IPSJ Transactions on Computer Vision and Applications Vol. 2 1–14 (Feb. 2010)

Research Paper

Full Pixel Matching between Images for Non-linear Registration of Objects

Yuichi Yaguchi,[†1,†2] Kenta Iseki[†1] and Ryuichi Oka[†1]

A two-dimensional continuous dynamic programming (2DCDP) method is proposed for two-dimensional (2D) spotting recognition of images. Spotting recognition is the simultaneous segmentation and recognition of an image by optimal pixel matching between a reference image and an input image. The proposed method performs optimal pixel-wise image matching and 2D pixel alignment, which are not available in conventional algorithms. Experimental results show that 2DCDP precisely matches the pixels of nonlinearly deformed images.

1. Introduction

Pattern matching for the 2D objects is a most important problem in image processing. One-dimensional nonlinear pattern matching almost always uses dynamic programming (DP)-based [21),28)], automaton-based [17)], or hash-based [1),35)] methods. In 2D pattern matching, optimal pixel matching between images is widely used in the image processing [8)] for tasks such as recognition [7)], retrieval [9)], registration [27),39)], and three-dimensional (3D) reconstruction from stereoscopic and/or time series images [20),24),31)].

Image registration is achieved by using feature point matching [13),30)], histogram matching [10)], template matching [26)], gradient-based matching [25)], baseline matching [36)], or a combination of these matching methods [12)]. Matching methods are usually categorized as suitable for either linear or nonlinear matching. It is always difficult to achieve perfect image matching. It should be noted that feature point matching is used in other matching methods as the starting point for matching because it is only weakly affected by pixel coordination and it is possible to identify stabilized feature points by sparse coding. At this time, finding feature points is a major problem. Lukas, et al. use object corners [30)], Lowe detects scale invariant feature transformations (SIFT) [13)], and other researchers use other feature point methods for the matching [5),14),15)]. Several feature point matching methods are robust against affine transformation variations or scaling because they take advantage of point-to-point matching algorithms.

We propose another nonlinear approach to 3D shape reconstruction without using a tracking procedure. Unlike the images in previous work that are assumed to be linear-transformed or affine-transformed, most real-world images are nonlinearly deformed when compared with those captured at a different time or from a different viewpoint. In addition, for strict matching, image registration can be made more precise and accurate if we match the images at the pixel level rather than at the feature point level. Segmentation, on the other hand, is a challenging problem that also needs to be solved. Our objective is to develop a method that is able to solve both these problems, namely nonlinear deformation and segmentation. Moreover, we aim to enhance the method to obtain optimal pixel correspondence by aligning the nonlinear deformation of pixels between images.

Our approach is based on previous studies of the 2D extension to DP matching (see **Fig. 1**) [18),34)]. There have been several studies on applying DP-based matching to 2D data such as real-world images. DP-based matching was originally developed for one-dimensional data sequences. Myers and Rabiner introduced dynamic time warping (DTW) [16)] for connected word recognition. Uchida and Sakoe developed 2D warping (2DW) by extending one-dimensional DTW [32),33)] (bottom-right of Fig. 1). They argued that 2DW has a pattern combination problem in the vertical and horizontal correlation [34)], so its calculation time becomes nondeterministic polynomial-time hard. Furthermore, 2DW requires a pre-segmentation of images for identifying the matching area because it needs fixed start and end points as its input. Thus, the result of 2DW matching is affected by background, and it is not enable to realize spotting recognition.

On the other hand, Continuous DP (CDP) [23)], a well-known spotting method, uses simultaneous recognition and segmentation, so there is no need to presegment the input time sequence. CDP has been applied to continuous sound [23),37)] and gesture recognition [22)]. It is superior to conventional DTW be-

†1 Department of Computer and Information Systems, University of Aizu
†2 JSPS Research Fellow DC2

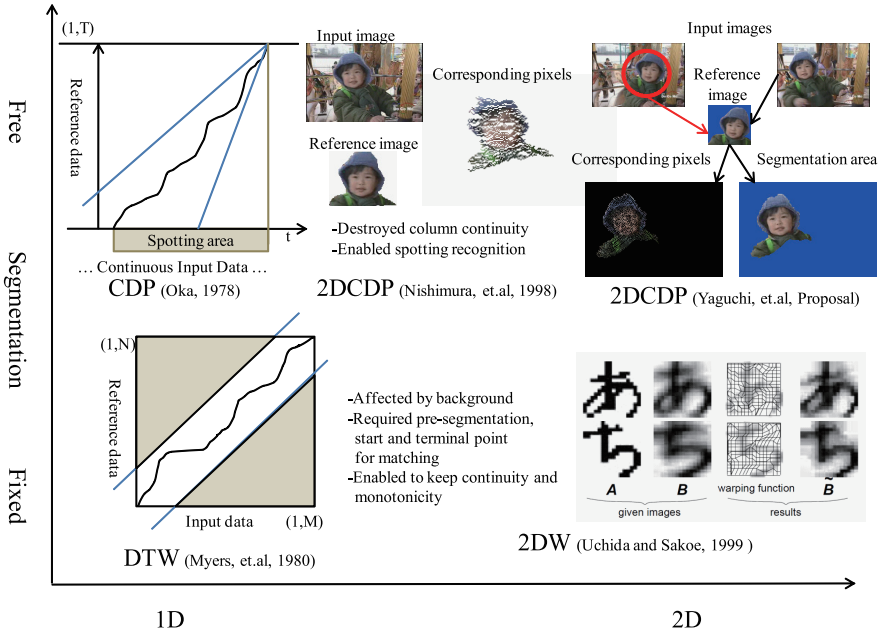

Fig. 1 State of arts for two-dimensional expansion of DTW and CDP.

cause it does not require pre-segmentation. Therefore, a 2D extension of CDP is able to overcome the problems of 2DW matching.

The first trial to 2D extension of CDP, proposed by Nishimura, et al. [18], applies CDP twice, firstly to calculate the differences of pixel intensity between input and reference images, accumulating a series of results for each row in the input image in the row direction, and secondly to accumulate the results for all rows aligned in the column direction. Therefore, this method is not considered a full 2D extension of CDP. It was extended by Suto, et al. for arbitrary-shaped queries [29]. Iwasa, et al. proposed a modification of Suto's method to enable continuous and monotonic pixel alignment [11]. However, these three methods still suffer pixel alignment errors because of the separation into column and row directions when accumulating the local distances between pixels in the two images (top-middle of Fig. 1). Moreover, Iwasa's method tends to miss matching pixels between images derived from a type of post-processing. To deal with the problems in all these methods, Yaguchi, et al. [38] proposed an accumulation and backtracking method to create a fully 2D extension of CDP.

Based on Yaguchi's approach, our method is a development that enables the simultaneous accumulation of local distances in both row and column directions. It optimally accumulates distances between corresponding pixels in two images, starting with the pixels in one corner of the reference image and moving toward those in the opposite corner. Because the pixels used in the reference image are positioned obliquely to each other, the total distances of pixels from the start to the end points can be obtained by simply adding up the distances in the row and column directions. Each pixel location in the input image is assumed to be the end point for the corresponding accumulation of local distances, and the optimal accumulation value is stored at that location. The location of the pixel in the local area of the input image that has the local minimum optimal accumulation value will be selected as the spotting point of the reference image. A segmented area of the input image is then extracted using backtracking of the matching paths, which are constructs of a mesh plane. This method is a true 2DCDP. It ensures complete 2D alignment of the pixels in the input image by matching to all pixels in the reference image. In addition, this 2DCDP achieves spotting recognition by extracting pixel correspondence between input and reference images, and recognizing labeled information in the reference image via the pixel correspondence of the two images.

