COMPUTER VISION AND IMAGE UNDERSTANDING
Vol. 70, No. 1, April, pp. 1–22, 1998
ARTICLE NO. IV970623

# Affine Matching of Planar Sets

Kenji Nagao*

*Multimedia Systems Research Laboratory, Matsushita Electric Industrial Co., Ltd.; and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139*

and

W. E. L. Grimson

*Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139*

Received May 8, 1995; accepted January 24, 1997

**To recognize an object in an image, we must determine the best-fit transformation which maps an object model into the image data. In this paper, we propose a new alignment approach to recovering those parameters, based on *centroid alignment* of corresponding feature groups built in the model and data. To derive such groups of features, we exploit a clustering technique that minimizes intraclass scatter in coordinates that have been normalized up to rotations using invariant properties of planar patches. The present method uses only a single pair of 2D model and data pictures even though the object is 3D. Experimental results both through computer simulations and tests on natural pictures show that the proposed method can tolerate considerable perturbations of features including even partial occlusions of the surface.** © 1998 Academic Press

## 1. INTRODUCTION

A central problem in object recognition is finding the best transformation that maps an object model into the image data. Alignment approaches to object recognition [12] find this transformation by first searching over possible matches between image and model features, but only until sufficiently many matches are found to explicitly solve for the transformation. Given such a hypothesized transformation, it is applied directly to the other model features to align them with the image. Each such hypothesis can then be verified by search near each aligned model feature for supporting or refuting evidence in the image (see, for example, [1] for a method to use the pose to focus the search for other matching features).

One of the advantages of Alignment approaches to recognition [12] is that they are guaranteed to have a worst case polynomial complexity. This is an improvement, for example, over correspondence space search methods such as Interpretation Trees [11], which in general can have an exponential expected case complexity. At the same time, the worst case complexity for alignment can still be expensive in practical terms. For example, to recognize an object with $m$ features from an image with $n$ features, where the projection model is weak perspective, we must search on the order of $m^3n^3$ possible correspondences [12], where $m$ and $n$ can easily be on the order of several hundred. Of course, there are also some other prevalent algorithms for object recognition, such as the Linear Combination method [26] or the Geometrical Hashing method [16], however, all of those are basically in the exhaustive search framework, thus suffering more or less from a similar practical computational problem.

One way to control this cost is to replace simple local features (such as vertices) used for defining the alignment, with larger groups (thereby effectively reducing the size of $m$ and $n$). In this paper, we examine one such method by using an invariant description of features from planar surfaces which undergo linear transformations in space. This invariant description is derived using the second order statistics of the features extracted from the planar patches. We employ this invariant representation in generating potentially corresponding partitions of the features in the model and the image data. This grouping of features allows us to derive a new alignment approach to object recognition based on centroid alignment of corresponding feature groups built on these invariant projections of the planar surface.

This method uses only a single pair of 2D model and data pictures even though the object is 3D. It is also quite fast; in our testing, it took around 30 ms (0.03 s) per sample model and data pair, each with 50 features. It is also demonstrated that our method can handle the considerable perturbations of the images caused even by occlusions of the surface. This is surprising, when we consider that our method solely relies on the (global) statistical information of the features extracted from the entire planar patches.

A work related to our method, in that it uses the whole image (moments of the image) as the feature instead of local features, is that of Cyganski *et al.* [7] based on tensor analysis. They

* Correspondence: nagao@ai.mit.edu.



1077-3142/98 $25.00

developed an efficient method to identify a planar object in 3D space and to recover the affine transformation which yielded the image data from the model. The basis of their method is the contraction operation of the tensors [17, 14] formed by the products of the contravariant moment tensors of the image with a covariant permutation tensor that produces unit rank tensors. Then, further combining those with zero-order tensors to remove the weight, they derived linear equations of the affine parameters to be solved. This method is quite elegant, but it needs at least fourth-order moments of the image (though it appears their updated version suffices with third-order moments [8]). Then, since the higher order moments are notorious for sensitivity to noise [20], it may be very fragile against the perturbations contained in the image data.

## 2. PROBLEM DEFINITION

Our problem is to recognize an object which has planar portions on its surface, using a single pairing of 2D views of the model and data as features. Thus, we assume that at least one corresponding region (which is from a planar surface of the object) including a sufficient number of features exists in both the model and data 2D views. Although we do not explicitly address the issue of extracting such regions from the data, we note that several techniques exist for accomplishing this, including the use of color and texture cues [22, 24], as well as motion cues (e.g., [25, 19]). Rather, we demonstrate in the experiments on natural pictures that our method can tolerate considerable deviations in such regions, including occlusions of the surface, and thus show that it does not require exact extraction of regions. We devise a method for finding an alignment between features of these planar regions. It is important to stress that our method is not restricted to 2D objects. Rather it assumes that objects have planar sections and that we are provided with 2D views of the object model that include such planar sections. Once we have solved for the transformation between model and image, we can apply it to all the features on a 3D object, either by using a full 3D model [12] or by using the Linear Combinations method on 2D views of the object [26].

