

# Visual Marker Detection and Decoding in AR Systems: A Comparative Study

Xiang Zhang    Stephan Fronz\*    Nassir Navab

Augmented Reality Group  
Siemens Corporate Research  
Princeton, NJ 08540, USA

{Xiang.Zhang,Nassir.Navab}@scr.siemens.com

## Abstract

*Visual markers are widely used in existing augmented reality (AR) applications [7, 12, 11, 19]. In most of such applications, the performance of an AR system depends highly on the tracking system for visual marker detection, tracking, and pose estimation. Currently, there are more than one marker based tracking/calibration systems available. It is thus desirable for the user to know which marker tracking system is likely to perform the best for a specific AR application. To this purpose, we compare several marker systems all using planar square coded visual markers. We present the evaluation results, both qualitatively and quantitatively, for the following properties: the **usability, efficiency, accuracy, and reliability**. For a particular AR application, there are different marker detection and tracking requirements. Therefore, the purpose of this work is not to rank the existing marker systems; instead, we try to analyze the strength and weakness of various aspects of the marker tracking systems and provide the AR application developers with this information.*

## 1 Introduction and Background

Augmented reality (AR) technology combines the virtual and the real worlds together to provide the viewers with enhanced views of the environment. Due to the progress of AR related research in the last decade, together with the advance of computer hardware and software, AR systems are getting more and more attention in such sectors as manufacturing, medical, military and entertainment (e.g., [7, 16]).

A typical AR system includes a display and a motion tracker with associated software. The software reads the tracking events to know the position of the display and renders the virtual objects. In order to render correctly, the virtual objects and the real world need to be registered. This registration implies that the geometry of the virtual camera where the augmentation takes place is known with respect to the real world.

Visual markers are widely used in the existing AR systems. For example, ArToolKit[7], Auto-assembly[12], Outdoor tracking[10], CyliCon [8, 1], ArLoc[18], and CyberCode[13] all use visual marker based systems for motion tracking and pose estimation. The performance of such AR systems depend highly on the performance of marker detection, decoding and pose estimation. Currently, there are more than a few existing marker tracking systems. Depending on the particular application, there are different tracking and pose estimation requirements. For instance, 3D graphic objects often need to be superimposed in good alignment with the real world. This requires the tracker to provide very accurate pose estimation. While in the case that only text information is displayed, the registration requirement is not as severe. In simple office applications a few markers may suffice. In large industrial applications hundreds or thousands of uniquely coded markers may become necessary. It is desirable to enable the AR application developer to determine which marker system is the most suitable for a given AR application.

Even though there is effort made to explore marker-less or feature based tracking and 3D reconstruction (e.g., [4] and [15]), AR applications are expected to continue to use visual markers for the foreseeable future.

In the AR applications, the square shaped vision markers are the most commonly used ones. The reason is probably that a square shape provides at least 4 co-planar corresponding points so that the camera calibration can be carried out with a single marker in the scene (see e.g. [20]). On the contrary, a circular marker can only provide one point correspondence, i.e. the center, unless the circle itself is used. If the center is used, at least 3 circular markers with known 3D positions have to be captured in the same image for pose estimation [6].

In this paper we focus our evaluation work on the marker tracking systems of square shaped planar visual markers. These are the marker tracking systems used in this comparative study:

\*Student intern from Technique University of Munich.

- ArToolKit(ATK) marker system [2].
- Institut Graphische Datenverarbeitung (IGD) marker system [3]
- Siemens Corporate Research (SCR) marker system [17]
- Hoffman marker system (HOM) used by SCR and Framatome ANP [1]

These marker systems were selected because they are actually used in AR applications and were available for analysis. In addition, these marker systems all allow users to create multiple codes and are quite similar and comparable to each other in many aspects. While we were unable to test all existing systems for this evaluation, we tried to use a representative set of marker based systems. Unsuccessful attempts were made to obtain the marker tracking system implemented at the Sony Computer Science Laboratory (CSL)[13]. However, the marker design of CSL system is quite similar to that of the IGD marker system included in this work.

We evaluated the marker tracking systems for the following properties: the **usability**, **efficiency**, **accuracy**, and **reliability**. These four major properties describe the performance, advantages, and disadvantages of these marker tracking systems.

The usability of a system describes whether it is easy for the users to integrate the system into their AR applications. It also considers the compatibility of the system with other computer platforms and operating systems. The usability can only be compared qualitatively.

The efficiency of a marker tracking system can be evaluated by computing its tracking time performance. This property can be evaluated by the required running time for marker detection and decoding or the frame rate when tracking the motion of markers in real-time videos.

The accuracy is defined in terms of errors in feature extraction from 2D images. We did not use the errors in pose estimation or back projection since they depend on camera internal parameters and calibration algorithms. While the *ground truth* of the feature correspondences are not available, we designed a precise and interactive procedure to extract the same features, used for 3D-2D correspondences, with the best possible accuracy. We consider these set of features as '*ground truth*' and use them in the evaluation. We then compute the average errors and the standard deviations of the corresponding features extracted by the marker tracking systems relative to this '*ground truth*'.

The reliability describes whether the system can detect and track the marker under non-ideal conditions. For example, marker detection in videos captured with a poorly focused camera or with large projective distortion. We also evaluate the ability of the system to detect and correctly decode the markers within small regions of interest.

For the rest of the paper, we first introduce the four markers detection and decoding systems in section 2. We then represent the quantitative results of the comparative

study in Section 3 followed by the qualitative discussion on the usability and scalability of these systems in section 4. Finally, we draw conclusions and provide some hints in the choice between different marker tracking systems based on applications.

