Detecting Markers in Blurred and Defocused Images

Masahiro Toyoura
University of Yamanashi
Kofu, Yamanashi, Japan
Email: mtoyoura@yamanashi.ac.jp

Haruhito Aruga
University of Yamanashi
Kofu, Yamanashi, Japan
Email: aruga@yamanashi.ac.jp

Matthew Turk
University of California, Santa Barbara
California, USA
Email: mturk@cs.ucsb.edu

Xiaoyang Mao
University of Yamanashi
Kofu, Yamanashi, Japan
Email: mao@yamanashi.ac.jp

***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;**

I. 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 [9], 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 [8] 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, since the high-frequency components are attenuated in blurred or defocused images.

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. For optimal performance under both static and dynamic conditions, the mono-spectrum marker may be combined with current markers. Our approach has a relatively high computational cost and thus requires GPU computing. We believe the rapidly growing availability of GPUs on mobile devices will make this approach to dealing with blurred and defocused images feasible.

II. Related Works

The physical marker of ARToolKit [8] is a popular tool for calibration, estimating the pose of the camera relative to the 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 [5], QR Code, and random dot markers [18] 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 [5] 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 [17] and Bokode tag [11]. The nested marker has a fractal structure so that it looks the same at long

978-1-4799-2245-1/13 $31.00 © 2013 IEEE
DOI 10.1109/CW.2013.58

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 camerafs focal length is set to infinity. These markers do not consider the possible loss of high-frequency components in images.

Natural feature points like SIFT [10] and SURF [3] also enable one to do calibration. The feature points are detected not in planar markers but in general objects. The systems like PTAM [9] or DTAM [12] 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 in order 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 [20] and fiducial image [14] 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 estimated position of a marker in blurred or defocused images was also less accurate than in images free of blurring or defocusing; these problems are addressed in the current paper by introduction of vertical and horizontal filters and combined use of mono-spectrum marker with the ARToolKit marker. The details are described in Section III.

Several template matching methods can deal with the blurring and defocusing problem. ESM-Blur [15] creates blurred templates in advance. Image resampling [6] 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 [19]. The method assumes that the detection failure occurs only in a few frame, but this assumption is not valid when the images are blurred or defocused.

The recovery of image quality is being studied actively [2], [16]. 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.

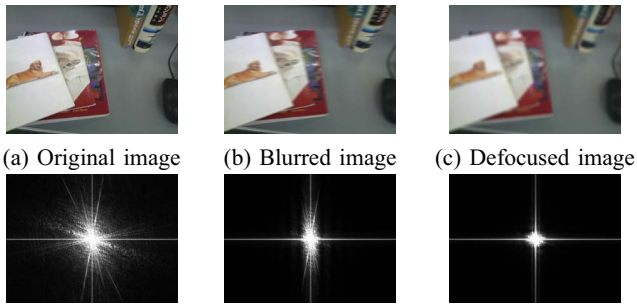
Blurring and defocusing parameters are used for displaying virtual objects on the observed images [13]. The parameters are extracted from the appearance changes of the special pattern marker of ARToolKit under the assumption that the blurring and defocusing are not large. The appearance of the virtual 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.

III. Mono-spectrum Marker Design and Detection

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



(a) Original image (b) Blurred image (c) Defocused image

(d) Spectrum of (a) (e) Spectrum of (b) (f) Spectrum of (c)

Figure 1. Change of frequency spectrum due to blurring and defocusing.

A defocused image is shown in Figure 1(c); its spectrum changes can be seen in Figure 1(f). The spectrum loses high-frequency components in all directions.

