Optimal Pixel Matching between Images

Yuichi Yaguchi, Kenta Iseki, and Ryuichi Oka

The University of Aizu

Tsuruga, Ikkimachi, Aizuwakamatsu, Fukushima, Japan

{d8101109,m5111118,oka}@u-aizu.ac.jp

<http://iplpcx1.u-aizu.ac.jp>

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

Keywords: Optimal Pixel Matching, DP, Spotting, Image Registration, Segmentation.

1 Introduction

Optimal pixel matching between images is widely used in image processing [1] for such tasks as recognition [2], retrieval [3], registration [4,5], and three-dimensional reconstruction from stereoscopic and/or time series images [6,7,8]. Image registration is done by using feature point matching [9,10], histogram matching [11], or based-on-correlation template matching [12]. Matching methods are usually divided into two categories: linear and non-linear. Feature point matching is non-linear, and histogram matching and based-on-correlation template matching are linear. Feature point matching using scale invariant feature transformation (SIFT) [10] is robust against variations in pixel shift and non-linear translation of feature points because it takes the advantages of point-to-point matching algorithm. Three-dimensional shape reconstruction using SIFT requires the use of a tracking procedure like Kanade-Lucas-Tomasi tracker [9]. We propose another non-linear approach to three-dimensional shape reconstruction without using a tracking procedure.

Unlike the images in the previous works that are assumed as linear-transformed or affine-transformed, most real-world images are non-linearly deformed compared with ones captured in a different time or from a different viewpoint. Additionally, for the 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, in the other hand, is a big challenging problem which also needs to be solved. Our objective is to develop a method which is able to solve both above mentioned problems: non-linear deformation and segmentation. Moreover, we aim to

enhance this method in order to obtain optimal pixel correspondence by aligning the non-linear deformation of pixels between images. Our approach is based on the previous studies on the two-dimensional extension of dynamic programming (DP) matching [13,14]. There have been several studies on applying DP-based matching to two-dimensional 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) [15] for connected word recognition. Uchida and Sakoe developed two-dimensional time warping (2DTW) by extending one-dimensional DTW [16]. They argued that 2DTW has a pattern combination problem in the vertical and horizontal correlation [13], so its calculating time becomes nondeterministic polynomial-time hard (NP-Hard). Furthermore, 2DTW requires the pre-segmentation of images for identifying the matching area because it needs fixed start and end points as its input. Continuous DP (CDP) [17], a well-known spotting method, uses simultaneous recognition and segmentation, so there is no need to segment the input time sequence to be matched in advance. CDP has been applied to continuous sound [17,18] and gesture recognition [19]. It is superior to conventional DTW because it does not require pre-segmentation. Thus, a two-dimensional extension of CDP is able to overcome the problems of 2DTW matching. The first two-dimensional CDP (2DCDP), proposed by Nishimura et al. [14], applies CDP two times: the first one is used to calculate the difference of pixel intensity between input and reference images, and then accumulate series of that results for each row in the input image on row direction, the second one is used to accumulate the results for all rows that align on column direction. Therefore, this method is not considered as a fully two-dimensional extension of CDP. It was extended by Suto et al. for arbitrary shaped queries [20]. Iwasa et al. proposed a modification of Suto's method to enable continuous and monotonic pixel alignment [21]. However, these three methods still suffer pixel alignment errors because of the separation of column and row directions when accumulating the local distances between pixels in the two images. Moreover, Iwasa's method tends to miss matching pixels between images derived from a kind of post-processing. To dispose of the problems of all these methods, Yaguchi et al. [22] proposed an accumulation and back tracking methods to create a fully two-dimensional extension of CDP.

Based on Yaguchi's approach, the method is developed for simultaneous accumulation of local distances in both the 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 starting to the ending points can be obtained by simply adding up the distances by 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 which has the local minimum optimal accumulation value will be selected for spotting point of the reference image. A segmented area

of the input image is then extracted using back tracking of matching paths which are construct of a mesh plane. This method is completely two-dimensional CDP. It ensures the completely two-dimensional alignment of the pixels in the input image by matching to all pixels in the reference image. In addition, 2DCDP attains the spotting recognition by extracting pixel correspondence between input and reference images, and recognizing labeled information of reference image through the pixel correspondence of the two images. The remainder of this paper is divided into three sections as follows: Section 2 describes the algorithm of our optimal pixel matching method. Section 3 describes our spotting recognition experiments and the results. Finally, section 4 summarizes the key points and mentions some future works.