## 2 The Marker Systems



Figure 1: Sample markers for ArToolKit.

The **ATK** markers come with the ArToolKit (ATK). Figure 1 shows two ATK markers. A marker is coded with the pattern inside the inner square of the marker. The marker decoding is based on a highly simplified template matching algorithm that compares three of the geometry invariants of the marker region with those of the patterns pre-registered in the system.

The source codes of ArToolKit for both Unix and Windows systems are generously available online [2] for free, together with documents, sample programs, and other utilities for camera calibration, virtual object overlay, etc. We found that the ATK package is well documented and very easy to use. The ATK marker system is widely used for AR application prototyping (see, e.g., [7]).

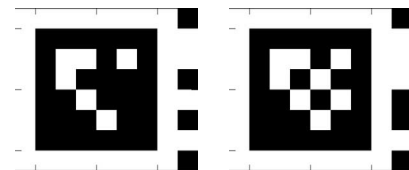


Figure 2: Sample markers designed at Siemens AG (HOM).

The **HOM** marker system (Figure 2) is developed originally by C. Hoffmann in 1994 at Siemens AG for photogrammetry purposes. This system has later been used in industrial documentation and maintenance applications. In addition to the main square, there is a side bar that provides 6 bits of encoding to enhance the reliability of marker recognition. Navab et al [9] used this marker system for 3D reconstruction and AR documentation based on high resolution (e.g., 3000×2000 pixels) images. These markers have been used by Siemens and Framatome ANP for camera calibration and as-built reconstructions of numerous power plants, chemical plants, and oil platforms. The original version of HOM system only offers an executable for processing static images. Recently, Siemens Corporate

Research (SCR) has compiled this into a software library that can be used for processing real-time video sequences.



Figure 3: Sample markers designed at IGD.

The **IGD** marker system is implemented at IGD, which is an ARVIKA (see [3]) partner. ARVIKA is the German government supported research project to develop AR related applications in industry. Many ARVIKA related applications are developed using the IGD marker system. Figure 3 shows two IGD markers. An IGD marker is a square divided into  $6 \times 6$  square tiles of equal size. The inner  $4 \times 4$  tiles are used to determine the orientation and the code of the marker. The precompiled libraries of the IGD marker system are available to ARVIKA participants.

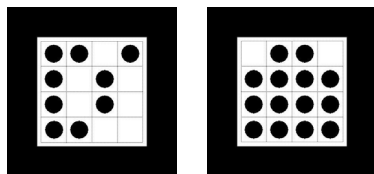


Figure 4: Sample markers designed at SCR.

The **SCR** marker system is developed at Siemens Corporate Research (SCR) for localization and tracking [16] in different AR applications [19]. This system is also used by some ARVIKA partners. The SCR markers are also coded with the inner matrix. Each SCR marker offers 8 corresponding feature points. The SCR system is currently implemented only as an OCX control to be used with Windows applications. The C++ classes are available. The system can therefore be adapted to other computer systems. Sample codes are available for integrating this marker tracking system with other applications (e.g., see [5]).

### 3 The Quantitative Study

**Computer:** The computer we used for this work is a Dell OptiPlex GX200 with 933MHz Pentium III processor and 512 MB SDRAM with Windows 2000.

**Cameras:** All the videos used in our evaluation are taken using one of the two cameras (1) the SONY EVI-D30 with wide range zoom and high speed auto-focus features; and (2) the IBM Net USB Camera Pro.

**Videos:** The evaluation is performed by processing video sequences of hundreds of frames. All video sequences in the same set for comparison are recorded under exactly

the same conditions. The procedure for capturing a set of video sequences for comparison is:

1. Close the window shade, turn on all light in the room, and fix the camera on a tripod.
2. Fix the markers on the background designed for the evaluation.
3. Capture the video sequence of a certain length at a certain rate, for example, 45 sec at 10 frame/sec.
4. During the video capturing, we hold the tripod and try to give slow micro movement to force the images to be different from frame to frame.
5. Only replace the markers on the background, take videos of the same length at the same rate for all other marker sets.
6. Use video editor to cut-off the first 30-80 frames of every video sequence, so that the rest of the video frames keep a stable intensity from the first to the last. All the markers are printed on paper of the same size.

The sizes of the images are either VGA ( $640 \times 480$  pixels) or half VGA ( $320 \times 240$  pixels). Figure 5 shows one set of images extracted from the marker system evaluation videos.

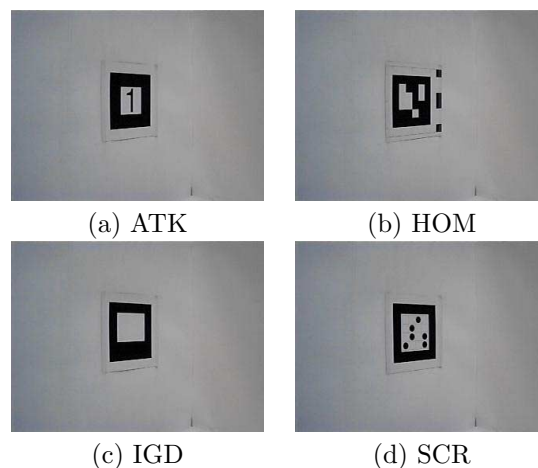


Figure 5: Images extracted from a set of video sequences for the evaluation.