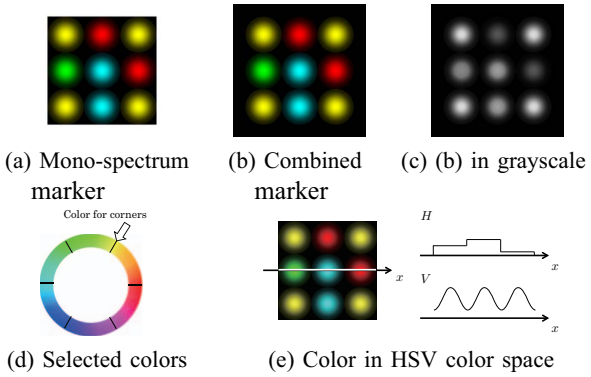
Short exposure time, adequate lighting, stable cameras and objects, and appropriate focal length would solve the above problem; however, these measures are impractical in most dynamic situations, especially in AR applications. Moving cameras often result in blurred images. Moreover the focal length is either fixed or adapts to one of many possibly scene objects. 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.

B. 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 Figure 2(a), a two-dimensional sinusoidal intensity pattern with multiple colors. 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.



(a) Mono-spectrum marker (b) Combined marker (c) (b) in grayscale

(d) Selected colors (e) Color in HSV color space

Figure 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 as shown in Figure 2(d). More number of colors will not be identified 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 as shown in Figure 2(e). 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:

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

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

C. Extraction of Mono-spectrum Marker

Figure 3 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.

Step1. Remove Zero Frequency Component

Let us denote the signal of mono-spectrum markers as $f(x)$ and the ones 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 Figure 3. 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 signals are projected on the images.

$$M: \quad \tilde{f}(x)+c, \quad (2)$$

$$N: \quad \tilde{g}(x)+c. \quad (3)$$

$\tilde{f}$ 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.

$$M: \quad (\tilde{f}(x)+c) * b_0(x) \approx \tilde{f}(x), \quad (4)$$

$$N: \quad (\tilde{g}(x)+c) * b_0(x). \quad (5)$$

Since $f(x)$ does not have a DC component, $\tilde{f}(x)$, which has a transformed single frequency and a little wider range, should not have it as well. Thus, $\tilde{f}(x)$ passes through the

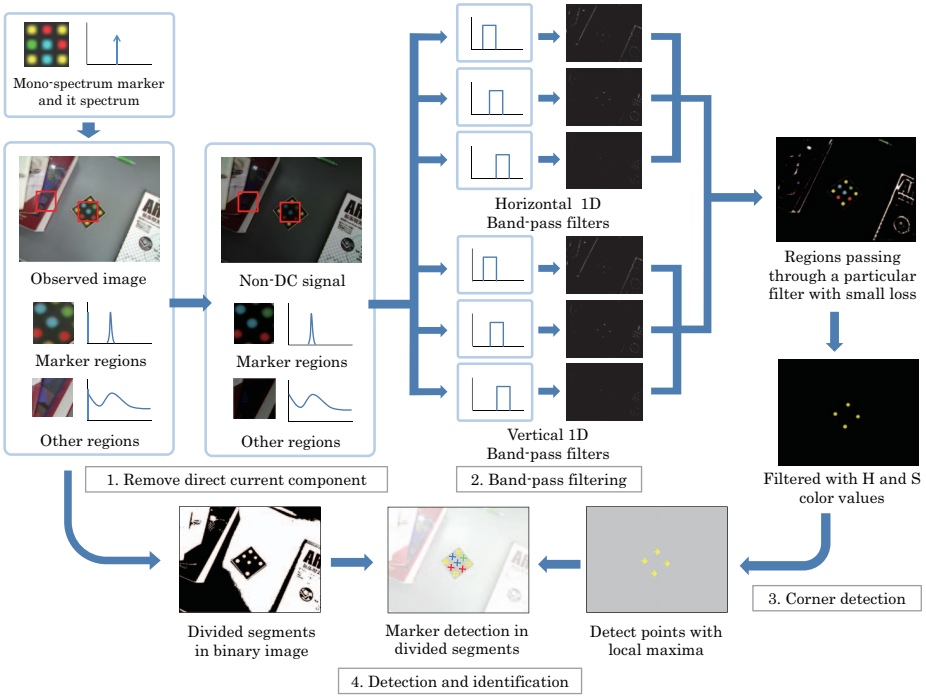


Figure 3. Extraction of mono-spectrum marker from observed images.