The remainder of this paper has three sections. Section 2 describes the algorithm for our optimal pixel matching method. Section 3 describes our spotting recognition experiments and their results. Finally, Section 4 summarizes the key points and identifies some future work.

2. 2DCDP: An Optimal Pixel Matching Method

2.1 The Road Map of the DP Algorithm

The DP algorithm is designed to solve sequential decision problems. Such problems are usually expressed in terms of an automaton or a tree structure. The DTW algorithm [16] is used to accumulate the minimum number of errors from the start to the end point under the principle of optimality. For large-

scale input data, DTW needs to extract short segments for matching. Then, when DTW processes the large-scale input data, it will set many start and end points in the input sequence, and will duplicate many processes in calculating the accumulation values. CDP is able to reduce the calculation time of duplicated processes compared with DTW, and enables start-point-free nonlinear sequential data matching [23].

In image processing, spotting recognition is used to identify segmentation and nonlinear pixel movement by using a reference image. The conventional 2DW method [34] is unable to segment into an input image because it requires pre-segmentation for matching, similarly to DTW. In this paper, we introduce a method that is able to perform spotting recognition, and we develop a 2D extension, derived from CDP, for spotting recognition.

2.2 Definition of the 2DCDP Algorithm

2DCDP is an extension of CDP [23] to 2D correlation, and is an effective algorithm for full-pixel matching (top-right part of Fig. 1). The pixel coordinates of input image S and reference image R are defined by:

$$S \triangleq \{(i,j)|1 \le i \le I, 1 \le j \le J\} \tag{1}$$
$$R \triangleq \{(m,n)|1 \le m \le M, 1 \le n \le N\}. \tag{2}$$

The pixel value at location (i,j) of the input image Sp is $Sp(i,j) = \{r,g,b\}$, and the pixel value at location (m,n) of the reference image Rp is $Rp(m,n) = \{r,g,b\}$, where r, g, and b are normalized red, green, and blue values respectively, and $(0 \le \{r,g,b\} \le 1)$.

We define the mapping $R \to S$, $(m,n) \in R$ and $(\xi(m,n), \eta(m,n)) \in S$ by:

$$(m,n) \Longrightarrow (\xi(m,n), \eta(m,n)), \tag{3}$$

set the end location for pixel matching as:

$$\xi(M,N) = \hat{i}, \quad \eta(M,N) = \hat{j}, \tag{4}$$

and the point of $(\hat{i},\hat{j})$ as a nomination of the spotting point.

Next, we set the local distance $d(i,j,m,n)$ as the difference value between $Sp(i,j)$ and $Rp(m,n)$, and set $w(i,j,m,n)$ as the weighted value of each local calculation. The accumulated local minimum $D(i,j,m,n)$ is used to evaluate the decision sequence, and is defined as:

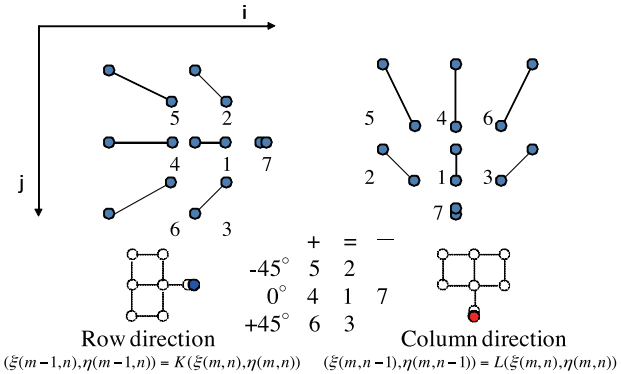

Fig. 2 Variation of candidate local paths. Paths are set as (1) same size, (2) same size and negative 45-degree rotation, (3) same size and positive 45-degree rotation, (4) doubled, (5) doubled and negative 45-degree rotation, (6) doubled and positive 45-degree rotation, and (7) a shrinking path.

$$D(\hat{i},\hat{j},m,n) = \tag{5}$$
$$\frac{1}{W} \min_{\xi,\eta} \left\{ \sum_{m=1}^{M} \sum_{n=1}^{N} w(\xi(m,n),\eta(m,n),m,n) d(\xi(m,n),\eta(m,n),m,n) \right\}.$$

Then $\xi^*(m,n)$ and $\eta^*(m,n)$ are used to represent the optimal solutions in $\xi(m,n)$ and $\eta(m,n)$ respectively, where W is the optimal accumulated weight:

$$W = \sum_{m,n} w(\xi^*(m,n),\eta^*(m,n),m,n). \tag{6}$$

To ensure monotonicity in non-linear pixel matching, $K(m,n) = \{\xi(m-1,n),\eta(m-1,n)\}$ and $L(m,n) = \{\xi(m,n-1),\eta(m,n-1)\}$ are used to define the sets of points that are movable in the i and j directions in the input image, taken from the movements in the m and n directions in the reference image (**Fig. 2**). Also, to ensure continuity in two-dimensional pixel correlation, the following equation decides the a suitable corresponding pixel of $(m-1,n-1)$ from three corresponding pixels (m,n), $(m-1,n)$ and $(m,n-1)$ (**Fig. 3**):

$$(\xi(m-1,n-1),\eta(m-1,n-1)) \in$$
$$K(m,n) \otimes L(m-1,n) \cap L(m,n) \otimes K(m,n-1). \tag{7}$$

Here, the operator $\otimes$ represents the connection between a set of points on the

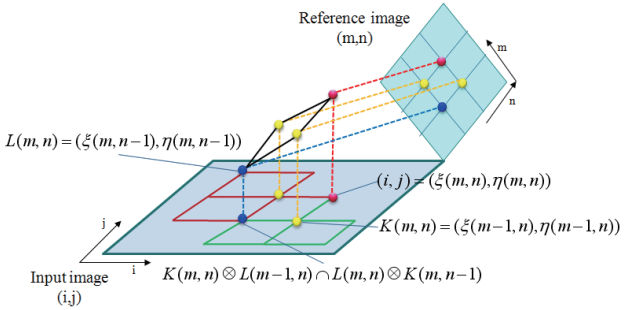

Fig. 3 Example explaining the roles of $L(m,n)$ and $K(m,n)$, which guarantee the 2D constraint between a reference image and an input image. This figure shows only one case (linear matching) among the possible cases for optimal matching of local images, which include many different cases of nonlinear optimal matching of local areas.

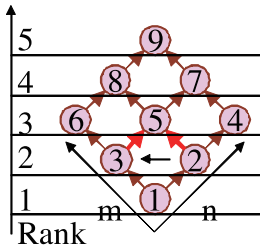

Fig. 4 Computation sequences and the rank of accumulation. A high-rank calculation node affects two lower-rank nodes directly, and all nodes that belong to the calculation node indirectly.

left and a set of points on the right.

To calculate the accumulated local distance, each accumulated local minimum $D(i,j,m,n)$ is derived from two previous accumulated local minimum $D(i',j',m-1,n)$ and $D(i'',j'',m,n-1)$. In this way, we define the rank $l=m+n$, as shown in **Fig. 4**, to smoothly calculate the accumulated local minimum.

