A High-Accuracy Rotation Estimation Algorithm Based on 1D Phase-Only Correlation

Sei Nagashima[1], Koichi Ito[1], Takafumi Aoki[1], Hideaki Ishii[2], and Koji Kobayashi[2]

[1] Graduate School of Information Sciences, Tohoku University, Sendai-shi, 980–8579 Japan
{nagasima, ito}@aoki.ecei.tohoku.ac.jp
[2] Yamatake Corporation, Fujisawa-shi, 251–8522 Japan

Abstract. This paper proposes a high-accuracy rotation estimation algorithm using 1D Phase-Only Correlation (POC). In general, the rotation angle between two images is estimated as follows: (i) convert the image rotation into the image shift by polar mappings of the amplitude spectra of images, and (ii) estimate the translational displacement between the polar mappings to obtain the rotation angle. The problem of rotation estimation between two images is replaced to 1D displacement estimation between pairs of horizontal lines at the same vertical position in two polar mappings. The proposed algorithm employs 1D POC instead of 2D matching for estimating a rotation angle. The use of 1D POC to estimate the rotation angle makes it possible to reduce the computational cost significantly without sacrificing the estimation accuracy.

1 Introduction

High-accuracy image matching is an important fundamental task in many fields, such as image sensing, image/video processing, computer vision, industrial vision, etc. Over the years, various techniques for image matching have been developed. Typical examples include methods using image correlation functions, Fourier-transform-based methods, image-feature-based methods, and others [1,2]. Among many methods, image matching techniques using Phase-Only Correlation (POC) (or simply a "phase correlation") have attracted much attention due to their high-accuracy and robust performance [3].

In our previous paper [4], we have proposed a set of techniques that can significantly enhance the accuracy of POC-based image matching. Using the proposed techniques, we can estimate image translation with 1/100-pixel accuracy, image rotation with 1/40-degree accuracy and image scaling with 1/10000-scale accuracy [4]. Recently, this method is extended to high-accuracy block matching for finding corresponding point pairs between two images with sub-pixel accuracy [5], which is successfully applied to the passive 3D measurement using stereo vision [6] and the video signal processing [7].

This paper proposes a high-accuracy rotation estimation algorithm using one-dimensional (1D) POC. In general, the rotation angle between two images is

M. Kamel and A. Campilho (Eds.): ICIAR 2007, LNCS 4633, pp. 210–221, 2007.

estimated as follows: (i) convert the image rotation into the image shift by polar mappings of the amplitude spectra of images, and (ii) estimate the translational displacement between the polar mappings to obtain the rotation angle [8]. The image rotation between two images is represented as the horizontal displacement between two polar mappings. Hence, the problem of rotation estimation is replaced to 1D displacement estimation between row lines in two polar mappings. In the proposed algorithm, we calculate the 1D POC functions between every pair of row lines in two polar mappings, and obtain the rotation angle between images by averaging all 1D POC functions.

This paper also proposes an effective line extraction algorithm to improve the estimation accuracy. The conventional methods using 2D image matching use all frequency components in images to estimate a rotation angle. The amplitude spectra, however, include meaningless frequency components for image rotation. In order to extract effective lines for estimating a rotation angle, we evaluate the noise immunity for row lines in a polar mapping. If a line has noise immunity, this line is extracted as an effective line for rotation estimation. Through a set of experiments, we demonstrate that the use of 1D POC to estimate the rotation angle makes it possible to reduce computational cost significantly without sacrificing estimation accuracy.

2 Basic Principle of Image Rotation Estimation

This section introduces the basic principle of the rotation estimation algorithm using 2D POC, which is proposed in [4,8]. The problem considered here is to estimate the rotation angle θ between two images that are translated and rotated each other. For estimating the image rotation, we employ the polar mapping of the amplitude spectrum to convert the image rotation into the image translation. Note that the amplitude spectra are not affected by the image shifts, and are rotated with respect to each other at the origin of the spatial frequencies by the same angle as their spatial domain counterparts [4,8]. The use of polar mapping[1], we can easily estimate the rotation angle θ.

In the following, we summarize the procedure for estimating the rotation angle θ (see [4] for detailed discussions).

