

Article

# Adaptive Image Matching Using Discrimination of Deformable Objects

Insu Won <sup>1</sup>, Jaehyup Jeong <sup>1</sup>, Hunjun Yang <sup>1</sup>, Jangwoo Kwon <sup>2,\*</sup> and Dongseok Jeong <sup>1,\*</sup>

<sup>1</sup> Department of Electronic Engineering, Inha University, 22212 Incheon, Korea; is.won@inha.ac.kr (I.W.); jaehyup@inha.edu (J.J.); pinion@inha.edu (H.Y.)

<sup>2</sup> Department of Computer Science and Information Engineering, Inha University, 22212 Incheon, Korea

\* Correspondence: jwkwon@inha.ac.kr (J.K.); dsjeong@inha.ac.kr (D.J.); Tel.: +82-32-860-7443 (J.K.); +82-32-860-7415 (D.J.)

Academic Editors: Ka Lok Man, Yo-Sub Han and Hai-Ning Liang

Received: 20 April 2016; Accepted: 13 July 2016; Published: 21 July 2016

**Abstract:** We propose an efficient image-matching method for deformable-object image matching using discrimination of deformable objects and geometric similarity clustering between feature-matching pairs. A deformable transformation maintains a particular form in the whole image, despite local and irregular deformations. Therefore, the matching information is statistically analyzed to calculate the possibility of deformable transformations, and the images can be identified using the proposed method. In addition, a method for matching deformable object images is proposed, which clusters matching pairs with similar types of geometric deformations. Discrimination of deformable images showed about 90% accuracy, and the proposed deformable image-matching method showed an average 89% success rate and 91% accuracy with various transformations. Therefore, the proposed method robustly matches images, even with various kinds of deformation that can occur in them.

**Keywords:** image matching; discrimination of deformable object; matching-pair clustering

---

## 1. Introduction

Computer vision lets a machine or computer see and understand objects, just like human vision. The purpose of computer vision is to recognize object in an image and/or to understand the relationships between objects. While recent research has focused on recognition based on big data and deep learning (DL) [1], traditional computer vision methods are still widely used in some specific areas. While DL is not yet used in many applications due to the requirement for high computing power and big data, traditional research based on hand-craft techniques, like feature detection and feature matching, are actively used in various applications, such as machine vision, image stitching, object tracking, and augmented reality. Image matching is matching similar images or objects, even under geometric transformations, such as translation; optical, scale, and rotation transformations; affine transformations, which are complex transformations; and viewpoint change. The new current challenge is image matching of deformable objects [2]. In the real world, deformable objects encompass the majority of all objects. In particular, research related to fashion items, such as clothes, is conducted because of the rapid growth in Internet shopping [3]. However, since deformable-object matching has a different target than the existing image-matching methods, feature-detection and -matching methods are not identical. As such, much research has been conducted into achieving both objectives at the same time, but without substantial results. Since deformable models cannot be defined by a specific transformation model, there are numerous difficulties in such research. Early deformable matching methods were researched for augmented reality, remote exploration, and image registration for medical images, and it was not until recently that research on deformable-object matching appeared.

A good matching algorithm is characterized by robustness, independence, and a fast matching speed [4]. Robustness is recognizing that two images are identical if they have exactly the same objects in them. However, the algorithm must recognize identical objects even under transformation. Independence is recognizing the differences between two images containing objects that are different. Lastly, fast matching is the property of rapidly determining a match. Without fast matching, an algorithm cannot process many images, and hence, cannot be a good algorithm. The biggest disadvantage to previous deformable object-matching algorithms is slow matching. Therefore, in this paper, as a solution to the problems of the aforementioned methods (and by considering these characteristics), we propose an efficient algorithm that operates the same way for both rigid and deformable objects.

The rest of this paper is composed as follows. Section 2 introduces the existing image-matching methods. Section 3 introduces the proposed algorithm, followed by Section 4, where the experimental results from image sets with various deformable objects are examined and analyzed. Finally, Section 5 evaluates the proposed algorithm, and concludes the paper.

## 2. Related Works

In this section, we introduce the known feature-matching methods for computer vision. Image-matching methods can be largely classified into matching rigid objects and matching deformable objects. Rigid object-matching methods mostly consist of those that examine geometric relationships based on feature correspondence, and that show good performance under transformations like viewpoint change and affine transformation. However, matching performance degrades when deformable objects are the target. For deformable object-matching, various methods are used depending on the specific image set. In other words, there is no generalized procedure. Moreover, there is the common problem of generally slow execution time. We introduce three categories of common feature-point matching methods, which are classified in terms of methodology [5].

### 2.1. Neighborhood-Based Matching

Early researchers used neighbor-pixel information around the points to find feature correspondence. Neighborhood-based methods include threshold-based, nearest neighbor (NN), and nearest neighbor distance ratio (NNDR) [6]. In the threshold-based method, if the distance between the descriptors is below a predefined threshold value, the features are considered to be matched. The problem in this method is that a single point in the source image may have several matched points in the target image. In the NN approach, two points are matched if their descriptor is in the nearest neighborhood and the distance is below a specific threshold. Therefore, the problem with the threshold-based method is solved. However, this method may result in many false positives between images, known as outliers. This is overcome by random sample consensus (RANSAC), which was proposed by Fischler and Bolles [7]. RANSAC is a good outlier-removal method, and is widely used for object detection and feature matching.

### 2.2. Statistical-Based Matching

Logarithmic distance ratio (LDR), proposed by Tsai et al. [8], is a technique for fast image indexing, which calculates the similarities in feature properties, including scale, orientation and position, and recognizes similar images based on lower similarity distance. This technique is effectively used to rapidly search similar image candidates in large image datasets. Lepsøy et al. proposed the distance ratio statistic (DISTRAT) method [9], which adopts LDR for image matching. They found that the LDR of inlier matching pairs has a specific ratio when LDR is calculated using feature coordinates only. Therefore, the final matching decision is based on a statistical analysis whereby the LDR histogram is narrower with more inliers. This method has the advantage of performance equivalent to RANSAC while having a faster matching speed, and was eventually selected as the standard in Moving Picture Expert Group (MPEG)—Compact Descriptors for Visual Search (CDVS) [10].

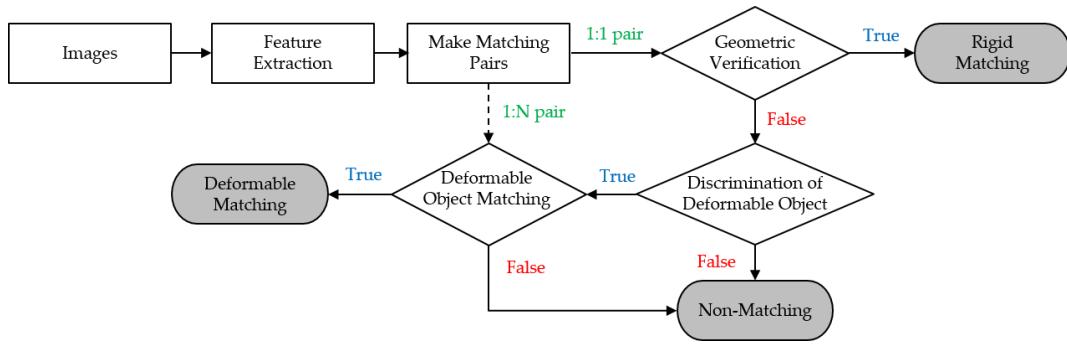
### 2.3. Deformable Object-Based Matching

The aforementioned matching algorithms are optimized for rigid objects. However, most real-world objects are deformable. Previously proposed feature-based deformable object-matching methods include transformation model-based [11], mesh-based [12], feature clustering [13], and graph-matching [14] methods. Transformation model-based methods require high complexity, because they operate under a pixel-based model. Therefore, feature-based methods that are not pixel-based were proposed. Agglomerative correspondence clustering (ACC), proposed by Cho and Lee [15], is a method that calculates the correspondence between each matching pair, and clusters matching pairs with a similar feature correspondence. ACC defines feature correspondence as the difference between the points calculated using the homograph model of matching pairs and the point that was actually matched. Then, matching pairs with a small difference are considered to have a locally similar homography, and are hierarchically clustered. Unlike previous methods, ACC will cluster matching pairs with a similar homograph by calculating the geometric similarity between each matching pair, rather than classifying matching pairs into inliers and outliers. While ACC has the disadvantages of appreciably high complexity and a high false positive rate, it shows the best performance when only the true rate is considered. As such, Improved ACC was introduced as an enhanced version of ACC [16].