Totally, an corresponding pixels set between input and reference images is detetcted into four-dimensional space at 2DCDP calculation as **Fig. 5**.

2.3 Implementation of Local Distance

The accumulation of the local distance in full-pixel matching requires simultaneous accumulation in the m and n directions for each pixel. In the accumulation calculation, the accumulated values are optimally selected in two directions. However, many formulas could be used to calculate the local distance value provided the value is normalized ($0 \le d \le 1$), because the accumulated distance is calculated by just adding up the local distance values. In our experiments, the pixel distance is as follows:

$$d(i,j,m,n)=\frac{1}{3}\sum_{k=1}^{3}|Sp_k(i,j)-Rp_k(m,n)|, \tag{8}$$

where the variable k indicates the k-th element of $Sp(i,j)$ and $Rp(m,n)$. Then the variance range of $d(i,j,m,n)$ is set as $0 \le d(i,j,m,n) \le 1$.

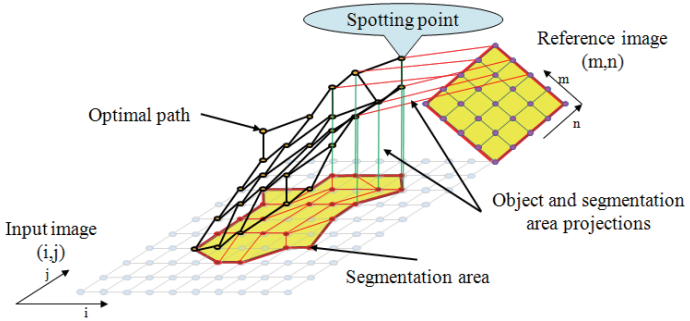

Fig. 5 Determination of a segmented area obtained by projecting optimal paths in the 3D space on the input image.

2.4 Algorithm for Optimal Local Distance Accumulation

2DCDP selects two local paths that are needed to check the connection of the four points (m,n), $(m-1,n)$, $(m,n-1)$, and $(m-1,n-1)$ that form a quadrangle (Fig. 3 and **Fig. 6**). As shown in Fig. 2, 2DCDP defines seven paths for each m and n direction as the local accumulation paths, namely (1) same size, (2) same size and minus 45-degree rotation, (3) same size and plus 45-degree rotation, (4) doubled, (5) doubled and minus 45-degree rotation, (6) doubled and plus 45-degree rotation, and (7) a shrinking path. Each accumulation point has four values, as shown in **Fig. 7**. If these four points (m,n), $(m-1,n)$, $(m,n-1)$, and $(m-1,n-1)$ keep to form a quadrangle similar to that in Fig. 3 at accumlation calculation, we need to check the whole enable patterns that are derived from the

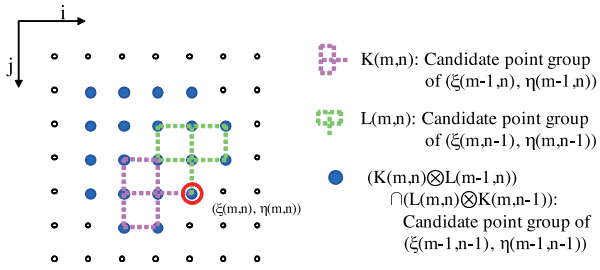

Fig. 6 Constraint conditions for neighboring pixels. Each i and j direction can connect seven candidate pixels. 2DCDP selects the node that has a minimal accumulation value from among these paths.

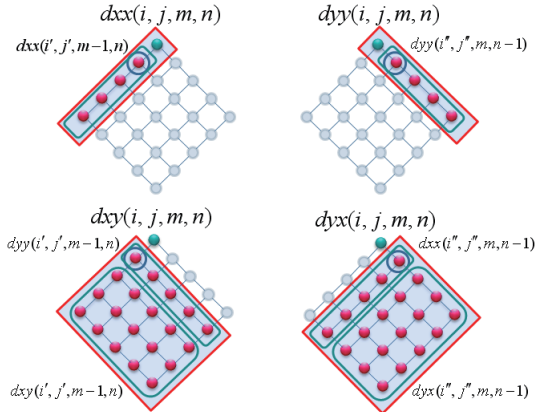

Fig. 7 Definition of the accumulation calculation of $D(\hat{i}, \hat{j}, m, n)$ projected in (m, n) space. This implementation avoids duplication of accumulating nodes that are connected indirectly.

above local accumulation paths, and the number of enable patterns are 165 which are derived from constraints condition expressed by Fig. 6. This checking procedure could spend much time on unnecessary recalculating operations. Therefore, we do not mention about optimality of "path direction" in this accumulation calculation. Alternatively, accumlation calculation keeps optimality of accumlated value. We set four values for the accumulating calculation of dxx, dxy, dyx, and dyy, as shown in Fig. 7, to take over low-level accumulation results and retain the path constraints. These four values enables to keep sum up equally. In addition, we set the path weights, as shown in Fig. 2, to simplify the algorithm. Then all path weight values will be set to $w(i, j, m, n) = 1$.

The algorithm for the accumulation of a local minimum is shown in terms of the following equations:

For $l = 2,\ \ l \leq M + N - 1,\ \ l = l + 1$
  For $m = 1$ and $n = l,\ \ m \leq M$ and $1 \leq N,\ \ m = m + 1$ and $n = n - 1$
    If $n > N$: continue.

Path selection:

$$(i', j', m-1, n) \triangleq \underset{\{i', j'\}}{\operatorname{argmin}} \left\{ \begin{array}{c} D(i-1, j, m-1, n) - dyx(i-1, j, m-1, n) \\ D(i-1, j-1, m-1, n) - dyx(i-1, j-1, m-1, n) \\ D(i-1, j+1, m-1, n) - dyx(i-1, j+1, m-1, n) \\ D(i-2, j, m-1, n) - dyx(i-2, j, m-1, n) \\ D(i-2, j-1, m-1, n) - dyx(i-2, j-1, m-1, n) \\ D(i-2, j+1, m-1, n) - dyx(i-2, j+1, m-1, n) \\ D(i, j, m-1, n) - dyx(i, j, m-1, n) \end{array} \right\}, \tag{9}$$

$$(i'', j'', m, n-1) \triangleq \underset{\{i'', j''\}}{\operatorname{argmin}} \left\{ \begin{array}{c} D(i, j-1, m, n-1) - dxy(i, j-1, m, n-1) \\ D(i-1, j-1, m, n-1) - dxy(i-1, j-1, m, n-1) \\ D(i+1, j-1, m, n-1) - dxy(i+1, j-1, m, n-1) \\ D(i, j-2, m, n-1) - dxy(i, j-2, m, n-1) \\ D(i-1, j-2, m, n-1) - dxy(i-1, j-2, m, n-1) \\ D(i+1, j-2, m, n-1) - dxy(i+1, j-2, m, n-1) \\ D(i, j, m, n-1) - dxy(i, j, m, n-1) \end{array} \right\}, \tag{10}$$

Accumulation of four values:

$$dxx(i, j, m, n) \triangleq d(i, j, m, n) + dxx(i', j', m-1, n) \tag{11}$$

$$dxy(i, j, m, n) \triangleq dxy(i', j', m-1, n) + dyy(i', j', m-1, n) \tag{12}$$

$$dyx(i,j,m,n) \triangleq dyx(i'',j'',m,n-1) + dxx(i'',j'',m,n-1) \quad (13)$$
$$dyy(i,j,m,n) \triangleq d(i,j,m,n) + dyy(i'',j'',m,n-1), \quad (14)$$