It is known that under the weak perspective projection model [12, 21, 15], corresponding image features $\{X\}$ and $\{X'\}$ in respective 2D views from the same planar surface are related by an affine transformation,

$$X' = LX + W, \tag{1}$$

where $L$ is a $2 \times 2$ matrix and $W$ is a 2D vector. Thus, the transformations we have to find are these affine parameters, which can be recovered by matching a small number of points across images. The direct use of 2D affine transformations in object recognition was made earlier by Huttenlocher [12]. The issue in which we are interested is whether there are properties of the affine transformation which we can use to efficiently and reliably find the parameters of that transformation.

## 3. A CLASS OF 2D PROJECTIONS OF PLANAR SURFACES INVARIANT TO LINEAR TRANSFORMATIONS

In this section, we introduce a class of transformations of 2D image features from 3D planar surfaces which yield a unique projection up to rotations in the image field, regardless of the pose of the surface in space. Our intention is to use this to introduce contraints on the affine relationship between a model and data defined as the sets of 2D image features.

When we are given potentially corresponding model and data feature sets, then because the translational terms can be removed using the centroid (first-order statistics) correspondences of the feature sets, we are only concerned with recovering the parameters $L_{ij}$, unless otherwise stated. The property of this affine transformation that we exploit for this objective is derived using up to second-order statistics, i.e., covariances, of the features, which is described in the following [5] results. Note that the basic idea is to apply an affine transformation to a point set so that the two major axes of the point set are equal. This involves a scaling along the direction of one of the principal axes of the point set, putting the point set into canonical form. If this is done to two point sets, then all that remains to align the two sets is a single rotation, which can be solved for.

LEMMA 1. *Suppose we apply* whitening transformations $A$, $A'$ *to feature sets* $\{X\}$, $\{X'\}$ *that are related by an affine transformation* $L$, *such that* $A\Sigma_X A^T = c^2 I$, $A'\Sigma_{X'} A'^T = c^2 I$, *to yield* $\{Y\}$, $\{Y'\}$. *Then, the resulting distributions* $\{Y\}$, $\{Y'\}$ *are related by an orthogonal transformation* $T$ *as illustrated in Fig.1,*

$$X' = LX \tag{2}$$

$$Y = AX \tag{3}$$

$$Y' = A'X' \tag{4}$$

$$Y' = TY, \tag{5}$$

*where*

$$A = cV\Lambda^{-\frac{1}{2}}\Phi^T \tag{6}$$

$$A' = cV'\Lambda'^{-\frac{1}{2}}\Phi'^T, \tag{7}$$

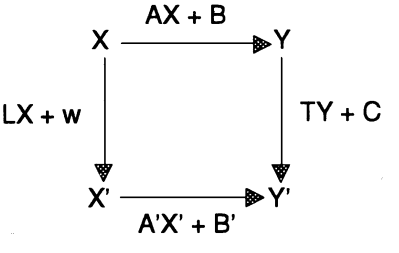


**FIG. 1.** Commutative diagram of transformations. Given model feature $X$ and corresponding data features $X'$, we seek conditions on the transformations $A$, $A'$ such that this diagram commutes.

*where $\Phi$ and $\Phi'$ are eigenvector matrices and $\Lambda$ and $\Lambda'$ are eigenvalue matrices of the covariance matrices of $X$ and $X'$, respectively, $[\cdot]^{-\frac{1}{2}}$ denotes the square root matrix of a positive definite matrix [13] and $[\cdot]^T$ is the matrix transpose,$V$ and $V'$ are arbitrary orthogonal matrices, and $c$ is an arbitrary scalar constant.*

Since we can control the selection of $A'$, $A$ such that $T = A'LA^{-1}$ satisfies $det[T] > 0$ (or $det[T] < 0$), where $det[L] > 0$, it can always represent a rotation (reflection) matrix. Therefore, the property stated in Lemma 1 implies that if we have a set of model features and data features related by an affine transformation (either due to a weak perspective projection of the object into the image, or due to a linear motion of the object image between two image frames), then if we transform both sets of features linearly in a well defined way (via (6) and (7)), we derive two distributions of features that are identical up to a rotation in the image field. This implies that the transformed distributions are unique up to their shapes. More importantly, it also provides an easy method for finding the related transformation.

A physical explanation of this property for the rigid object case is given using Fig. 2 as follows. Suppose the upper pictures show the surfaces in space at the model and the data poses as well as the respective orthographic projections. Looking at the major and minor axes of the 2D model and the data, we can change the pose of the planes so that the major and minor axes have the same length in both the model and data, as depicted in the lower pictures. This is nothing but a normalization of the feature distributions, and the normalized distributions are unique up to a rotation, regardless of the pose of the plane, i.e., no matter whether it is from the pose for the model or for the data.