Input

images $f(n_1, n_2)$ and $g(n_1, n_2)$

Output

relative rotation angle θ between $f(n_1, n_2)$ and $g(n_1, n_2)$

Step 1

Calculate 2D DFTs of the discrete images $f(n_1, n_2)$ and $g(n_1, n_2)$ to obtain $F(k_1, k_2)$ and $G(k_1, k_2)$, where we assume that the index ranges are $n_1 = -M, \cdots, M$, $n_2 = -M, \cdots, M$, $k_1 = -M, \cdots, M$ and $k_2 = -M, \cdots, M$. The image size is $N \times N$ ($N = 2M + 1$). In order to reduce the effect of discontinuity at

[1] In this paper, we call the polar mapping of the amplitude spectrum as polar mapping.

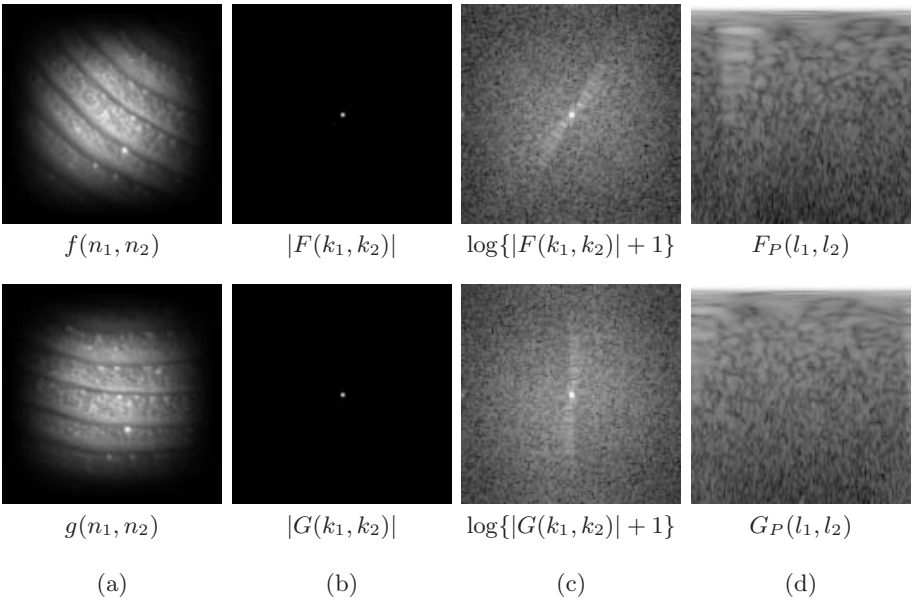

Fig. 1. Example of rotation estimation: (a) two input images, (b) amplitude spectra of the images, (c) enhanced amplitude spectra, and (d) polar mappings of the enhanced amplitude spectra

images border, we also apply 2D Hanning window to the input images $f(n_1, n_2)$ and $g(n_1, n_2)$ as shown in Fig. 1(a).

Step 2

Calculate the amplitude spectra $|F(k_1, k_2)|$ and $|G(k_1, k_2)|$. For natural images, most energy is concentrated in low-frequency domain. Hence, we had better to use $\log\{|F(k_1, k_2)|+1\}$ and $\log\{|G(k_1, k_2)|+1\}$ (Fig. 1(c)) instead of $|F(k_1, k_2)|$ and $|G(k_1, k_2)|$ (Fig. 1(b)).

Step 3

Calculate the polar mappings $F_P(l_1, l_2)$ and $G_P(l_1, l_2)$ of $\log\{|F(k_1, k_2)| + 1\}$ and $\log\{|G(k_1, k_2)| + 1\}$ (Fig. 1 (d)), where the index ranges of the transformed image are $l_1 = -M, \cdots, M$ and $l_2 = -M, \cdots, M$.

Step 4

Estimate the image displacement between $F_P(l_1, l_2)$ and $G_P(l_1, l_2)$ using the 2D POC function to obtain the rotation angle θ, where the displacement in the horizontal direction (l_2 direction) corresponds to the image rotation.