Accumulation of local minimum value:

$$D(i,j,m,n) \triangleq dxx(i,j,m,n) + dxy(i,j,m,n) + dyx(i,j,m,n) + dyy(i,j,m,n). \quad (15)$$

Equations (9)–(15) imply that an accumulated value $D(i,j,m,n)$ is recursively calculated by $D(i',j',m-1,n)$ and $D(i'',j'',m,n-1)$ following the application of DP. The path configuration in Fig. 2 enables infinite path shrinking. Therefore, in our experiment, we counted the number of times shrinking occurred and set a limit for the number of consecutive path-shrinkage occurrences. Finally, the optimal spotting point corresponding to (i,j) in the input image is given by:

$$\begin{aligned} &D(\hat{i},\hat{j},m,n) = \\ &\min_{\xi,\eta}\left\{\sum_{m=1}^{M}\sum_{n=1}^{N} dxx(\xi(m,n),\eta(m,n),m,n) + dxy(\xi(m,n),\eta(m,n),m,n)\right. \\ &\left.+dyx(\xi(m,n),\eta(m,n),m,n) + dyy(\xi(m,n),\eta(m,n),m,n)\right\} \\ &= \min_{\xi,\eta}\left\{\sum_{m=1}^{M}\sum_{n=1}^{N} 2d(\xi(m,n),\eta(m,n),m,n)\right\} \\ &= 2\min_{\xi,\eta}\left\{\sum_{m=1}^{M}\sum_{n=1}^{N} d(\xi(m,n),\eta(m,n),m,n)\right\}. \end{aligned} \quad (16)$$

This equation follows Eq. (5).

2.5 Correction of Mesh Structure Using Backtracking

After the spotting point has been determined, we need to extract the spotting area from the four-dimensional (4D) accumulated local minimum space. In the CDP algorithm, the backtracking part traces only the connected local paths from the spotting point. However, the connected local paths in 2DCDP sometimes conflict with the constructed mesh structure in the m and n directions. On the other hand, each matching point $D(i,j,m,n)$ has an optimal accumulated value from the start to that point. Therefore, the algorithm for finding the optimal path from two points is expressed via the following equation:

$$(i^*,j^*) \in K(\xi^*(m+1,n),\eta^*(m+1,n)) \otimes L(\xi^*(m,n+1),\eta^*(m,n+1)) \quad (17)$$
$$(\xi^*(m,n),\eta^*(m,n)) = \underset{\{i^*,j^*\}}{\operatorname{argmin}}\{D(i^*,j^*,m,n)\}. \quad (18)$$

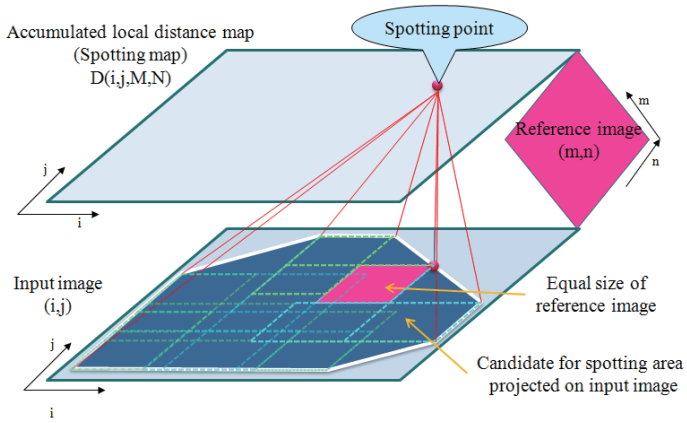

Fig. 8 A spotting point and its spotting area. The maximum size of spotting is 12 times the size of the reference image.

The candidate spotting area in the input image is about 12 times larger than for the reference image (**Fig. 8**) because the implementation allows 45-degree rotation and doubled size for each of the connected paths. The problem with backtracking is that it is able to select a shrinking path without any limitation, which can cause overshrinking of the spotting area. Therefore, we implement a controlling variable to limit the number of consecutive occurrences of shrinking.

Finally, a set P, containing the segments, is defined as:

$$P \subset \{(\xi^*(m,n),\eta^*(m,n))|1 \le m \le M, 1 \le n \le N\}. \quad (19)$$

2.6 Extraction of Object from Reference Image

When the value of the local distance of pixels in a discarded area is set to a maximum value, the local distance value of pixels in the background will be the same, and the arbitrary shape can be cut off from the reference image as shown in **Fig. 9**. In our experiment, the value of the local distance was set to 1.

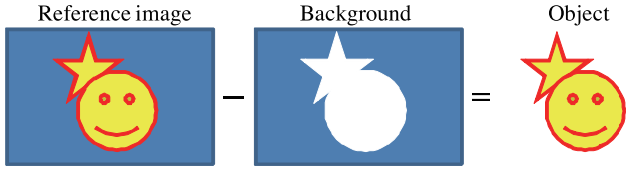

Fig. 9 Arbitrary shape matching overview. A reference image is composed of object areas and a background area. During the local minimum calculation, the out-of-mask area (background) in the reference image sets the maximum value for the distance dividing background and foreground. After the accumulating calculation, the accumulated value of the out-of-mask area is subtracted and these nodes are deleted from the matching result.

2.7 Calculation Time and Memory Amount

Assume that 2DCDP takes unit time to calculate the local distance and accumulation at each element in a 4D tensor field. Then the time needed for the 2DCDP calculation is $O(n^4)$ because the number of elements in the tensor field is $I \times J \times M \times N$. In this algorithm, backtracking needs the value for each accumulated local minimum $D(i, j, m, n)$. Therefore, the amount of memory required is also $O(n^4)$.

3. Experiments

3.1 Spotting Recognition Experiment

3.1.1 Methods and Materials

To evaluate our optimal pixel matching method experimentally, we used a single OS-implemented thread (Mac OS X running on an Xserve containing dual 2.8 GHz quad-core Xeon processors and 32 GB SDRAM). In the first experiment, spotting recognition used an arbitrary-shaped query extracted from the original image (the image on the top in **Fig. 10**). In the second experiment, multi-answer spotting recognition used real-world data. The third experiment applied 2DCDP to images from nature.

Exp. 1: Spotting recognition used four input images (Fig. 10) as follows:

Input 1: Spotting recognition uses affine transformed images from the top and bottom halves of an image extracted from the original image.

Input 2: Spotting recognition uses affine transformed images from the left and right halves of an image extracted from the original image.

