# Painting Alive: Handheld Augmented Reality System for Large Targets

Jae-In Hwang, Min-Hyuk Sung, Ig-Jae Kim, Sang Chul Ahn, Hyoung-Gon Kim, and Heedong Ko

Imaging Media Research Center, Korea Institute of Science and Technology

**Abstract.** This paper presents a handheld augmented reality (AR) system and an authoring method which provides alive contents in large targets. In the general augmented reality tools, they are not designed for large targets but for only adequate size of target which fits in the screen. Therefore we designed and built a vision-based AR system and an authoring method that can handle much larger targets than the view frustum.

**Keywords:** augmented reality.

## 1 Introduction

Recently, handheld augmented reality (AR) technology is growing drastically with the advances of mobile devices such as smartphones and tablet-PCs. There are many diversities of application for handheld AR such as games, visual information providers, advertisements, and so on. One of the adequate applications is mobile tour guide in the museum or sightseeing places [1]. While working on our project named Mobile Augmented Reality Tour (MART) since 2009, we have found several interesting research issues about handheld AR. Because the goal of the project was providing augmented contents through mobile devices during tours in the museum or places, there were many targets that have various sizes and forms. In cases of small targets, we could apply existing augmented reality tracking algorithms or tools such as SIFT [2] or SURF [3]. But most of the tracking methods were designed for the screen-fit size object. Then what happens if we can see just ten percent of the object through camera in the mobile device? The system would have difficulties in recognizing and tracking the object. So the augmented contents would not appear or could be placed on the wrong position.

In the project MART, we had technical challenges to adding augmenting contents on the "Painting of Eastern Palace" which is 576 centimeters in width and 273 centimeters in height. The behavior of tourist using handheld AR tour guide is not predictable. They could focus on any certain part of the painting from various viewing positions and directions. So the technical challenge here was building handheld AR tracking system with the unpredicted view of the large target. In the remaining part of this paper, we will show details of the tracking system. Moreover, we also present about the authoring method for providing various multimedia contents for the objects.

R. Shumaker (Ed.): VAMR/HCII 2013, Part I, LNCS 8021, pp. 32–38, 2013.

**Fig. 1.** Example of large target for handheld AR, Painting of Eastern Palace (National Treasure of South Korea, located at the museum of Donga University, 576cm×273cm)

## 2 Vision-Based Tracking for Large Targets

### 2.1 Divide and Conquer Method in Handheld AR

The basic idea of the large target tracking is simple. We divide the large target into multiple pieces and store feature points in the database. At the beginning of the execution of the AR program, we compare input features with features of database. We can find which part we are looking at by comparing feature points of each piece with input feature point set. This step is called target recognition. Once we find the piece what we looking at, we compute position and pose of the camera. The feature matching process is described in the Sec. 2.3. When we lost the target, we do the same procedure again. The time for finding target piece depends on the numbers and resolution of the pieces. In our case, we divided our target into four parts, and recognizing each piece takes less than 50 milliseconds. (If it takes more, the latency of the tracking would be noticeable.) So, it would take less than one second to recognize the current target and compute camera poses among 7 to 8 large targets.

To avoid the failure of the matching caused by the change of user's viewpoint, we trained different scale of the target. This procedure produces more image pieces than just dividing. In our case, we added 4-5 more pieces in case when the user wants to see specific part of the painting. As the result, it took within one second to find the part when we lost them.

### 2.2 Tracking States

In cases of handheld AR, abrupt movement of handheld cameras becomes the main factor that makes the tracking procedure unstable. Particularly in our

