www.ioirp.com
International Journal of Innovative Research in Computer Science and Engineering (IJIRCSE)
ISSN: 2394-6364, Volume – 1, Issue – 7. June 2015

Multiobject Tracking in Vision Systems

[1]Saranya Raj, [2]R. Sureshkumar, [3]R. Saravanakumar
[1]M.E, Dept. of Computer Science and Engineering, Maharaja Prithvi Engineering College, Coimbatore, India
[2]Assistant Professor, Dept. of Computer Science and Engineering, Maharaja Prithvi Engineering College, Coimbatore, India
[3]Assistant Professor, M.E, Dept. of Information Technology, Maharaja Prithvi engineering college, Coimabatore, India
[1]sharroooss@gmail.com, [2]sweetsuresh17@gmail.com, [3]mskumarinfotech@gmail.com

***Abstract*— The object tracking is to associate objects in consecutive video frames. Sometimes the associate can be difficult when the objects are moving fast relative to the frame rate. To track objects even under frequent occlusions, the system demonstrated the effectiveness and usefulness of each part of the system. SIFT, SURF and MRF algorithms which helps to manage different possible motions of the object. The algorithm can handle images with blurring and which helps to find the keypoints between images. The algorithm detects and tracks the multiple objects in a random manner and can maintain the time consuming problem. The method yields good tracking performance in a large variety of highly dynamic scenarios.**

Keywords— Image matching, local feature, SIFT, SURF, MRF

I. INTRODUCTION

Detection of moving objects and motion-based tracking are important components of many computer vision applications. That can include security control, traffic monitoring, and automotive safety. Object detection which includes locating objects in the frame of a consecutive video. In each tracking method that require an object detection mechanism. The mechanism is either in every frame or when the object first appears in the video.

Videos are sequences of images, each of which can be called as frame. It can displayed in fast enough frequency so that human eyes can percept the continuity of its content. All image processing techniques can be applied to individual frames in consecutive videos. The two consecutive frames with their content are usually closely related. At the first level, there are raw pixels with color or brightness information. The further processing features such as edges, corners, lines, curves, and color regions in each video. In the higher abstraction layer that may combine and interpret these kinds of features as objects and their attributes. At the highest level, there are the human level concepts which involves one or more objects and relationships among them.

The goal of SURF (SPEED UP ROBUST FEATURES) is to increase the speed in every step. SURF is good at handling images with some kinds of blurring and rotation. The goal of SIFT (SCALE INVARIANT FEATURE TRANSFORM) is to define keypoint detection and description in the video. SIFT which helps to find the keypoint between two images which are matched by identifying their nearest neighbours. The goal of MRF (MARKOV RANDOM FIELD) is to select random object in the video sequence.

II. RELATED WORK

SURF’s detection scheme is based on the concept of automatic scale selection, proposed by Lindeberg in 1998. In this work, Lindeberg experimented with using the determinant of the Hessian matrix for a 2-D Gaussian, as well as the Laplacian (i.e. the Hessian’s trace), to detect blob-like structures in images. Mainly motivated by Lindeberg’s findings, the authors of SURF chose the determinant of Hessian as their target feature. Other well-known feature schemes include the famed Harris corner detector (which relies on eigenvalues of the second moment), the entropy-based salient region detector proposed by Kadir and Brady, and the edge-based work of Jurie and Schmid. Furthermore, SURF’s detector extends on Lowe’s idea of using the Difference of Gaussian as an approximation of the Laplacian of Gaussian filter.

The core concept of feature comparison is to find nearest neighbours for a numerical vector. However, given a large feature count and a high vector dimensionality, finding true nearest neighbours can be extremely slow. Beyond the mainstream approximation search algorithms using hash tables and K dimensional trees (i.e. k-d trees), various improvements have been proposed by the image feature community.

International Journal of Innovative Research in Computer Science and Engineering (IJIRCSE)
www.ioirp.com ISSN: 2394-6364, Volume – 1, Issue – 7. June 2015

III. METHODOLOGY

A. Methodology Analysis

The greatest characteristic of SIFT algorithm is scale invariance. Inorder to make scale invariance, SIFT uses DoG (Difference of Gaussian) function. SURF uses different methods,location description and descriptior generation.First, confirm that you have the correct template for your paper size.

B. Discussion

In a practical implementation, finding the exact nearest neighbour for a large number of high-dimensional vectors can be incredibly slow. Therefore, deployment-grade feature frameworks can only afford to look for approximate nearest neighbours when used in real-time applications. However, because speed is not one of our project design goals, we conducted all of our experiments using exact feature neighbour matches. Nevertheless, we have also thoroughly investigated the design, implementation and performance of approximate nearest neighbours using K-dimensional trees.

IV. EXPERIMENTS

In this section, we conduct experiments to investigate the performance of SIFT and its variants in different situations: scale and rotation change, blur change, illumination change,and affine change. We also investigate the time consumption of each algorithm in different situations.