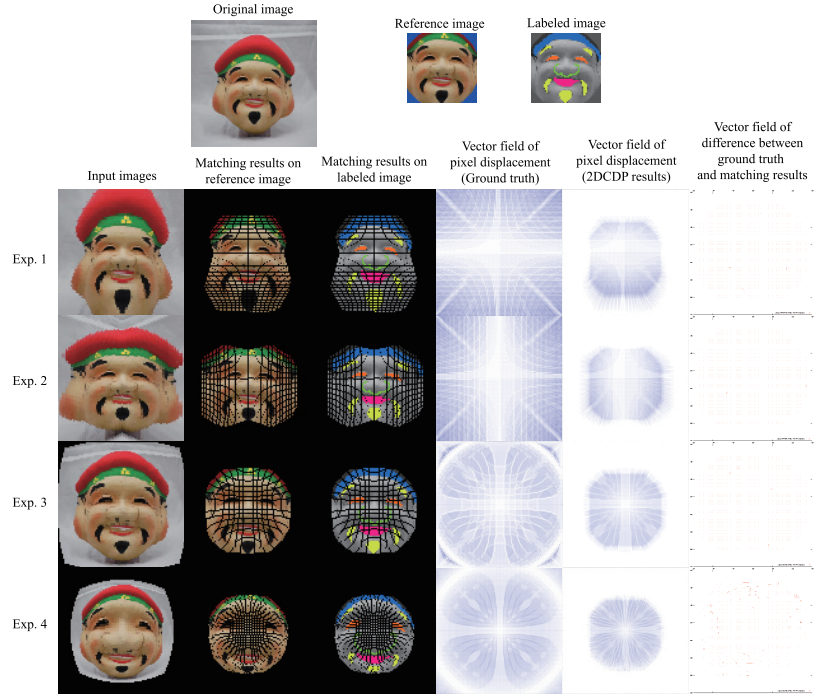

Fig. 10 Experimental results for 2DCDP. Image 1: Vertically divided affine image. Image 2: Horizontally divided affine image. Image 3: Thick lens distortion; applied transform $a_1 r - a_2 r^3$, $a_1 = 0.3$, $a_2 = 0.0001$. Image 4: Thick lens distortion; applied transform $a_1 r - a_2 r^3$, $a_1 = 0.6$, $a_2 = 0.0005$.

Input 3: Spotting recognition uses a distorted image, such as one captured through a thick lens, expressed by $a_1 r - a_2 r^3, a_1 = 0.3, a_2 = 0.0001$, extracted from the original image.

Input 4: Spotting recognition uses a distorted image, such as one captured through a thick lens, expressed by $a_1 r - a_2 r^3, a_1 = 0.6, a_2 = 0.0005$, extracted from the original image.

Exp. 2: Spotting recognition uses an input image constructed from several images selected from a movie and another picture ($I = 320, J = 240$) and a reference image from another frame of the movie ($M = 63, N = 61$).

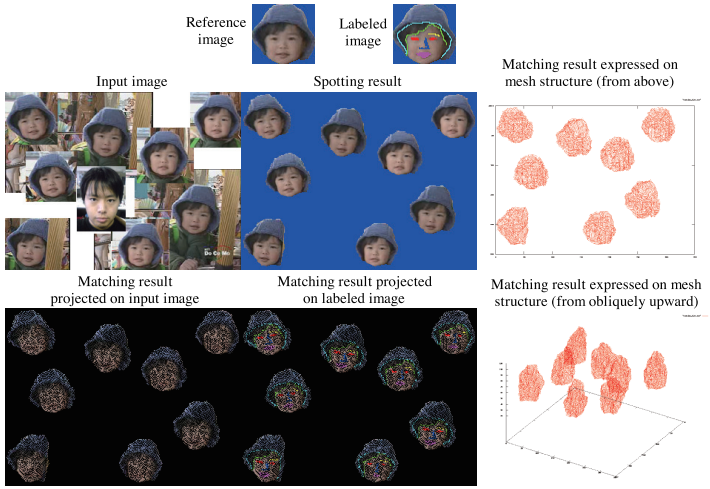

Fig. 11 (Left): Extraction of multiple objects from an image using a face query. Each face object is extracted from a different frame in a time-varying sequence and placed in an image. The reference image is also extracted from a frame image. The method extracts only eight similar face objects. (Upper Right): Eight 2D mesh images. Each mesh indicates the nonlinear and optimal correspondence for all pixels between a reference image and a spotted area of the input image. (Lower Right): Eight sets of 3D paths obtained by the application of 2DCDP.

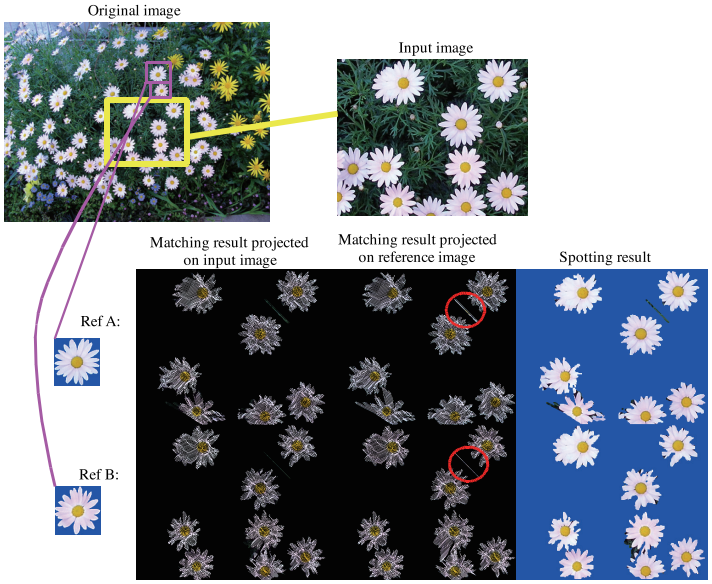

Fig. 12 Multi-object extraction using two different queries about flowers. Reference A can detect seven out of eleven objects. Reference B can detect eight out of eleven objects. Two different queries can spot seven or eight objects because of color differences between the references. A red circle identifies an area with pixel matching errors.

Exp. 3: Image spotting from nature using 2DCDP, as shown in **Figs. 12** and **13**.

In Experiment 1, we used a 100×100-pixel image for the input and a 55×55-pixel image for the reference. In Experiment 2, we used several frames from a video database [2] and cut-and-pasted other face-image frames that had several margins. In these experiments, the shrinking limit was set to 2.

3.1.2 Experimental Results

For Experiment 1, Fig. 10 shows the ground truth (labeled as "Vector field of pixel displacement (ground truth)"), the pixel movement (labeled as "Vector field of pixel displacement obtained by 2DCDP"), and the difference between the ground truth and the pixel movement (labeled as "Vector field of difference between ground truth and matching result"). The accuracy rate results are shown in **Table 1**. Input 4 showed that, although some pixel movements exceeded the limited path constraint in the ground truth, this method was still effective because it is a method for finding global optimality. Experiment 2 showed that, for several extracted facial areas, it was able to find multiple candidates for each area and that each area had a pixel-to-pixel relationship between the subject and reference images. **Figure 11** shows that every result successfully indicated the borderlines between hair, face, eyes, nose, and mouth. The calculation time and memory usage is shown in **Table 2**. In Experiment 3, Fig. 12 also shows that 2DCDP is able to extract multiple spotting areas and capture different results using two different reference images because these two reference images have differences in color and shape. In Fig. 12, we obtain nine objects in each trial, with some spotting errors indicated by the red circle in the figure. Figure 13 indicates that 2DCDP is able to extract an object that has undergone a perspective transformation. This is a significant feature for image-based modeling, because this