filter $b_0(x)$ in M. On the other hand, $g(x)$ would have a DC component, 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: $\quad sum = sum + s(x+xw, y+yw) * b_0(xw, yw).$
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 $(i \in (1,\cdots,n_v) = \mathcal{N}_v)$ and b_j^h $(i \in (1,\cdots,n_h) = \mathcal{N}_h)$ 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.

$$
\begin{aligned}
&R_p = \{(x,y) \ \ | (s*b_0)(x,y) > th_p \},\\
&R_v = \\
&\left\{(x,y) \ \middle| \ \exists i \in \mathcal{N}_v, \left|\left| \frac{(s*b_0*b_i^v)(x,y) - (s*b_0)(x,y)}{(s*b_0)(x,y)} \right|\right| < th_r \right\},\\
&R_h = \\
&\left\{(x,y) \ \middle| \ \exists j \in \mathcal{N}_h, \left|\left| \frac{(s*b_0*b_j^h)(x,y) - (s*b_0)(x,y)}{(s*b_0)(x,y)} \right|\right| < th_r \right\},\\
&R = R_p \cap R_v \cap R_h,
\end{aligned}
\tag{6}
$$

where th_p is the threshold of absolute brightness and th_r is the threshold of relative remaining power. In experiment, th_p was set to 0.05 and th_r was set to 0.20. The thresholds should be appropriately defined depending on the average brightness of captured images.