2 2DCDP: An Optimal Pixel Matching Method

2.1 The Road Map of DP Algorithm

DP algorithm is designed to solve sequential decision problems. Such problems are usually expressed as an automaton or a tree structure. DTW algorithm [15] is used to accumulate the minimum amount of errors from the start to the end point under the principle of optimality. For large-scale input data, DTW needs to extract short segment for matching. Then, if DTW processes the large-scale input data, it will set many start and end points in the input sequence, and will duplicate many processes to calculate the accumulation values. CDP is able to reduce the calculation time of duplicated processes in DTW, and enables start-point-free non-linear sequential data matching [17].

In image processing, spotting recognition is used to identify segmentation and non-linear pixel movement by using a reference image. The conventional 2DTW method [13] is unable to segment into an input image because it requires pre-segmentation for matching like DTW. In this paper, we introduce a method which is able to perform spotting recognition, and we developed a two-dimensional extension derived from CDP for spotting recognition.

2.2 Definition of 2DCDP Algorithm

2DCDP supports full-pixel matching and it is extended from CDP into two-dimensional correlation. The pixel coordinates of input image S and reference image R are defined by:

$$S \triangleq \{(i, j) | 1 \leq i \leq I, 1 \leq j \leq J\} \quad (1)$$

$$R \triangleq \{(m, n) | 1 \leq m \leq M, 1 \leq n \leq N\}. \quad (2)$$

The pixel value at location (i, j) of an input image Sp is $Sp(i, j) = \{r, g, b\}$, and the pixel value at location (m, n) of an reference image Rp is $Rp(m, n) = \{r, g, b\}$. Values r , g and b express red, green and blue respectively, and each value has the range $(0 \leq \{r, g, b\} \leq 1)$. We define the mapping $R \rightarrow S$, $(m, n) \in R$ and $(\xi(m, n), \eta(m, n)) \in S$ by:

$$(m, n) \implies (\xi(m, n), \eta(m, n)), \quad (3)$$

Fig. 1. Image spotting: recognize input image using reference image and simultaneous optimal pixel matching and segmentation. There is no limit to number of matching segments.

and we set the end location of pixel matching as

$$\xi(M, N) = i, \quad \eta(M, N) = j. \quad (4)$$

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

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

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

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

Fig. 2. Relation of accumulation, spotting point and projecting spotting area

Fig. 3. Definition of rank in accumulation and computation sequences

Fig. 4. Constraint of pixel connection: each i and j direction can connects 7 candidate pixels

To ensure continuity and monotonicity, $K(m, n) = \{\xi(m-1, n), \eta(m-1, n)\}$ and $L(m, n) = \{\xi(m, n-1), \eta(m, n-1)\}$ are used to define the sets of points in the input image that are movable in the m and n directions from the reference image. The following relationship is required for point $(m-1, n-1)$ corresponding to (m, n) (see Figure 2 and Figure 4).

$$(\xi(m-1, n-1), \eta(m-1, n-1)) \in K(m, n) \otimes L(m-1, n) \cap L(m, n) \otimes K(m, n-1) \quad (7)$$

The operator $\otimes$ represents the connection between a set of points on the left and a set of points on the right.

To calculate accumulated local distance, each accumulated local minimum $D(i, j, m, n)$ is derived from two previous accumulated local minimums $D(i', j', m-1, n)$ and $D(i'', j'', m, n-1)$. Thus, we define rank $l = m + n$ as shown in Figure 3 in order to smoothly calculate the accumulated local minimum.

2.3 Implementation of Local Distance

The accumulation of the local distance in optimal pixel matching requires simultaneous accumulation in m and n directions for each pixel. In the accumulation

Fig. 5. Local path direction: seven candidate paths at 45-degree left and right rotations; doubled path expansion and shrinkage for each m and n direction

Fig. 6. Definition of local path weight: all paths determined as path weight becomes 3 in each m and n value increment

calculation, the accumulated values are optimally selected by two directions. However, there is no difference in the pixel distance between m and n directions because the data consists of only the distance values of a corresponding pixel.

