

# Mono-spectrum marker: an AR marker robust to image blur and defocus

Masahiro Toyoura · Haruhito Aruga ·  
Matthew Turk · Xiaoyang Mao

© Springer-Verlag Berlin Heidelberg 2013

**Abstract** Planar markers enable an augmented reality (AR) system to estimate the pose of objects from images containing them. However, conventional markers are difficult to detect in blurred or defocused images. We propose a new marker and a new detection and identification method that is designed to work under such conditions. The problem of conventional markers is that their patterns consist of high-frequency components such as sharp edges which are attenuated in blurred or defocused images. Our marker consists of a single low-frequency component. We call it a mono-spectrum marker. The mono-spectrum marker can be detected in real time with a GPU. In experiments, we confirm that the mono-spectrum marker can be accurately detected in blurred and defocused images in real time. Using these markers can increase the performance and robustness of AR systems and other vision applications that require detection or tracking of defined markers.

**Keywords** Augmented reality · Spectrum analysis · Planar marker

## 1 Introduction

Estimating the relative position and orientation of a camera from images is a fundamental requirement of augmented reality (AR). The technique is used for table top interfaces [10], registration for AR surgery systems [4] and many other

applications, such as estimating robot position and posture or tracking of moving objects. A planar marker is an efficient and effective tool for this estimation in many cases.

However, these applications tend to produce images that are blurred due to camera motion or defocused due to variations in scene depth with a fixed focal distance, and these artifacts cause problems with typical marker detection and identification methods. We propose a new planar marker for mobile cameras and moving objects which can be reliably detected even in blurred or defocused images.

Conventional markers, such as those used in ARToolKit [9] and QR Code, have patterns with high-frequency components such as edges or corners. These attributes make their markers hard to detect in blurred or defocused images, which have attenuated high-frequency components.

The problem can be solved with the marker having a characteristic frequency spectrum. In this paper, we describe the methods for making the new marker and detecting it in blurred and defocused images. We call the marker a mono-spectrum marker. The key idea is that the mono-spectrum marker consists of only low-frequency components. The frequency components pass through the low-pass filtering of blurring and defocusing with a small loss. The markers can be detected in real time with GPUs by analyzing their frequency components. Our approach has a relatively high computational cost, and thus requires GPU computing. We believe that the rapidly growing availability of GPUs on mobile devices will make this approach to deal with blurred and defocused images feasible.

## 2 Related works

The physical marker of ARToolKit [9] is a popular tool for calibration, estimating the pose of the camera relative to the

---

M. Toyoura (✉) · H. Aruga · X. Mao  
University of Yamanashi, 4-3-11, Kofu,  
Yamanashi 400-8511, Japan  
e-mail: mtoyoura@yamanashi.ac.jp

M. Turk  
University of California, Santa Barbara, CA 93106-5110, USA

scene. The marker consists of a black bold frame and an internal binary pattern. It is detected in camera images, and its position and orientation are estimated by locating the four corners of the black frame. The internal pattern helps to identify the marker. Other markers, such as the ARTag [6], QR Code, and random dot markers [19] also consist of binary edges, corners or dots of black and white.

These conventional markers are difficult to detect and identify under conditions of image blur and defocusing, which attenuates the high-frequency components from their sharp edges and corners. Fiala [6] reported how much Gaussian noise affects marker detection, but did not address the effect of image blurring and defocusing.

Changes in marker appearance are considered in the nested marker [18] and Bokode tag [12]. The nested marker has a fractal structure so that it looks the same at long distance as it does at short distance, to provide distance invariance. The Bokode tag consists of a light-emitting component, which is best detected when the camera's focal length is set to infinity. These markers do not consider the possible loss of high-frequency components in images.

Natural feature points such as SIFT [11] and SURF [3] also enable one to do calibration. The feature points are detected not in planar markers but in general objects. Systems like PTAM [10] or DTAM [13] for registration of real environment and virtual objects are constructed based on the map of natural features. The applications are different from those of planar markers, since the feature points of the planar markers are registered in a database in advance to estimate the position and posture of target objects. Like our marker, SIFT and SURF were designed with frequency changes in mind. They provide frequency-shift-free tracking. However, they do not consider the loss of high-frequency components altogether, but rather just changes in frequency components at least when the environmental map is reconstructed. The Fourier tag [21] and fiducial image [15] use frequency components for identification, but the components do not contribute to robust detection or identification in blurred or defocused images.

We also proposed a new AR marker in [1], but this marker could not be correctly detected when viewed with a large distortion caused by perspective projection when the plane containing the marker is tilted with respect to the camera plane. The problem is addressed in the current paper by the introduction of vertical and horizontal filters. The details are described in Sect. 3.

Several template matching methods can deal with the blurring and defocusing problem. ESM-Blur [16] creates blurred templates in advance. Image resampling [7] addresses the problem by sequentially updating templates. The practical disadvantage of the template matching method is that only a few kinds of markers can be used. Conventional markers for AR support hundreds of markers or more, as does our mono-spectrum marker.

The failure of marker detection can be mitigated by interpolating the results of detection in frames [20]. The method assumes that the detection failure occurs only in a few frames, but this assumption is not valid when the images are blurred or defocused.

The recovery of image quality is being studied actively [2, 17]. Blurred or defocused images are *degraded images*, and the aim is to restore such images so that they are free of blurring and defocusing. The degradation is modeled with a point spread function (PSF). The images are restored by estimating the PSF and its inverse function. However, restoration works well only when the blurring and defocusing are small. That means images taken with actively moving cameras cannot be effectively restored. Note that our marker can be used together with the restoration methods.

Blurring and defocusing parameters are used for displaying virtual objects on the observed images [14]. The parameters are extracted from the appearance changes of the special pattern marker of ARToolKit under the assumption that the blurring and defocusing are not large. The appearance of the virtual object is adjusted with the extracted parameters. Our marker can be used to estimate parameters from blurrier and more defocused images, making the appearance adjustment method more usable.