#### 3.1 Processing time for marker detection and decoding

Table 1 presents the average time for the marker tracking, which includes necessary image processing, marker detection and decoding, and feature correspondence extraction of all recognized markers. In Table 1, *Size* is the video/image size in pixels; *ROM/MPF* are the region of marker(s)<sup>1</sup> and number of markers per frame.

<sup>1</sup>In this paper, region of marker(s) (ROM) stands for the smallest rectangular region that contains all the markers in the image.

For each system except the HOM, we run the marker extraction 20 times for each frame. The processing time is then averaged from processing video sequences of hundreds of frames. Figure 6 shows images extracted from the videos used for the timing evaluation. Even with the generous help from Didier Striker at IGD, we were unable to detect multiple markers in the same image using the IGD system. That is why the corresponding cells of the Table 1 are blank for IGD markers.

Table 1: Average processing time for marker recognition (*ms/frame*).

Size	ROM/MPF	Atk	Hom	Igd	Scr	ScrT
320	$68 \times 68/1$	4.1	5.1	6.2	11.6	3.5
$\times$	$61 \times 70/1$	4.1	4.9	6.4	11.9	3.1
240	$188 \times 148/3$	7.1	10.3	—	14.9	—
	$257 \times 207/10$	23.9	35.5	—	21.9	—
640	$200 \times 200/1$	13.1	13.6	19.8	58.2	22.1
$\times$	$514 \times 414/10$	41.6	51.0	—	72.9	—
480	$258 \times 218/10$	33.3	41.3	—	58.5	—

From Table 1 we observe that:

- The processing time for half VGA resolution videos is generally very small, i.e., less than ten milliseconds. This means that when a USB camera is used for video capturing, the performance bottleneck is usually not the marker extraction.
- The best running time performance for detection and decoding of a single marker is obtained by ATK. The processing time of both the ATK and the HOM marker systems appear to be highly dependent on the number of markers in an image. The SCR system perform worst. It is however less dependent on the number of markers.

In Table 1, there is also a column '*ScrT*' to present the time for the processing when the tracking feature of the SCR marker system is turned on. When the '*tracking*' is enabled, this marker system tracks a single marker based on the marker features extracted from the previous image. The performance of this system dramatically increases in this case. The current versions of the other systems do not take advantage of the interframe '*tracking*' information.

### 3.2 Accuracy of feature points

Since that the ground truth of the correspondence is not available for error evaluation, we can only compare the feature points extracted from the marker tracking systems with the 'ground truth' such as the corners extracted using a well established corner detection algorithm. In the evaluation, we use the corner detection implemented in the OpenCV library (OCV) (see [14]) to compute the correspondence of the corners. Another 'ground truth' we used for the evaluation is obtained as follows:

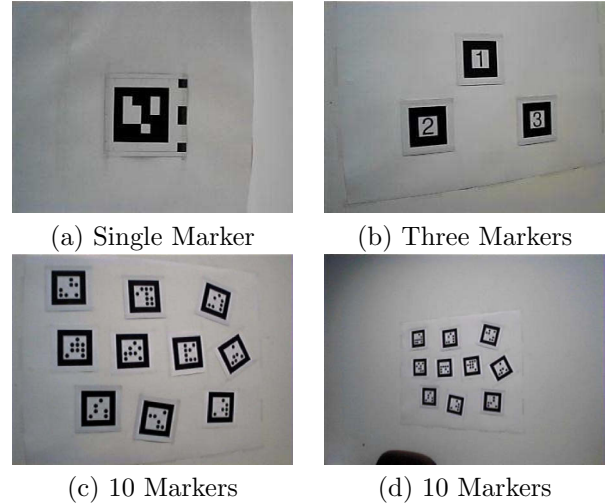


Figure 6: Sample images for marker detection timing.

- use the edge points detected around each corner of a marker to fit two straight lines in that small region;
- compute the intersection of the two straight lines as the 'ground truth' point for comparison (see Figure 7).

The reason for creating the second 'ground truth' for comparison is that the process of edge detection, least square line fitting, and intersection (LIT) can act as filters to eliminate some of the image noise and obtain better feature points.

All figures in this section are color coded. Therefore they need to be printed in color pages.

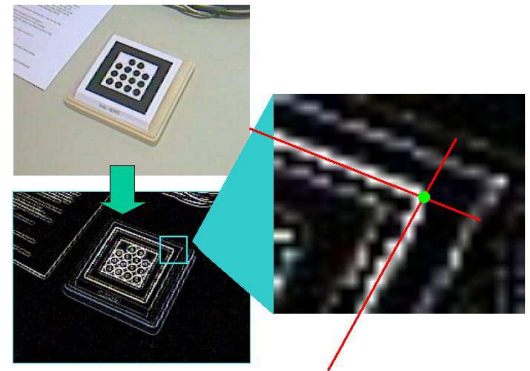


Figure 7: Extracting feature points for comparison using line fitting.

All videos used in this evaluation are half VGA size, well focused, and with a single marker of  $70 \times 70$  pixels appearing at the center of the images. With this, the effect of the radial distortion is reduced to a minimum. Thus the *true* feature points should be very close to the LIT or OCV points. Figures 8 to 10 are the '*error*' distribution of the

extracted feature points relative to the LIT points. The average distance and the standard deviation are presented in Table 2. In the figures and the table, the *angle* is the viewing angle of the camera shown in Figure 15.