## 3. Proposed Algorithm

RANSAC shows robust performance against geometric transformation of rigid objects. However, it does not offer good performance when matching deformable objects. Meanwhile, deformable object-matching methods generally have high complexity, which presents difficulty in applications that require fast matching. The easiest solution to this issue is to first perform rigid object-matching and match the remaining non-matched points using deformable object-matching methods. However, this is an inefficient solution. Therefore, it is more effective to selectively adopt the matching method, as long as it can discriminate deformable objects. As a solution, we propose discrimination of deformable transformations based on statistical analysis of matching pairs, and the subsequent use of the corresponding matching method. For example, if there are no inliers at all from among numerous matching pairs, then these are likely to be non-matching pairs. Moreover, even if there are some inliers, they are unlikely to be a match if the inlier ratio is low. Since the inlier ratio is low for deformable objects, it is impossible to obtain good results. This paper proposes discrimination of possible deformable objects through statistical analysis of such matching information and supervised learning.

Figure 1 shows a diagrammatic representation of the proposed image-matching process. First of all, features are detected from the image, and candidate matching pairs are found. Final matching is examined through geometric verification. The algorithm is an adaptive matching method wherein the possibility of the candidate matching pair being a deformable object is examined through discrimination of the deformable object (using the proposed algorithm) within the candidate matching pairs that do not satisfy geometric verification. Deformable object-matching is only performed on matching pairs that are determined to be deformable objects. If the algorithm cannot discriminate the deformable objects well, it is an inefficient algorithm. Therefore, how well the algorithm discriminates deformable objects significantly affects the overall performance.



**Figure 1.** Proposed method for image matching using discrimination of deformable object images.

### 3.1. Feature Detection, and Making a Matched Pair

The proposed method requires feature coordinates and properties for deformable-object matching. Therefore, a scale-invariant feature transform (SIFT) detector is used, which is a point-based detector that includes the feature's coordinates, scale, and dominant orientation [17]. Speeded up robust features (SURF), maximally stable extremal regions (MSER), and affine detectors, which have similar properties, can also be used. SIFT is a set of orientation histograms created on a  $4 \times 4$ -pixel neighborhood with eight bins each. The descriptor then becomes a vector of all the values of these histograms. Since there are  $4 \times 4 = 16$  histograms, each with eight bins, the vector has 128 elements. The features extracted from each image are  $F(i) = \{c_i, s_i, o_i, desc_i\}$  ( $i = 0 \sim N$ ) where  $c_i$  is a coordinate,  $s_i$  is the scale,  $o_i$  is the dominant orientation, and  $desc_i$  denotes 128-dimensional descriptors. To form matching pairs, the features detected from each image are compared. The metric used for the comparison is Euclidean distance, which is given by Equation (1):

$$d(f_{R(i)}, f_{Q(j)}) = \sqrt{\sum_{k=1}^{128} (f_{R(i)}^k - f_{Q(j)}^k)^2} \quad (1)$$

which obtains the Euclidean distance between the  $i$ -th feature vector of the reference image,  $f_{R(i)}$ , and the  $j$ -th feature vector of the query image,  $f_{Q(j)}$ . The simplest feature matching sets a threshold (maximum distance) and returns all matches from other images within this threshold. However, the problem with using a fixed threshold is that it is difficult to set; the useful range of thresholds can vary greatly as we move to different parts of the feature space [18]. Accordingly, we used NNDR for feature matching [17] as follows:

$$NNDR = \frac{d_1}{d_2} = \frac{\|D_A - D_B\|}{\|D_A - D_C\|} \quad (2)$$

where,  $d_1$  and  $d_2$  are the nearest and second nearest neighbor distances,  $D_A$  is the target descriptor,  $D_B$  and  $D_C$  are its closest two neighbors, and the symbol  $\|\bullet\|$  denotes the Euclidean distance. Lowe demonstrated that the probability of a false match (e.g., a feature with a similar pattern) significantly increases when  $NNDR > 0.8$  [17]. Thus, matching pairs with an NNDR higher than 0.8 are not employed. Numerous studies showed that forming 1:1 matching pairs using NNDR leads to the best performance. However, matching for deformable objects the single matching-pair can be outliers, which would disrupt performance. Therefore, considering deformable-object matching, up to  $k$  candidates are selected in decreasing order of ratio, rather than selecting a single candidate with NNDR. A feature point forms 1: $k$  matching pairs using  $k$ -NNDR. For rigid-object matching, matching pairs with a  $k = 1$  are used, and in the deformable object-matching method,  $k = 2$  or 3 is used.

The  $N \times M$  matching pairs formed as such undergo an overlap check process, as follows:

$$ovlp[i, j] = \begin{cases} 1 & \text{if } M_i(p_i) = M_j(p_j), \\ 0 & \text{otherwise.} \end{cases} \quad (0 \leq i, j \leq NM) \quad (3)$$

Matching pairs ( $M_k$ ) consist of each of the feature points from the reference and query images. In Equation (3),  $M_k(p_k)$  denotes the positions of the two feature points. Here,  $p_k = [p_k^R, p_k^Q]$ , where  $p_k^R$  is the position of the feature point extracted from the reference image, and  $p_k^Q$  is the position of the feature point extracted from the query image. If  $p_i^R$  is equal to  $p_j^R$ , or if  $p_i^Q$  is equal to  $p_j^Q$  when comparing the  $i$ -th matching pair ( $M_i$ ) with the  $j$ -th matching pair ( $M_j$ ), it is recognized as a repetition, and 1 is assigned to  $ovlp[i, j]$ . In this manner, 1 or 0 is assigned to all  $ovlp[i, j]$ , eventually generating an  $NM \times NM$  overlap matrix, which has  $ovlp[i, j]$  as elements. The generated overlap matrix is used for clustering during deformable-object matching.

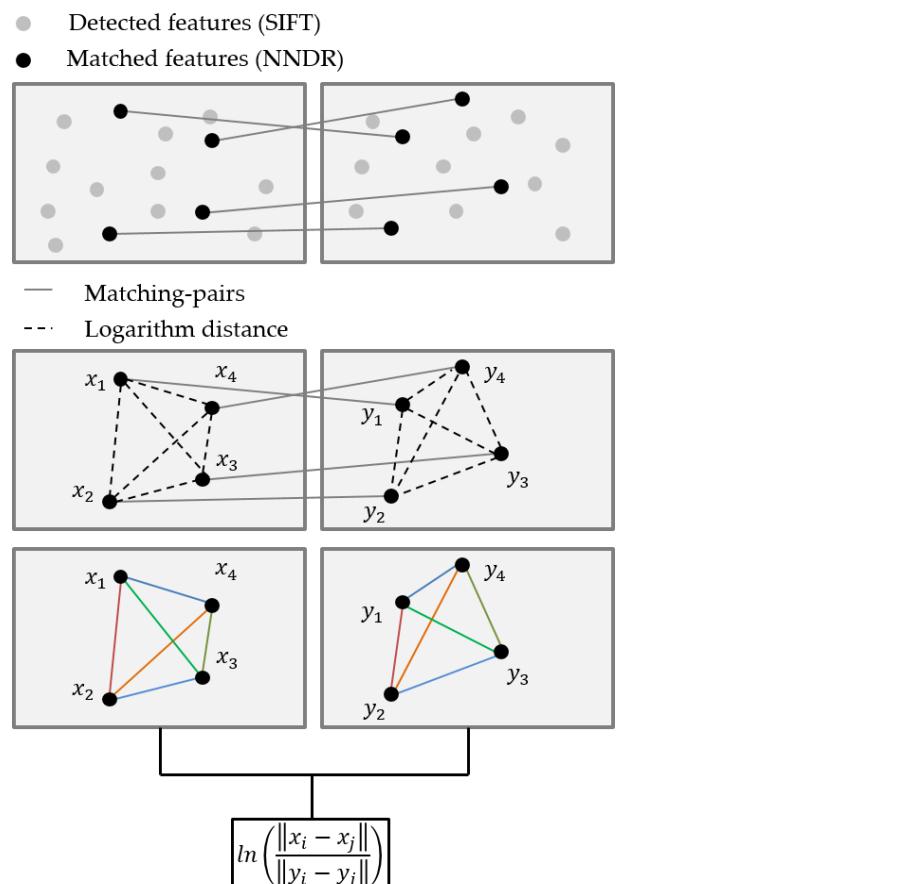
### 3.2. Geometric Verification for Rigid Object-Matching

After forming matching pairs, image matching is determined through geometric verification of 1:1 matching pairs. Letting the matching pairs be  $(x_1, y_1), \dots, (x_n, y_n)$ , the LDR set  $Z$  is obtained with the following equation:

$$z_{ij} = \ln \left( \frac{\|x_i - x_j\|}{\|y_i - y_j\|} \right), \quad Z = \{z_{ij} | i \neq j\} \quad (4)$$

where  $x_{i,j}$  is a coordinate of the reference image, and  $y_{i,j}$  and  $x_{i,j}$  are the matched-feature coordinates.