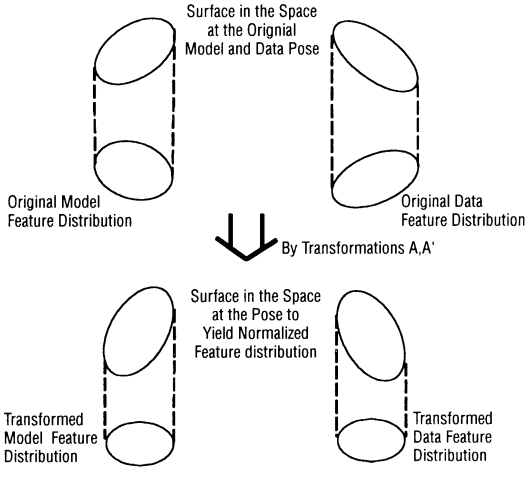


**FIG. 2.** Physical explanation of the invariant projection. The upper pictures show the surfaces in space at the model and the data poses, as well as their orthographic projections to the image field. The lower pictures show the surfaces and their projections at the poses yielding normalized distributions.

The value of the invariant description introduced above is further recognized by observing the following property regarding the availability of other invariant representations.

PROPOSITION 1. *As long as we are provided only up to second-order statistics of the image feature sets, the only available class of linear transformations that can perform invariance up to orthogonal transformation is the one described above that decorrelates the given distributions.*

A proof of this proposition is given in the Appendix. The content of this propostion well coincides with the following intuitive observation: Since the constraint by second-order statistics $\Sigma_{X'} = L\Sigma_X L^T$ provides only three equations for four unknowns $L_{ij}, (i, j \in 1, 2)$ because of its symmetry, we can never solve for all of these parameters by only using the covariances. In this context, the orthogonal matrix $T$ accounts for the remaining one degree of freedom. If we could generate an invariant distribution up to rotations using only second-order moments and yet without whitening the distribution, we would be able to determine the matrix $T$, e.g., the rotation angle, from the principal axes of the thus normalized distribution, thereby solving for the affine parameters. This is apparently a contradiction.

## 4. ALIGNMENT USING A SINGLE 2D MODEL VIEW

In this section, we show how we can align the 2D model view of the planar surface with its 2D images using the affine invariant description of the features described in the last section.

### *4.1. Using the Centroid of Corresponding Feature Groups*

If the model and data features can be extracted with no errors, and if the surface is completely planar, then applying the presented transformation to model and data features will yield new feature sets with identical shapes (up to an image plane rotation). Thus, in this case, our problem, i.e., recovering the affine parameters which generated the data from the model, is quite straightforward. One way to do this is simply to take the most distant features from the centroid of the distribution both in the model and data and then to do an alignment by rotating the normalized model to yield a complete coincidence between each model and data feature. Then, we can compute the affine parameters which result in that correspondence.

However, the real world is not so cooperative. Errors will probably be introduced in extracting features from the raw image data, and, in general, the object surfaces may not be as planar as we expect. Further, the target object region may be partially occluded by other surfaces. To overcome these complications, we propose a robust alignment algorithm that makes use of the correspondences of the centroids of corresponding feature groups in the model and data. Here we have a popular convenient property [7]:

LEMMA 2. *When the motion of the object in space is limited to linear transformations, the centroid of its orthographic projection to a 2D image field, i.e., centroids of image feature*

*positions, is transformed by the same transformation as that by which each image feature is transformed.*

We further note the following useful property regarding the stability of the centroid.

LEMMA 3. *When the perturbations of the features (due to inaccuracies of the feature positions, missing features, occlusions, or deviations from coplanarity of the features) are zero-mean, the centroid is still transformed by the same linear transformation, although each feature point is no longer guaranteed to be aligned by the same transformation.*

Note that these properties are generally true for any object surface and its motions. The planarity of the surface does not matter. In the case when the object happens to be planar, as the motion of the 2D image feature is described by a 2D affine transformation, the centroid of the features is also transformed by the same affine transformation.

In [23], the use of region centroids was proposed in the recognition of planar surfaces. Unlike our approach for using feature group centroids, however, their method can only be applied to planar objects, as described in the paper.

### *4.2. Grouping by Clustering of Features*

Since affine parameters can be determined from three point correspondences, our problem becomes one of obtaining at least three corresponding positions in model and data, in the presence of perturbations. Based on the observations made in the preceding sections, we propose to group the model and data features using their normalized coordinates, so that we can extract a single feature from each of a small number of groups. The goal is to use such groups to drastically reduce the complexity of alignment based approaches to recognition, by finding groups whose structure is reproducible in both the model and the data and then only matching distinctive features of such groups.