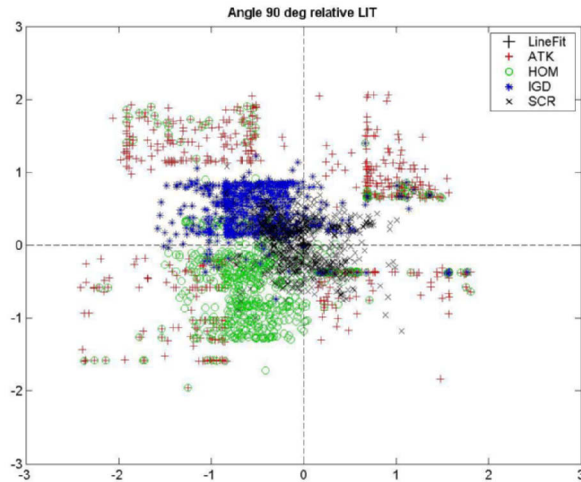


Figure 8: 'Error' distribution related to LIT points (camera angle: 90°).

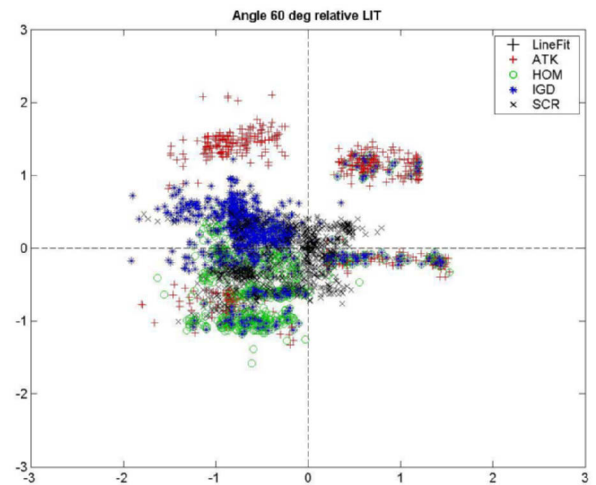


Figure 9: 'Error' distribution related to LIT points (camera angle: 60°).

Table 2: 'Errors' related to the LIT points (Average Distance)/(Standard Deviation) (in pixels).

angle	Atk	Hom	Igd	Scr
90°	1.43/0.57	0.98/0.48	0.84/0.29	0.46/0.26
75°	1.43/0.51	0.84/0.43	0.85/0.31	0.58/0.34
60°	1.27/0.43	0.88/0.35	0.84/0.33	0.61/0.34
45°	1.57/0.42	0.92/0.46	0.99/0.44	0.63/0.40
30°	1.28/0.39	0.92/0.37	0.94/0.38	0.63/0.32
Avg.	1.40/0.46	0.91/0.42	0.89/0.35	0.58/0.33

Figures 11 to 13 are the 'error' distribution of the system extracted feature points related to the OCV points. The average distance and the standard deviation are presented in Table 3.

From both the comparisons to the LIT and OCV results, we observe that the error of feature extraction is higher for the ATK system. The reason is that ATK system directly extracts the features from the binary images. This method saves computational effort, but leaves larger error in the feature extraction. We also applied the ATK marker detection to a poorly focused image, then we found that the difference between the ATK results and the OpenCV corner detection results were more than 2 pixels, as presented in Figure 14. The rest of marker systems in this study perform well when comparing to both the LIT and the OCV results.

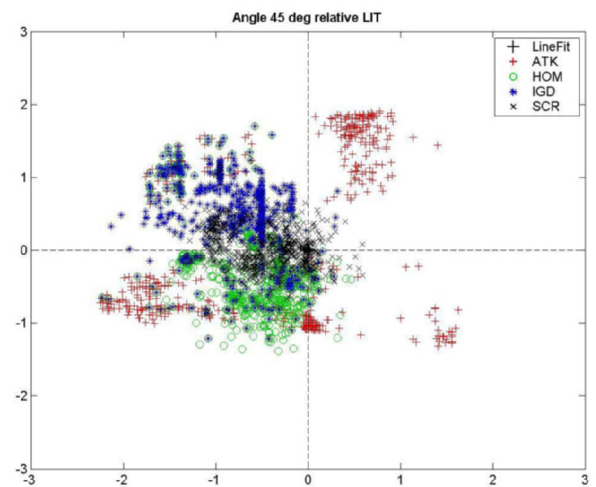


Figure 10: 'Error' distribution related to LIT points (camera angle: 45°).



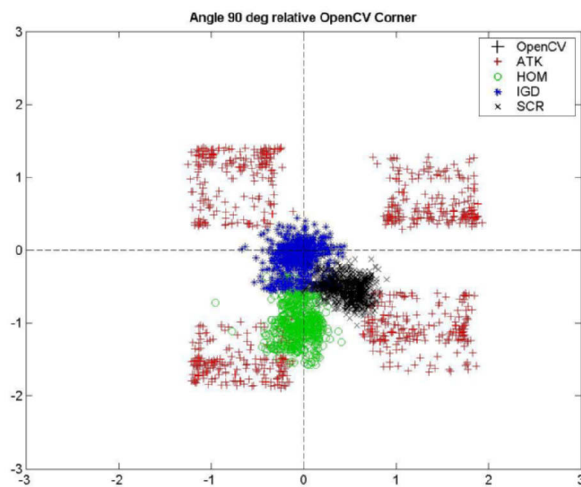


Figure 11: 'Error' distribution related to OCV points (camera angle: 90°).