3 Rotation Estimation Algorithm Using 1D Phase-Only Correlation

This section presents a rotation estimation algorithm using a 1D POC function. The image rotation is represented as horizontal displacements between two polar

mappings. The problem of rotation estimation is replaced to 1D displacement estimation between row lines in two polar mappings. The use of 1D POC function makes it possible to reduce the computational cost significantly for estimating the rotation angle. In this section, we first define the 1D POC function and then introduce the details of the proposed rotation estimation algorithm.

3.1 1-Dimensional Phase-Only Correlation (1D POC)

Consider two 1D image signals, $f(n)$ and $g(n)$, where we assume that the index range is $n = -M, \cdots, M$ for mathematical simplicity, and hence the signal length is $N = 2M+1$. The discussion could be easily generalized to non-negative index ranges with power-of-two signal length. Let $F(k)$ and $G(k)$ denote the Discrete Fourier Transforms (DFTs) of the two signals. $F(k)$ and $G(k)$ are given by

$$F(k) = \sum_{n=-M}^{M} f(n) W_N^{kn} = A_F(k) e^{j\theta_F(k)}, \tag{1}$$

$$G(k) = \sum_{n=-M}^{M} g(n) W_N^{kn} = A_G(k) e^{j\theta_G(k)}, \tag{2}$$

where $W_N = e^{-j\frac{2\pi}{N}}$, $A_F(k)$ and $A_G(k)$ are amplitude components, and $e^{j\theta_F(k)}$ and $e^{j\theta_G(k)}$ are phase components.

The cross-phase spectrum (or normalized cross spectrum) $R(k)$ is defined as

$$R(k) = \frac{F(k)\overline{G(k)}}{\left|F(k)\overline{G(k)}\right|} = e^{j\theta(k)}, \tag{3}$$

where $\overline{G(k)}$ denotes the complex conjugate of $G(k)$ and $\theta(k) = \theta_F(k) - \theta_G(k)$. The POC function $r(n)$ is the Inverse Discrete Fourier Transform (IDFT) of $R(k)$ and is given by

$$r(n) = \frac{1}{N} \sum_{k=-M}^{M} R(k) W_N^{-kn}. \tag{4}$$

If there is a similarity between two signals, the POC function gives a distinct sharp peak. (When $f(n) = g(n)$, the POC function becomes the Kronecker delta function.) If not, the peak drops significantly. The height of the peak can be used as a good similarity measure for signal matching, and the location of the peak shows the translational displacement between the two signals. Figure 2 shows an example of the POC function, where (a) and (b) are an example signal and a displaced version of the signal, respectively. Figure 2 (c) is the corresponding POC function.

Now consider $f_c(t)$ as a signal defined in continuous space with a real number index t. Let δ represent a displacement of $f_c(t)$. So, the displaced signal can be

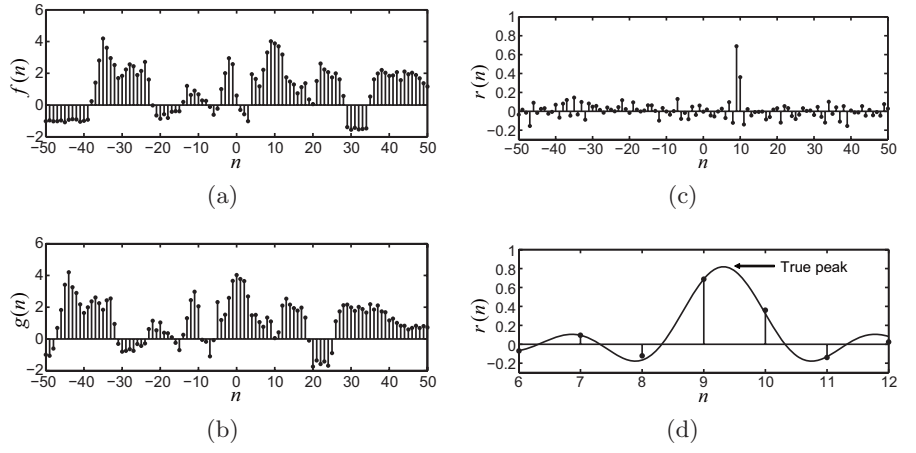

Fig. 2. Displacement estimation using 1D POC function: (a) original signal $f(n)$, (b) displaced version of the original signal, $g(n)$, (c) 1D POC function $r(n)$, and (d) function fitting for estimating peak position with sub-pixel accuracy