One way to group features is to employ clustering techniques. In the selection of a clustering algorithm, taking into account the use of the property described in the last section, that is, the normalized model and data features are unique up to rotations and translations, we set the following two criteria: (a) invariance of the clustering criterion to rotations and translations of the $x$, $y$ coordinate system and (b) low computational cost. The criterion (b) is critical, because if the computational cost of clustering is similar to those of conventional feature correspondence approaches, the merit of our method will be greatly decreased.

As the basic principle of the clustering algorithm, we have adopted the *nearest-mean iteration procedure*, which is also the basis of the well known *Kmean or ISODATA* procedure [9, 10, 18]. It is a realization of minimizing the intraclass covariances of the features given below, which is apparently invariant to rotations, by an iterative procedure.

Specifically, let the criterion be

$$J = trace[K_w], \tag{8}$$

where

$$K_w = \sum_{i=1}^{M} Q(\omega_i) K_i, \tag{9}$$

where $Q(\omega_i)$ is the probability density function of the $i$th cluster, $M$ is the number of clusters, and $K_i$ is the intraclass covariance of the $i$th cluster. Therefore, the clustering algorithm attempts to reduce the sizes of clusters, i.e., the variances of the features contained. In other words, it tries to find chunks of features concentrated in a small area. We use this clustering mechanism in the normalized coordinate space of the features produced by using the transformations described in the last section, where the model and the data feature distributions have the same shape up to a rotation (for the detailed description of this algorithm, see Fukunaga [9]).

For the reasons described above, when the feature distributions are concentrated in some local parts, since the model and the data feature distributions should have the same or at least similar shape, the algorithm can yield the corresponding partition of the features quite stably even under some collapse of the data. In fact, this is demonstrated in the experiments on natural pictures with considerable partial occlusions. Of course, even in cases where the feature distributions do not have local concentrations, as long as the damage of the correspondences of the extracted features between the model and the data are not serious, e.g., without occlusion, the clustering algorithm can yield similar segmentations of the features by devising the way of giving initial clusters.

Since the nearest-mean iteration procedure, starting from the initial clustering, proceeds like a steepest descent method for ordered data, it is computationally very fast. It runs in $O(N)$ time in terms of the number of features $N$ to be classified, when we set the upper limit on the number of iteration. We should also note that, although it is not guaranteed that it will reach the real minimum of $J$, we know that our aim is not to minimize/maximize some criterion exactly, but simply to yield the same cluster configuration both in model and data clustering. Minimization of a criterion is nothing more than one attempt to do this.

### *4.3. Aligning a Model View with the Data*

Now we can describe a basic algorithm for aligning a 2D view of a 3D model object with its novel view, which is assumed to be nearly planar. Note that to determine the best affine transformation, we must examine all the feature parts isolated from the image data, as we do not know which group in the data actually corresponds to the planar surface which has been extracted to form the model.

- Step 0: For a feature set from a 2D view of a model, compute the matrices given in (6) where $V$ may be set to $I$ and generate the normalized distribution. Cluster based on nearest-mean iteration to yield at least three clusters. Compute the centroid of each cluster reproduced in the original coordinates. This process can be done off-line.

- Step 1: Given a 2D image data feature set, do the same thing as step 0 for the data features.
- Step 2: Compute the affine transformation for each of the possible combinations of triples of the cluster centroids in model and data.
- Step 3: Do the alignment on the original coordinates and select the best-fit affine transformation.

Step 1 is $O(N)$. In Step 2, computation of affine parameters must be done for only a small number of combinations of clusters of model and data features. So, it runs in constant time. Step 3 is, like all other alignment approaches, on the order of the image size. Thus, this alignment algorithm is computationally an improvement over the conventional ones for object recognition.

We stress again that our method is not restricted to planar objects. We simply require a nearly planar surface on an object to extract the alignment transformation. This transform can then be applied to a full 3D model or used as part of a Linear Combinations approach to sets of views of a 3D model to execute 3D recognition.

As another way of using the clustering technique, one might consider that it may suffice to generate only two clusters for model and data. Then, we can rotate the model so that the centroid of the model cluster coincides with that of the data cluster, recovering the rotation matrix $T$ and thus affine parameters by $L = A'^{-1}TA$. In some situations, this may work fine, but in others this may become errorneous. Since the proposed transformation that normalizes a distribution is computed solely from the covariance matrix of the given feature distribution, it is affected by the errors included in the given feature set. In other words, when a feature set includes some errors the normalized distribution of it is distorted. For example, when data feature sets have some missing features from the model data set, the normalized distributions of the model and the data are distorted with respect to each other in addition to the missing features and no longer coincide by a rotation. In particular, this becomes serious when some portions of the planar patch are dropped due to unstable region extraction or simply because of occlusion.