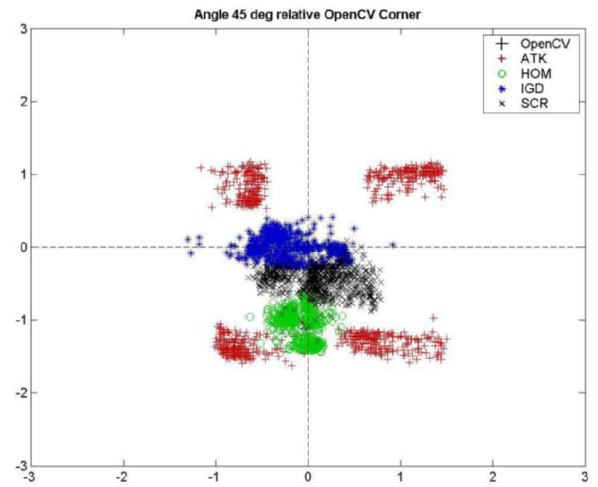


Figure 13: 'Error' distribution related to OCV points (camera angle: 45°).

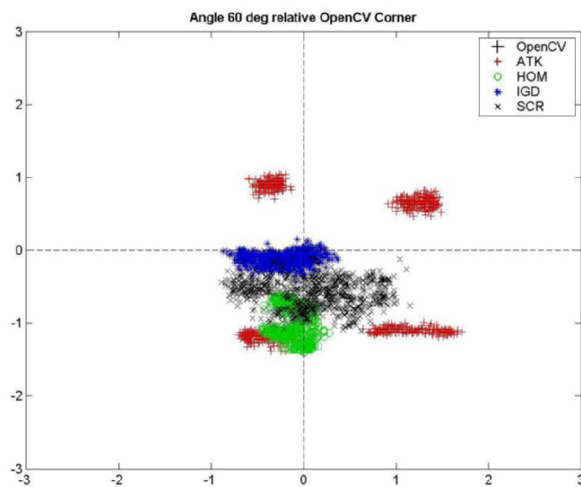


Figure 12: 'Error' distribution related to OCV points (camera angle: 60°).

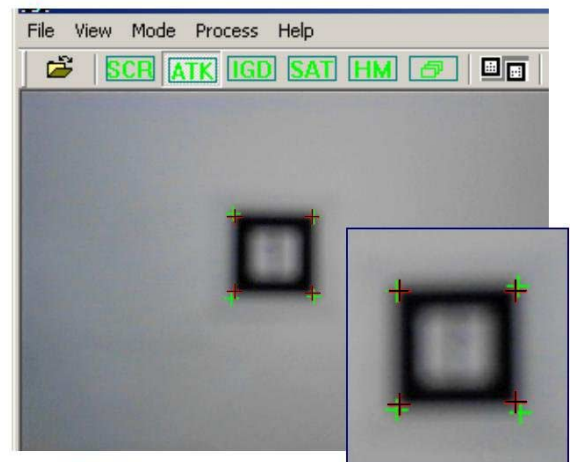


Figure 14: A screen snapshot when apply the ATK marker recognition to a bad focused image, the red crosses are the OpenCV corner detection results and the green crosses are the correspondences from the ATK system. The differences here are  $\geq 2$  pixels.

Table 3: 'Error' related to the LIT points (Average Distance)/(Standard Deviation) (in pixels).

angle	Atk	Hom	Igd	Scr
90°	1.55/0.32	1.22/0.14	0.17/0.10	0.59/0.14
75°	1.44/0.07	1.12/0.05	0.22/0.11	0.59/0.09
60°	1.42/0.12	1.17/0.05	0.37/0.13	0.78/0.24
45°	1.50/0.21	1.05/0.13	0.39/0.11	0.53/0.12
30°	1.23/0.18	1.16/0.06	0.44/0.24	0.71/0.17
Avg.	1.43/0.18	1.14/0.09	0.32/0.14	0.64/0.15

### 3.3 Marker recognizability

Marker recognizability evaluates the ability of a marker system to reliably detect *and* decode markers under various unfriendly conditions. In this work, we tested the recognition with small region of marker (ROM), the marker recognition rate against projective distortion, and the marker recognition rate from videos recorded by not well focused cameras.

**Recognition under projective distortion:** Figure 15 shows the setup for recording videos that the markers of different projective distortion. The projective distortion level is represented by the viewing angles from 90° to 15°. The comparison results are presented in Table 4. Note that the same videos were also used to evaluate the feature extraction accuracy of the marker systems presented above.

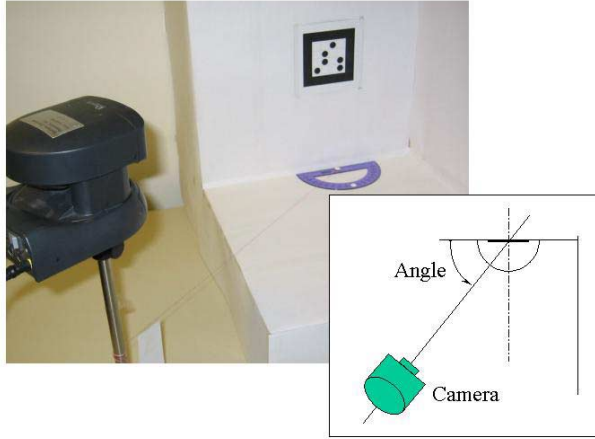


Figure 15: Setup for recording videos with different projective distortion.