A. Experiment Description

SIFT and its variants are implemented in Matlab 2013 and executed on a Dell PC. It has a Pentium(R) Dual-Core CPU E5300@2.60GHz, and 4G memory, running Windows 7. In order to conduct empirical comparative analysis of SIFT and its variants, we use image data sets.

In this section, we conduct experiments to investigate the performance of SIFT and its variants in different situations: scale and rotation change, blur change, illumination change,and affine change. We also investigate the time consumption of each algorithm in different situations In all experiments, we follow the traditional common approach, using KNN (k-nearest neighbor algorithm) to match keypoints andto eliminate mismatches, in SIFT and its variants. Specifically, they use KNN to match keypoints on a KD-tree (short for *k-dimensional tree*).In keypoint matching, target image keypoints are used as abenchmark. The goal of SIFT and its variants is to search keypoints, which are the nearest neighbor and the second nearest neighbor to target image keypoints,

B. Performance Evaluation under Different Situations

- The algorithm now allowed almost any feature pairs to contribute to recognition, the weaker features were correctly matching with other weak ones among all images. Because these weak pairs boosted all recognition scores by similar amounts, they in fact cancelled each other out.
- The average position of the correct match in the image versus image setup was noticeably worse than in the image versus object setup. This observation strengthens our previous argument: when the weaker features were allowed to pair up, the system became incapable of distinguishing between objects in the images. However, this phenomenon is negated by pooling all the bad feature matches and good feature matches together in the image versus object setup.

V. EXPERIMENT SUMMARIZATION

In the above sub-sections, we empirically compared the performance of SIFT and its variants in four different situations, i.e., scale and rotation invariance, blur invariance, illumination invariance, and affine invariance. We also investigated the time consumption of each algorithm in the above four situations. In this sub-section, we analyze the experimental results qualitatively. Thus, we can have general ideas of the performance of each algorithm in different situations. We rate the experimental results in four grades, i.e., Best, Better, Good, and Common.

A. SURF and SIFT

Speeded-Up Robust Features (SURF) is a newly-developed framework, which we believe is very likely to becoming the next *de facto* feature detector in the industry. Compared to its main competition, SIFT, SURF has been shown by its authors to offer both faster and more robust performance. These improvements are made possible by using ingenious box filter and integral image tricks to find features quickly, and by using Haar wavelets to describe them robustly. Using our object recognition task, we have shown how SURF can be used to robustly detect objects in images taken under different extrinsic and intrinsic settings. For this project, we have developed a full-fledged SURF framework using MATLAB, based solely on the original SURF authors' publications. However, we made some explicit choices in our implementation in some aspects of the algorithm that we did not fully understand, or for which we identified problems in our experiments. Therefore, it is no surprise that our version of SURF yields poorer recognition rates than the original authors' version. Nevertheless, by writing all the algorithms ourselves, we have discovered a tremendous amount of useful information on SURF, as well as on image features in general.

Specifically, our experiments suggest that we should employ a relatively lax comparator threshold, because even though this setting will allow weak features with smaller determinant of Hessian values, our results suggest that their effects cancel out at values near 0.8. However, these large values are better than smaller ones, since we want to have a sufficient number of features to compare with in each image. The same reasoning further suggests choosing a small detector threshold to create more features per image. Furthermore, our trials have shown that comparing images to objects has the added benefit that the positions of the correct object in erroneous matches often have only slightly smaller recognition scores, which we can use in conjunction with Machine Learning techniques to boost the performance. On the other hand, by matching images to other images, our system can detect both the object identity, as well as the image orientation. We have thus designed two systems, each having their own unique benefits! We would like to conclude this report by discussing some potential improvements. First of all, because we made many assumptions in our descriptor part, we would like find out which of our assumptions are actually consistent with the original authors' implementation.

One simple adjustment is to perform bilinear interpolation when sampling pixels, instead of choosing the nearest neighbour. Within our detector, we have disables the fourth octave and did not search through the up-scaled image at all, because our source images had very limited resolutions. We believe that we can learn a lot more about the detector by using larger-sized images. Another related feature that we disabled was the sign of the Laplacian – we are interested to examine the difference in performance resulting from using this feature. Given time constraints, we did not implement our comparator to truly compare query images with database objects, but instead opted for the simpler approximation of grouping feature pairs together. We suspect that a true setup will improve results in general, but we are curious to find out its side-effects. Finally, because our comparator codebase does not depend on SURF features, we can trivially use other types of features, such as SIFT, to assess the performance of object matching among different frameworks. So that we can compare the benefits and drawbacks of using features versus using pixels directly. In the end we are extremely satisfied with the amount of knowledge that we gained by exploring the various aspects of this powerful feature framework for this project.