It might also be possible, after generating two clusters and recovering the rotation angle, to find correspondences of the features in the normalized coordinate space and then recover the affine parameters using the established correspondences on the original coordinates of the features. This may work fine as long as the contamination of the data feature is small enough with respect to the density of the feature distributions, so that the unique and correct correspondences are obtainable by aligning the model with the data using the recovered rotation angle. However, when the collapse of the data becomes large and the distortion of the shapes of the normalized model and data features is unignorable, we will have to pay some additional computational cost for finding the feature correspondences. For example, as a possible algorithm: First, we try to recover the rotation angle between the normalized model and data features, then, aligning the model with the data using the computed rotation angle, we can just limit the possible match between the model and data features by setting some allowable distance between them in the normalized space. Finally, we apply some conventional exhaustive search procedure to find the best match in the limited candidates. Thus, this may be interpreted as a coarse-to-fine approach to finding the best-fit transformation from the model to the data. If the collapse of the data features get even more serious such that, as described, the transformed (for normalization) model and the data features no longer correspond by rotations, computing the rotation angle might have very little effect for finding correspondences between the model and data features.

In contrast to those other candidate methods, since in the proposed algorithm the cluster centroids are used in the original coordinates to directly recover the affine parameters, it is not disturbed by the distortion of the normalized distributions described above in recovering the parameters, as long as the generated model and the data clusters are still correspondent. This is the strong merit of our clustering plus centroid alignment based method in dealing with the inaccuracy of the earlier feature extraction process.

## 5. EXPERIMENTAL RESULTS

In this section, experimental results on both computer simulated data and real natural pictures show the effectiveness of the proposed algorithm for recognizing planar surfaces. In the computer simulation and the following first part of the tests on real data, we deal with perturbations of the data due to inaccuracies of feature location, missing features, and surface deviation from planarity. Then, in the second part of the tests on real pictures, we demonstrate the case in which, in addition to these kinds of perturbations, the data is also partially occluded.

### *5.1. Computer Simulations*

Analyses are made using canonical statistical tools, that is, random patterns for model features, random values for affine parameters by which to yield data features, and Gaussian perturbations. Gaussian perturbations simulate the feature extraction errors and the depth perturbations of the object surface in space from planarity. We also study the case in which missing of features happens randomly.

As the model features were generated simply randomly, the distributions tended to be fairly regular. Thus, this simulates the case where perturbations are included in the relatively regularly distributed feature data. As argued in the last section, this is a slightly hard situation, so we had to devise an implementation of the algorithm to recover the affine parameters stably. After all, by using some different initial clusters in conjunction with the alignment framework, it has turned out that generating three or four clusters for each given initial cluster provides sufficiently good performance.

*Algorithm Implementation.* To obtain this minimum number of clusters in model and data, we adopted a hierarchical application of the nearest-mean procedure, each separating the

given whole features into two clusters. This is because in testing the nearest-mean procedure, we found that the accuracies for generating three clusters at once severely declined from those for generating two clusters. Therefore, the actual method we took for feature clustering was: (1) first do clustering on the original complete feature set to yield two clusters for model and data, and (2) then, do clustering again for each of the clusters generated in the first clustering to yield two subclusters from each cluster. As we do not know which clusters correspond with each other in model and data, all the possible combinations of the correspondences between the centroids of model and data cluster and subclusters were examined, which counted 8 matches. In addition, as we found in the course of the experiments that the nearest-mean procedure is slightly sensitive to the variation of initial clusterings, we used several different initial clusters generated as described below and selected the best-fit affine parameters $L_{ij}$: we first compute the line, say $l_0$, that passes through the centroid of the distributions to be classified and is perpendicular to the line passing through the centroid and the most distant feature position from the centroid, then rotate $l_0$ around the centroid by 45°, 90°, 135°, respectively, to yield $l_i$, $\{i = 1, 2, 3\}$. Then, each feature is classified according to which side of the line it is located, producing two initial clusters. This was done for each $l_i$. Although this clearly gives 4 times as many combinations, that is 32 matches, requiring additional computation, the accuracy in recovering the affine parameters is drastically improved as examined by the comparison with the results by a single pair of initial clusters.

In Fig. 3, intermediate results of the hierarchical procedures described above are shown. Note how there is a clear match, up to a rotation in the image plane, between the clusters of the bottom two figures in both the left and right column. In each of the following experiments, 100 sample model and data sets each with 50 features were used, and the average of their results were taken.

*With errors in extracting features.* In Fig. 4, errors in recovering the affine parameters $L_{ij}$ both of single pair of initial clusters and of 4 of them are plotted versus the rate of the Gaussian deviation to the average distance between closest features of the data. Errors are measured based on the formula

$$\text{error} = \sqrt{\frac{\sum_{i,j}(\hat{L}_{i,j} - L_{ij})^2}{\sum_{i,j} L_{ij}^2}}, \tag{10}$$

where $\hat{L}_{ij}$ is the recovered values for affine parameters.