Moreover,  $Z$  denotes the LDR sets of all matching pairs. A schematic of this process is in Figure 2.



**Figure 2.** LDR histogram calculation process.

Let all features within one image follow the same bivariate normal distribution, with variance  $\sigma_x^2$  in the query image and  $\sigma_y^2$  in the reference image. Let  $a^2$  be the proportion of the variances:

$$\frac{\sigma_x^2}{\sigma_y^2} = a^2 \quad (5)$$

Then the LDR has the following probability distribution function (PDF):

$$f_Z(z; a) = 2 \left( \frac{ae^z}{e^{2z} + a^2} \right)^2 \quad (6)$$

In addition, we examine the upper and lower bounds of the LDR for objects of this type. This behavior is studied by first forming LDR histogram  $h(k)$  for the matching pairs by counting the occurrences over each bin:

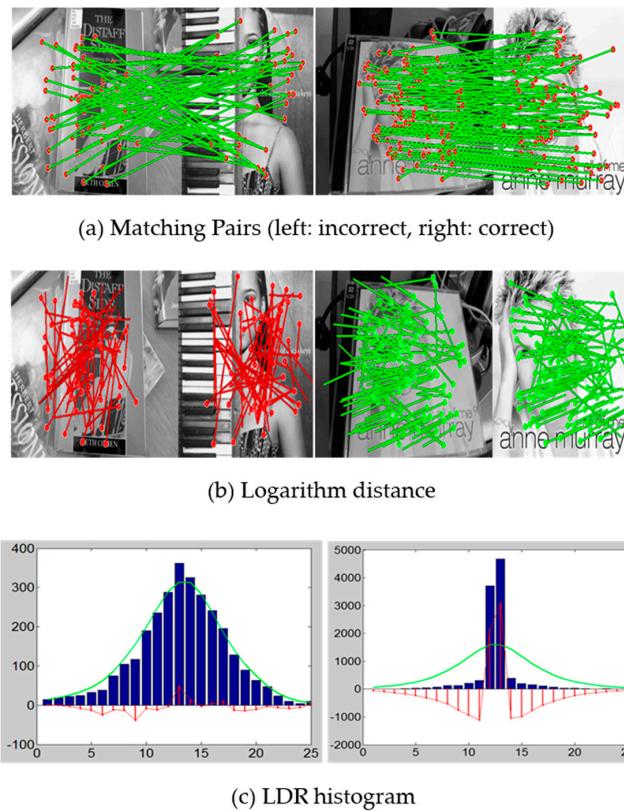
$$h(k) = \#(Z \cap \zeta_k) \quad (7)$$

The bins  $\zeta_1, \dots, \zeta_K$  are adjacent intervals. The inlier behavior can be expressed by the following double inequality:

$$a||x_i - y_i|| \leq ||y_i - y_j|| \leq b||x_i - x_j|| \quad (8)$$

where  $a$  and  $b$  define the boundaries of the LDR for inliers. The LDR is restricted to an interval. The inliers would contribute to bins contained in  $(-\ln b, -\ln a)$  which for most cases is a limited portion of the histogram. We used the interval  $(a, b)$ , which is  $(-2.6, 2.6)$ .

The comparison of LDR histograms between incorrectly matched images and correctly matched images is presented in Figure 3. Since a geometric transform occurs mostly in rigid object images, each matching-pair will have a similar logarithm distance and will become narrow. Exploiting this, the match between two images can be determined.



**Figure 3.** (a) Results of feature matching using NNDR (**left**: incorrectly matched images, **right**: correctly matched images); (b) the lines point to the positions of the matched features in the other image; and (c) the LDR histogram of the model function.

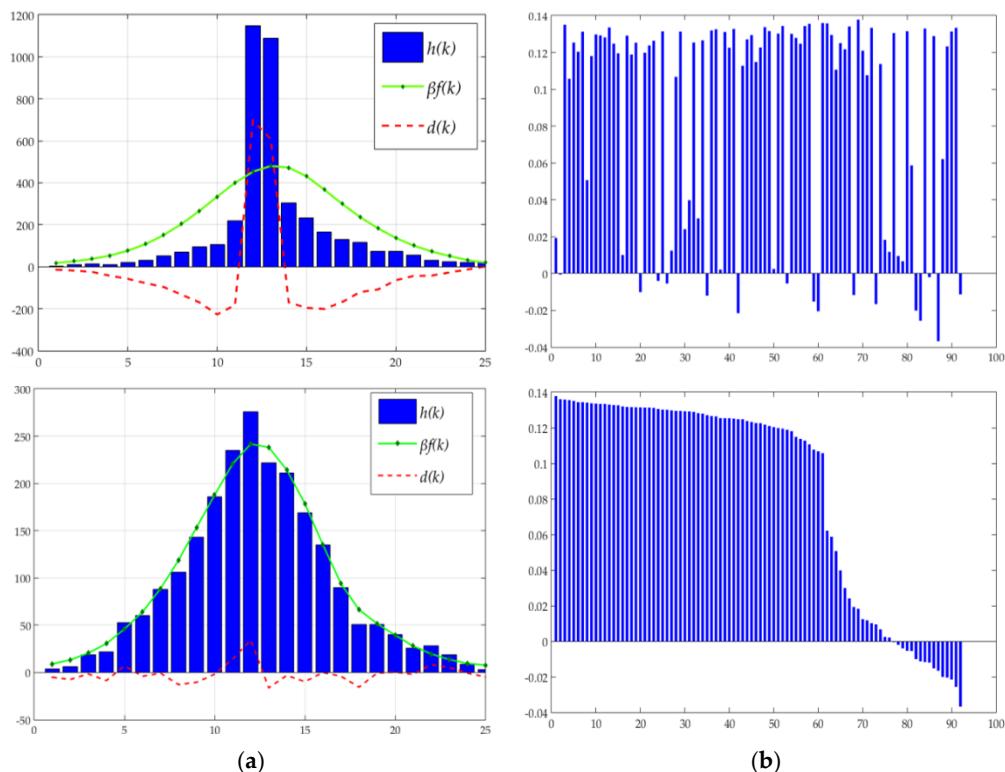
Pearson's chi-square test is utilized to compare the correlation of  $h(k)$  and  $f(k)$ . Let the LDR histogram have  $k$  bins. The histogram will be compared to the discretized model function  $f(k)$ . We used these quantities to formulate the test. Equation (9) is a formula to calculate the goodness-of-fit parameter for  $c$ . A high value for  $c$  if the shape of the LDR histogram differs much from that of  $f(k)$ , implied that many of the matches are inliers:

$$c = \sum_{k=1}^K \frac{(h_k - n f_k)^2}{n f_k} \geq \chi_{1-a, K-1}^2 \quad (9)$$

where  $n$  is the total number of matching pairs, and  $\chi_{1-a, K-1}^2$  is the threshold of  $\chi^2$  having  $K-1$  degrees of freedom. The threshold determination method is based on an experiment by Lepsøy [9]. To obtain the false positive rate below,  $1 - a = 0.01$  (1%), set the threshold to 70. If  $c$  is higher than the threshold,  $d(k)$  is computed using Equation (10) to find the inliers:

$$d(k) = h(k) - \beta f(k), \quad \beta = \frac{\sum_{k=1}^K h(k) f(k)}{\sum_{k=1}^K (f(k))^2} \quad (10)$$

where  $\beta$  is a weight to normalize  $h(k)$  and  $f(k)$ . Figure 4a shows the comparisons of  $h(k)$  to  $f(k)$  and to  $d(k)$ , respectively, between matching images and non-matching images. We can see that the LDR of matching pairs is narrow with numerous inliers, and that a match can rapidly be identified by calculating this difference.



**Figure 4.** Analysis for geometric verification based on LDR ( $k = 25$ ): (a) example of LDR histogram  $h(k)$ , model function  $f(k)$ , and difference  $d(k)$  (**top**: correctly matched images, **bottom**: incorrectly matched images); and (b) eigenvector  $r$  for finding inliers (**top**), and the results in descending order (**bottom**).

After geometric verification based on LDR, matrix  $D$  is formed by quantizing  $d(k)$  in order to predict the number of inliers, and eigenvalue  $u$  and dominant eigenvector  $r$  are calculated using  $Dr = ur$ , used in Equation (11).

$$inlier_n = 1 + \frac{\mu}{\max_{k=1,\dots,K} d(k)} \quad (11)$$

Subsequently, eigenvector  $r$  is sorted in descending order, and the upper-range matching pairs corresponding to the calculated number of inliers are finally determined to be the inliers. Figure 4b shows a diagrammatic representation of this process.