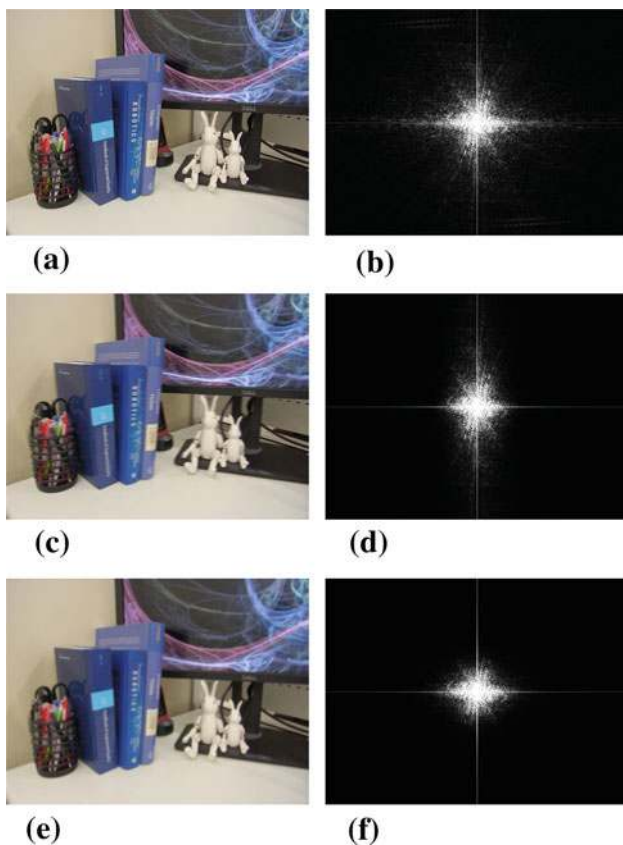
### 3 Mono-spectrum marker design and detection

#### 3.1 Inhibition of high-frequency components by image blurring and defocusing

Images are blurred when the relative position between the camera and an observed object changes during exposure. Blur happens when either the camera or the object is moving. Although a shorter exposure time can solve the problem in some cases, it tends to produce underlit, and thus noisy, images.

Figure 1b depicts the frequency spectrum of the observed image shown in Fig. 1a. With the camera moving horizontally, we obtain a blurred image Fig. 1c, and its spectrum changes as in Fig. 1d. The spectrum loses the high-frequency components along the x-axis.

Images are defocused when the focal length of the camera is incorrectly set for the current distance between the camera and the object. The degree of defocusing depends on the difference between the focal distance and the distance between the camera and the object. Short exposure time, adequate lighting, stable cameras and objects, and appropriate focal length would solve the above problem; however, these measures are impractical in most dynamic situations, especially in AR applications. Moving cameras often result in blurred images. Moreover, the focal length is either fixed or adapts to one of many possibly scene objects.



**Fig. 1** Change of frequency spectrum due to blurring and defocusing. **a** Original image, **b** spectrum of **a**, **c** blurred image, **d** spectrum of **c**, **e** defocused image, **f** Spectrum of **e**

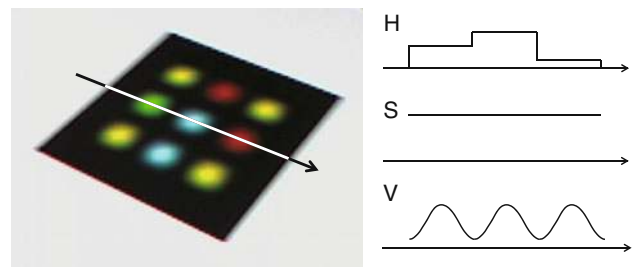
A defocused image is shown in Fig. 1e; its spectrum changes can be seen in Fig. 1f. The spectrum of a defocused image loses high-frequency components in all directions.

Conventional markers are designed without consideration of the circumstances where high-frequency components are lost. They are difficult to detect in blurred or defocused images.

### 3.2 Design of the mono-spectrum marker

We present the design of our marker, which is designed to be invariant to the blurring and defocusing artifacts described above. The extraction method covers both *detection* and *identification* of the marker in the observed images.

The mono-spectrum marker consists of the pattern shown in Fig. 2, a two-dimensional sinusoidal intensity pattern with multiple colors that looks like a pattern of blurry dots to the observer. We designed the marker so that its brightness changes at a single low frequency. This low-frequency component is little affected by blurring and defocusing. Hence, the regions in the observed images corresponding to the markers also have a single low-frequency component.



**Fig. 2** Design of the mono-spectrum marker

The color value of each position of the mono-spectrum marker is given in hue, saturation, and value (HSV) color space. The values of  $H$ ,  $S$  and  $V$  are determined as follows.

A special  $H$  value, yellow in our experiment, is reserved for the corner dots of the marker. That is, colors excluding yellow are used for dots that are not corners. The markers are identified through the alignment of the colors of dots. Individual colors should be distinguishable even when they are slightly changed in the images. We empirically used 6 colors for making markers. A larger number of colors will not be distinguishable in various illumination conditions.

The value of  $S$  is set to the maximum value for each position.

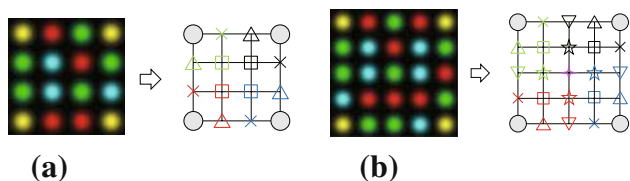
The value of  $V$  changes with the single low-frequency component. The regions of the markers are distinguished from other regions by using the characteristic frequency spectrum. Let us denote by  $r$  the distance from the center of region, and  $L$  the width and height of a square region. The value of  $V \in [0, 1]$  is defined as follows:

$$V = \begin{cases} (1 - \alpha) \left( \cos \frac{2\pi r}{L} \right) + \alpha, & (r \leq \frac{L}{2}) \\ 0, & (\text{otherwise}) \end{cases} \quad (1)$$

where  $\alpha$  is the offset of the brightness. By adjusting  $\alpha$ , we can give the center of each dot a brightness which can be retained in both the printed paper and captured image. An appropriate value for  $\alpha$  can be found by checking whether the dots are clearly perceivable in the printed paper, like the one shown in Fig. 2. We set  $\alpha = 0.1$  in our experiments. Image blurring and defocusing can be assumed to be low-pass filters with a certain cut-off frequency value. The frequency components of the center of the regions should be smaller than the cut-off frequency value. If they are filtered out by blurring or defocusing, the marker detection will fail.