The ATK system associates each extracted marker with a confidence value  $cf$  for marker decoding. When  $cf \geq 0.50$ , the ATK system regards the marker as being recognized. In the HOM system, the marker recognition confidence is evaluated with 7 levels, labeled from 0 to 6, with 0 being the highest and 6 being the lowest. When the

Table 4: Marker recognition rate under projective distortion (%).

angle	Atk	Hom	Igd	Scr
90°	100	100	100	100
75°	100	100	100	100
60°	100	100	100	100
45°	100	100	100	98
30°	100	100	100	95
15°	71/( $cf \geq 0.50$ ) 16/( $cf \geq 0.67$ ) 8/( $cf \geq 0.75$ )	100	0	7

confidence value is  $\leq 2$ , the marker is recognized with very high confidence. Only when the marker is detected and decoded with high confidence, the SCR system regards the marker as being recognized, nothing will be reported otherwise. This feature reduces false recognition, but it could also result in a relatively lower recognition rate.

**Recognition with videos of multiple markers:** Table 5 presents the marker recognition rate for videos captured with multiple markers.

Table 5: Marker recognition rate with multiple markers (%).

Size	ROM/MPF	Atk/cf	Hom	Scr
640 × 480	(514 × 414)/10	90/( $cf \geq 0.50$ ) 59/( $cf \geq 0.67$ ) 46/( $cf \geq 0.75$ )	100	81
	(258 × 218)/10	83/( $cf \geq 0.50$ ) 38/( $cf \geq 0.57$ ) 29/( $cf \geq 0.75$ )	100	72
320 × 240	(257 × 207)/10	83/( $cf \geq 0.50$ ) 39/( $cf \geq 0.67$ ) 14/( $cf \geq 0.75$ )	100	52
	(188 × 148)/3	100/( $cf \geq 0.50$ ) 86/( $cf \geq 0.67$ ) 58/( $cf \geq 0.75$ )	100	93

We registered 21 markers to the ATK system to test whether the system can distinguish between markers that have similar patterns. Figure 16 shows some of the marker recognition problems we found when checking the details of the output from the ATK system. For example, the marker with a pattern of the number '3' is recognized as the marker with pattern of '2' with a high confidence value ( $cf = 0.85$ ) from the ATK system. We have not found false recognition problem from the output of other systems. The results presented in Table 4 and 5 are the output directly from the respective marker systems which does not distinguish the false recognitions as shown in Figure 16. We are currently working on eliminating the false recognition to present the correct counting.

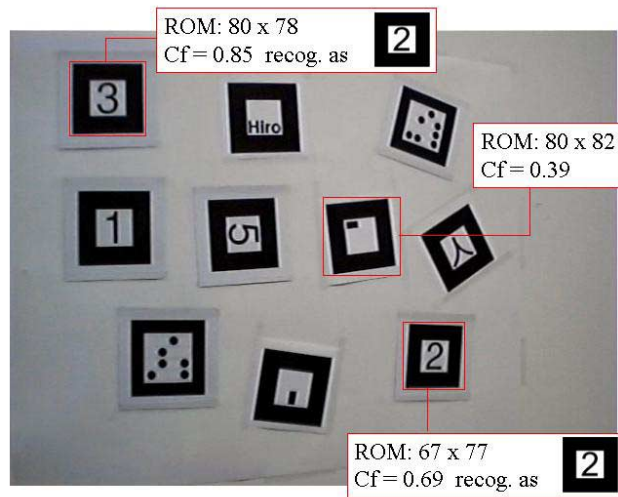


Figure 16: Marker recognition problems of the ATK system

**Recognition with small region of marker<sup>2</sup> (ROM):** Here we evaluate the performance of these systems when the image (pixel) size of the marker is small. We use the SONY EVI-D30 camera to record the video sequences for the evaluation. During the recoding of the video sequences we only change the zoom of the camera so that the marker appear first with a large ROM (Figure 17.(a)), then we smoothly zoom out the marker to a very small ROM (Figure 17.(b)). We keep all other factors, such as the depth from the camera to the marker and the physical size of the markers, that affect the marker image size (in pixels) to be unchanged. The high-speed auto-focus feature of SONY EVI-D30 keeps the camera well focused from the start to the end. The results are shown in Figure 18. All the video sequences we used in the evaluation are of half VGA size.

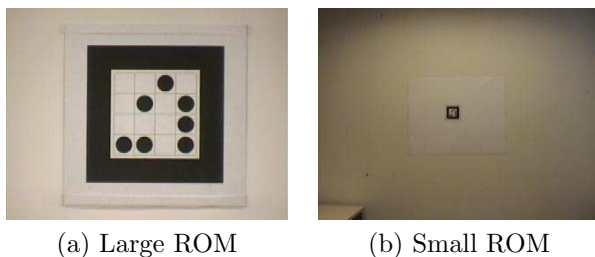


Figure 17: Zoom out the marker from large to small.

The results presented in Figure 18 show that the ATK system performed the best in this experiment. It can detect and decode a marker of  $14 \times 14$  pixels from a  $320 \times 240$  pixels.

<sup>2</sup>In this paper, region of marker (ROM) stands for the smallest rectangular region that contains all the markers in the image.

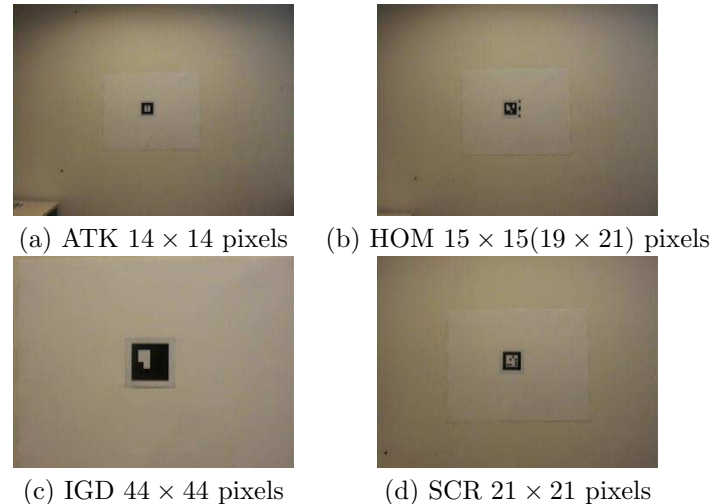


Figure 18: Marker recognition with small region of interest (image size  $320 \times 240$  pixels).

