SELECTION AND IMPLEMENTATION OF AFFINE INVARIANT REGION DETECTOR FOR AR SYSTEM

Miroslav Juhas

Doctoral Degree Programme (1), FEEC BUT

E-mail: xjuhas00@stud.feec.vutbr.cz

Supervised by: Ilona Kalová

E-mail: kalova@feec.vutbr.cz

Abstract: The augmented reality is actually being used in advertisement, but specially designed image patterns and camera calibration-based methods are required. Augmented reality in industrial context proved to be beneficial in development, service, maintenance, manufacturing and design, but in fact it is rarely used in special cases and research projects. Computer vision based image registration is suitable for the industrial system but camera calibration or camera affine transformation-based registration methods are inappropriate. In this paper the affine invariant region detector as a basis for future industrial, camera calibration free and registration free augmented reality system is selected. Selected algorithm is implemented in C++ and speed of method is tested on actual performance computer system with parallel speed up aimed to achieve real-time processing speed of the algorithm.

Keywords: MSER implementation, Linear Time MSER, AR, Augmented Reality, Segmentation

1. INTRODUCTION

In augmented reality (AR) system, as for example presented in [1], the image is processed for feature points, the extracted point's description matched with model in database and the object pose is estimated. The object is then tracked and corresponding augmented object presented to user. Because of nature of the problem, where common object with unknown pose and size is searched in image, the invariant object description is challenging.

The feature point appearance is subjected to perspective transformation, which can be simplified to affine transformation for points lying on planar surfaces of the object. The feature point can be described in affine invariant way, or can be extracted in affine invariant way and described with relatively simply descriptor.

The affine invariant feature point extraction is more transparent than its complex invariant description. In this way it is more suitable for augmented reality systems. Invariant feature point extraction can significantly reduce amount of different viewpoints needed for complete object description and accelerate registration process.

Computation time needed for invariant feature point extraction algorithms are the main problem of their application for augmented reality systems [1]. On the other hand, some recently developed algorithms [4][6][7] in combination with ongoing progress in performance of computer systems should be suitable for augmented reality systems.

2. REGION DETECTOR

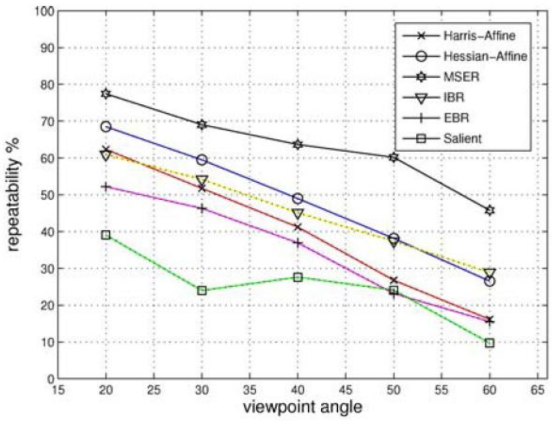
2.1. AFFINE COVARIANT DETECTOR SELECTION

Appearance of the object in image acquired by a camera is subject of shift in brightness intensity and size, rotation and viewpoint changes. The optimal region detector should be invariant to those distortions. A lot of region detectors, for instance SIFT[3], can meet only subset of those requirements.

Although the requirements are looking too difficult to fulfill, the comparison of six affine covariant region detectors was presented in [6]. The compared detectors were invariant to brightness changes and general affine transformations. The compared detectors were:

- Harris-affine [5] and Hessian-affine – methods which detects interest points in scale-space, an elliptical region is determined with the second moment matrix of the intensity gradient.
- EBR – edge geometry based region detector.
- IBR – intensity extrema-based region detector.
- MSER [4] – maximally stable extremal region detector.
- Salient region detector

The detectors were compared very closely in characteristics such as computation time, viewpoint angle and scale change dependence, region count, size and stability, light, noise, compression and scene type dependencies and many others.



Detector	Run time (min:sec)	Number of regions
Harris-Affine	0:01.43	1791
Hessian-Affine	0:02.73	1649
MSER	0:00.66	533
IBR	0:10.82	679
EBR	2:44.59	1265
Salient Regions	33:33.89	513

Figure 1: Repeatability of detectors in structured scene for viewpoint changes and computation times for sample image (800x640) presented in [6].

In comparison with many cases the highest score was obtained by MSER detector, followed by Hessian-affine. MSER and IBR performed well on images containing homogenous regions with distinctive boundaries. Other detectors were more suitable for other scene types. No detector outperforms other detectors for all scene types and all types of transformations. The MSER detector produces fewer regions than other detectors, but this is balanced by significantly higher positive matches produced by MSER detector for descriptor used in [6].

Because of high repeatability, short computation time and usability for scenes with homogenous regions widely presented on artificial industrial objects the MSER detector were selected as invariant region detector for augmented reality system. IBR, EBR and Salient Regions where excluded because of long computation times and Harris-affine and Hessian-affine because of high false matches count.