### 3.3 Number of possible marker patterns

Corner dots have a reserved color. Figure 3 shows rotationally identical dots. The dots indicated with the same shape are rotationally identical for the rotation angles of  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ . A group of same color dots matches another group of dots when the marker is rotated. A  $4 \times 4$  marker has 4 groups of 3 dots. We denote by  $d$  the number of dots on an



**Fig. 3** Groups of dots that are rotationally identical. **a**  $4 \times 4$  Dot marker, **b**  $5 \times 5$  dot marker

edge. In case of the  $4 \times 4$  marker,  $d$  is 4. According to the Cauchy–Froebenius theorem [5], when  $d$  is odd, a marker has 4 groups of  $u_1 = (d - 1)(d + 1)/4 - 1$  dots together with an independent dot at the center of marker. When  $d$  is even, a marker has 4 groups of  $u_2 = (d/2)^2 - 1$  dots. The number of patterns that  $m$  colors are assigned to the groups of dots is calculated as the following.

$$\begin{cases} m \left( \frac{m^{4u_1} + m^{2u_1} + 2m^{u_1}}{4} \right), & (\text{If } d \text{ is odd}) \\ \left( \frac{m^{4u_2} + m^{2u_2} + 2m^{u_2}}{4} \right). & (\text{If } d \text{ is even}) \end{cases} \quad (2)$$

The number of patterns is 825 when  $m = 5$  and  $d = 3$ , and 544, 207, 356 when  $m = 6$  and  $d = 4$ .  $m$  and  $d$  should be set in consideration of the size of markers, robustness of marker extraction, and required number of patterns.

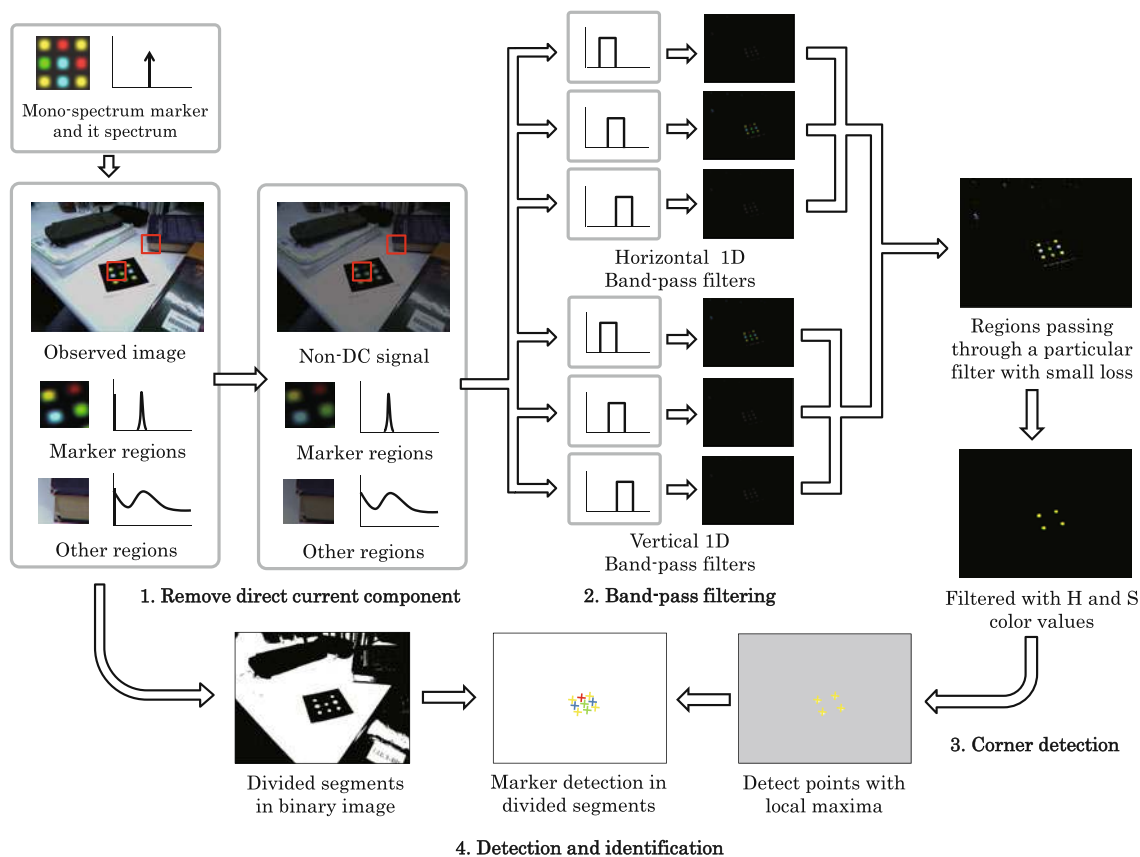
### 3.4 Extraction of mono-spectrum marker

Figure 4 shows an overview of marker detection and identification. We remove the zero frequency components (Step 1) and band-pass filtering of each region (Step 2) in the observed images. The corner dots are detected by referring to the colors of the regions (Step 3). Valid corner dots are selected by a segmentation of the original image. The mono-spectrum markers are then detected and identified (Step 4). We describe each step in detail below.

#### Step 1 Remove zero frequency component.

Let us denote the signal of mono-spectrum markers as  $f(x)$  and that of other objects as  $g(x)$ .