**Recognition with poorly focused videos:** We use USB camera to record the videos for this evaluation so we can manually adjust the focus. First, we manually adjust the camera focus to the best level, then record the video sequences for each marker system. Then we turn the camera focusing dial for about  $30^\circ$  each step to force it out of focus. At each focus level, we record a half VGA video sequence for each marker system. Figure 19 shows the images extracted from the videos of different focus levels. The marker recognition results are presented in Table 6. It is intriguing to see that the confidence value (cf) provided by the ATK system for the worst focused video is higher than those of some of the better focused videos. This suggests that the confidence value obtained by the ARK system may not be quite reliable.

Table 6: Recognition rate with poorly focused videos (%).

Focus	Atk	Hom	Igd	Scr
Perfect	100 (cf=0.79)	100	100	100
Good	100 (cf=0.81)	100	100	100
Bad	100 (cf=0.63)	100	28	97
Worse	100 (cf=0.56)	0	12	0
Worst	100 (cf=0.73)	0	0	0

## 4 The Qualitative Discussion and Summary

The ATK marker system is compatible with most of the computer systems. It is well documented and easy to used. The ATK marker detection and decoding are fast and stable. It scored the best in several aspects, such as the processing speed for a single marker and the recognition with a small region, for the tests presented in this



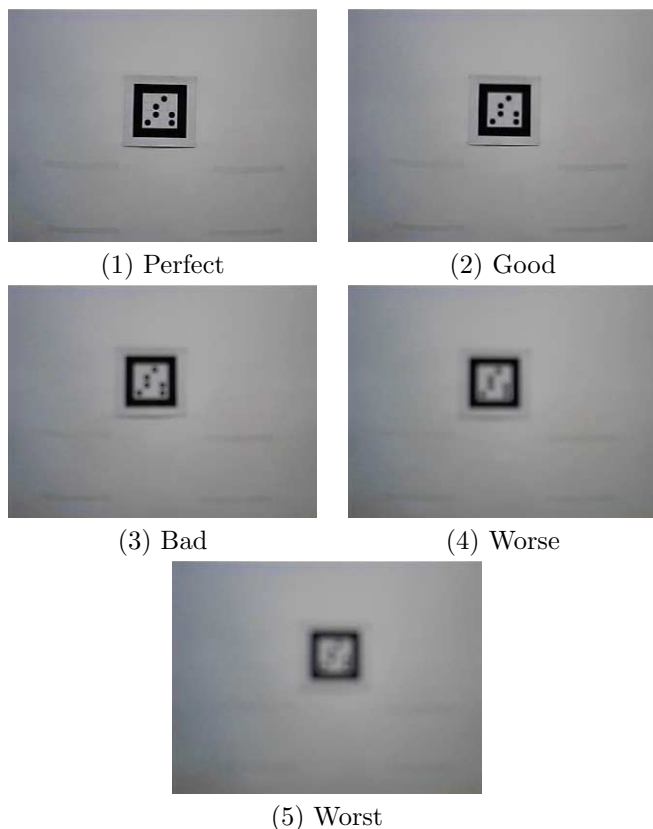


Figure 19: Video sequences of different focus levels for marker recognition (ROM  $55 \times 55$  pixels, image size  $320 \times 240$  pixels).

paper. It is suitable for prototyping and has been used widely in various AR applications. For applications that require hundreds or thousands of different markers such as maintenance and localization in large industrial environment or office buildings, however, it is quite tedious to register every marker and then modify the marker registration file manually as required by the system.

The ATK system uses a highly simplified template-matching algorithm for the marker decoding. The algorithm compares several invariants of a marker detected in the image with those of the patterns registered in the system. The advantage of this approach is high processing speed. On the other hand, such simplification also causes problems such as the false marker recognition shown in Figure 16. To solve the false recognition problem, a rigorous template-matching algorithm is needed which could be computationally expensive, especially when the number of pre-registered markers is large.

The ATK system directly use the binary image that produced for marker detection to extract the image feature points. Since the edge positions of a marker are dependent on the threshold for binarizing the image, it may cause system errors in the accuracy of the extracted feature points as shown in Figure 14.

The HOM marker system performs well in many aspects. The marker is designed with systematic coding as both the IGD and SCR markers are. In addition, there is extra visual structure on each marker that increases decoding reliability. A nice feature offered by the HOM marker system is that it provides 7 confidence levels, namely from 0 to 6, to indicating the reliability of marker detection and decoding result for each marker detected. When the confidence level is  $\leq 2$ , the marker is recognized with high reliability. If the confidence level is  $\geq 5$ , the marker decoding results is usually not quite reliable. From our testing and real application in industries, we observe that the HOM confidence level is reliable and consistent with the image quality of the markers. This is very different from what we observed from the confidence level offered by the ATK system, which is not so reliable and consistent in some cases as shown previously.