B. MRF

This algorithm can select the random objects in the video sequence. A Markov random field is similiar in its representation of dependencies; the differences being that Bayesian networks are directed and acyclic, whereas Markov networks are undirected and may be cyclic. Thus, a Markov network can represent certain dependencies that a Bayesian network cannot (such as cyclic dependencies); on the other hand, it can't represent certain dependencies that a Bayesian network can (such as induced dependencies).

C. Comparisom on Time Consumption

1) Positioning Figures and Tables: Place figures and tables at the places where they needed. All tables should be in Classic 1 format with borders to heading and subheading columns. Large figures and tables may span across both columns. To do so select text above one column table and conver it in two column and then select text below one column table and convert it into two column. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 1", even at the beginning of a sentence. Also Submit figures and tables on separate page at the end of manuscript with their lables.

TABLE I.

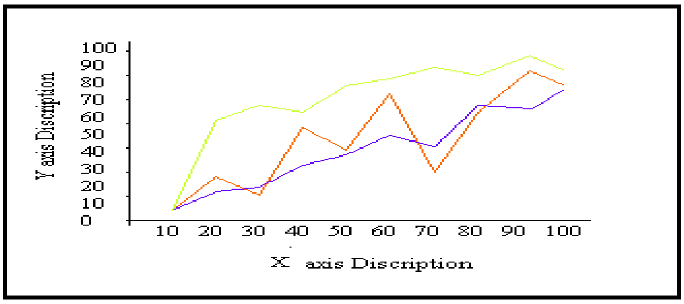
Algorithms	***Step***	***Levven***	***Graffitti***	***Average***
SURF, SIFT and MRF	3.5872	2.3241	2.8211	3.1387

We suggest that you use border for graphic (ideally 300 dpi), with all fonts embedded) and try to reduce the size of figure to be adjust in one column.

Figure and Table Labels: Use 8 point Times New Roman for Figure and Table labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader.

TABLE II. QUALITATIVE SUMMARIZATION

	Situations	
	Scale and Rotation	***Blur***
SURF, SIFT, MRF	Best	Better



a.

Fig. 1. SURF, SIFT and MRF Algorithms based on time

VI. CONCLUSIONS

This type of tracking systematically analyzed the major members of the SIFT family, including SIFT, SURF and MRF. They are image local feature description algorithms which are based on scale-space, keypoints and random selection. Here, the tracking empirically evaluated their performance in different situations: scale and rotation change, blur change, illumination change, and affine change. With the large computation of SIFT and its variants, the algorithms investigated their time consumption empirically in different situations. In the experimental results, the qualitative analysis and evaluation of the performance of each algorithm can also included, which provided the general ideas and suggestions of how to choose the best algorithm for a specific real-world problem.

References

[1] Ouyang, W., Tombari, F., Mattoccia, S., Di Stefano, L., Cham, W.-K. (2012). Performance evaluation of full search equivalent pattern matching algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (1), 127-143.

[2] Birinci, M., Diaz-de-Maria, F., Abdollahian, G. (2011). Neighborhood matching for object recognition algorithms based on local image features. In *IEEE Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE),* 4-7 January 2011. IEEE, 157-162.

[3] Mian, A., Bennamoun, M., Owens, R. (2010). On the repeatability and quality of keypoints for local featurebased 3D object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89 (2-3), 348-361.

[4] Mikulka, J., Gescheidtova, E., Bartusek, K. (2012). Soft-tissues image processing: Comparison of traditional segmentation methods with 2D active contour methods. *Measurement Science Review*, 12 (4), 153-161.

[5] Kim, D., Rho, S., Hwang, E. (2012). Local featurebased multi-object recognition scheme for surveillance. *Engineering Applications of Artificial Intelligence*, 25 (7), 1373–1380.

[6] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[7] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[8] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[9] Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60 (2), 91-110.

[10] Tuytelaars, T., Mikolajczyk, K. (2008). Local invariant feature detectors: A survey. *Foundations and Trends* in Computer Graphics and Vision, 3 (3), 177-280.

[11] Juan, L., Gwun, O. (2009). A comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing*, 3 (4), 143-152.

[12] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[13] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[14] Younes, L., Romaniuk, B., Bittar, E. (2012). A comprehensive and comparative survey of the SIFT algorithm - feature detection, description, and characterization. In *Proceedings of the International* Conference on Computer Vision Theory and Applications (VISAPP). SciTePress, Vol. 1, 467-474.

[15] [Morel, J.M., Yu, G. (2009). ASIFT: A new framework for fully affine invariant image comparison. *SIAM* Journal on Imaging Sciences, 2 (2), 438-469.

[16] Rabbani, H. (2011). Statistical modeling of low SNR magnetic resonance images in wavelet domain using Laplacian prior and two-sided Rayleigh noise for visual quality improvement. *Measurement Science Review*, 11 (4), 125-130.