represented as $f_c(t-\delta)$. Assume that $f(n)$ and $g(n)$ are spatially sampled signals of $f_c(t)$ and $f_c(t-\delta)$, and are defined as

$$f(n) = f_c(t)|_{t=nT}, \tag{5}$$

$$g(n) = f_c(t-\delta)|_{t=nT}, \tag{6}$$

where T is the sampling interval and the index range is given by $n = -M, \cdots, M$. For simplicity, we assume $T = 1$. The cross-phase spectrum $R(k)$ and the POC function $r(n)$ between $f(n)$ and $g(n)$ will be given by

$$R(k) = \frac{F(k)\overline{G(k)}}{\left|F(k)\overline{G(k)}\right|} \simeq e^{j\frac{2\pi}{N}k\delta}, \tag{7}$$

$$\begin{aligned} r(n) &= \frac{1}{N}\sum_{k=-M}^{M} R(k)W_N^{-kn} \\ &\simeq \frac{\alpha}{N}\frac{\sin\{\pi(n+\delta)\}}{\sin\left\{\frac{\pi}{N}(n+\delta)\right\}}, \end{aligned} \tag{8}$$

where $\alpha = 1$. The above Eq. (8) represents the shape of the peak for the POC function between the same signals that are slightly displaced with each other. This equation gives a distinct sharp peak. The peak position δ of the POC function corresponds to the displacement between the two signals. We can prove that the peak value α decreases (without changing the shape of the function itself), when small noise components are added to the original signals. Hence, we assume $\alpha \leq 1$ in practice. For the signal matching task, we evaluate the similarity between the two signals by the peak value α, and estimate the displacement by the peak position δ.

Listed below are important techniques for improving the accuracy of 1D image matching.

(i) Function fitting for high-accuracy estimation of peak location
We use Eq. (8) — the closed-form peak model of the POC function — directly for estimating the peak location by function fitting. By calculating the POC function, we can obtain a data array of $r(n)$ for each discrete index n. Figure 2 (d) shows the POC function around the correlation peak, where the black dots indicate the discrete data values from $r(n)$. The solid line in Fig. 2 (d) represents the estimated shape of the POC function. Thus, it is possible to find the location of the peak that may exist between image pixels by fitting the function Eq. (8) to the calculated data array around the correlation peak, where α and δ are fitting parameters.
(ii) Spectral weighting technique to reduce aliasing and noise effects
For natural images, the high frequency components typically may have less reliability (low S/N ratio) compared with the low frequency components. We could improve the estimation accuracy by applying a low-pass-type weighting function $H(k)$ to $R(k)$ in the frequency domain and eliminating the high frequency components with low reliability. The simplest weighting function is defined as

$$H(k) = \begin{cases} 1 & |k| \leq U \\ 0 & \text{otherwise} \end{cases}, \tag{9}$$

where U is integer satisfying $0 < U \leq M$. The cross-phase spectrum $R(k)$ is multiplied by the weighting function $H(k)$ when calculating the 1D IDFT. Then the modified $r(n)$ will be given by

$$r(n) \simeq \frac{\alpha}{N} \cdot \frac{\sin\{\frac{V}{N}\pi(n+\delta)\}}{\sin\{\frac{\pi}{N}(n+\delta)\}}, \tag{10}$$

where $V = 2U + 1$. In our experiment, we choose $U = \lfloor M/2 \rfloor$. When using the spectral weighting technique, Eq. (10) should be used for function fitting instead of Eq. (8).

3.2 High-Accuracy Image Rotation Estimation Algorithm

To estimate the rotation angle θ, we calculate 1D POC functions for every pairs of row lines in two polar mappings, and then summarize all 1D POC functions to obtain θ. In the calculation of 1D POC functions, we need to use only the effective lines (effective frequency bands), since the amplitude spectra include meaningless frequency components for image rotation. The proposed algorithm consists of two steps: (i) effective line extraction and (ii) rotation estimation using 1D POC. The followings are detailed procedures for the proposed algorithm.