In our experimental implementation, the pixel distance is as follows:

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

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

2.4 Algorithm for Optimal Local Distance Accumulation

2DCDP selects two local paths that need to check the connection of the four points (m, n) , $(m-1, n)$, $(m, n-1)$ and $(m-1, n-1)$ that become a quadrangle

Fig. 7. Definition of four accumulated values of accumulating combination

Fig. 8. Definition of accumulation calculation of $D(i, j, m, n)$

(Figure 7). 2DCDP defines seven paths for each m and n direction for the local accumulation paths, as shown in Figure 5: (1) same size, (2) same size and a minus 45-degree rotation, (3) same size and a plus a 45-degree rotation, (4) doubled, (5) doubled and minus a 45-degree rotation, (6) doubled and plus a 45-degree rotation, and (7) a shrinking path, and each accumulation point has four values as shown in Figure 7 and 8. If these four-point (m, n) , $(m-1, n)$, $(m, n-1)$ and $(m-1, n-1)$ make a quadrangle like Figure 4, we need to check 165 patterns that are derived from local accumulation paths above. This checking procedure takes long time because of the recalculating operations. Therefore, we set four values for accumulating calculation dx , dx , dy and dy as Figure 8 to overload low-level accumulation results and keep the path constraints. Next, we set the path weight as Figure 5 to simplify the algorithm. Then, any path weight value will be set as $w(i, j, m, n) = 1$.

The algorithm for accumulation local minimum is shown as following equations:

For $l = m + n, 2 \leq l \leq M + N, l = l + 1$

For $m = 1$ and $n = l, 1 \leq m \leq M$ and $1 \leq n \leq N, m = m + 1$ and $n = n - 1$

Path selection:

$$(i', j', m-1, n) \triangleq \arg \min \left\{ \begin{array}{l} D(i-1, j, m-1, n) - dyx(i-1, j, m-1, n) \\ D(i-1, j-1, m-1, n) - dyx(i-1, j-1, m-1, n) \\ D(i-1, j+1, m-1, n) - dyx(i-1, j+1, m-1, n) \\ D(i-2, j, m-1, n) - dyx(i-2, j, m-1, n) \\ D(i-2, j-1, m-1, n) - dyx(i-2, j-1, m-1, n) \\ D(i-2, j+1, m-1, n) - dyx(i-2, j+1, m-1, n) \\ D(i, j, m-1, n) - dyx(i, j, m-1, n) \end{array} \right\}, \quad (9)$$

$$(i'', j'', m, n-1) \triangleq \arg \min \left\{ \begin{array}{l} D(i, j-1, m, n-1) - dxy(i, j-1, m, n-1) \\ D(i-1, j-1, m, n-1) - dxy(i-1, j-1, m, n-1) \\ D(i+1, j-1, m, n-1) - dxy(i+1, j-1, m, n-1) \\ D(i, j-2, m, n-1) - dxy(i, j-2, m, n-1) \\ D(i-1, j-2, m, n-1) - dxy(i-1, j-2, m, n-1) \\ D(i+1, j-2, m, n-1) - dxy(i+1, j-2, m, n-1) \\ D(i, j, m, n-1) - dxy(i, j, m, n-1) \end{array} \right\}, \quad (10)$$

Accumulation four values:

$$dx(i, j, m, n) \triangleq d(i, j, m, n) + dx(i', j', m-1, n) \quad (11)$$

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

$$dyx(i, j, m, n) \triangleq dyx(i'', j'', m, n-1) + dx(i'', j'', m, n-1) \quad (13)$$

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

Accumulation local minimum value:

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

Fig. 9. Spotting point and spotting area candidate: Candidate spotting area in input image is about 12 times larger than reference image

The equations (9)-(15) imply that an accumulated value $D(i, j, m, n)$ is recursively calculated by $D(i', j', m-1, n)$ and $D(i'', j'', m, n-1)$ as a result of the application of DP. The path configuration in Figure 5 enables infinite path shrinking. Therefore, in our experiment, we counted the number of times of shrinking and set the limitation for the consecutive times of shrinking.

Finally, the optimal spotting point corresponding to pixel (i, j) in the input image is given by:

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

This equation follows the equation (5).

2.5 Correction of Mesh Structure Using Back Tracking

After the spotting point has been determined, we need to extract spotting area from four-dimensional accumulated local minimum space. The back tracking is used to optimally accumulate the local distance. The back tracking in CDP traces only the connected path. However, the connected path in 2DCDP sometimes twists between m and n directions. Therefore, each matching point $D(i, j, m, n)$ has an optimal accumulated value from start to that point. Thus, the algorithm for finding the optimal path from two points is expressed as following equation:

$$(i^*, j^*) \in K(\xi^*(m+1, n), \eta^*(m+1, n)) \otimes L(\xi^*(m, n+1), \eta^*(m, n+1)) \tag{17}$$

$$(\xi^*(m, n), \eta^*(m, n)) = \min_{i^*, j^*} \{D(i^*, j^*, m, n)\}. \tag{18}$$

Candidate spotting area in input image is about 12 times larger than reference image (Figure 9) because of the implementation allows 45-degrees rotation and doubled size each connected paths. The problem of the back tracking is: it is able to unlimitedly select a shrinking path that can cause over-shrinking spotting area. Then, we implement a control variable to limit the consecutive times of shrinking.