Geometric transform together with a perspective projection changes not only the frequency, but also the amplitude of the signal. The illumination also adds an offset to the signal. Therefore, the signal of the observed marker usually has a wider range in the frequency domain as shown in the left column of Fig. 4. In the regions corresponding to the mono-spectrum marker  $M$  and the ones of the other regions  $N$ , the following signals of different features are observed since the



**Fig. 4** Extraction of mono-spectrum marker from observed images

signals are projected on the images:

$$M : \tilde{f}(x) + c, \tag{3}$$

$$N : \tilde{g}(x) + c. \tag{4}$$

$\tilde{f}$  and  $\tilde{g}$  means that  $f$  and  $g$  are transformed with a perspective projection.  $c$  is the DC (Direct Current) offset.

In Step 1, we remove the DC components from the original signals. The high-pass filter  $b_0(x)$  that cuts off only the DC component is convolved with the observed signals.

$$M : (\tilde{f}(x) + c) * b_0(x) \approx \tilde{f}(x), \tag{5}$$

$$N : (\tilde{g}(x) + c) * b_0(x). \tag{6}$$

Since  $f(x)$  originally consists of a single frequency,  $\tilde{f}(x)$  should have a transformed single frequency and a little wider range. Thus,  $\tilde{f}(x)$  passes through the filter  $b_0(x)$  in  $M$ . On the other hand,  $g(x)$  would consist of various frequency components, and  $\tilde{g}(x)$  should not pass through the filter  $b_0(x)$  in  $N$ . The GPU convolves the band-pass filters in the space domain in real time as described in Algorithm 1.

---

**Algorithm 1** Remove zero frequency component

---

```

1:  $x = threadid.x, y = threadid.y, sum = 0.0.$ 
2: for  $(xw, yw) \in filterWindow_0$  do
3:    $sum = sum + s(x + xw, y + yw) * b_0(xw, yw).$ 
4: end for
5:  $s0(x, y) = sum.$ 

```

---

*Step 2* Band-pass filtering.

In the next step, all small regions are band-pass filtered.

Although  $f(x)$  is known in the original marker image,  $\tilde{f}(x)$  is unknown in the observed image.  $\tilde{f}(x)$  no longer appears to be mono-spectrum in the observed image because of perspective or the inclination of marker plane. In addition, neither the location nor the range in the frequency domain is known for the marker in the observed image. To solve the problem, we use multiple vertical and horizontal band-pass filters that have different pass bands. We still can expect that most frequency components are located within a certain range of the transformed single frequency at least along the vertical and horizontal lines. There should be one or more vertical and horizontal filters that pass the signal with a small loss.

Denoting  $b_i^v$  ( $i \in (1, \dots, n_v) = \mathcal{N}_v$ ) and  $b_j^h$  ( $i \in (1, \dots, n_h) = \mathcal{N}_h$ ) as vertical and horizontal band-pass filters, and  $s(x)$  as the observed image; the following region  $R$  is calculated as the candidate region of the mono-spectrum marker.

$$diff_i^v(x, y) = \left\| \frac{(s * b_0 * b_i^v)(x, y) - (s * b_0)(x, y)}{(s * b_0)(x, y)} \right\|,$$

$$diff_i^h(x, y) = \left\| \frac{(s * b_0 * b_j^h)(x, y) - (s * b_0)(x, y)}{(s * b_0)(x, y)} \right\|,$$

$$R_p = \{(x, y) \mid (s * b_0)(x, y) > th_p\},$$

$$R_v = \{(x, y) \mid \exists i \in \mathcal{N}_v, diff_i^v(x, y) < th_r\},$$

$$R_h = \{(x, y) \mid \exists j \in \mathcal{N}_h, diff_j^h(x, y) < th_r\},$$

$$R = R_p \cap R_v \cap R_h, \tag{7}$$

where  $th_p$  is the threshold of absolute brightness and  $th_r$  is the threshold of relative remaining power. The thresholds should be appropriately defined depending on the average brightness of captured images. Auto white balance function helps to define the appropriate values of  $th_p$  and  $th_r$ , and the function is generally implemented in recent consumer cameras. All the experimental results in Sect. 4 are given with the same parameters, which  $th_p$  was set to 0.05 and  $th_r$  was set to 0.20.

The band-pass filters  $b_i^v$  and  $b_j^h$  are designed in the frequency domain. The shape of each filter is determined by a particular band to be passed. An inverse Fourier transformation generates  $b_i^v$  and  $b_j^h$  in the space domain. The filter passing the signal with period of  $N_1$  pixels to  $N_2$  pixels in the space domain is designed such that values are 1 for  $1/N_2$  to  $1/N_1$  in the frequency domain.

The pseudo-code for detecting the candidate regions of the mono-spectrum markers is shown in Algorithm 2.

---

**Algorithm 2** Detect marker candidate regions

---

```

1:  $candidateRegion = \phi.$ 
2:  $x = threadid.x, y = threadid.y.$ 
3: if  $s0(x, y) > th_p$  then
4:   for  $i \in \mathcal{N}$  do
5:      $sum^v = 0.0.$ 
6:     for  $xw \in filterWindow_i^v$  do
7:        $sum^v = sum^v + s0(x + xw, y) * b_i^v(xw).$ 
8:     end for
9:     if  $\|sum^v - s0(x, y)\|/s0(x, y) < th_r$  then
10:       $sum^h = 0.0.$ 
11:      for  $yw \in filterWindow_j^h$  do
12:         $sum^h = sum^h + s0(x, y + yw) * b_j^h(yw).$ 
13:      end for
14:      if  $\|sum^h - s0(x, y)\|/s0(x, y) < th_r$  then
15:         $candidateRegion = candidateRegion \cup (x, y).$ 
16:      end if
17:    end if
18:   end for
19: end if

```

---

*Step 3* Corner detection.

The corner dots of the mono-spectrum markers have the selected  $H$  values and large  $S$  values. The corner dots are detected by searching for these values in the detected marker regions in Step 2. The centers of the dots have local maxima. A window with size corresponding to the filter frequency is



set around each pixel of the marker regions. If a pixel has the maximum value in its window, the pixel is a candidate of the corner dots.