(i) Effective line extraction
This step is to extract lines (frequency bands) of the registered image $f(n_1, n_2)$ which are effective to estimate the rotation angle. We first rotate $f(n_1, n_2)$ to

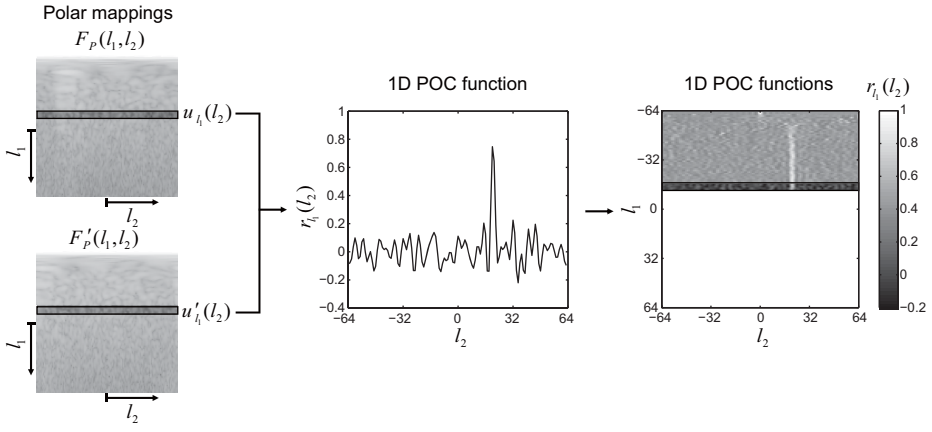

Fig. 3. Example of calculating the 1D POC function between a pair of row lines in two polar mappings

have $f'(n_1, n_2)$ and then estimate the rotation angle between $f(n_1, n_2)$ and $f'(n_1, n_2)$ using 1D POC function. The 1D POC functions are calculated for each pairs of row lines in two polar mappings generated from $f(n_1, n_2)$ and $f'(n_1, n_2)$. If the peak location of 1D POC function shows the correct rotation angle, the line is extracted as the effective line for estimating the rotation angle. The above procedure is equivalent to measure the noise immunity for row lines in the polar mapping, where the noise is caused by image rotation. Note that different types of noise (e.g., white noise, etc.) can be also used to measure the noise immunity.

The following is the detailed procedure for this step.

Input

registered image $f(n_1, n_2)$

Output

indices $\boldsymbol{l}$ of effective lines

Step 1

Generate the rotated image $f'(n_1, n_2)$ by rotating the registered image $f(n_1, n_2)$ by Θ degree, where $\Theta = 30$ in this paper.

Step 2

Calculate the polar mappings $F_P(l_1, l_2)$ and $F'_P(l_1, l_2)$ of $f(n_1, n_2)$ and $f'(n_1, n_2)$, respectively.

Step 3

Extract 1D image signals $u_{l_1}(l_2)$ and $u'_{l_1}(l_2)$ in the horizontal direction (l_2 direction) from $F_P(l_1, l_2)$ and $F'_P(l_1, l_2)$, respectively, as shown in Fig. 3. Next, we calculate the 1D POC functions $r_{l_1}(l_2)$ between $u_{l_1}(l_2)$ and $u'_{l_1}(l_2)$, and obtain the displacement δ_{l_1} and the correlation peak value α_{l_1}. The number of 1D POC functions is N.