2.2. DETECTOR IMPLEMENTATION

The MSER algorithm [4] can be briefly described as image thresholding on multiple thresholds (in fact full range of image gray values). In many images, local binarization is stable over a large range of thresholds in certain regions. Those regions are maximally stable if function (1), where Q is a size of region on threshold i, has a local minimum at i.

$$q(i) = \frac{\left(Q_{i+\Delta} - Q_{i-\Delta}\right)}{Q_i} \tag{1}$$

The set of all extremal regions can be computed in $O(n \log \log n)$ time, where n is the number of pixels. The computation time is none linearly dependent on number of pixels. The algorithm was improved to linear time dependency by Nistér and Stewénius [7] and this type was selected for implementation because of faster and linear time computation. For the algorithm block scheme see **Figure 2**.

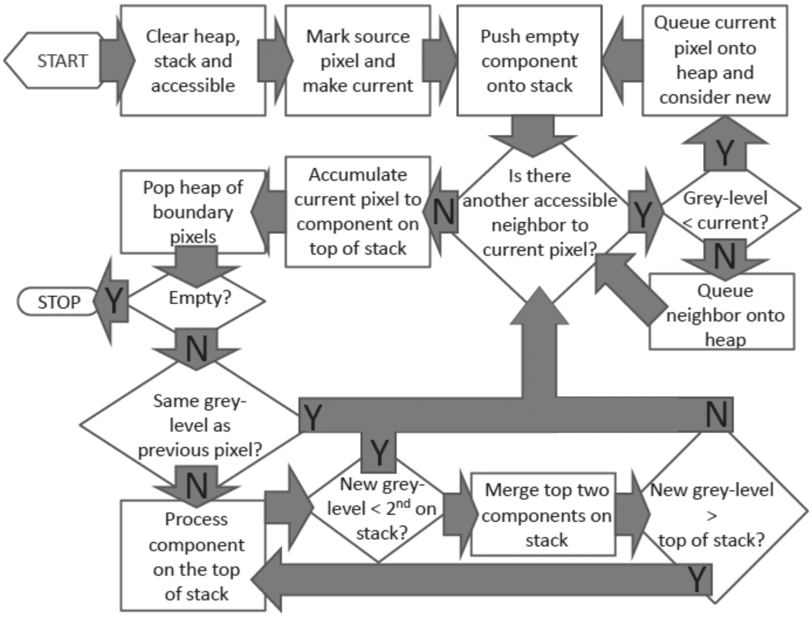


Figure 2: State graph for the algorithm by Nistér and Stewénius [7]

The algorithm was implemented in as C++ object according to state graph and authors description. The algorithm is performed over grayscale image. In this place same brief description of algorithm and requirements for high performance should be mentioned.

The algorithm uses 3 data structures: mask of accessible pixels, priority (inverse of gray level) heap of boundary pixels and stack of component information (holding pixels in component and history of the component). For fast computation a dynamic allocation during processing can't be used. Heaps of sizes of number of pixels must be pre-allocated for history, boundary heap and pixel components. The components are then picked from the heaps during execution. This approach improved computation time for 0.5Mpx test image from approx. 3500ms to 200ms.

The priority heap of boundary pixels is also problematic. The darkest pixel must be returned. Pop of the boundary heap occurs very often and searching for the darkest pixel on the heap significantly slows down the algorithm. In [7] the processors instructions are utilized for processing of a binary mask of occupied boundary pixels gray levels. This approach may be fast, but is too low level and platform dependent. Instead of binary mask the data structure keeping information about the last known darkest gray level value and corresponding actual count of pixels on the heap for this level was used. The data structure is updated when new darkest pixel comes to the heap or in case that darkest heap level becomes empty of pixels. This approach reduces number of necessary searching for the darkest gray level on boundary heap. There was benefit in speed up from approx. 200ms to 85ms.

The block „Process component on top of the stack“ represents extremal region stability test. If the region is stable (local minimum of (1) is lower than selected value) the contour is generated. When merging two components simply history from winner (larger component) is taken.

2.3. ALGORITHM SPEED EVALUATION

The speed of implemented algorithm was tested on set of 20 pictures. The processor platform was Intel i7-720QM running at 1.6GHz. A dependency on resolution was evaluated. For linearity test the pictures were resampled to required resolutions. The results are on **Figure 3**.