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

```
candidateRegion = φ.
x = threadid.x, y = threadid.y.
if s0(x,y) > th_p then
  for i ∈ N do
    sum^v = 0.0.
    for xw ∈ filterWindow_i^v do
      sum^v = sum^v + s0(x+xw, y) * b_i^v(xw).
    end for
    if ||sum^v − s0(x,y)||/s0(x,y) < th_r then
      sum^h = 0.0.
      for yw ∈ filterWindow_j do
        sum^h = sum^h + s0(x, y+yw) * b_j^h(yw).
      end for
      if ||sum^h − s0(x,y)||/s0(x,y) < th_r then
        candidateRegion = candidateRegion ∪ (x,y).
      end if
    end if
  end for
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 dots pixels can be also calculated in real-time with the use of 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. 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 Figure 2(a) has 3×3 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 [7]. The brightness peaks of parabolas define final positions of the corner dots.

D. Combined Use with ARToolKit

The position of the marker in images is defined by the pixels having the local peak of V. Since the local peaks are detected from one low-frequency component, the detection is less accurate than the detection from edges or corners with high-frequency components. For instance, the position of an ARToolKit marker is defined by the corner positions. That means ARToolKit's accuracy is superior to our mono-spectrum marker when the images are neither blurred nor defocused. Detecting the peaks from a low-frequency component tends to be affected by image flicker. When both marker and camera are stable, ARToolKit gives better position estimation.

Accordingly, we proceeded to combine our marker with the ARToolKit marker. An example of such a combination marker is shown in Figure 2(b). The ARToolKit marker possesses a black bold frame, and each marker has a unique internal pattern. The marker shown in Figure 2(b) is generated from the one shown in Figure 2(a) with a black bold frame. Detection and identification of markers are done using a binary image. The marker image shown in Figure 2(b) is transformed into the grayscale one shown in Figure 2(c). When the image is binarized, it becomes an ARToolKit marker.

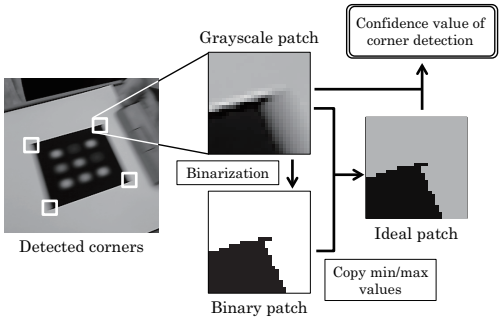


Figure 4. Confidence value of corner detection.

The value indicates the amount of blurring or defocusing. A binarized image is used for corner detection in ARToolKit. When no blurring and defocusing are observed, the corners in the original images should be same to the ideal image in Figure 4. The ideal image is generated by binarised region painted with minmum and maximum color values in the grayscale patch of original image. The confidence value is

estimated by summing errors between the grayscale patch and ideal patch and normalizing the sum.

The threshold of confidence value was set to 0.90 in practical use. If the value is smaller than the threshold, we use the estimated position by ARToolKit, and if not, we use the position of the mono-spectrum marker. Note that we did not apply the confidence value in the experiment in the next section, since we aim to show the advantages of mono-spectrum marker from ARToolKit marker.

IV. 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 640×480 pixels and frame rate was 15 fps. We use the auto-white-balance function of the camera. The marker was detected using a mono-spectrum method as well as ARToolKit method running on a desktop PC (OS: Windows XP, CPU: Intel Core 2 Duo 2.66GHz, GPU: NVidia GeForce GTX295 896MB). 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 to 15 pixels, and the other passed a signal with 15 to 31 pixels. Thus, a 3×3 marker should be observed as 21 to 93 pixels in width and height.