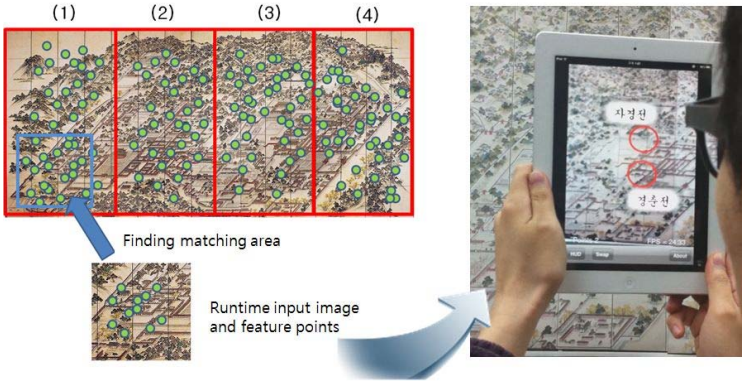


**Fig. 2.** Feature matching method for large targets

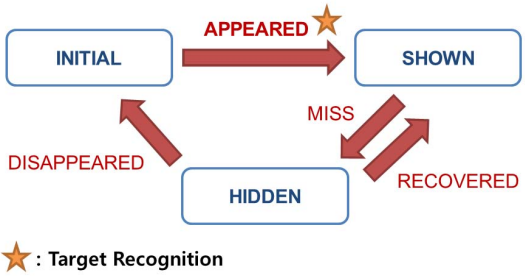


**Fig. 3.** Tracking states and transitions between them

system, it can be triggered to find the new target piece even for one moment miss of the current target. The same situation of momentary target miss can also be occurred due to camera noises that are usually observed in low quality cameras used by mobile devices. In our system, we avoid this problem by defining three states in the tracking procedure as shown in Fig. 3. From the "Initial" state, we move to the state "Shown" when a target piece is found in the recognition step. When the found target disappear in the camera view, we do not directly move to the "Initial" state, but temporary move to a state "Hidden". This "Hidden" state indicates that the current target momentary disappeared but will be shown soon. The target recognition step is executed only when we have moved back to the "Initial" state. This hidden-state system may delay the shift from one target piece to the other, but this artifact is merely noticeable if we define the lasting time in "Hidden" state as short. In our implementation, we set 5 frames for the "Hidden" state period.

**Fig. 4.** Example of 2D image strip for animation

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | image | king | king | 566 | 574 | 128 | 128 | | | | |
| 2 | image | red_men | redman_bow | 196 | 788 | 150 | 150 | 10 | 0.02 | | |
| 3 | image | yellow_men | yellowman_bow | 530 | 788 | 150 | 150 | 10 | 0.02 | | |
| 4 | image | spear_men_1 | 2spearman | 838 | 470 | 186 | 186 | 3 | 0.1 | | |
| 5 | image | spear_men_2 | 3spearman | 698 | 618 | 220 | 220 | 3 | 0.1 | | |
| 6 | image | spear_men_3 | 4spearman | 236 | 546 | 254 | 254 | 3 | 0.1 | | |
| 7 | image | text_box_01 | text_box | 360 | 50 | 560 | 272 | | | | |
| 8 | text | scene_txt_01 | scene_01 | 380 | 68 | 520 | 236 | 24 | 0.0005 | | |
| 9 | sound | scene_wav_01 | scene_01 | | | | | | | | |
| 10 | | | | | | | | | | | |

**Fig. 5.** Spreadsheet style layout of contents

### 2.3 Matching and Tracking Method

We used modified version of Histogrammed Intensity Patch (HIP) to match and tracking in real time [4]. In the training step, we generated a set of training images by artificially warping a reference image with various scales, rotations, and other affine transformations. Local image patches are extracted from training images and grouped when they are obtained from close position with similar warping. In each group of local patches, we create a simple integrated patch which of each pixel is quantized into 5 levels using histograms. When, these quantized patches are also produced in runtime, it can be done much faster since we only handle one specific viewpoint. Moreover, the matching between patches can also be computed quickly using bit-wise operations.

## 3 Multimedia Contents Layout Authoring for Handheld AR

### 3.1 Multimedia Contents Types

There are many different types of digital contents which can be presented through handheld AR. In our system, we decided to show 2D animation which can be blended easily on the old painting. Also, we added audio/text narrations which can deliver historical stories.

2D animation can be made with series of images for each frame. As shown in Fig. 4, the images are stitched in one image. The animation is shown during run-time by putting each part of images onto the frame buffer.

**Fig. 6.** Result of the layout authoring

### 3.2 Layout Authoring

We use spreadsheet style layout for the contents authoring. Using the layout style, it is very easy and intuitive to locate and display various digital contents on the screen. In the Fig. 5, the first column represents the type of the contents. The second column contains object names and third column shows names of resource files. Also, the next 4 columns indicate the left upper position (x, y) and size (x, y), respectively. The last 2 columns are optional for animation; the number of strip frames and animation speed. In case of the text, we can animate text scrolling by putting animation speed at the column 'I'.