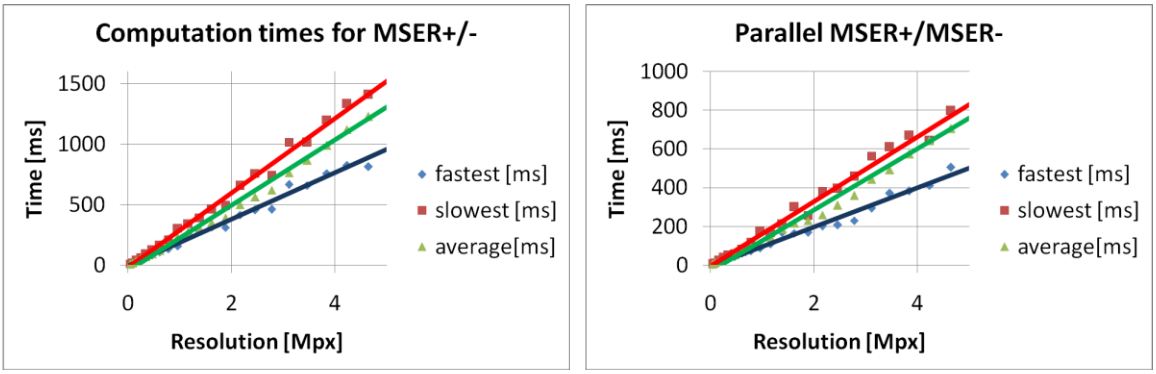


Figure 3: Algorithm performance over set of tested images, the time for the elliptical affine covariant region computation is not included

(1Mpx)	MSER+/-	Parallel MSER	Elliptical region	Total time	Possible parallel time
Average [ms]	248	143	153	401	153
Fastest [ms]	190	99	76	266	99
Slowest [ms]	299	164	77	376	164

Table 1: Computation times for 1Mpx image

In algorithm the covariant region detection is performed twice. One is for original image (MSER+) and one is for inverted image (MSER-). The average performance of implementation is 248ms for 1Mpx image, which gives performance of 4Mpx/s. The MSER+ and MSER- could be computed parallel. Parallel computation takes average 143ms, giving 7Mpx/s performance. The computation time is not half because of different processing intensity for MSER+ and MSER-. The computation time is given by waiting for slowest to finish. The average speed up by parallelization is by 1.75, in worst case by 1.6 for some scenes.

The elliptical affine covariant region fitting is comparable with MSER detector in computation time. The best way to achieve good performance is to compute elliptical region parallel with MSER. With this parallelization, the average possible performance is about 6.5Mpx/s.

The MSER algorithm was implemented with and without OpenCV support. There are almost no performance differences between implementations since OpenCV library is used only as an interface for image input and contour output but not for computation.

The test proved linearity and dependency on image content. The algorithm is fastest for simple images with distinctive boundaries and homogenous regions. The minimum performance was observed for complex images with high region count, weak boundaries and color gradients presented in scene.

2.4. PROPOSED ALGORITHM SPEEDUP

The MSER detector can be hardly accelerated because of simplicity. The down sampled images could be used for segmentation and region described in full sized image. This approach uses MSER size invariance property.

Another possibility is in massive parallelization. The parallelization should be possible for linear time MSER detector when image sliced in brightness levels and processed parallel. With this approach the MSER stability test cannot be performed immediately because of regions splits into multiple slices. The component tree must be generated instead of direct region stability test and MSER output. After processing the component tree from all slices should be merged and the region stability test performed over complete component tree.

3. CONCLUSION

The MSER detector was selected as a basis for AR system because of affine covariance, high speed and stability. The size invariance property along with low dependency on viewpoint angle and scene rotation enables minimization of object description. With MSER detector only few viewpoints representations for object should be sufficient for registration and tracking. The implementation proved high performance and linearity. The average speed of 6.5Mpx/s is near the real-time processing, it means 21fps for resolution 640x480. In AR high resolution images are common. In this case massive parallelization should be possible or MSER detector could be used for reliable registration and frequent tracker corrections.

ACKNOWLEDGEMENT

This work was supported by grant "Modern Methods and Approaches in Automation" from the Internal Grant Agency of Brno University of Technology (grant No. FEKT-S-10-12).

REFERENCES

[1] Lepetit, V.,Vacchetti, L., Thalmann, D., Fua, P.: Fully Automated and Stable Registration for Augmented Reality Applications, In: Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR ’03), 2003

[2] Li Yi-bo, Kang Shao-peng, Qiao Zhi-hua, Zhu Qiong: Development Actuality and Application of Registration Technology in Augmented Reality, In: 2008 International Symposium on Computational Intelligence and Design, vol. 2, 2008, pp.69-74

[3] Lowe, D.: Distinctive image features from scale invariant keypoints. International Journal of Computer Vision 60(2), 2004, pp. 91–110

[4] Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: British Machine Vision Conference, 2002, pp.384–393

[5] Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Proceedings of the 7th European Conference on Computer Vision, 2002, Copenhagen, Denmark.

[6] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky,F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. In: International Journal of Computer Vision 65(1–2), 2005, pp. 43–72

[7] Nister, D., Stewenius, H.: Linear Time Maximally Stable Extremal Regions. In:ECCV, 2008 II: pp. 183–196