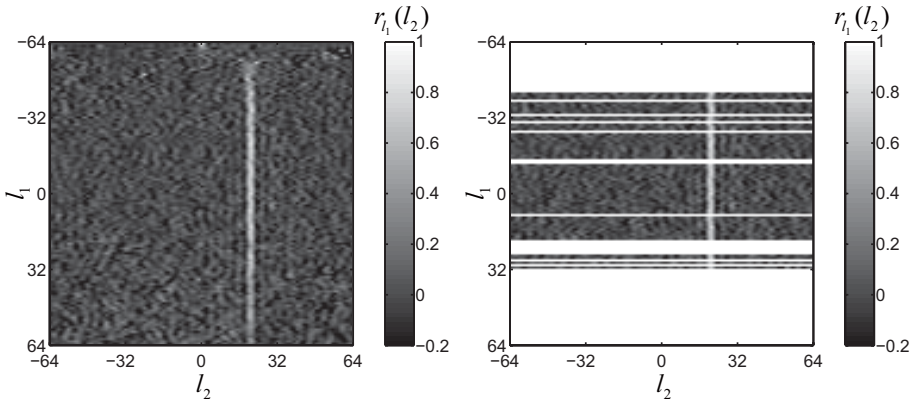

Fig. 4. Examples of 1D POC functions for all row lines (a) and only for effective row lines (b) (the correct displacement is $l_2 = 20$)

Step 4

Select indices $\boldsymbol{l}$ by comparing between δ_{l_1} and Θ as follows:

$$\boldsymbol{l} = \{l_1 : \ |\delta_{l_1} - N\Theta/\pi| < \delta_{th}, -M \leq l_1 \leq M\}, \tag{11}$$

where $\delta_{th} = 1$ in this paper. Note that $N\Theta/\pi$ indicates the displacement between two polar mappings which is equivalent to the rotation angle Θ.

Step 5

Update the indeces $\boldsymbol{l}$ by extracting upper half of the correlation peak value α_i.

Figures 4 (a) and (b) show examples of 1D POC functions for all row lines and only for effective row lines, respectively. In this example, the correct displacement, i.e., the correct rotation angle, is $l_2 = 20$. As is observed in these figures, all of 1D POC functions in Fig. 4 (b) indicate the correct displacement, while some of 1D POC functions in Fig. 4 (a) indicate the wrong displacements.

(ii) Rotation estimation using 1D POC function

This procedure is to estimate the rotation angle using the effective lines obtained from (i).

Input

registered image $f(n_1, n_2)$, input image $g(n_1, n_2)$ and indices $\boldsymbol{l}$ of effective lines

Output

rotation angle θ

Step 1

Calculate the polar mappings $F_P(l_1, l_2)$ and $G_P(l_1, l_2)$ of $f(n_1, n_2)$ and $g(n_1, n_2)$, respectively.

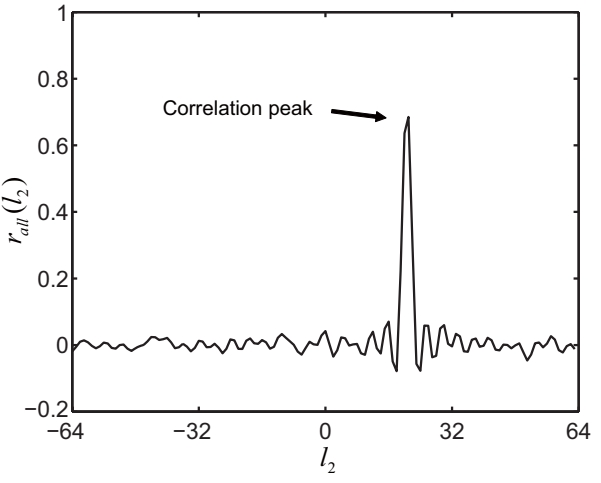

Fig. 5. Example of the averaged 1D POC function $r_{all}(l_2)$

Step 2

Extract 1D image signals $u_{l_1}(l_2)$ and $v_{l_1}(l_2)$ in the horizontal direction (l_2 direction) from $F_P(l_1, l_2)$ and $G_P(l_1, l_2)$, respectively, where $l_1 \in \boldsymbol{l}$. Next, we calculate the 1D POC functions $r_{l_1}(l_2)$ between $u_{l_1}(l_2)$ and $v_{l_1}(l_2)$, where the number of 1D POC functions equals to the number of effective lines.

Step 3

Calculate the average of all 1D POC functions $r_{l_1}(l_2)$, which denotes $r_{all}(l_2)$ (Fig. 5). Next, we obtain the displacement δ from the peak location of the averaged 1D POC function $r_{all}(l_2)$. Then, we calculate the rotation angle θ as $\theta = \delta\pi/N$.