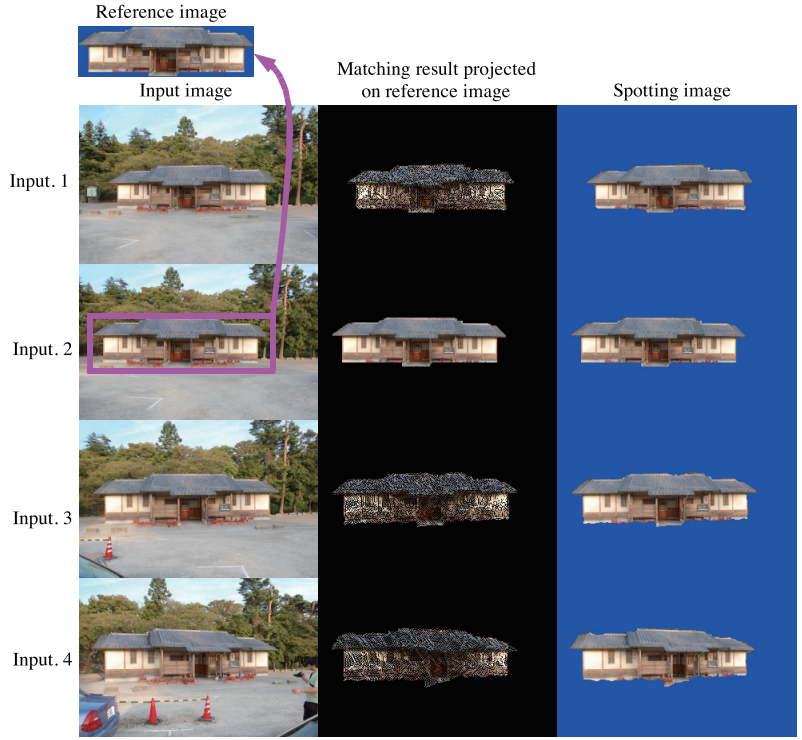

Fig. 13 Extracting a building from different frames in a stream of moving image.

full-pixel matching will easily enable the reconstruction of a 3D shape from two or more images.

3.2 Extraction Pixel Flow Experiment

To compare optimal pixel matching methods, this experiment compares pixel flow using 2DCDP with several optical flow methods, namely block matching (BM), the Horn & Schunck method (HS), the Lukas & Kanade method (LK), and SIFT matching flow. The 2DCDP pixel flow uses 240×180, 120×190, and 40×30 pixels of images. The other methods use 740×480 and 240×180 pixels of images in the comparison experiments.

For the experimental environment, we used a DELL Precision system (CPU: Xeon 3.16 GHz dual CPU, Memory: 64 GB, OS: Cent OS). For the comparisons, we used Autopano-SIFT [19] as the SIFT application and the OpenCV [6] Library for the development of each optical flow algorithm. For the comparison material, we used a movie that included a TV program [3].

Table 1 Performance in Experiment 1: Accuracy rate of pixel movement was calculated to be less than $\sqrt{2}$ of pixel movement error.

	Input 1	Input 2	Input 3	Input 4
No. of corresponding pixels	2741	2741	2724	1820
No. of corresponding errors	1	0	3	20
Accuracy rate of pixel movement	99.963%	100.00%	99.890%	98.901 %
Calculation time (s)	4.931	4.954	4.954	4.855

Table 2 Performance in Experiments 2 and 3: Calculation time and memory size increase $O(N^4)$.

	Fig. 11	Fig. 12 Ref. 1	Fig. 12 Ref. 2	Fig. 13 Average
Input image size	320×240	416×339	416×339	300×199
Reference image size	63×61	96×98	90×96	219×63
Calculation time (sec)	59.823	235.000	261.478	159.500
Memory usage (GByte)	5.3	23.9	21.9	14.8

Figures 14–17 are examples of extracting pixel flow using 2DCDP. Figure 14 indicates that 2DCDP was able to track a series of corresponding pixels derived by transformation of the foreground image. Occlusion pixels in the input image, and those assigned to the border of background and foreground images, did not violate constraints on continuity and monotonicity. A masked image example is shown in Fig. 15. 2DCDP was able to extract an optimal matching path locally, because the 2DCDP method extracts via global optimality but the paths are decided by local optimality.

Figure 16 shows global optimality. Optical flow methods could not track global variations of the image [4]. In addition, the LK method needed some texture information to track pixels. The BM method was able to track objects at first but it became difficult to make correct correspondences eventually, because the BM method cannot check the constraints of continuity and monotonicity. 2DCDP was able to track pixel-by-pixel. Figure 17 shows extraction in adverse conditions. The adverse aspects are the different background, the illumination, and

Fig. 14 Pixel tracking using 2DCDP(1): The second and fourth columns show the result of tracking multiple objects from the first and third columns. Our proposed method extracts precise pixel flows caused by the motion of multiple objects. Occlusion and emerging new objects in a stream of time-varying images make up a small part of the pixel flow image.

Fig. 15 Pixel tracking using 2DCDP(2): The result of pixel tracking using arbitrary shapes. 2DCDP was able to track object deformation.

the shape of foreground, but the foreground object is the same. In this example, the results for a background that included the reference show that pixel flow was not extracted by any method. However, 2DCDP was able to extract the object using a masked reference image.

Figures 18 and **19** are comparisons of optical flow. For SIFT matching, we barely obtain the motion of the human body in Fig. 19. In Fig. 18, we cannot obtain corresponding points of the objects and have many wrong correspondences. Furthermore, for optical flow in the LK, BM, and HS methods, we obtain the direction of movement of the object (roller coaster) in Fig. 18, but we have many wrong correspondences in Fig. 19 because there is much variation in the images. In particular, the HS method has many wrong correspondences in Fig. 19. On the other hand, for 2DCDP, we obtain packed flow for every object with both

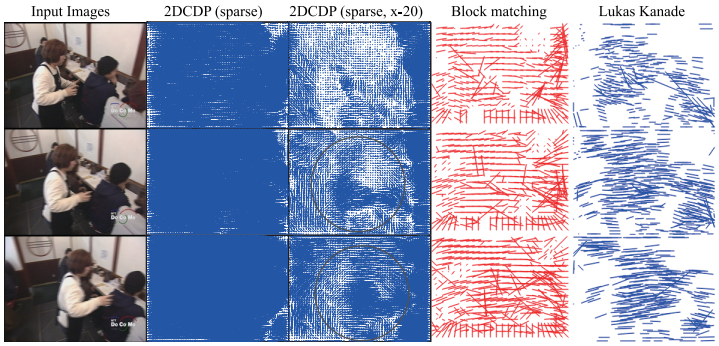

Fig. 16 Pixel tracking using 2DCDP (3): The result of macro-level camera motion (panning) and comparison results for the LK and BM methods. The LK method was not able to extract flow on a textureless plain but 2DCDP and the BM method could. In addition, 2DCDP was able to fix correspondences via the constraints of continuity and monotonicity. Third columns indicates our method can track object motion into 20 pixels panning of x-axis.

240×180 and 120×90 image sizes, as shown in Figs. 18 and 19. The texture of the object is continuous (except if self-occlusion occurs), and if occlusion occurs, the occluded points form on the edge, so, in 2DCDP, the flow is organized by the constraint conditions on pixel connections. However, for cases where the texture information is low in the visible part lost in the reference, the result will be matching errors. In addition, if we use small images such as 40×30 with 2DCDP, we obtain a very harsh flow for the resolution of the object because, although we can obtain global flow, the variation of motion will become bigger. We should consider these problems in terms of tracking area and fineness of texture.

Comparing execution speed, 2DCDP takes about 44 seconds for a 240×180 image, about 6 seconds for 120×90, and about 0.1 seconds for 40×30. SIFT takes about 30 seconds for 720×480, and 8 seconds for 240×180. The BM method takes 63 seconds for 720×480 with a 10×10 window size. The LK method takes less than 0.1 seconds for a 720×480 image.