The SCR marker system is the slowest for the detection and feature extraction on a single frame. However, it uses the temporal tracking information to perform faster than other systems on video sequences and the performance degrades minimally as the number of markers per frame increases. It performs reasonably well in other aspects such as for the accuracy of feature point extraction, and for the reliability of detection and decoding. With a systematic coding, there is no need of any pre-registration for using the SCR markers. The same applies to HOM and IGD markers. With a  $4 \times 4$  coding matrix (Figure 4), there could be more than 10,000 unique codes. Each SCR marker offers 8 feature points for pose estimation resulting in a better pose estimation and superimposition with the same accuracy in feature extraction. However, the SCR system is currently available only as an OCX control for Windows applications and not for Unix applications.

The IGD marker system is widely used in many ARVIKA projects. As with those of HOM and SCR markers, IGD markers are also systematically coded. The marker feature extraction is accurate and the processing speed is reasonable. It is a little inconvenient that the user needs to create a wrapper library and compile it with the Intel C++ compiler in order to use this marker system for Windows applications. Even though we were unable to detect multiple markers from the same image using this system, we believe the functionality was implemented and blame the failure to our own unfamiliarity to the IGD system.

## 5 Conclusion and Future Works

Visual markers are widely used in existing augmented reality (AR) applications. In this paper, we compared ATK ARtoolkit, IGD, HOM and SCR systems. We presented the evaluation results, both qualitatively and quantitatively, in terms of the usability, efficiency, accuracy, and reliability of these systems. This provides an analysis of the strength and weakness of various aspects of the marker tracking systems. For a particular AR appli-

cation, there are different marker detection and tracking requirements. The evaluation results presented in this paper can guide the readers in choosing the right marker tracking system for their specific application. The evaluation of the marker systems with camera distortions, under various lighting conditions, and with complicated multiple marker configurations in 3D space are yet to be carried out. These define part of our future work.

## References

- [1] M. Appel and N. Navab. Registration of technical drawings and calibrated images for industrial augmented reality. In *IEEE Workshop on Applications of Computer Vision*, 2000.
- [2] ArToolKit. [www.hitl.washington.edu/research/shared\\_space/download/](http://www.hitl.washington.edu/research/shared_space/download/).
- [3] ARVIKA. <http://www.arvika.de/www/index.htm>.
- [4] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2000.
- [5] S. Goose, S. Sudarsky, X. Zhang, and N. Navab. SEAR: Towards a mobile and context-sensitive speech-enabled augmented reality. In *Proc. of IEEE International Conference on Multimedia & Expo.*, 2002.
- [6] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the three point perspective pose estimation problem. *Int. J. of Comp. Vision*, 13(3):331–356, 1994.
- [7] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proc. IEEE International Workshop on Augmented Reality*, pages 125–133, 1999.
- [8] N. Navab, B. Basclé, M. Appel, and E. Cubillo. Scene augmentation via the fusion of industrial drawings and uncalibrated images with a view to marker-less calibration. In *Proc. IEEE International Workshop on Augmented Reality*, pages 125–133, San Francisco, CA, USA, October 1999.
- [9] N. Navab, E. Cubillo, B. Basclé, J. Lockau, K. D. Kamsties, and M. Neuberger. CyliCon: a software platform for the creation and update of virtual factories. In *Proc. of the 7th IEEE Int'l. Conference on Emerging Technologies and Factory Automation*, pages 459–463, Barcelona, Spain, 1999.
- [10] U. Neumann, S. You, Y. Cho, J. Lee, and J. Park. Augmented reality tracking in natural environments. In Y. Ohta and H. Tamura, editors, *Mixed reality: merging real and virtual worlds*, pages 101–130. Ohmsha, Ltd. Tokyo, 1999.
- [11] I. Poupyrev, D. Tan, M. Billinghurst, H. Kato, H. Regenbrecht, and N. Tetsutani. Developing a generic augmented reality interface. *Computer*, pages 44–50, March 2002.
- [12] D. Reiners, D. Stricker, G. Klinker, and S. Müller. Augmented reality for construction tasks: Doorlock assembly. In *Proc. IEEE International Workshop on Augmented Reality*, 1998.
- [13] J. Rekimoto. Matrix: A realtime object identification and registration method for augmented reality. In *Asia Pacific Computer Human Interaction*, 1998.
- [14] OpenCV: Open source computer vision library. <http://www.intel.com/research/mrl/research/opencv/>.
- [15] D. Stricker and T. Kettenbach. Real-time and markerless vision-based tracking for outdoor augmented reality applications. In *IEEE Int. Symp. on Augmented Reality*, 2001.
- [16] X. Zhang, Y. Genc, and N. Navab. Mobile computing and industrial augmented reality for real-time data access. In *Proc. of the 7th IEEE Int'l. Conference on Emerging Technologies and Factory Automation*, 2001.
- [17] X. Zhang, Y. Genc, and N. Navab. Taking AR into large scale industrial environments: Navigation and information access with mobile computers. In *IEEE Int. Symp. on Augmented Reality*, 2001.
- [18] X. Zhang and N. Navab. Tracking and pose estimation for computer assisted localization in industrial environments. In *IEEE Workshop on Applications of Computer Vision*, pages 214–221, 2000.
- [19] X. Zhang, N. Navab, and S. Liou. E-commerce direct marketing using augmented reality. In *Proc. of IEEE International Conference on Multimedia & Expo.*, 2000.
- [20] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proc. Int. Conf. Comp. Vision*, pages 666–673, 1999.

## Acknowledgements

We thank Dr. Y. Genc, Mr. V. Kumar, and Dr. C. Akinlar for their helps related to this work.