.A. Dang†, Pao Sriprasertsuk†, Wataru Kameyama†

1. INTRODUCTION

In this research, in order to speed up the SURF keypoint matching for AR application, multiple scalespaces of different sample resolutions are constructed. Using a simple ultrasonic range-finder, only the appropriate scalespace is used for the detection process. By implementing this methodology, our prototype system is able to save at least 30 percent of computational resources and to increase greatly the accuracy of object recognition. To maintain the high performance, the database needs to be carefully constructed taking into account the input image quality and the number of scalespaces.

2. SURF SCALESPACE

Inspired by SIFT (Scale-invariant feature transform)[1], SURF (Speed-Up Robust Feature)[2] is a well-known local descriptor algorithm published in 2008 by Herbart Bay. By constructing scalespace for the keypoint extraction, SURF becomes to have ability to detect image even at different scale. This feature is ideal for AR application. However, it also generates large amount of data that not only consumes great quantity of computational resources but also decreases the matching accuracy when scale difference is significant.

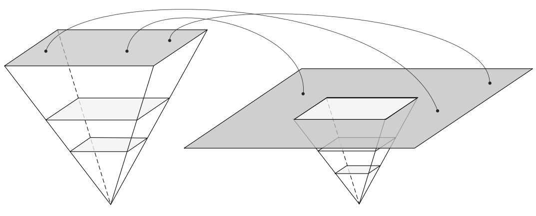

Figure 1: SURF Matching in Typical AR Scenario

Figure 1 shows the SURF scalespace pyramid in a typical scenario of AR application. In AR application, the objects that need to be detected usually appear on a small area of query image while a sample image is usually stored at the highest resolution as possible. Hence, there is a high probability that false matching between the top scales of the sample image and the query image resulting into inaccurate recognition. Our survey conducted during this research shows that 50% of target objects only cover 30% or less of query image area in actual scenario. On the other hand, 38% of query is close-up image which also can result into inaccuracy in the case that sample image is not detail enough.

† GITS, Waseda University

This issue of false matching can be solved by using an appropriate scale of sample and query if the relative scale between the query and the sample image is known. Also note that Figure 1 ignores the lower scale of surrounding scene on query image to ensure its readability.

3. SCALE FILTERING SURF

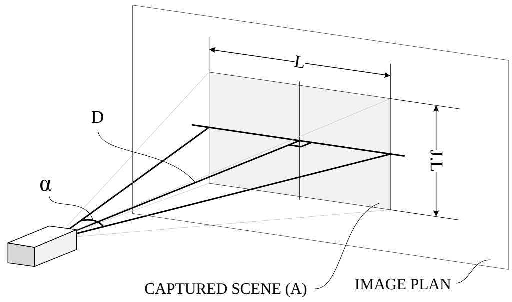

Figure 2: Typical Image Acquisition Scenario

In order to improve the overall system performance, we propose a methodology to minimize the number of redundant keypoints by matching the scale of the sample and the query image using the data obtained from a simple range finder.

$$Sf = \frac{rJD^2 tan^2\left(\frac{\propto}{2}\right)}{Rjd^2 tan^2\left(\frac{\beta}{2}\right)} \qquad (1)$$

The scale decision/scale filtering process is based on the result obtained from Formula (1), where Sf is the "*Scale factor", D* is the distance collected by range-finder, R and J are the resolution and aspect ratio of sample image. Corresponding lower case parameters (r and j) are those of query image. Finally, $\propto$ and β are angel of view of sampling camera and query camera.

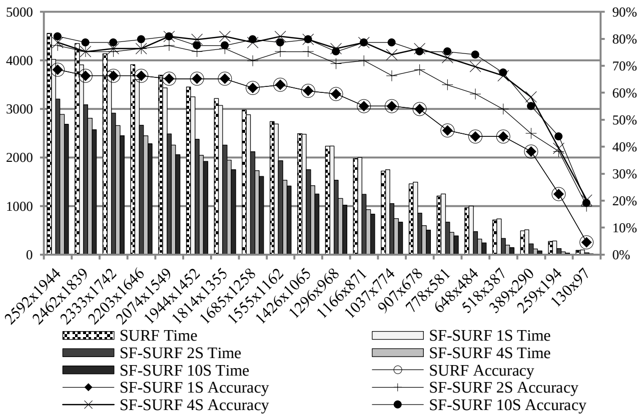

Figure 3: Experimental Results

The verification test has been conducted with various query sizes, and the results of our verification are shown in Figure 3. From the results, it is clear that the matching time of the system has decreased significantly. The system saves 37% computational time at the resolution of 2592x1944 pixels and up to 93% at the resolution of 648x484 pixels.