Finally, set P , containing segments is defined as:

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

In addition, when recognizing multiple segments, back tracking from the minimum order of spotting points, and sometimes skipping the trial if the trial contains the element of earlier segments, is exclusive.

2.6 Arbitrary Shape of Reference Image

When the value of local distance of pixels in discarded area is set to maximum value, the local distance value of pixels in background will have the same value. Therefore, the arbitrary shape can be cut off from the reference image as shown in Figure 10. In our experiment, local distance is set to 1.

Fig. 10. Image of arbitrary reference image matching

2.7 Calculation Time and Memory Amount

2DCDP takes one time to calculate local distance and accumulation at every element in four-dimensional tensor field. Thus, the time needed 2DCDP calculation is $O(N^4)$ because the amount of element in the tensor field is $I \times J \times M \times N$. In this algorithm, back tracking needs each value of accumulated local minimum $D(i, j, m, n)$. Therefore, the amount of memory size is also $O(N^4)$.

3 Spotting Recognition Experiments

3.1 Methods and Materials

To experimentally evaluate our optimal pixel matching method, we used only a single OS-implemented thread (MacOS X running on an Xserve with a dual 2.8-GHz Xeon processor 4-cores and 32-GB SDRAM). In the first experiment, spotting recognition was done using an arbitrary shaped query extracted from the original image (image on the top of Figure 11). In the second experiment, multi-answer spotting recognition was done using real world data. The third experiment is used to apply 2DCDP for nature images.

Fig. 11. Experimental result of 2DCDP; Exp. 1: vertically divided affine image. Exp. 2: horizontally divided affine image. Exp. 3: thick lens distortion; applied transform $a_1r - a_2r^3$, $a_1 = 0.3$, $a_2 = 0.0001$. Exp. 4: thick lens distortion; applied transform $a_1r - a_2r^3$, $a_1 = 0.6$, $a_2 = 0.0005$.

Exp. 1: Spotting recognition used four input images (Figure 11) as follows:

Input 1: Spotting recognition is done by using affine transformed image in top and bottom halves of image extracted from original image.

Input 2: Spotting recognition is done by using affine transformed image in left and right halves of image extracted from original image.

Input 3: Spotting recognition is done by using a distorted image such as one captured through a thick lens, as expressed by $a_1r - a_2r^3$, $a_1 = 0.3$, $a_2 = 0.0001$, extracted from original image.

Input 4: Spotting recognition is done by using a distorted image such as one captured through a thick lens, as expressed by $a_1r - a_2r^3$, $a_1 = 0.6$, $a_2 = 0.0005$, extracted from original image.

Exp. 2: Spotting recognition is done by using input image constructed by several images picked up from a movie and another picture ($I = 320$, $J = 240$) and reference image bring another frame into the movie ($M = 63$, $N = 61$).

Exp. 3: Some nature image spotting using 2DCDP as Figure 13 and Figure 14.

In experiments 1, we used 100×100 pixels image for input and 55×55 pixels image for reference. In experiment 2, we used several frames from a video database [23] and cut and pasted other face-image frames that had several margins. In these experiments, the limitation of shrinking was set to 2.

3.2 Experimental Results

For experiments 1, Figure 11 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

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 (sec)	4.931	4.954	4.954	4.855

Fig. 12. Exp. 2: multi-extraction using face query. Eight objects extract from input image. Each face pick up from other frame into a movie. Reference image is also pickup from other frame into same movie.

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

	Fig. 12	Fig. 13 Ref. 1	Fig. 13 - Ref. 2	Fig 14 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

Fig. 13. Multi-object extraction using flower query: Program tries to extract 9 objects. Two different query can spotting 7 or 8 objects because of color difference between references. Red circle pointed area of matching miss.

area and that each area had a pixel-to-pixel relationship between the subject and reference images. Figure 12 shows that every result was success to indicate the borderline between hair, face, eyes, nose and mouth. The calculation time and memory usage is shown in Table 2. In the experiment 3, Figure 13 also shows that 2DCDP is able to extract multiple spotting area and capture different results using two different reference images because these two reference images have difference in color and shape. In the Figure 13, we obtain 9 objects in each trial and some spotting errors are indicated in red circle in the figure. Figure 14 indicates that 2DCDP is able to extract the object which has perspective transformation. This is a special feature for image based modeling because this full-pixel matching easily allows the reconstruction of 3D shape from two or more images.