The average distance between closest feature points was estimated by

$$\text{average distance} = \sqrt{\frac{det[L]A}{\pi N}}, \tag{11}$$

where $A$ is the area occupied by the model distribution and $N$ is the number of the features included. The perturbation rate used to generate Gaussian deviations was taken to be the same in both the $x$ and $y$ coordinates to simulate the errors in feature extraction. In Fig. 4 we note that errors are almost proportional to the perturbation rate. In Fig. 5, examples of the reconstructed data distributions, with different errors in recovering the affine parameters, were superimposed on the data with no perturbations. The effect of using 4 pairs of initial clusters are drastic in terms of the accuracy in recovering affine parameters. Although, in the case with a single pair of initial clusters the average errors increased as perturbations in the data features grew larger, errors are still small for most samples as we can see in Table 1. In almost all cases when the recovery of $L_{ij}$ results in large errors, the first clustering failed due to the distortion of the normalized feature configurations caused by the perturbations. The ratio of this kind of failure increased as the perturbation percentage grew, so this is the main reason for the error elevations. When we attempted the first level clustering with 4 pairs of initial clusters, the error ratio was drastically reduced as we see in Fig. 4 and Table 2. Presumably, this trend will continue as we increase the number of the pairs of initial clusters.

From Figs. 4 and 5, our algorithm, especially with the multiple pairs of initial clusters, is found to be quite robust against considerable perturbations caused by the errors in feature extraction.

*Depth perturbation from planarity.* In the same way, Fig. 6 shows estimation errors for the simulated case where the surface has depth perturbations from planarity. As described previously, perturbations in the image field caused by depth variation occur in the direction of the translational component of the affine transformation. Therefore, the perturbation was taken only for the $x$ coordinate. Similar results were obtained for other directions of perturbation.

From Fig. 6, again, we can see that the accuracy was drastically improved by using multiple pairs of initial clusters and this accuracy ensures the stability of our algorithm against perturbations caused by the depth variations of the points from planarity. Thus, our method can be used to obtain approximate

**TABLE 1**
**Number of Samples with Errors vs Perturbation by Single Initial Clustering**

| | Percentage of missing features | | | | | | |
|---|---|---|---|---|---|---|---|
| Recovery rates | 5 | 10 | 15 | 20 | 25 | 30 | 35 |
| –0.01 | 73 | 52 | 30 | 21 | 7 | 3 | 0 |
| 0.01–0.05 | 12 | 17 | 27 | 31 | 36 | 37 | 31 |
| 0.05–0.1 | 8 | 10 | 14 | 16 | 15 | 14 | 14 |
| 0.1–0.2 | 2 | 3 | 5 | 4 | 8 | 10 | 11 |
| 0.2–0.3 | 2 | 2 | 3 | 6 | 7 | 5 | 7 |
| 0.3–0.4 | 0 | 2 | 3 | 5 | 3 | 5 | 5 |
| 0.4– | 3 | 14 | 18 | 17 | 24 | 26 | 32 |

*Note.* The number of the samples with errors out of 100 model and data pairs are shown versus perturbation rate. The first column shows the recovery errors, and the first row shows the perturbation percentages included in the data features. Clustering was done with only $l_0$ for generating the initial clusters.