In the first experiment, images $s(t)$ $(t = 0, \cdots, 510)$ 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 σ_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 σ_x defining the degree of defocusing.

The results of marker detection for the blurred images and the defocused images are shown in Figure 5. The top images in Figure 5 are the ones of 90^{th} frame in the movie. Figures 5(b) and 5(c) are examples of blurred images and Figures 5(d) and 5(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 Figure 5. ARToolKit markers tend to be detected on farther positions in defocused images.

Figure 6(a) shows the average error of the extracted four corners from the blurred images in case that 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 6(a) 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, ARToolKit gave a larger error. The error of the mono-spectrum marker was not negligible, although it was smaller than that of ARToolKit. The combined use of the mono-spectrum marker and the ARToolKit marker can accurately extract the marker. When more blurred as shown in Figure 6(b), the markers could not be detected in several frames since the low-frequency components of the mono-spectrum marker were severely attenuated with significant blurring. The average and variance of the error are listed in Figure 7(a). Note that the frames of ARToolKit were not used in this case.

Figure 6(c) 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 Figure 6(d), the ARToolKit marker was not detected in several frames, since the low-frequency components were severely attenuated as well as the case of blurring. The optimal solution is to combine mono-spectrum markers and ARToolKit markers, as they have complementary strengths.

Processing times are listed in Figure 7(b). For 640×480 pixel images, 15 or more frames were processed per second, so 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 to 15 pixels and the ones with a period of 15 to 31 pixels in the above experiments. More filters gives a wider depth range of detection, but also uses 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 to 63 pixels was employed as the third one, the 3×3 marker can be observed as 189×189 pixels. Figure 7(b) shows the relationship between the number of pairs of filters and the processing time. For the image resolution of 640×480 pixels, the processing time was 5.3 ms per band-pass filter on average. For 320×240 pixel images, the processing time was 1.5 ms per band-pass filter on average.

More experimental results are shown in the supplemental movie. The movie shows the marker detection in various

illumination scenes and various scales. The images are captured by a moving camera. The markers could be robustly detected even in blurred or defocused images.

We also confirmed the validity of combined use with ARToolKit in the movie. We captured real blurred and defocused images and evaluated the detection rate as an ARToolKit marker and mono-spectrum marker. In the blurred images, ARToolKit marker was missed in 22 images out of 500, and mono-spectrum marker was correctly detected in all images. In the defocused images, both of ARToolKit marker and mono-spectrum marker were detected in all images, although the position of the ARToolKit marker was farther than the real position.

V. 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] H. Aruga, M. Toyoura, and X. Mao. A trackable ar marker in blurred or defocused images. In *Meeting on Image Recognition and Understanding (MIRU)*, pages 496–503, 2011. (In Japanese)