Lastly, the weights of the matching pairs and the inliers are calculated, and matches are determined by calculating the ratios of the weights. Weights are based on the conditional probability of the inliers when the ratio between the NN distance and second distance during feature matching is given. Equation (12) shows the corresponding conditional probability function.

$$p(c|r) = \frac{p(r|c)}{p(r|c) + p(r|\bar{c})} \quad (12)$$

where  $c$  and  $\bar{c}$  denote the probability of inliers and outliers, respectively, and  $r$  is the distance ratio.

### 3.3. Discrimination of Deformable Object Images

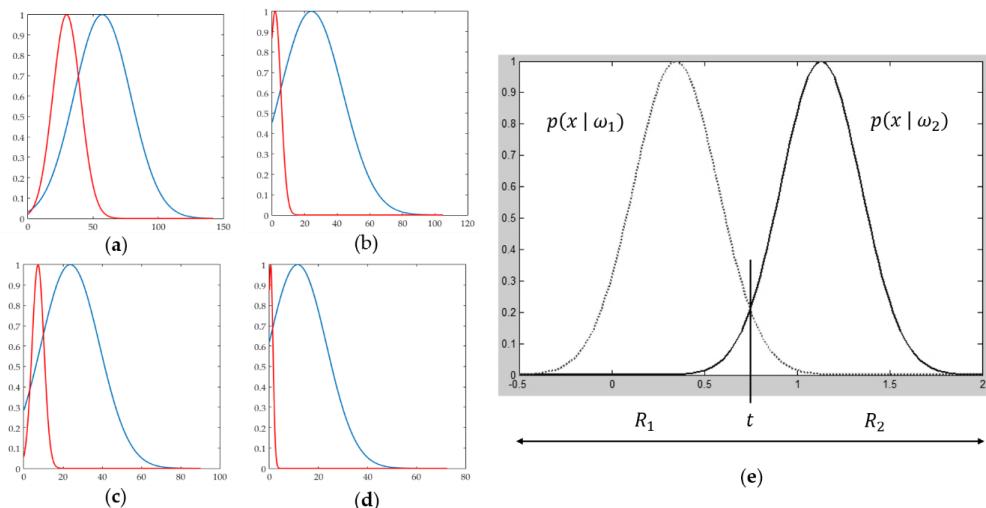
After generating matching pairs from the two images,  $x$  and  $y$ , and performing geometric verification, the total number of matching pairs, the number of inliers, the matching-pair weight, and the inliers' weights are calculated using the results. These can be defined as matching information  $M(x,y) = \{match_n, inlier_n, weight\}$ . We analyzed which statistical distribution  $M(x,y)$  has, using the example images. Non-matching and deformable-object images were used as training images, and the mean,  $\mu$ , and standard deviation,  $\sigma$ , of each property were calculated to form normal distribution  $N(\mu, \sigma^2)$ . However, since the non-matching and deformable-object images show similarity in  $M(x,y)$  from numerous aspects, discrimination of the two classes is vague. Therefore, the following is proposed to find the model that minimizes the error. Analyzing numerous training images, we observed that deformable-object matching does not lead to good results if the number of inliers is less than a certain number. In other words, there must be more than a certain number of inliers. Figure 5b was obtained from experiment using more than four inliers corresponding to minimum error. Subsequently, the ratio between the number of inliers with the greatest difference in mean between the two classes, and the matching-pair weight with the least difference in variance is calculated, and  $t$  with the minimum error is obtained using a Bayesian classifier.

Let the training set be  $X = (x_1, c_1), (x_2, c_2), \dots, (x_n, c_n)$ ,  $x_i$  for the  $i$ -th properties, and let  $c_i$  be the class index of  $x_i$ , which represents non-matching ( $w_1$ ) and deformable-matching ( $w_2$ ).  $X$  is defined as the ratio between the inlier number from matching information and matching-pair weight, as shown in Equation (13):

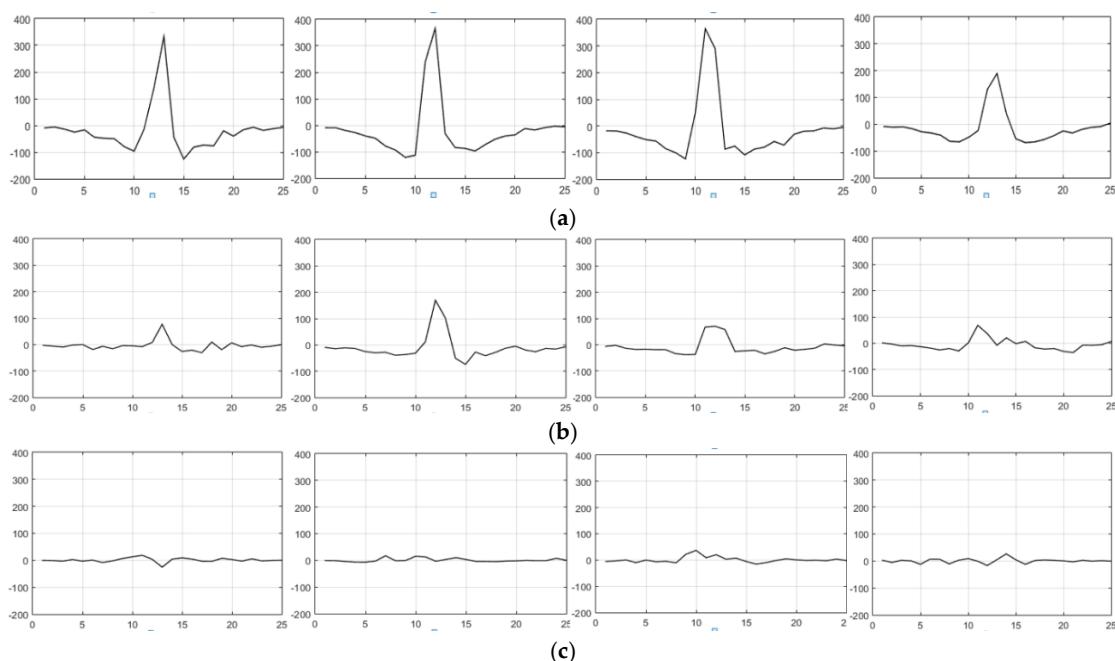
$$x = \frac{inlier_n}{dist_n} \quad (13)$$

Figure 5 shows the graph obtained by separating matching information into that of incorrectly matched images and deformable matching images. Figure 5a–d shows each property of the matching information; however, it is difficult to distinguish the deformable matching pairs from non-matching pairs. Figure 5e shows the normal distribution of  $x$  for the matching pairs with more than four inliers after rigid-object matching. We can see that the distinction is clearer than in the previous graphs. Therefore,  $t$  with the minimum error was found and is used for the classifier.

However, discrimination of deformable objects does not exhibit good performance if it is based on statistical analysis only. This is because a deformable transformation cannot be defined as a certain model. Therefore, the pattern for  $d(k)$ , which was used for geometric verification, was identified through machine learning-based training, and the result was used. Figure 6 shows a graphical illustration of  $d(k)$  from matching rigid-object, deformable-object, and non-matching-pair images.



**Figure 5.** Normal distribution model of matching information from matching pairs of deformable and non-matching images: (a) number of matching pairs; (b) number of inliers; (c) sum of all matching pairs' distances; (d) sum of inliers' distances; and (e) matching information of  $x$ .



**Figure 6.** Comparison of  $d(k)$  between matching images: (a)  $d(k)$  of rigid matching pair; (b)  $d(k)$  of deformable-object matching pair; and (c)  $d(k)$  of non-matching pair.

Figure 6c appears to be completely different from the case above. It exhibits irregular patterns over the entire range, most of which are small in size. This implies that the LDR and the PDF are almost identical. Exploiting this characteristic,  $d(k)$  of a deformable matching-pair and  $d(k)$  of a non-matching pair were extracted from the training set, and the characteristics of each pattern were classified using support vector machine (SVM) [19], which is one of the supervised learning algorithms.

The probability of a deformable transformation is determined from the results of the previous statistical analysis and machine learning, and the deformable objects are finally discriminated through voting. The voting method combines the results obtained from various conditions, rather than

from using the results of the one with the best classification performance, which is appropriate for unpredictable deformable transformations.

### 3.4. Deformable-Object Matching

Deformable-object images are discriminated, and finally, deformable object-matching is performed. Rigid object-matching methods mostly calculate a homograph in order to examine the geometric consistency of the features in the whole image. However, while deformable objects can have locally similar geometry, it is difficult to calculate a single homograph from the whole image. Therefore, we used a method to cluster matching pairs with similar geometric models, and we propose a method with enhanced performance through the use of clustering validation.