Fig. 14. Extracting building from different frame into motion picture

4 Conclusion

We developed and tested a two-dimensional continuous dynamic programming (2DCDP) method for spotting recognition of images. It achieves simultaneous segmentation and image recognition due to continuous and monotonic pixel-to-pixel matching. Our testing demonstrated that it is robust against non-linear deformation of images. Our future work would enable this method to use other indicators instead of only RGB in above experiments. We also includes investigating applications of 2DCDP such as finding errors in medical images from non-linear image registration, three-dimensional reconstruction, and recognition of facial expressions in our future plan.

References

1. Forsyth, D., Ponce, J.: Computer Vision: A Modern Approach. Prentice Hall Professional Technical Reference, Englewood Cliffs (2002)
2. Brunelli, R., Poggio, T.: Face recognition: features versus templates. IEEE Trans. on PAMI 15(10), 1042–1052 (1993)
3. Geiger, D., Gupta, A., Costa, L., Vlontzos, J.: Dynamic programming for detecting, tracking, and matching deformable contours. IEEE Trans. on PAMI 17(3), 294–302 (1995)
4. Zitová, B., Flusser, J.: Image registration methods: a survey. Image and Vision Computing 21(11), 977–1000 (2003)
5. Pluim, J., Maintz, J., Viergever, M.: Mutual-information-based registration of medical images: a survey. IEEE Trans. on Medical Imaging 22(8), 986–1004 (2003)

6. Ohta, Y., Kanade, T.: Stereo by intra-and inter-scanline search. *IEEE Trans. on PAMI* 7(2), 139–154 (1985)
7. Okutomi, M., Kanade, T.: A multiple-baseline stereo. *IEEE Trans. on PAMI* 15(4), 353–363 (1993)
8. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. *IJCV* 9(2), 137–154 (1992)
9. Tomasi, C., Kanade, T.: Detection and tracking of point features. *School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91-132* (1991)
10. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. *IJCV* 60(2), 91–110 (2004)
11. Hashizume, C.: Vinod, V.V., Murase, H.: Robust object extraction from color images under illumination changes. Technical report of IEICE. PRMU 97(325), 33–40 (1997)
12. Pereira, S., Pun, T.: Robust template matching for affine resistant image watermarks. *IEEE Trans. on Image Processing* 9(6), 1123–1129 (2000)
13. Uchida, S.: Dp matching: Fundamentals and application. Technical report of IEICE. PRMU, pp. 31–36 (December 2006)
14. Nishimura, T., Oka, R.: Spotting Image Recognition using Two-Dimensional Continuous Dynamic Programming. Technical Report of IEICE. PRMU, 1–7 (July 1997)
15. Myers, C., Rabiner, L., Rosenberg, A.: Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Trans. on ASSP* 28(6), 623–635 (1980)
16. Uchida, S., Sakoe, H.: Piecewise linear two-dimensional warping. *Systems and Computers in Japan* 32(12), 1–9 (2001)
17. Oka, R.: Spotting method for classification of real world data. *The Computer Journal* 41(8), 559–565 (1998)
18. Yaguchi, Y., Watanabe, Y., Naruse, K., Oka, R.: Speech and Song Search on the Web: System Design and Implementation. In: *CIT 2007*, pp. 270–278 (2007)
19. Oka, R., Nishimura, T., Yabe, H.: On Spotting Recognition of Gesture Motion from Time-varying Image. *Trans. of IPSJ* 43, 54–68 (2002)
20. Suto, N., Nishimura, T., Fujii, R.H., Oka, R.: Spotting Recognition of Concave and Convex Reference Image with Pixel-wise Correspondence using Two-dimensional Continuous Dynamic Programming. In: Technical report of IEICE. PRMU (July 2003)
21. Iwasa, Y., Oka, R.: Algorithm for Guaranteeing Monotonuous Contiguity of Pixel Correspondence in Spotting Recognition of Image. In: *MIRU 2005*, pp. 997–1004 (July 2005)
22. Yaguchi, Y., Iseki, K., Oka, R.: Two-dimensional Continuous Dynamic Programming for Spotting Recognition of Image. In: *MIRU 2008*, pp. 708–714 (July 2008)
23. Babaguchi, N., Etoh, M., Satoh, S., Adachi, J., Akutsu, A., Arika, Y., Echigo, T., Shibata, M., Zen, H., Nakamura, Y., Minoh, M.: Video Database for Evaluating Video Processing. In: Technical Report of IEICE. PRMU (June 2002)