4 Experiments and Discussions

This section describes a set of experiments for evaluating estimation performance of the proposed algorithm. In this experiment, we have estimated the rotation angle between two images taken by a CCD camera (JAI CVM10 with Sony VCL-16WM lens). The target object is a wood cube with the size of 10 cm × 10 cm × 10 cm, which is mounted on a micro stage that allows precise rotation as shown in Fig. 6 (a). The distance from the camera to the cube is 70 cm, and the size of the cube in the captured images is about 350 × 350 pixels. We use sub-images with a wooden texture extracted from the center of camera images. These sub-images contain only the texture of the wood cube and do not contain background textures as shown in Fig. 6 (b). We have rotated the micro stage from 0 to 90 degrees with each step of 1 degree, and took time-averaged images (30 frames at each position). The total number of the images used in this experiment is 91.

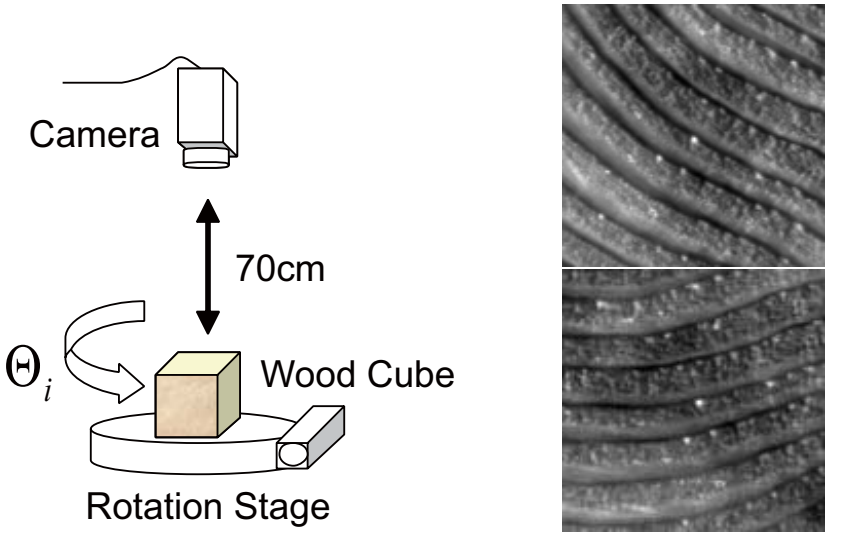

Fig. 6. Experimental system setup (a) and examples of captured image (b)

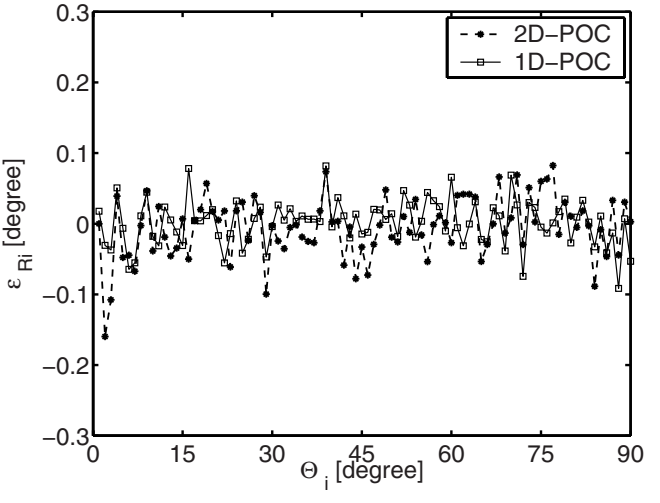

Fig. 7. Errors in rotation estimation for the proposed algorithm and the conventional algorithm, where the image size is 128×128 pixels

We compare two algorithms: (A) 2D POC-based rotation estimation algorithm and (B) the proposed algorithm (1D POC-based rotation estimation algorithm). The estimation error for the rotation angle is evaluated by

$$\epsilon_{Ri} = \theta_i - \Theta_i,$$

Table 1. Errors in rotation estimation for the proposed algorithm and the conventional algorithm, when changing the image size into 64×64, 128×128 and 256×256 pixels