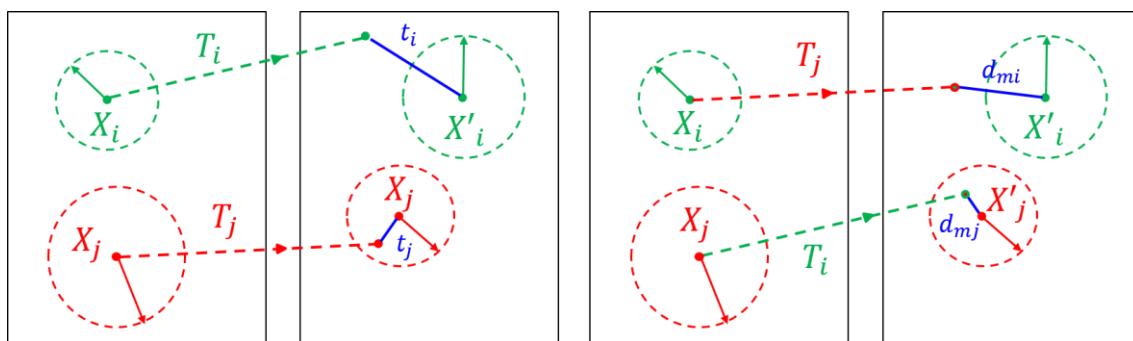
Letting two matching pairs be  $M_i$ , and  $M_j$ , transformations  $T_i$  and  $T_j$  and translations  $t_i$ , and  $t_j$  can be calculated from the characteristics of each matching pair, using the enhanced weak geometric consistency (WGC) [20], as shown in Equation (14).

$$\begin{bmatrix} x'_q \\ y'_q \end{bmatrix} = s' \begin{bmatrix} \cos\theta' & -\sin\theta' \\ \sin\theta' & \cos\theta' \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix}, \quad t = |q'(x_q, y_q) - q(x_q, y_q)| \quad (14)$$

where  $s$  denotes scale,  $\theta$  is the dominant orientation, and  $t_x$  and  $t_y$  represent coordinate translations. Using Equation (9), the matching pairs can be expressed as  $M_i = ((x_i, y_i), (x'_i, y'_i), T_i)$  and  $M_j = ((x_j, y_j), (x'_j, y'_j), T_j)$ , and the geometric similarity of the two matching pairs is calculated using Equation (15):

$$d_{geo} = (M_i, M_j) = \frac{1}{2} (\|X'_j - T_i X_j\| + \|X'_i - T_j X_i\|) = \frac{1}{2} (d_{mi} + d_{mj}) \quad (15)$$

where,  $X_k = [x_k, y_k]^t$ ,  $X'_k = [x'_k, y'_k]^t$ ,  $k = i, j$ . If transformation models  $T_i$  and  $T_j$  are similar,  $d_{geo}(M_i, M_j)$  will be close to 0. A graphical representation of this is shown in Figure 7.



**Figure 7.** Example of a geometric similarity measure.

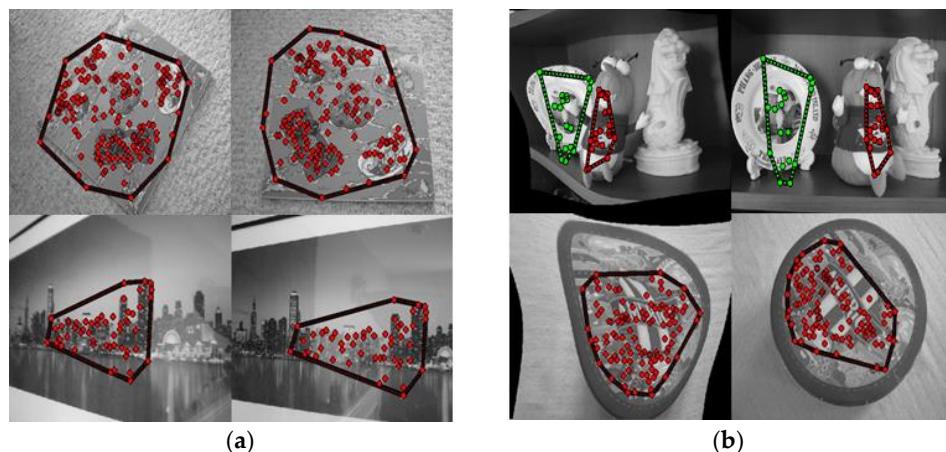
Using this relationship, it can be assumed that the matching pairs with a small  $d_{geo}(M_i, M_j)$  have a similar geometric relation. Therefore, similarity is computed by calculating the geometric transformation between each matching pair, rather than by defining a transformation model of the whole image.

Figure 8 shows an example of an affine matrix where the geometric similarity between the matching pairs is calculated. The matrix is formed by calculating the geometric similarity between each matching pair, assuming there are 10 matching pairs. In the example, the matching pairs with high geometric similarity are the 9th and 10th matching pairs. Deformable objects are found through hierarchical clustering of matching pairs with high similarity in the calculated affine matrix.

	1	2	3	4	5	6	7	8	9	10
1	0	20.0946	94.4845	8.8386	6.1508	6.0669	4.1845	71.1195	8.6838	6.6399
2	20.0946	0	86.6645	29.9761	34.2802	43.8197	24.0208	84.3051	45.7754	44.5732
3	94.4845	86.6645	0	69.7688	144.3691	106.6573	82.3050	23.9875	79.6483	85.3052
4	8.8386	29.9761	69.7688	0	9.6952	6.1954	7.4311	50.4994	5.8437	8.3202
5	6.1508	34.2802	144.3691	9.6952	0	10.3589	10.9666	104.7653	14.0340	16.5099
6	6.0669	43.8197	106.6573	6.1954	10.3589	0	8.0793	74.7833	8.0507	5.1691
7	4.1845	24.0208	82.3050	7.4311	10.9666	8.0793	0	62.8971	9.2976	8.3856
8	71.1195	84.3051	23.9875	50.4994	104.7653	74.7833	62.8971	0	58.6221	62.7673
9	8.6838	45.7754	79.6483	5.8437	14.0340	8.0507	9.2976	58.6221	0	1.8499
10	6.6399	44.5732	85.3052	8.3202	16.5099	5.1691	8.3856	62.7673	1.8499	0

**Figure 8.** Example of an affinity matrix (10 matching pairs).

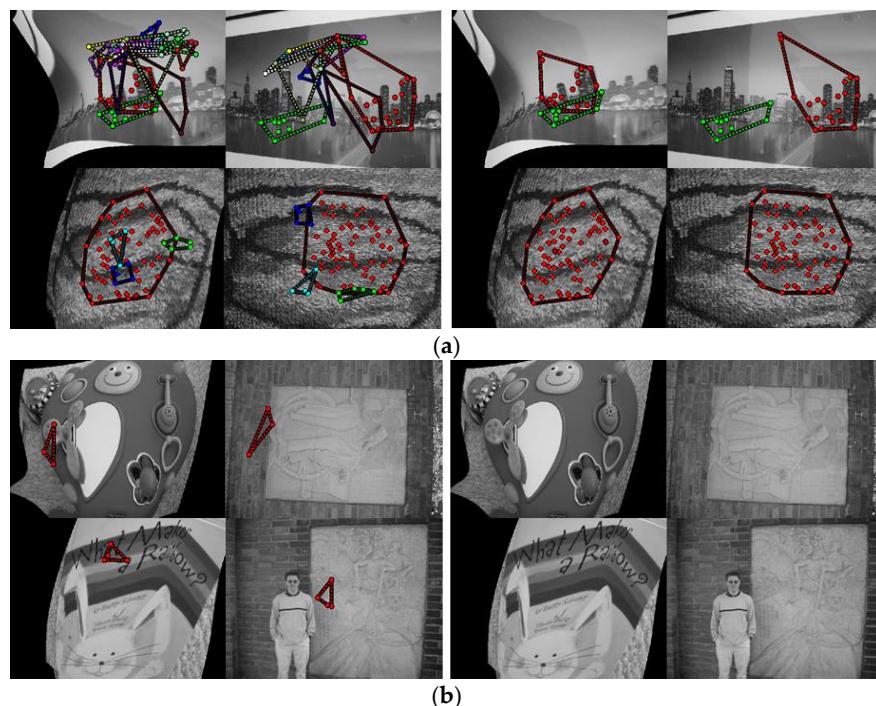
Hierarchical clustering will group the clusters through linkages by calculating the similarity within each cluster. In order to minimize the chain effect during linkage, a  $k$ -NN method was used. While this is similar to a single-linkage method, it was extended to have  $k$  linkages, and has the advantage of robustness against the chain effect. Figure 9 shows the results of hierarchical clustering based on geometric similarity between matching pairs. Figure 9a shows the matching result when there is a geometric transformation in an image containing a rigid object. We can see from the figure that the entire object is grouped into a single large cluster due to similar geometric relations. In Figure 9b, we can see that the objects are matched separately, with each object clustered distinctly under a deformable transformation.