We avoid false detections of corner dots by first smoothing the observed images to remove the noise of the observed images. The set of the candidate corner dot pixels can be also calculated in real time with the use of the GPU.

#### Step 4 Detection and identification.

Valid corner dots are selected from candidates of detected corner dots in this step. The binarized image of the original image is divided into segments. The number and positions of candidates are checked in each segment. If three or fewer candidates are detected in a segment, these candidates are removed. False-positive candidate corners may be extracted especially from a cluttered background, counting four corners in a binarized segment guarantees that only the true corners of mono-spectrum markers are detected. The ARToolKit method removes falsely detected corners in a similar manner. The method can detect multiple markers in an observed image.

The centers of the four corner dots give a homography which projects from the marker coordinates to the coordinates of the observed image. The position and posture of the marker are represented with the homography. The pattern of the marker is also recognized with the homography. The marker in Fig. 2 has  $3 \times 3$  dots. The  $H$  values of the dots

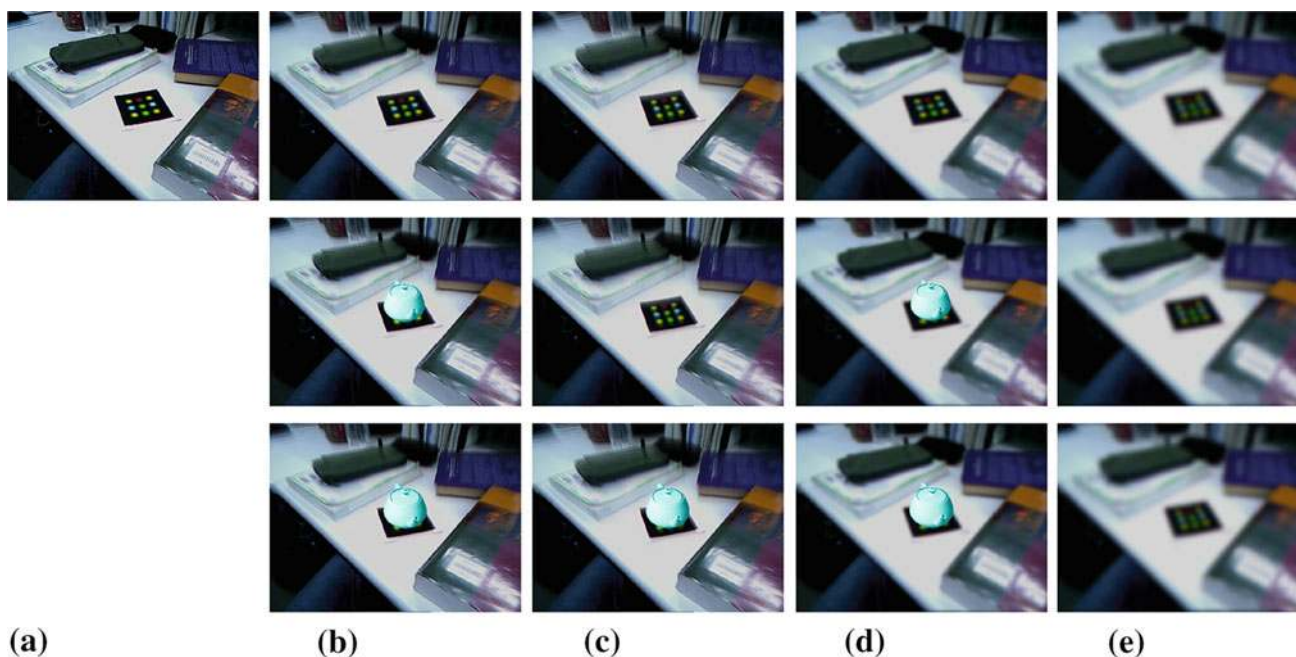
are put on a lattice that is defined by the four corners. The  $H$  values represent the marker pattern. The marker in the image is identified by matching it with stored patterns of markers.

After the identification, the centers of the four corners are refined by parabola fitting [8]. The brightness peaks of parabolas define final positions of the corner dots.

## 4 Experimental results

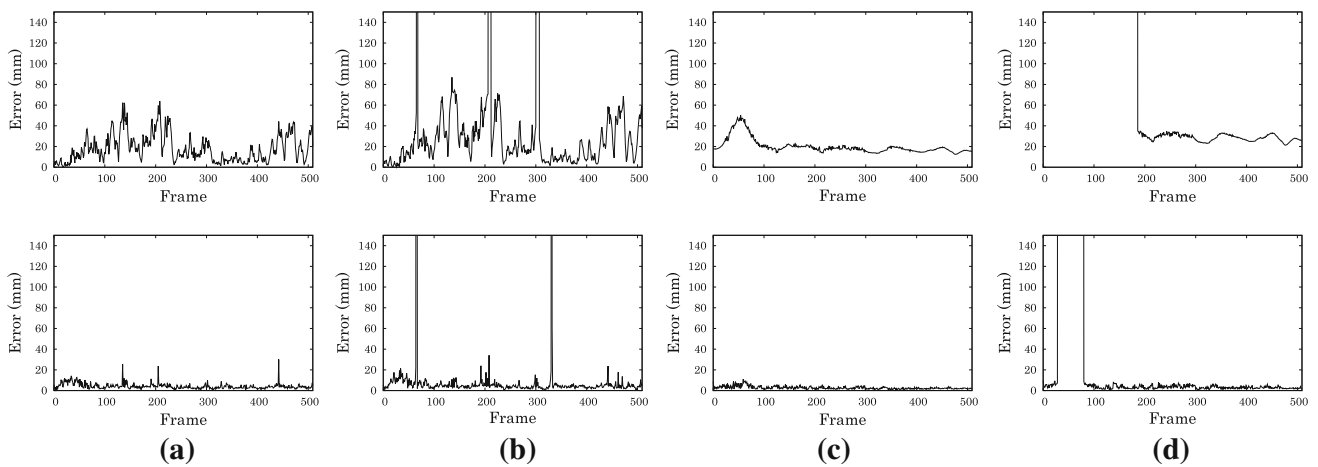
In order to assess the viability and performance of the new mono-spectrum marker, we ran two experiments: one with artificially blurred and defocused images (so we could control the magnitude of the artifacts), and one with real-world blurred and defocused imagery.