4. DATABASE CONSTRUCTION

4.1 Optimization

When implementing this system, beside the predefined specific domain for sample and query detail level, clustering is another solution that can improve the system performance. In this solution, as sample images with a similar detail level are grouped, the system only needs to resize the query one time for each cluster.

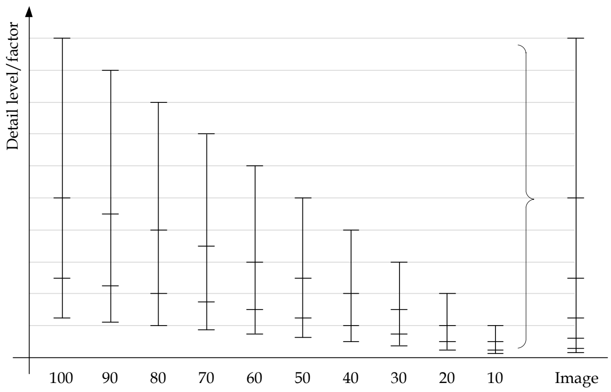

Figure 4: Adding Additional Octaves Effectively Covering Larger Domain

The above method is simple and easy to implement, but it is not optimized for memory usage because similar keypoints are created in adjacent scales. Furthermore, the scalespace pyramid is flat-toped. Hence, in some case, it cannot cover the full range of possible query size/detail.

Because the detail factor is obtained in this system, the scalespace construction process can automatically adjust the number of octaves of each sample image so that they can cover larger domain without creating extra scalespaces. Figure 4 illustrates this process. Instead of constructing additional ten scalespaces, the extra octaves in scalespace can cover the same domain.

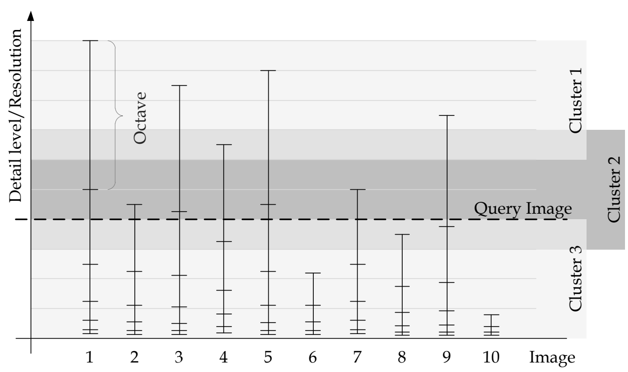

Figure 5: Sophisticated Implementation Scenario

Since all the keypoints are stored in one scalespace for each sample image, the matching process needs to filter each keypoint individually based on the detail factor of scale layer where they are extracted from. This leads to further improvement by clustering database based on the detail factor of keypoints.

Figure 5 illustrates the database clustering and the adaptive scalespace construction in a sophisticated implementation scenario. Although clustering is not necessary for filtering process, it is essential for practical matching implementation which requires clustering and training to enhance the system performance, especially for large size of database. For example, FLANN is a popular algorithm for matching using this strategy[3].

4.2 Content Development

The proposed system requires a range sensor for both sample collection and query acquisition processes which is not practical for implementation. Furthermore, the available sample data cannot be utilized under this scheme. Fortunately, it is possible to estimate scale factor of ordinary sample image based on scale factor of matched keypoints using the following Formula (2):

$$Df = \frac{\sigma k(NK + I)}{\delta K(nk + i)} * df \qquad (2)$$

Where Df is the detail factor of sample image; K and σ are the number of layers per octave and standard deviation of Gaussian kernel of SURF configuration applied on the sample image, N and I are the current octave and the layer level (inside octave) of keypoints extracted from sample image. df, n, k, i, δ are corresponding parameters of query side. The octave and the layer level are zero based counting. Since SURF uses the box filters instead of the Gaussian blur, the approximate Gaussian derivative is calculated by the following Formula (3):

$$\omega_{app} = s\frac{\varphi}{S} \qquad (3)$$

Where, ω_{app} is the approximation of the Gaussian derivative of the current layer, φ is the base filter scale (Gaussian derivative at the base filter), S is the base filter size, and s is the current filter size [2]. This implementation only requires a range sensor on AR device. The average of multiple estimations would result into the proximate scale factor of the sample.

5. CONCLUSION

The experiment shows that implementing a range-finder to AR device not only saves computational resources but also greatly increases the overall accuracy of the object recognition. But it requires extra effort on database construction to maximize the performance and the usability of the system.

REFERENCE

[1]. Lowe, David G. "Object recognition from local scale-invariant features". Proceedings of the International Conference on Computer Vision. 2. pp. 1150–1157, 1999

[2]. Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008

[3]. M. Muja, D. Lowe (2009). “Fast approximate nearest neighbors with automatic algorithm configuration”. VISAPP09