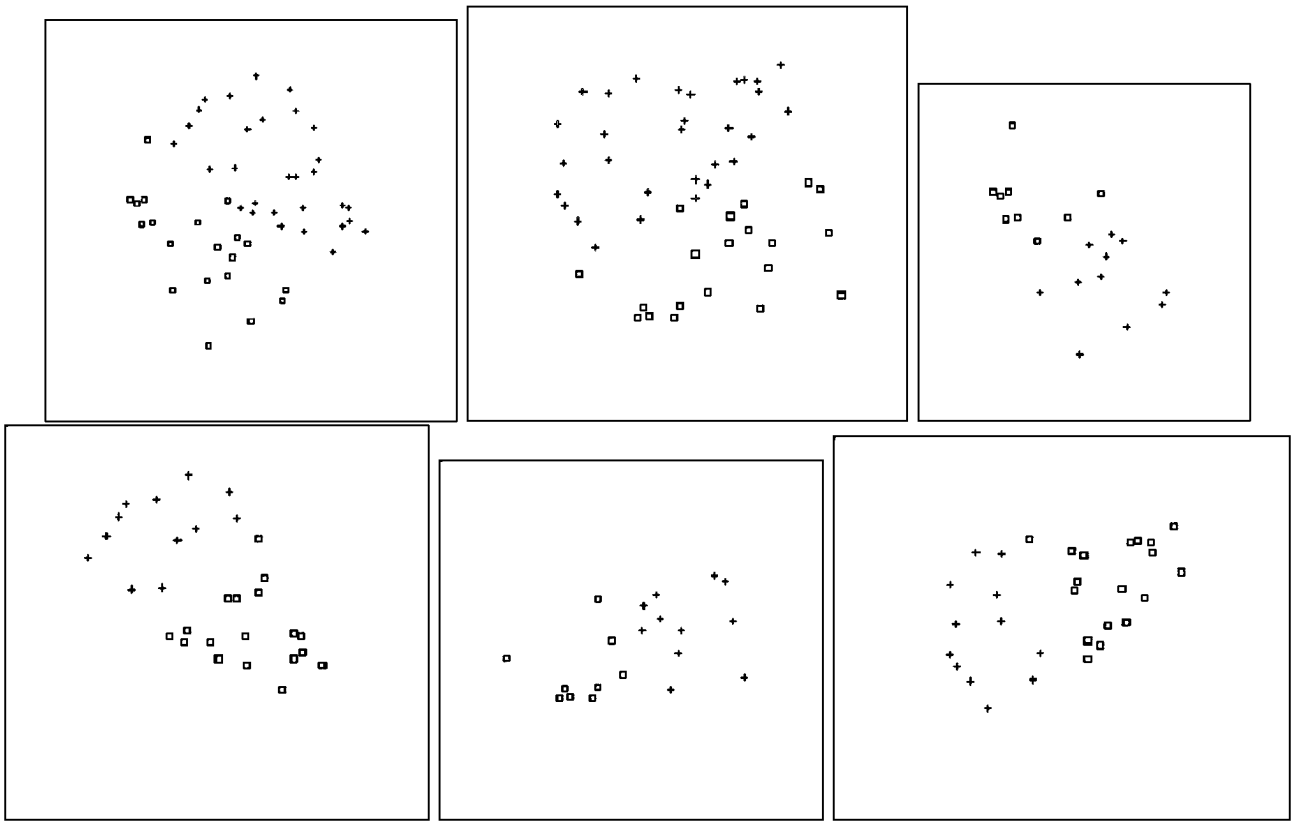

**FIG. 3.** An example of hierarchical clustering. Upper left: results of the first clustering of the transformed model features. Upper right: results of the first clustering of the transformed data features. Middle: subclusters yielded by the second clustering of the first clustering results of the model. Lower: subclusters yielded by the second clustering of the first clustering results of the data.

affine parameters for object surfaces with small perturbations from planarity.

*Involving missing features.* In Fig. 7 and Tables 3 and 4, the errors in recovering affine parameters are shown versus the rate of the number of the missing features in the data, which is to simulated the unstable input from the feature extraction as well as cases involving occlusions. Although the errors increased as the missing features increased, again, we could drastically improve the accuracy by introducing multiple pairs of initial clusters.

**TABLE 2**
**Number of Samples with Errors vs Perturbation with 4 Initial Clusterings**

| | Percentage of missing features | | | | | | |
|---|---|---|---|---|---|---|---|
| Recovery rates | 5 | 10 | 15 | 20 | 25 | 30 | 35 |
| –0.01 | 98 | 94 | 82 | 52 | 26 | 15 | 5 |
| 0.01–0.05 | 2 | 2 | 12 | 39 | 59 | 66 | 71 |
| 0.05–0.1 | 0 | 0 | 0 | 1 | 0 | 1 | 3 |
| 0.1–0.2 | 0 | 0 | 1 | 0 | 1 | 1 | 2 |
| 0.2–0.3 | 0 | 0 | 0 | 1 | 3 | 4 | 4 |
| 0.3–0.4 | 0 | 2 | 2 | 2 | 4 | 4 | 6 |
| 0.4– | 0 | 2 | 3 | 5 | 7 | 9 | 9 |

*Note.* The number of the samples with errors out 100 model and data pairs are shown versus perturbation rate. The first column shows the recovery errors, and the first row shows the perturbation percentages included in the data features. The first clusterings were tried using 4 different pairs of initial clusters produced by using lines $l_i$ $\{i = 0, 1, 2, 3\}$.

**TABLE 3**
**Number of Samples with Errors vs Rate of Missing Features using 1 Initial Clustering**

| | Percentage of missing features | | | | |
|---|---|---|---|---|---|
| Recovery rates | 5 | 10 | 15 | 20 | 25 |
| –0.01 | 7 | 0 | 0 | 0 | 0 |
| 0.01–0.05 | 24 | 29 | 21 | 15 | 1 |
| 0.05–0.1 | 19 | 24 | 34 | 33 | 28 |
| 0.1–0.2 | 4 | 17 | 2 | 8 | 18 |
| 0.2–0.3 | 6 | 17 | 8 | 11 | 8 |
| 0.3–0.4 | 11 | 4 | 1 | 14 | 10 |
| 0.4– | 29 | 9 | 34 | 19 | 35 |

*Note.* The number of the samples with errors out of 100 model and data pairs are shown versus the rate of missing features in the data. Each model has 50 features. The first column shows the recovery errors, and the first row shows the percentages of missing features. Clustering was done with only $l_0$ for producing the initial clusters.