4. Conclusion

We have developed and tested a 2DCDP method for spotting recognition of images. It achieves simultaneous segmentation and image recognition caused by its continuous and monotonic pixel-to-pixel matching. Our testing has demonstrated that it is robust against nonlinear deformation of images. Our future work will enable this method to use other indicators besides RGB in the above experiments. Our plan for the future also includes investigating applications of 2DCDP, such as finding errors in medical images from nonlinear image registration, 3D reconstruction, and recognition of facial expressions.

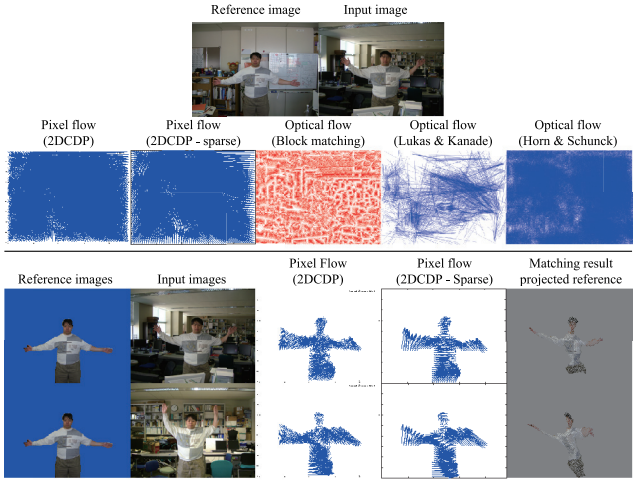

Fig. 17 Pixel tracking using 2DCDP (4): The result of pixel tracking for images of the human body. The first column is the original image sequence. The second is the sequence of images composed of the optical flows obtained from two successive images of the original image. The third is the sequence of the segmented part from the input frame, using 2DCDP with increments in the reference image. The fourth is the sequence of images showing the difference values for pixels obtained from pixel correspondence. A black part indicates a small value for the difference.

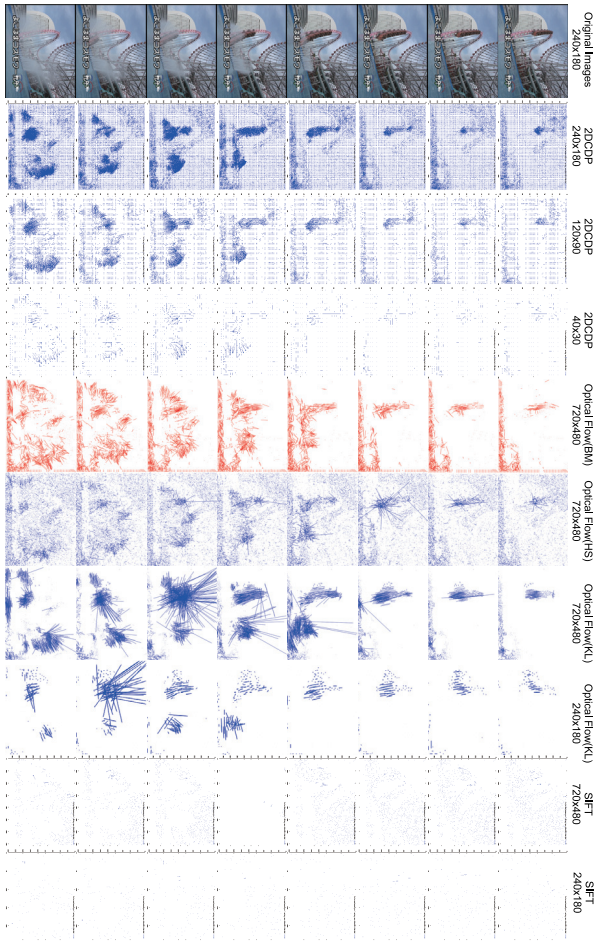

Fig. 18 Comparison (1) of optical flow among different methods. The time-varying image indicates the scene obtained from a roller coaster and spray. The SIFT method cannot track most of the pixels of the moving object. The other methods, including 2DCDP, can track many pixels of the moving object.

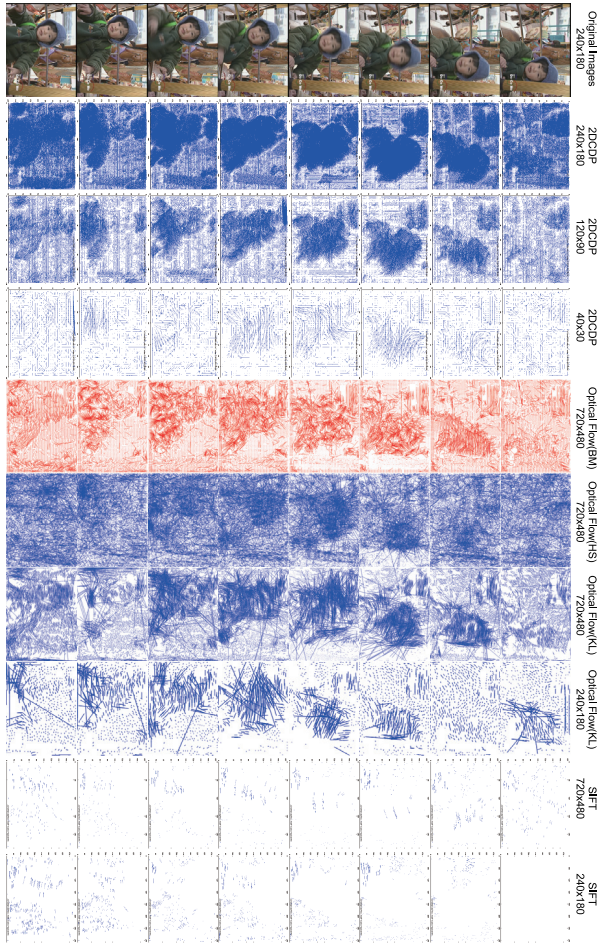

Fig. 19 Comparison (2) of optical flow among different methods. The LK, HS, and BM methods make incorrect pixel correspondences because they lack guarantees for the two characteristics of continuity and monotonicity between the two images to be matched. The proposed method does not completely succeed in making a perfect optimal flow, but it adapts its tracking to the variation caused by deformation of shape of the foreground object. The SIFT method can track every object, but the number of tracking points is quite small compared with other methods.