	2D-POC		1D-POC	
Image size [pixel]	RMS [degree]	Max. [degree]	RMS [degree]	Max. [degree]
64×64	0.1637	0.4926	0.1433	0.3168
128×128	0.0433	0.1597	0.0324	0.0915
256×256	0.0286	0.0803	0.0241	0.0598

Table 2. Computational cost for estimating a rotation angle between two images

	Polar mapping generation	Displacement estimation	Total
2D-POC	$2N$	$4N$	$6N$
1D-POC	$2N$	$N/2 + 1$	$2.5N + 1$

where θ_i [degree] and Θ_i [degree] are the estimated angle and the actual angle of the rotation stage for the i-th micro step, respectively.

Figure 7 shows the errors in rotation estimation, where "2D-POC" corresponds to 2D POC-based rotation estimation and "1D-POC" corresponds to 1D POC-based rotation estimation, and the image size is 128×128 pixels. Table 1 shows the RMS and maximum errors in rotation estimation for both algorithms when changing the image size into 64×64, 128×128 and 256×256 pixels, respectively. Here, the RMS error represents the *Root Mean Square* error. As is observed in the above experiments, the proposed algorithm exhibits efficient performance comparable with that of the conventional 2D POC-based algorithm.

We evaluate the amount of computation required for estimating a rotation angle. In this experiment, we use only one registered image so that we do not take account of computation time spent in the effective line extraction described in Sect. 3.2 (i). We evaluate the amount of computation required only for calculating DFT, since 90% of the whole computation time is spent in the calculation of DFT when estimating a rotation angle between two images. We assume that the image size is $N \times N$ pixels and the row-column decomposition DFT is used for calculating DFTs of images. Thus, a 2D DFT of $N \times N$-pixel image is computed by $2N$ 1D N-point DFTs. Let the number of lines extracted in the proposed algorithm be $N/2$, since the computational cost of the proposed algorithm is depending on the number of lines.

Table 2 summarizes the computational cost for estimating the rotation angle between two images. By using 1D POC function, significant reduction in computational cost is expected in comparison with the conventional 2D POC-based algorithm, where the computational cost can be reduced to about 50% of the conventional algorithm. As is observed in the above experiments, the use of 1D POC for estimating a rotation angle makes it possible to improve the estimation accuracy as well as to reduce the computational cost.

5 Conclusion

This paper has presented a high-accuracy rotation estimation algorithm using 1D POC. The use of 1D POC in rotation estimation makes it possible to significantly reduce the computational cost without sacrificing the estimation accuracy compared with the conventional 2D POC-based algorithm. We will expect that the proposed algorithm is useful for blurred images, i.e., low-S/N images, since the algorithm can select the effective lines (effective frequency bands) automatically.

References

1. Brown, L.G.: A survey of image registration techniques. ACM Computing Surveys 24, 325–376 (1992)
2. Zitova, B., Flusser, J.: Image registration methods: a survey. Image and Vision Computing 21, 977–1000 (2003)
3. Kuglin, C.D., Hines, D.C.: The phase correlation image alignment method. In: Proc. Int. Conf. Cybernetics and Society, pp. 163–165 (1975)
4. Takita, K., Aoki, T., Sasaki, Y., Higuchi, T., Kobayashi, K.: High-accuracy subpixel image registration based on phase-only correlation. IEICE Trans. Fundamentals E86-A, 1925–1934 (2003)
5. Takita, K., Muquit, M.A., Aoki, T., Higuchi, T.: A sub-pixel correspondence search technique for computer vision applications. IEICE Trans. Fundamentals E87-A, 1913–1923 (2004)
6. Muquit, M.A., Shibahara, T., Aoki, T.: A high-accuracy passive 3D measurement system using phase-based image matching. IEICE Trans. Fundamentals E89-A, 686–697 (2006)
7. Loy, H.C., Aoki, T.: Robust motion estimation for video sequences based on phase-only correlation. In: Proc. of the 6th IASTED Int. Conf. Signal and Image Processing, pp. 441–446 (2004)
8. Castro, E.D., Morandi, C.: Registration of translated and rotated images using finite Fourier transforms. IEEE Trans. Pattern Anal. Machine Intell. 9, 700–703 (1987)