[2] H. Asai, Y. Oyamada, J. Pilet, and H. Saito. Cepstral analysis based blind deconvolution for motion blur. In *IEEE International Conference on Image Processing (ICIP)*, 2010.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[4] C. Bichlmeier, F. Wimmer, S. Heining, and N. Navab. 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)*, pages 129–138, 2007.

[5] M. Fiala. Artag, a fiducial marker system using digital techniques. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 590–596, 2005.

[6] E. Ito, T. Okatani, and K. Deguchi. Robust planar target tracking and pose estimation from a single concavity. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[7] K. Kanatani and Y. Sugaya. Performance evaluation of iterative geometric fitting algorithms. *Computational Statistics and Data Analysis*, 52:1208–1222, 2007. Parabola Fitting.

[8] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *International Workshop on Augmented Reality*, pages 85–94, 1999.

[9] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

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

[11] A. Mohan, G. Woo, S. Hiura, Q. Smithwick, and R. Raskar. Bokode: Imperceptible visual tags for camera based interaction from a distance. *ACM Transactions on Graphics*, 28(3), 2009. Article 98.

[12] R. A. Newcombe, S. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *International Conference on Computer Vision (ICCV)*, pages 2320–2327, 2011.

[13] B. Okumura, M. Kanbara, and N. Yokoya. Augmented reality based on estimation of defocusing and motion blurring from captured images. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 219–225, 2006.

[14] C. Owen, F. Xiao, and P. Middlin. What is the best fiducial? In *IEEE International Workshop on Augmented Reality Toolkit*, 2002.

[15] Y. Park, V. Lepetit, and W. Woo. Esm-blur: Handling and rendering blur in 3d tracking and augmentation. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 163–166, 2009.

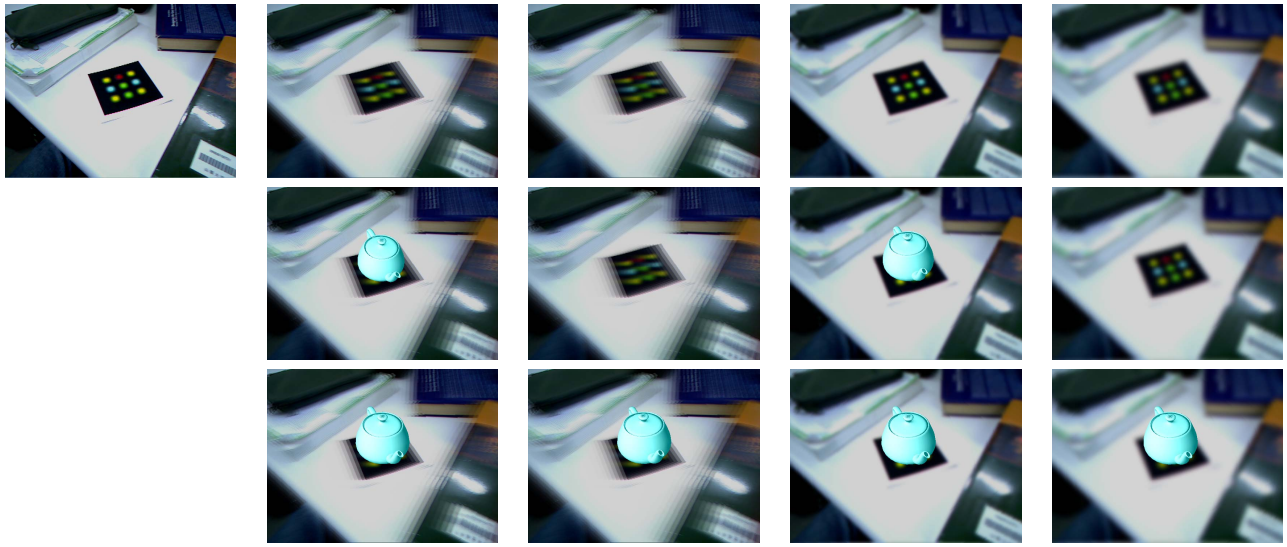
[16] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(3), 2008.

[17] K. Tateno, I. Kitahara, and Y. Ohta. A nested marker for augmented reality. In *IEEE Virtual Reality*, pages 259–262, 2007.

[18] H. Uchiyama and H. Saito. Random dot markers. In *IEEE Virtual Reality (IEEE VR)*, 2011.

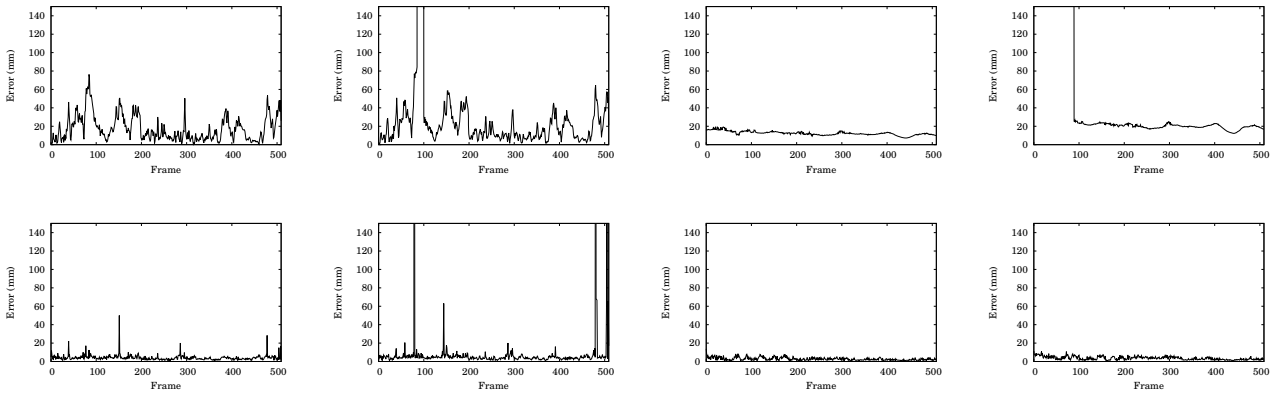
[19] Y. Uematsu and H. Saito. Improvement of accuracy for 2d marker-based tracking using particle filter. In *International Conference on Artificial Reality and Telexistence (ICAT)*, pages 183–189, 2007.

[20] A. Xu and G. Dudek. Fourier tag: A smoothly degradable fiducial marker system with configurable payload capacity. In *Canadian Conference on Computer and Robot Vision (CRV)*, pages 40–47, 2011.



(a) Original image (b) $\sigma_t = 4.0$ (c) $\sigma_t = 7.0$ (d) $\sigma_x = 4.0$ (e) $\sigma_x = 7.0$

Figure 5. Results of marker detection on blurred and defocused images. (Top-left) Original image of 90^{th} 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. The marker positions were correctly estimated.

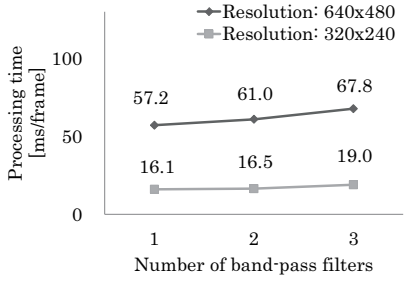


(a) $\sigma_t = 4.0$ (b) $\sigma_t = 7.0$ (c) $\sigma_x = 4.0$ (d) $\sigma_x = 7.0$

Figure 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.

	Resolution of images					
	640 x 480			320 x 240		
	Ave.	Var.	Time	Ave.	Var.	Time
ARToolKit marker (Blur, σ_t=4.0)	24.3	356.2	9.2	25.6	364.7	5.0
ARToolKit marker (Defocus, σ_t=7.0)	28.6	10.4	7.9	30.6	113.7	3.8
Mono-spectrum marker (Blur, σ_t=4.0)	5.3	19.8	58.3	7.1	76.8	17.1
Mono-spectrum marker (Defocus, σ_t=7.0)	3.4	2.7	56.8	3.9	3.9	16.2

(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.

Figure 7. Errors and processing time.