References

1) Attig, M., Dharmapurikar, S. and Lockwood, J.: Implementation results of bloom filters for string matching, *Field-Programmable Custom Computing Machines, 2004, FCCM 2004, 12th Annual IEEE Symposium on*, pp.322–323 (2004).
2) Babaguchi, N., Etoh, M., Satoh, S., Adachi, J., Akutsu, A., Ariki, Y., Echigo, T., Shibata, M., Zen, H., Nakamura, Y. and Minoh, M.: Video Database for Evaluating Video Processing, *Technical Report of IEICE, PRMU* (2002).
3) Babaguchi, N., Etoh, M., Satoh, S., Adachi, J., Akutsu, A., Ariki, Y., Echigo, T., Shibata, M., Zen, H., Nakamura, Y., Minoh, M. and Matsuyama, T.: Video Database for Evaluating Video Processing, *Technical Report of IEICE, PRMU*, Vol.102, No.155, pp.69–74 (2002).
4) Barron, J., Fleet, D. and Beauchemin, S.: Performance of optical flow techniques, *International journal of computer vision*, Vol.12, No.1, pp.43–77 (1994).
5) Bay, H., Tuytelaars, T. and Van Gool, L.: Surf: Speeded up robust features, *Lecture Notes in Computer Science*, Vol.3951, p.404 (2006).
6) Bradski, G.: OpenCV: Examples of use and new applications in stereo, recognition and tracking, *Proc. Intern. Conf. Vision Interface* (*VI'2002*) (2002).
7) Brunelli, R. and Poggio, T.: Face recognition: features versus templates, *IEEE Trans. PAMI*, Vol.15, No.10, pp.1042–1052 (1993).
8) Forsyth, D. and Ponce, J.: *Computer Vision: A Modern Approach*, Prentice Hall Professional Technical Reference (2002).
9) Geiger, D., Gupta, A., Costa, L. and Vlontzos, J.: Dynamic programming for detecting, tracking, and matching deformable contours, *IEEE Trans. PAMI*, Vol.17, No.3, pp.294–302 (1995).
10) Hashizume, C., Vinod, V.V. and Murase, H.: Robust object extraction from color images under illumination changes, *Technical report of IEICE, PRMU*, Vol.97, No.325, pp.33–40 (1997).
11) Iwasa, Y. and Oka, R.: Algorithm for Guaranteeing Monotonuous Contiguity of Pixel Correspondence in Spotting Recognition of Image, *MIRU2005*, pp.997–1004 (2005).
12) Kannala, J. and Brandt, S.: Quasi-dense wide baseline matching using match propagation, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, pp.1–8 (2007).
13) Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints, *IJCV*, Vol.60, No.2, pp.91–110 (2004).
14) Mikolajczyk, K. and Schmid, C.: A performance evaluation of local descriptors, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.27, No.10, pp.1615–1630 (2005).
15) Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T. and Gool, L.: A comparison of affine region detectors, *International Journal of Computer Vision*, Vol.65, No.1, pp.43–72 (2005).
16) Myers, C., Rabiner, L. and Rosenberg, A.: Performance tradeoffs in dynamic time warping algorithms for isolated word recognition, *IEEE Trans. ASSP*, Vol.28, No.6, pp.623–635 (1980).
17) Navarro, G.: A Guided Tour to Approximate String Matching, *ACM Computing Surveys*, Vol.33, No.1, pp.31–88 (2001).
18) Nishimura, T. and Oka, R.: Spotting Image Recognition using Two-Dimensional Continuous Dynamic Programming, *Technical Report of IEICE, PRMU*, pp.1–7 (1997).
19) Nowozin, S.: Autopano–Sift, making panoramas fun (2005). http://user.cs.tu-berlin.de/˜nowozin/autopano-sift/
20) Ohta, Y. and Kanade, T.: Stereo by intra-and inter-scanline search, *IEEE Trans. PAMI*, Vol.7, No.2, pp.139–154 (1985).
21) Oka, R.: Continuous words recognition by use of continuous dynamic programming for pattern matching, *Acous. Soc. J., SIG-S, $78-20*, pp.145–152 (1978).
22) Oka, R., Nishimura, T. and Yabe, H.: On Spotting Recognition of Gesture Motion from Time-varying Image, *Trans. IPSJ*, Vol.43, pp.54–68 (2002).
23) Oka, R.: Spotting Method for Classification of Real World Data, *Comput. J.*, Vol.41, No.8, pp.559–565 (1998).
24) Okutomi, M. and Kanade, T.: A multiple-baseline stereo, *IEEE Trans. PAMI*, Vol.15, No.4, pp.353–363 (1993).
25) Paillou, P. and Gelautz, M.: Relief reconstruction from SAR stereo pairs: the "optimalgradient" matching method, *IEEE Trans. Geoscience and Remote Sensing*, Vol.37, No.4, pp.2099–2107 (1999).
26) Pereira, S. and Pun, T.: Robust template matching for affine resistant image watermarks, *IEEE Trans. Image Processing*, Vol.9, No.6, pp.1123–1129 (2000).
27) Pluim, J., Maintz, J. and Viergever, M.: Mutual-information-based registration of medical images: A survey, *IEEE Trans. Medical Imaging*, Vol.22, No.8, pp.986–1004 (2003).
28) Sakoe, H. and Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition, *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol.26, No.1, pp.43–49 (1978).
29) Suto, N., Nishimura, T., Fujii, R.H. and Oka, R.: Spotting Recognition of Concave and Convex Reference Image with Pixel-wise Correspondence using Two-dimensional Continuous Dynamic Programming, *Technical report of IEICE, PRMU* (2003).
30) Tomasi, C. and Kanade, T.: Detection and tracking of point features, *School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91–132* (1991).
31) Tomasi, C. and Kanade, T.: Shape and motion from image streams under orthography: A factorization method, *IJCV*, Vol.9, No.2, pp.137–154 (1992).

32) Uchida, S. and Sakoe, H.: Handwritten character recognition using monotonic and continuous two-dimensional warping, *5th International Conference on Document Analysis and Recognition* (*ICDAR–99*), pp.499–502 (1999).
33) Uchida, S. and Sakoe, H.: Piecewise linear two-dimensional warping, *Systems and Computers in Japan*, Vol.32, No.12, pp.1–9 (2001).
34) Uchida, S.: DP Matching: Fundamentals and Application, *Technical report of IEICE, PRMU*, pp.31–36 (2006).
35) Wu, S. and Manber, U.: Agrep—a fast approximate pattern-matching tool, *Proc. USENIX Technical Conference*, pp.153–162 (1992).
36) Xiao, J. and Shah, M.: Two-frame wide baseline matching, *Proc. 9th IEEE International Conference on Computer Vision, 2003*, pp.603–609 (2003).
37) Yaguchi, Y., Watanabe, Y., Naruse, K. and Oka, R.: Speech and Song Search on the Web: System Design and Implementation, *CIT 2007*, pp.270–278 (2007).
38) Yaguchi, Y., Iseki, K. and Oka, R.: Two-dimensional Continuous Dynamic Programming for Spotting Recognition of Image, *MIRU2008*, pp.708–714 (2008).
39) Zitová, B. and Flusser, J.: Image registration methods: A survey, *Image and Vision Computing*, Vol.21, No.11, pp.977–1000 (2003).

(Received March 31, 2009)
(Accepted October 26, 2009)
(Released February 4, 2010)

(Communicated by *Long Quan*)

Yuichi Yaguchi was born in 1981. He received his B.S. and M.S. from University of Aizu in 2006 and 2008 respectively. He has been studying for Ph.D. at University of Aizu since 2008, and engaging JSPS research fellow DC2 since 2009. His current research interests are the algorithm of spotting recognition for high-dimensional data and the system construction for multi-media information retrieval. He is a member of IEICE, and a student member of IEEE.

Kenta Iseki was born in 1983. He received his B.S. and M.S. from University of Aizu in 2007 and 2009 respectively. He has been working on NEC Corp. since 2009.

Ryuichi Oka was born in 1945. He received his M.E. and Ph.D. from The University of Tokyo in 1970 and 1984 respectively, and was engaged in research on character recognition and speech recognition with Electrotechnical Laboratory from 1970. He worked for Real World Computing Project of MIT Japan from 1993 to 2002, and became a professor at University of Aizu in 2002. His current research interests are the speech retrieval, the image understanding, the Web mining and the free-viewpoint 3D-TV. He is a senior member of edditorial committee of JSAI, and a member of IEEE, IEICE, ASJ.