Fig. 6 shows the result of the layout authoring. We placed scrolling textbox, animated characters and a narration sound. When the user looks the painting through the handheld camera, multimedia contents appear instantly. As described on the spreadsheet-style layout, king and other characters are located. The textbox shown in the figure is an animated textbox, therefore it scrolls itself as time goes on.

### 3.3 Illumination Adaptation

In some practical applications, we may experience a decline in the tracking performance due to illumination of real situations. For example of museums, the lighting is generally very dark for protecting displayed stuffs from being exposed to strong lights. It means that the camera captured image can look quite different with the trained image as both images cannot be recognized as the same one in the tracking procedure. A common solution for this case is to simply use the camera captured image from the training step. In this case, however, it is too difficult to place virtual objects in proper positions of the augmenting 3D space. As an example, it is practically impossible to capture the image from the exact front view. This indicates that the real image plane may be a bit tilted in the captured image as shown in Fig. 7 (a). Hence, we cannot put a virtual object to be exactly fit on the real image plane.

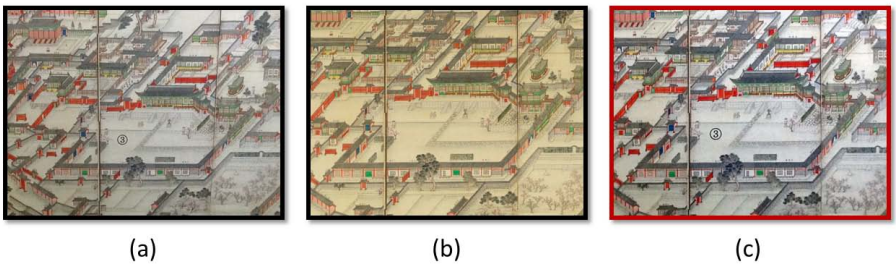


**Fig. 7.** Tone mapping. (a) A captured image, (b) The ground truth image, (c) The transformed image from (a) to (b)

To solve this issue, we focused on the trade-off between robustness and computation time of feature descriptors. Our modified HIP descriptor provides real-time speed, but cannot overcome the difference of illumination unless all diverse lighting conditions are considered in the training step. (Even if they were so, considering more conditions lead to slower speed in runtime.) On the other hand, some other descriptors such as SIFT that are too slow to be used in real-time applications may work robustly even for the difference of illumination. Therefore, as a pre-processing step, we transform the given captured image using SIFT [2] to be looked the same with the ground-truth image. Indeed, we perform this as a sort of tone mapping process. As shown in Fig. 7, the transformed image (c) not only has the exactly same arrangement of scenes with the ground truth (b), but also reflects the illumination of the real situation as (a).

## 4 Conclusion and Future Work

In this paper, we presented a handheld AR system which can show information for the very large target. We presented the method for the matching and tracking for large targets by divide and conquer. Also, we presented the spreadsheet style based layout authoring for AR contents. By building total procedure, we could show various augmented contents without much efforts. Developed system has been installed at the National Palace Museum of Korea. We have several future plans to improve current system. In the current system, we can detect a few but not many large targets such as more than hundred targets. We could do that by adding detection module which can detect numerous targets before tracking stage.

**Acknowledgement.** This research is supported by Ministry of culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA), under the Culture Technology (CT) Research & Development Program 2009.

## References

1. Vlahakis, V., Ioannidis, M., Karigiannis, J., Tsotros, M., Gounaris, M., Stricker, D., Gleue, T., Daehne, P., Almeida, L.: Archeoguide: an augmented reality guide for archaeological sites. IEEE Computer Graphics and Applications 22(5), 52–60 (2002)
2. Ke, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, June-2 July 2004, vol. 2, pp. 506–513 (2004)
3. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Speeded-up robust features (surf). Computer Vision and Image Understanding 110(3), 346–359 (2008)
4. Taylor, S., Rosten, E., Drummond, T.: Robust feature matching in 2.3 $\mu$s. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009, pp. 15–22 (June 2009)