A marker was observed with a USB camera (Qcam S7500 of Logicool Corporation) whose resolution was  $640 \times 480$  pixels and frame rate was 15 fps. We used the auto-white-balance function of the camera. The marker was detected using our mono-spectrum method as well as ARToolKit method running on a desktop PC (OS: Windows XP, CPU: Intel Core 2 Duo 2.66 GHz, GPU: NVidia GeForce GTX295 896 MB). Our program was implemented with CUDA 4.0. Two pairs of vertical and horizontal band-pass filters were used for detecting the mono-spectrum marker. One pair of the filters passed a signal with a period of 7–15 pixels, and the other passed a signal with 15–31 pixels. Thus, a  $3 \times 3$

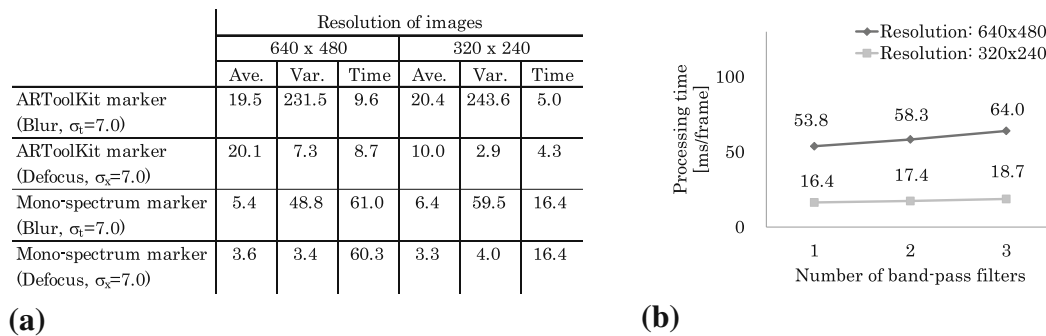


**Fig. 5** Results of marker detection on blurred and defocused images. (*top-left*) Original image of 67th frame. (*top*) Blurred ((**b**) and (**c**)) or defocused ((**d**) and (**e**)) images. (*middle*) Teapots placed on the marker positions detected as an ARToolKit marker. The marker was not detected when images were blurred and defocused, and detected

on farther positions when images were defocused. (*bottom*) Teapots placed on the marker positions detected as a mono-spectrum marker in the blurring image. The marker positions were correctly estimated. The marker could not be detected in severely defocused image. **a** Original image, **b**  $\sigma_t = 4.0$ , **c**  $\sigma_t = 7.0$ , **d**  $\sigma_x = 4.0$ , **e**  $\sigma_x = 7.0$



**Fig. 6** Results of marker detection. (top) Average errors of four corner positions of detected markers as ARToolKit markers. (bottom) Errors as mono-spectrum marker. The  $x$ -axis is the number of frames, and the  $y$ -axis is the error. **a**  $\sigma_t = 4.0$ , **b**  $\sigma_t = 7.0$ , **c**  $\sigma_x = 4.0$ , **d**  $\sigma_x = 7.0$



**Fig. 7** Errors and processing time. **a** Errors and processing times (ms/frame) of marker detection (Not including frames in which markers could not be detected.), **(b)** processing time versus number of band-pass filters

marker should be observed as 21–93 pixels in width and height.

In the first experiment, images  $s(t)$  ( $t = 0, \dots, 510$ ) were captured with a moving camera. The movement of the camera was small so that the images would not be blurred. The focal length of the camera was appropriately set, so the images would not be defocused.

Next, the images were artificially blurred and defocused in the following way. Blurred images  $s_B(x, t)$  were generated by summing the weighted values in  $H(-t)G(0, \sigma_t^2)$  of pixels of the neighboring frames.  $H(-t)$  is the Heaviside function and  $G(0, \sigma_t^2)$  is a Gaussian function with variances  $\sigma_t$  defining the degree of blurring.

The defocused images  $s_D(x, t)$  were generated by summing the weighted values in  $G(0, \sigma_x^2)$  of pixels of the neighboring pixels with variances  $\sigma_x$  defining the degree of defocusing.