**Figure 9.** Deformable image-matching results using hierarchical clustering of geometric similarity: (a) matching results for rigid objects with geometric transformation; and (b) matching results of a deformable object.

Hierarchical clustering eventually forms a single cluster. Therefore, clustering must be stopped at some point. During the linkage of clusters, clustering is stopped when the geometric similarity of the matching pair exceeds a set threshold. However, if such a thresholding method is the only one used, the number of clusters becomes excessive, with most of the clusters likely being false clusters. Therefore, clustering validation is used to remove false clusters. If the number of matching pairs that form the clusters is too low, it is less likely that the resulting clusters become objects. Hence, two methods are used for clustering validation. First, a cluster is determined to be a valid cluster only if the number of matching pairs that form the cluster is greater than  $\tau_m$ . Secondly, a cluster is determined to be a valid cluster if the area of the matching pairs that form the cluster is larger than a certain portion of the entire area ( $\tau_a$ ). The area of the matching pairs that form the cluster is calculated using a convex hull. Figure 10 shows the results of removing the invalid clusters through clustering validation for each case of inlier matching and outlier matching. From inlier matching, it was observed

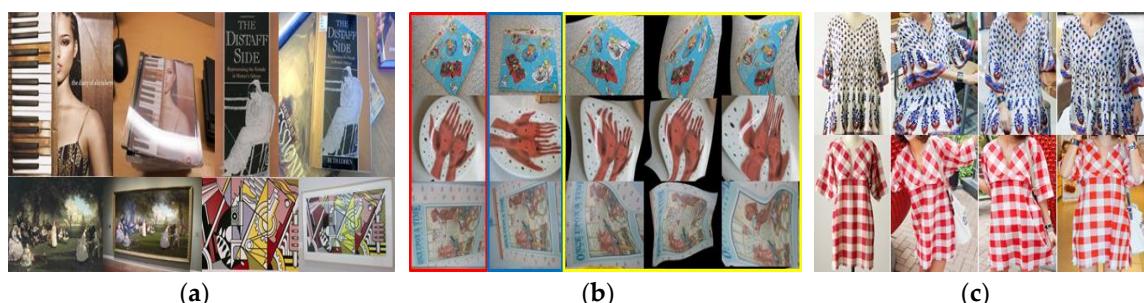
that the clusters with a small area are removed, and for outlier matching, the accuracy was enhanced by preventing the false positives that occur due to small clusters.



**Figure 10.** Comparison results from before (left) and after (right) clustering validation: (a) inlier-matching pairs; and (b) outlier-matching pairs.

#### 4. Experiment Results

In order to evaluate the performance of the proposed matching method, the Stanford Mobile Visual Search (SMVS) dataset from Stanford University was used [21]. SMVS includes images of CDs, DVDs, books, paintings, and video clips, and is currently the standard image set for performance evaluation of image matching under MPEG-7 CDVS. The annotations of matching pairs and non-matching pairs were compiled in SMVS, through which true positives and false positives can be evaluated. Additionally, in order to evaluate deformable object-matching, performance was evaluated with varying intensities of deformation using thin plate spline (TPS) [22]. Deformation intensity was varied in three levels (light, medium, and heavy), and 4200 query images from SMVS were extended to 12,800 images. Lastly, 600 clothing images, which are natural deformable objects without artificial deformation, were collected and used. Figure 11 shows examples of SMVS images, deformed SMVS images with each level of intensity, and clothing images.



**Figure 11.** Examples of images for the matching test: (a) SMVS datasets; (b) deformable transformation images (normal, light, medium, and heavy); and (c) clothing images.

True positive rate (TPR) and false positive rate (FPR) were used to evaluate matching performance. TPR is an equation that examines robustness from among the characteristics of matching algorithms, with greater value implying better performance. In contrast, FPR examines independence from among the characteristics of matching algorithms, with a smaller value implying better performance. Moreover, for objective comparison, accuracy was used, which is defined in relation to TPR and FPR.

$$\text{TPR} = \frac{TP}{TP + FN} = \frac{TP}{P}$$

$$\text{FPR} = \frac{FP}{FP + TN} = \frac{FP}{N}$$

$$\text{Accuracy} = \frac{TP + TN}{P + N}$$

For a speed performance test, we used an Intel Xeon E3-1275 (8 core) CPU and with a clock speed of 3.5 GHz and 32 GB RAM running the Windows 7 (64-bit).

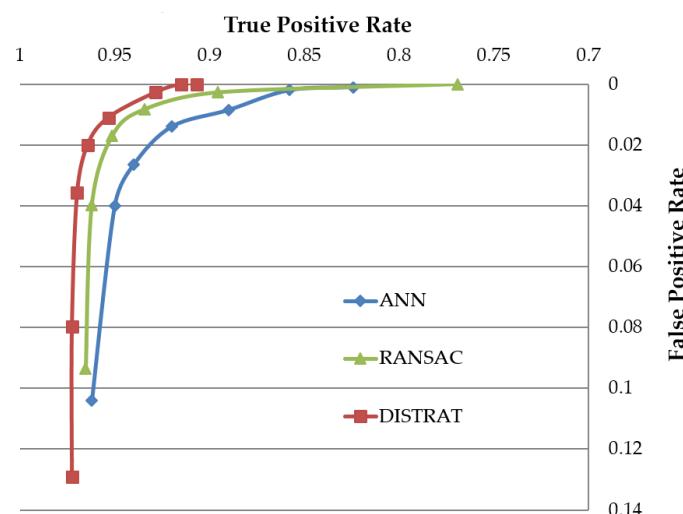
#### 4.1. Geometric Verification Test for Rigid Object Matching

First, an  $NNDR$  higher than 0.8 was excluded in feature matching, and a Euclidean distance less than 0.6 was not used for fast calculation. This method was verified by Mikolajczyk and has been applied in most feature-matching cases [18]. Inliers were determined through LDR, while the matching score for a matching decision was calculated as follows:

$$\text{matching score} = \frac{\omega}{(\omega + T_m)} \quad (16)$$

where,  $\omega$  is the sum of inlier distances, and  $T_m$  is the threshold value. Matching was determined for a matching score  $>0.5$ . Accordingly, the experiment was conducted by altering the  $T_m$  value. The optimum value for  $T_m$  was determined through the receiver operating characteristic (ROC) curve, and set at  $T_m = 3$ , allowing a comparison of the performances of standard matching methods. A rigid-object image from the SMVS dataset was employed as an experimental image.

Figure 12 illustrates the ROC curves of Approximate Nearest Neighbor (ANN) [23], RANSAC [7], and DISTRAT [9], which were employed in this study as rigid-object image-matching methods. A superior performance is indicated by a position closer to the top-left of the graph. DISTRAT and RANSAC exhibit similar performance in rigid-object matching. However, DISTRAT, which is based on a statistical algorithm, has the advantage of a very fast matching speed compared to RANSAC, which is based on an iteration algorithm.



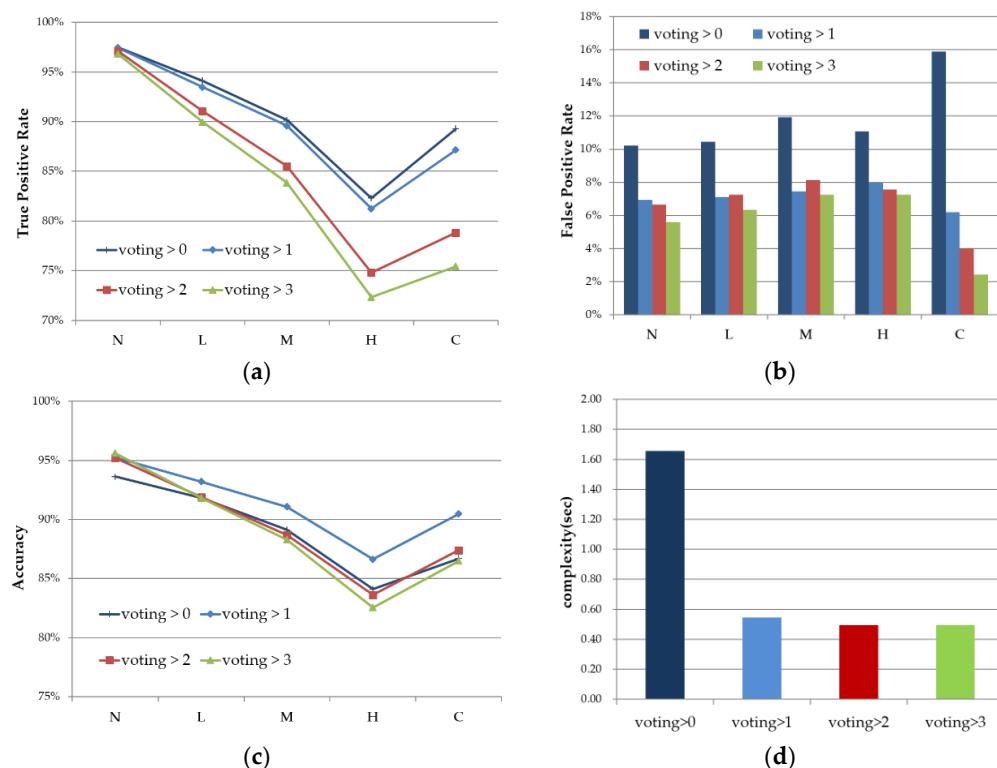
**Figure 12.** ROC curve comparison of the representative rigid object-matching methods.

#### 4.2. Discriminating Deformable Objects Using Voting Methods

The voting method was used for discrimination of deformable object images. Two additional methods were used as conditions in each voting, as well as SVM and the statistical model described in Section 3.3. The additional methods are the matching score used in rigid-object matching, and the sum of inliers distance. Matching score employs Equation (16). A low matching score indicates a small number of inliers or insufficient weight for an image-matching determination. However, it is necessary to slightly lower the matching parameter for deformable object images. Thus, a value that is lower than 0.5, which is used in rigid-object matching, was selected, and the probability of deformable object-matching was determined. The value was set at 0.2 and employed for the experiment. Second, the ratio of the total matching-pair weight and inlier weights was calculated. This value represents the proportion of inlier weights in the total matching pairs, regardless of the number of inliers. A value of 0.28 was selected through experiment as the method for determining matching-pairs having the probability of a deformable object.

Results of the proposed discrimination process for deformable objects are shown in Figure 13. The University of Kentucky (UKY) dataset was used as the training image set [23]. The UKY dataset consists of 10,200 images of 2550 objects.

Four conditions were used for voting, and the voting value was varied throughout the experiments, where  $voting > 0$  implies conducting deformable-object matching for all cases that have not been rigid-matched. In contrast,  $voting > 3$  indicates no deformable-object matching in most cases.

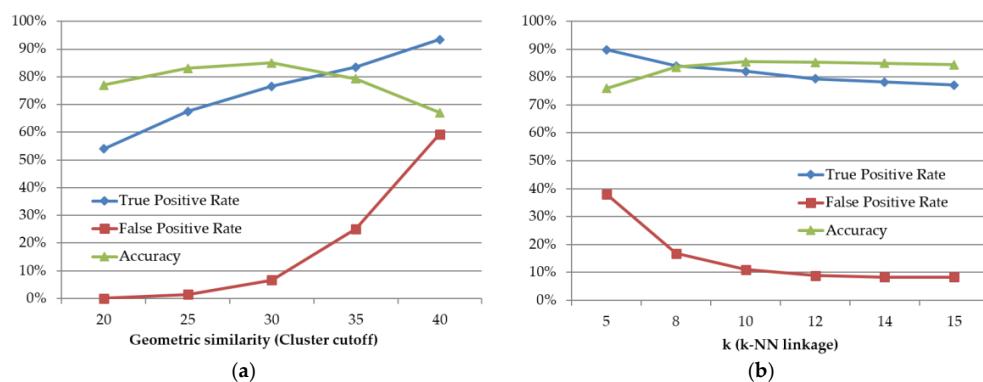


**Figure 13.** Performance per voting values (N: normal images; L: light deformable images; M: medium deformable images; H: heavy deformable images; and C: clothing images): (a) true positive rate for voting values; (b) false positive rate for voting values; (c) accuracy for voting values; and (d) execution time for voting values.

While TPR is highest  $voting > 0$ , the execution time is inefficient, the FPR is high, and accuracy is low, since all of the numerous deformable object-matching processes were run. If the voting value increases, the TPR and accuracy decrease, overall, since the possibility of deformable objects decreases. Therefore, the best performance for average accuracy was when  $voting > 1$ .

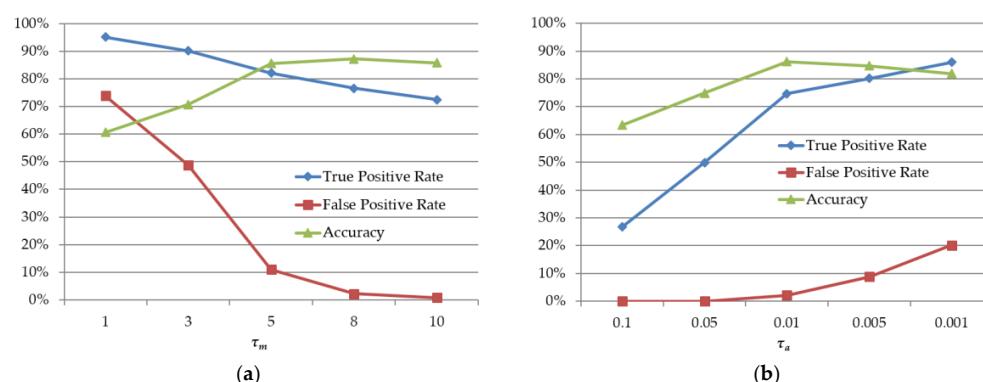
#### 4.3. Deformable Object-Matching Performance Test

Hierarchical clustering was used for deformable-object matching. Hierarchical clustering will create a single cluster without cutoff. Figure 14a presents experimental results obtained by changing the cutoff. As the cutoff value increases, TPR and FPR increase. Accuracy was calculated to find the optimum cutoff value. It was confirmed through experiment that the best accuracy was found at a cutoff of 30. Moreover, linkage for the clustering employed a strong  $k$ -NN in a chain effect, and an experiment was conducted to determine the optimum  $k$ . The results are presented in Figure 14b. For a higher  $k$ , the number of false positives is reduced, but complexity increases. Because there is no significant effect on accuracy when  $k$  is higher than 10, a  $k$  value equal to 10 was employed.



**Figure 14.** Experimental results according to parameters in clustering of geometric similarity for deformable-object matching: (a) experimental result based on the cutoff of geometric similarity; and (b) experimental result based on  $k$  of a  $k$ -NN linkage.

An experiment was conducted using the two proposed methods for clustering validation. When clusters were formed using hierarchical clustering, the number of matching pairs contained in each cluster and the area of each cluster consisting of matching pairs in the image are determined as shown in Figure 15. Each cluster can be viewed as an object in an image. When the number of matching pairs constituting an object is small, or the area is narrow, the resulting object has a low value. Hence, cluster validation was performed using this result. Figure 15a illustrates the experiment regarding false clusters when the number of matching pairs contained in a cluster has a small threshold value ( $\tau_m$ ). The optimum value was determined through experiment to be  $\tau_m = 5$ . Figure 15b shows the calculated ratio of each cluster area and the entire image area. A false cluster was deemed to occur when the ratio ( $\tau_a$ ) is low. From the experiment, the best value was confirmed to be  $\tau_a = 0.01$ , and hence, this value was employed.



**Figure 15.** Results of the parameter experiment for cluster validation: (a) threshold of matching-pairs constituting a cluster; and (b) experimental results based on the ratio of clusters to the entire image area.

#### 4.4. Performance Evaluation for the Proposed Matching Method

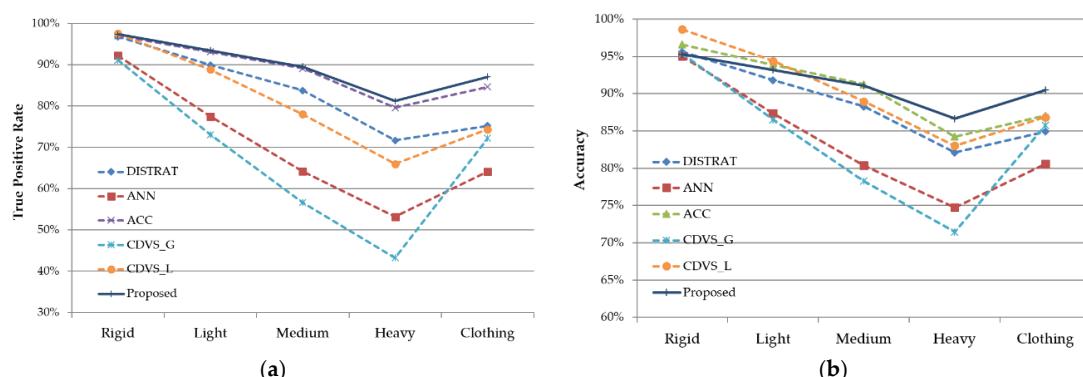
Lastly, performance evaluation using all test images was compared to that of various matching methods. TPR, FPR, accuracy and execution time were compared, as shown in Table 1. The parameters for performance evaluation were determined through the experiments described in the previous Sections 4.1–4.3. The main parameters are as follows. A value of  $T_m = 3$  for the matching score was employed for rigid-object matching. Moreover, rigid matching was deemed to occur when the calculated matching score was higher than 0.5; voting > 1, which determines the best performance, was employed for voting on discrimination of deformable object images. Finally, cutoff = 30 was employed for cluster matching in deformable-object matching, while values of  $\tau_m = 5$  and  $\tau_a = 0.01$  were employed for cluster validation. For the performance comparison, the same parameters were employed for DISTRAT [9] and ACC [15].