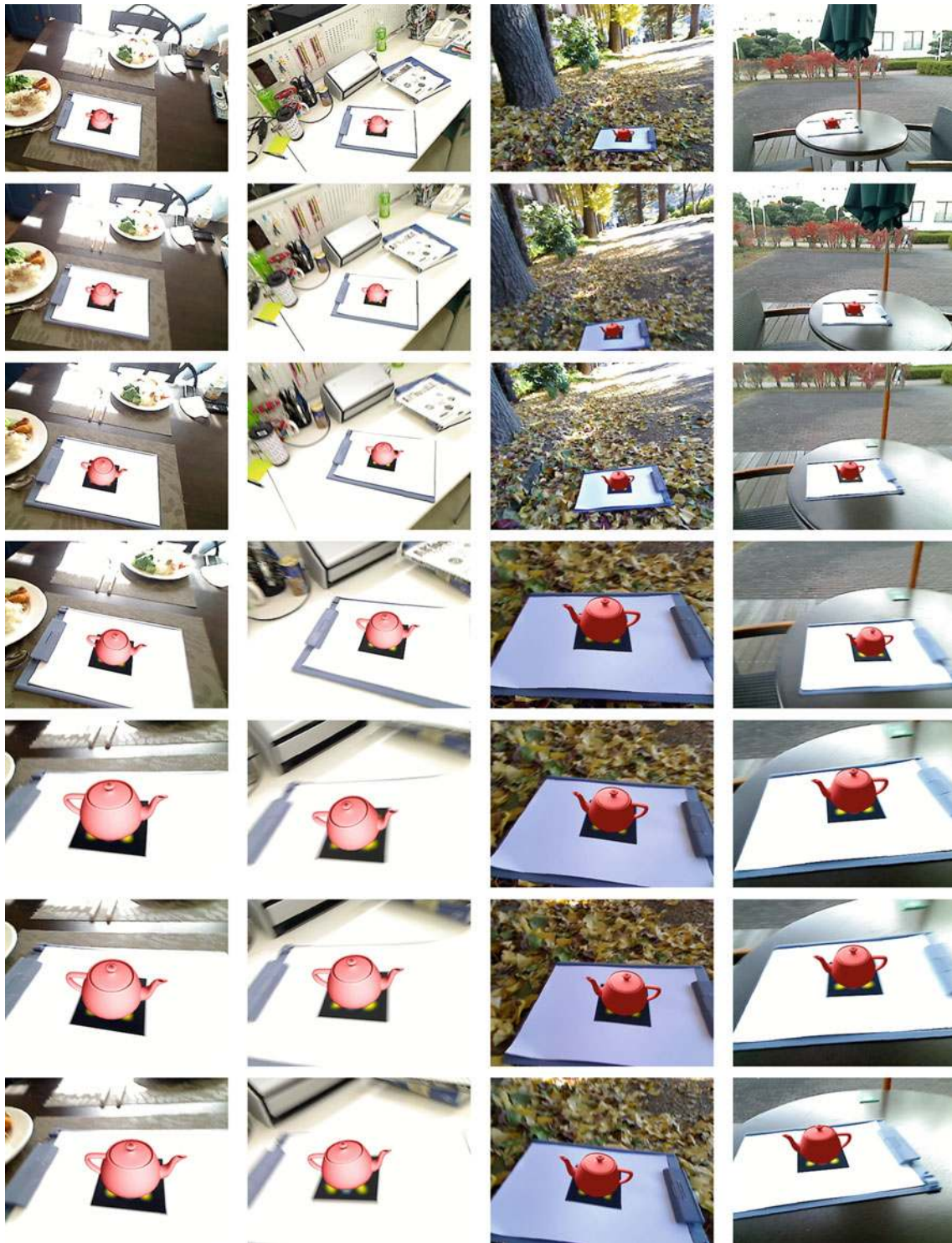
The results of marker detection for the blurred images and the defocused images are shown in Fig. 5. The top images in Fig. 5 are the ones of 67th frame in the movie. Figure 5b and c are examples of blurred images, and Fig. 5d and e

are defocused ones. ARToolKit could not detect markers in blurred images with  $\sigma_t = 7.0$  and defocused images with  $\sigma_x = 7.0$  of Fig. 5. ARToolKit markers tend to be detected on farther positions in defocused images. The position of mono-spectrum markers were estimated more correctly than those of ARToolKit markers.

Figure 6a shows the average error of the extracted four corners from the blurred images when the marker was extracted using ARToolKit. We assumed that the correct position of the corners of the marker was the one estimated with ARToolKit from the original observed images without blurring or defocusing. The error was calculated for the estimated position from the blurred or defocused images. The error is infinity when no marker is detected.

Figure 6a shows that the ARToolKit marker and the mono-spectrum marker could be detected the all frames. Since blurring makes it difficult to detect edges perpendicular to the moving direction of the camera, mono-spectrum marker gave a smaller error, although it was not negligible. When more blurred as shown in Fig. 6b, the markers could not be detected in several frames since the low-frequency components of the





**Fig. 8** Marker detection in various illumination and various scales, indoor and outdoor. The images were captured by Logicool Qcam Pro 9000 with auto white balance, autoexposure and autofocus functions. The teapots are placed at the center of the detected markers. Even though several images are blurred or defocused, the markers could be robustly

detected. Especially, the images in the *third column* include cluttered backgrounds, which consist of the same frequency components with mono-spectrum dots. The mono-spectrum marker could be correctly detected even in the images



mono-spectrum marker were severely attenuated with significant blurring. The average and variance of the error are listed in Fig. 7a.

Figure 6c shows the errors in defocused images. Although the ARToolKit marker could be detected in all frames, the error was large on average. The mono-spectrum marker was extracted in all frames, and the error was smaller than the error of ARToolKit. Image defocusing makes the ARToolKit marker appear smaller and results in a larger error. The error of the mono-spectrum marker is the same level as in the case of image blurring. When more defocused as shown in Fig. 6d, both of markers were not detected in several frames, since the low-frequency components were severely attenuated as well as the case of blurring.

Processing times for our mono-spectrum marker approach are listed in Fig. 7b. For  $640 \times 480$  pixel images, 15 or more frames were processed per second, therefore the operation was almost real time. We used two pairs of vertical and horizontal band-pass filters for the signal with a period of 7–15 pixels and the ones with a period of 15–31 pixels in the above experiments. More filters give a wider depth range of detection, but also use more processing time. Consequently, we examined the change in processing time by varying the number of filters. Since a bandpass filter with a period of 31–63 pixels was employed as the third one, the  $3 \times 3$  marker can be observed as  $189 \times 189$  pixels. Figure 7b shows the relationship between the number of pairs of filters and the processing time. For the image resolution of  $640 \times 480$  pixels, the processing time was 5.1 ms per band-pass filter on average. For  $320 \times 240$  pixel images, the processing time was 1.2 ms per band-pass filter on average.

More experimental results are shown in Fig. 8. Figure 8 shows the marker detection in various illumination scenes including in indoor and outdoor, and various scales. The images are captured by a moving camera. The markers could be robustly detected even in blurred or defocused images. The results shown in Fig. 8 were generated with the parameter setting. The number and supporting range of the filters do not require further tuning in the cases. The additional filter supporting a period of 31–63 pixels enables us to detect a  $3 \times 3$  marker observed as  $189 \times 189$  pixels, although more number of filters takes more time as mentioned above.

## 5 Conclusions

We proposed a mono-spectrum marker that can be accurately extracted even from blurred or defocused images. The marker has only a low-frequency component. Band-pass filters that pass different frequency components are used for extracting the mono-spectrum marker. In an experiment, the marker was accurately extracted from blurred or defocused images in real time.

Processing without GPUs is also a future topic. GPUs were used for band-pass filtering. They can filter in parallel all pixels. ARToolKit has an advantage in processing time and required PC specs. The need to use GPUs would restrict the applications of the mono-spectrum marker. The disadvantage could be overcome with digital signal processors (DSPs). DSPs are used in mobile devices for processing voice signals, and filtering voice signals is a major function. Such a function could be used for extracting the mono-spectrum marker. On the other hand, GPUs are also mounted on into recent mobile devices. They would be useful for extracting mono-spectrum markers.

## References

1. Aruga, H., Toyoura, M., Mao, X.: A trackable ar marker in blurred or defocused images. In: Meeting on Image Recognition and Understanding (MIRU), pp. 496–503 (2011)
2. Asai, H., Oyamada, Y., Pilet, J., Saito, H.: Cepstral analysis based blind deconvolution for motion blur. In: IEEE International Conference on Image Processing (ICIP) (2010)
3. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Surf: speeded up robust features. *Comput. Vis. Image Underst.* **110**(3), 346–359 (2008)
4. Bichlmeier, C., Wimmer, F., Heining, S., Navab, N.: Contextual anatomic mimesis: Hybrid in-situ visualization method for improving multi-sensory depth perception in medical augmented reality. In: International Symposium on Mixed and Augmented Reality (ISMAR), pp. 129–138 (2007)
5. Dixon, J.D., Mortimer, B.: *Permutation Groups*. Springer, Berlin (1996)
6. Fiala, M.: Artag, a fiducial marker system using digital techniques. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 590–596 (2005)
7. Ito, E., Okatani, T., Deguchi, K.: Robust planar target tracking and pose estimation from a single concavity. In: International Symposium on Mixed and Augmented Reality (ISMAR) (2011)
8. Kanatani, K., Sugaya, Y.: Performance evaluation of iterative geometric fitting algorithms. *Comput. Stat. Data Anal.* **52**, 1208–1222 (2007)
9. Kato, H., Billinghurst, M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: International Workshop on Augmented Reality, pp. 85–94 (1999)
10. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: International Symposium on Mixed and Augmented Reality (ISMAR) (2007)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
12. Mohan, A., Woo, G., Hiura, S., Smithwick, Q., Raskar, R.: Bokode: imperceptible visual tags for camera based interaction from a distance. *ACM Trans. Graph.* **28**(3), 98 (2009)
13. Newcombe, R.A., Lovegrove, S., Davison, A.J.: Dtam: dense tracking and mapping in real-time. In: International Conference on Computer Vision (ICCV), pp. 2320–2327 (2011)
14. Okumura, B., Kanbara, M., Yokoya, N.: Augmented reality based on estimation of defocusing and motion blurring from captured images. In: International Symposium on Mixed and Augmented Reality (ISMAR), pp. 219–225 (2006)
15. Owen, C., Xiao, F., Middlin, P.: What is the best fiducial? In: IEEE International Workshop on Augmented Reality Toolkit (2002)

16. Park, Y., Lepetit, V., Woo, W.: Esm-blur: handling and rendering blur in 3d tracking and augmentation. In: International Symposium on Mixed and Augmented Reality (ISMAR), pp. 163–166 (2009)
17. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. *ACM Trans. Graph.* **27**(3), 73 (2008)
18. Tateno, K., Kitahara, I., Ohta, Y.: A nested marker for augmented reality. In: IEEE, Virtual Reality, pp. 259–262 (2007)
19. Uchiyama, H., Saito, H.: Random dot markers. In: IEEE Virtual Reality (IEEE VR) (2011)
20. Uematsu, Y., Saito, H.: Improvement of accuracy for 2d marker-based tracking using particle filter. In: International Conference on Artificial Reality and Telexistence (ICAT), pp. 183–189 (2007)
21. Xu, A., Dudek, G.: Fourier tag: a smoothly degradable fiducial marker system with configurable payload capacity. In: Canadian Conference on Computer and Robot Vision (CRV), pp. 40–47 (2011)



**Masahiro Toyoura** received the B.Sc. degree in Engineering, M.Sc. and Ph.D. degrees in Informatics from Kyoto University in 2003, 2005 and 2008, respectively. He is currently an assistant professor at Interdisciplinary Graduate School of Medical and Engineering, University of Yamanashi, Japan. His research interests are augmented reality, computer and human vision. He is a member of ACM SIGGRAPH and IEEE Computer Society.



**Haruhito Aruga** received the B.Sc. and M.Sc. degrees in Engineering from University of Yamanashi, Japan. His research interests include augmented reality and computer vision.



**Matthew Turk** is a professor of Computer Science and former chair of Media Arts and Technology at UC Santa Barbara, where he co-directs the UCSB Four Eyes Lab. He received a Ph.D. from the Massachusetts Institute of Technology in 1991 and has worked at Martin Marietta Aerospace, LIFIA/ENSIMAG (Grenoble, France), Teles Research, and Microsoft Research. He is on the editorial board of the *ACM Transactions on Intelligent Interactive Systems* and the *Journal of Image and Vision Computing*. He is a Fellow of the IEEE and the recipient of the 2011–2012 Fulbright-Nokia Distinguished Chair in Information and Communications Technologies.



**Xiaoyang Mao** received her B.Sc. in Computer Science from Fudan University, China, M.Sc. and Ph.D. in Computer Science from University of Tokyo. She is currently a professor at Interdisciplinary Graduate School of Medical and Engineering, University of Yamanashi, Japan. Her research interests include texture synthesis, non-photorealistic rendering and their application to scientific visualization. She is a member of ACM SIGGRAPH and IEEE Computer Society.