**Table 1.** Performance results of image-matching methods (averages).

Methods	TPR	FPR	Accuracy	Matching Time (s)
DISTRAT [9]	83.51%	6.41%	88.55%	0.446
ANN [24]	70.27%	3.03%	83.62%	0.536
ACC [15]	86.45%	6.46%	89.99%	3.759
CDVS(Global) [10]	67.27%	0.35%	83.46%	0.003
CDVS(Local) [10]	74.94%	0.28%	87.33%	0.005
Proposed	89.78%	7.12%	91.33%	0.521

Compared to various matching methods, the average accuracy was highest with the proposed method, with no significant difference in execution time. DISTRAT [9] and ANN [24] are representative rigid object-matching methods, and show good results with normal images, where there is only geometric transformation rather than deformable transformation. However, a dramatic decrease in performance was seen with an increase in the intensity of deformable transformation. Moreover, while the most recent method, CDVS [10], shows extremely fast execution time, since its objective is in retrieval rather than matching (and hence, compressed descriptors are used), this method also shows a decrease in performance because it is intended for rigid objects. While the clustering method [15] has a TPR equivalent to the proposed method, its FPR and time complexity are high, which will present difficulties in actual applications.

Figure 16 shows a comparison of various matching methods with different image types. TPR was observed to be the best in the proposed method. While it can be seen that both TPR and accuracy with the previous methods indicate a significant decrease in performance with an increasing intensity of deformation, the proposed method shows the least amount of decrease, implying that it can be used for robust matching of any transformation.



**Figure 16.** Performance results of the matching methods: (a) comparison of true positive rate; and (b) comparison of accuracy.

## 5. Conclusions

This paper introduced a method of matching images with geometric transformation and deformable transformations using the same features. Previous methods were specialized for certain transformations, and do not show good performance with other transformations. As a solution to this issue, we proposed an adaptive image-matching method that adopts discrimination of deformable transformations. The possibility of the occurrence of a deformable transformation is discriminated using the results from statistical analysis of the matching information and supervised learning. The proposed method showed discrimination performance of more than 90% accuracy, on average. Moreover, we proposed a method of calculating the geometric similarity of each matching pair, matching the pairs by clustering those with high similarity. Previous clustering-based methods have an issue with high time complexity. However, since the use of the aforementioned method of discriminating deformable transformations leads to most non-matching images being removed, and matching concerns only images with a deformable transformation, the discrimination method brings an increase in efficiency to the overall matching method. While the proposed image-matching method exhibits a TPR of 89%, since it can generate both rigid- and deformable-matching results, it was observed that FPR does not increase, since most false positives are removed through discrimination of deformable transformations.

Future research tasks include improving the accuracy of the deformable transformation discrimination method, and developing a deformable image-matching method with a fast execution time. The accuracy of the deformable transformation discrimination method greatly influences performance. In addition to local features, which were used as a solution to the problem of accuracy, the use of global features, like color or texture, can also be considered. Since this paper intends to use single-feature detection, it is believed that the use of such additional information in actual applied systems in the future can lead to improved performance. Moreover, if it is possible to perform fast indexing that can also be used for deformable transformations, it is expected that the system can be extended to an image-retrieval system that is also robust to image transformations.

**Acknowledgments:** This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2010-0020163).

**Author Contributions:** Insu Won and Dongseok Jeong provided the main idea of this paper, designed the overall architecture of the proposed algorithm. Jaehyup Jeong and Hunjun Yang conducted the test data collection and designed the experiments. Insu Won and Jangwoo Kwon wrote and revised the manuscript. All authors read and approved the final manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
2. Won, I.; Yang, H.; Jang, H.; Jeong, D. Adaptive Matching Method of Rigid and Deformable Object Image using Statistical Analysis of Matching-pairs. *J. Inst. Electron. Inform. Eng.* **2015**, *52*, 102–110. [[CrossRef](#)]
3. Liu, S.; Song, Z.; Liu, G.; Xu, C.; Lu, H.; Yan, S. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3330–3337.
4. Na, S.; Oh, W.; Jeong, D. A Frame-Based Video Signature Method for Very Quick Video Identification and Location. *ETRI J.* **2013**, *35*, 281–291. [[CrossRef](#)]
5. Kahaki, S.; Nordin, M.J.; Ashtari, A.H.; Zahra, S.J. Deformation invariant image matching based on dissimilarity of spatial features. *Neurocomputing* **2016**, *175*, 1009–1018. [[CrossRef](#)]
6. Awrangjeb, M.; Lu, G. Contour-Based Corner Detection and Robust Geometric Point Matching Techniques. Ph.D. Thesis, Monash University, Melbourne, Australia, 2008.

7. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
8. Tsai, S.S.; Chen, D.; Takacs, G.; Chandrasekhar, V.; Vedantham, R.; Grzeszczuk, R.; Girod, B. Fast geometric re-ranking for image-based retrieval. In Proceedings of the International Conference on Image Processing (ICIP), Hong Kong, 26–29 September 2010; pp. 1029–1032.
9. Lepšøy, S.; Francini, G.; Cordara, G.; Gusmão, D.; Buarque, P.P. Statistical modelling of outliers for fast visual search. In Proceedings of the International Conference Multimedia and Expo (ICME), Barcelona, Spain, 11–15 July 2011; pp. 1–6.
10. Duan, L.; Chandrasekhar, V.; Chen, J.; Lin, J.; Wang, Z.; Huang, T.; Girod, B.; Gao, W. Overview of the MPEG-CDVS Standard. *IEEE Trans. Image Process.* **2016**, *25*, 179–194. [[CrossRef](#)] [[PubMed](#)]
11. Zhu, J.; Hoi, S.C.; Lyu, M.R. Nonrigid shape recovery by Gaussian process regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 1319–1326.
12. Pilet, J.; Lepetit, V.; Fua, P. Real-time nonrigid surface detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–26 June 2009; pp. 822–828.
13. Kettani, O.; Ramdani, F.; Tadili, B. An Agglomerative Clustering Method for Large Data Sets. *Int. J. Comp. Appl.* **2014**, *92*, 1–7. [[CrossRef](#)]
14. Zhou, F.; Torre, F. Deformable graph matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2922–2929.
15. Cho, M.; Lee, J.; Lee, K.M. Feature correspondence and deformable object matching via agglomerative correspondence clustering. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan, 27 September–4 October 2009; pp. 1280–1287.
16. Yang, H.; Won, I.; Jeong, D. On the Improvement of Deformable Object Matching. In Proceedings of the Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV), Okinawa, Japan, 4–6 February 2014; pp. 1–4.
17. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
18. Krystian, M.; Cordelia, S. A Performance Evaluation of Local Descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1615–1630.
19. Chapelle, O.; Haffner, P.; Vapnik, V.N. Support vector machines for histogram-based image classification. *IEEE Trans. Neural Netw.* **1999**, *10*, 1055–1064. [[CrossRef](#)] [[PubMed](#)]
20. Zhao, W.; Wu, X.; Ngo, C. On the annotation of web videos by efficient near-duplicate search. *IEEE Trans. Multimed.* **2010**, *12*, 448–461. [[CrossRef](#)]
21. Chandrasekhar, V.R.; Chen, D.M.; Tsai, S.S.; Cheung, N.; Chen, H.; Takacs, G.; Reznik, Y.; Vedantham, R.; Grzeszczuk, R.; Bach, J. The Stanford mobile visual search data set. In Proceedings of the Second Annual ACM Conference on Multimedia Systems, Santa Clara, CA, USA, 23–25 February 2011; pp. 117–122.
22. Bookstein, F.L. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* **1989**, *10*, 567–585. [[CrossRef](#)]
23. Nister, D.; Stewenius, H. Scalable recognition with a vocabulary tree. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 17–22 June 2006; pp. 2161–2168.
24. Arya, S.; Mount, D.M.; Netanyahu, N.S.; Silverman, R.; Wu, A.Y. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM* **1998**, *45*, 891–923. [[CrossRef](#)]



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).