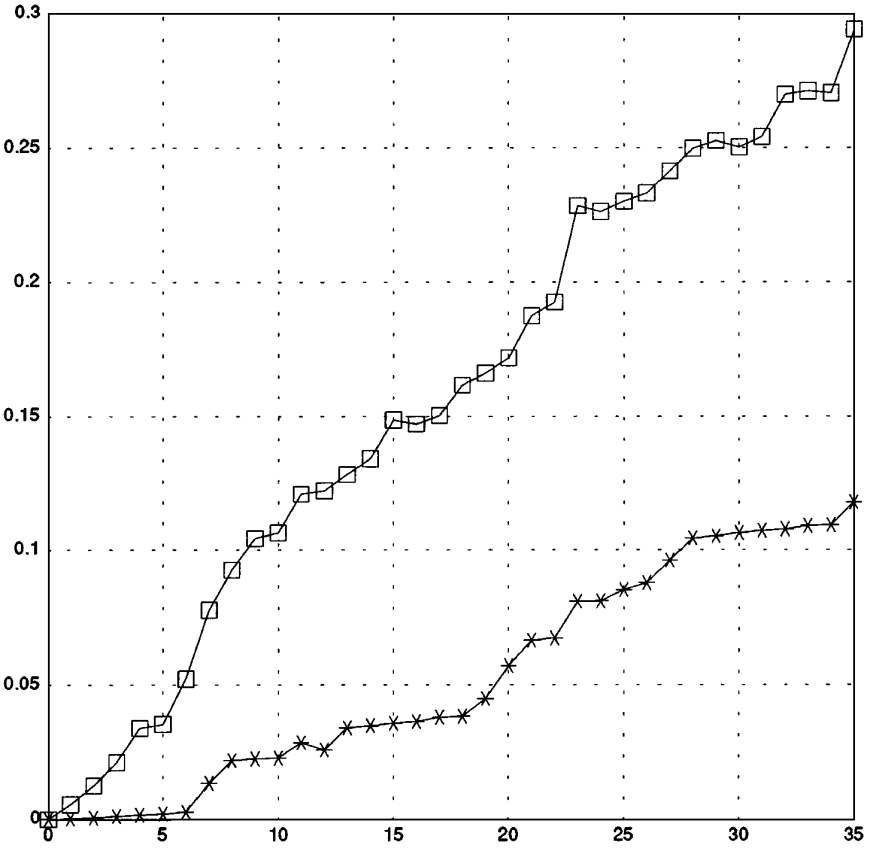


**FIG. 4.** Recovery error versus the rate of perturbation. Errors in recovering affine parameters $L_{ij}$ from the data extracted with errors. The horizontal axis shows the percentage of the Gaussian deviation to the average distance between closest features and the vertical axis shows the error in recovering $L_{ij}$. One hundred model and data pairs were used for each of the perturbation ratios, and 50 features were included in the model and data. The results by a single pair of initial clusters and those by multiple pairs are plotted respectively with the box and the star. Errors are almost proportional to the perturbation rate.

*Computational cost.* The computational cost for recovering affine parameters when we used a single pair of initial clusters in the first level clustering and that of four pairs were on average 10 and 30 ms, respectively, on a SPARCstation IPX. Compared with conventional approaches to object recognition, this is a noticeable improvement.

**TABLE 4**
**Number of Samples with Errors vs Rate of Missing Features using 4 Initial Clusterings**

| | Percentage of missing features | | | | |
|---|---|---|---|---|---|
| Recovery rates | 5 | 10 | 15 | 20 | 25 |
| –0.01 | 11 | 5 | 0 | 0 | 0 |
| 0.01–0.05 | 72 | 84 | 57 | 47 | 14 |
| 0.05–0.1 | 2 | 6 | 23 | 39 | 55 |
| 0.1–0.2 | 3 | 0 | 4 | 4 | 3 |
| 0.2–0.3 | 2 | 0 | 1 | 10 | 3 |
| 0.3–0.4 | 0 | 0 | 0 | 0 | 5 |
| 0.4– | 10 | 5 | 15 | 0 | 20 |

*Note.* The number of the samples with errors out of 100 model and data pairs are shown versus the rate of missing features in the data. Each model has 50 features. The first column shows the recovery errors, and the first row shows the percentages of missing features. The first level clusterings were tried using 4 different pairs of initial clusters produced by lines $l_i$ $\{i = 0, 1, 2, 3\}$.

### *5.2. Tests on Natural Pictures: Without Occlusions*

The proposed algorithm was tested on real pictures taken under natural lighting conditions. As you will note in the following, the objects to be recognized here all have more or less planar portions on their surfaces. The actual extraction of such planar patches was done manually although we expect that this could be done using color/motion cues within some error range. Therefore, the given input image data does not any local occlusions, although some perturbations due to other factors are included in the extracted features. We used the same implementation of the algorithm as that used for the computer simulation, in which four pairs of initial clusters were applied. The feature extractions from the given gray level images were performed by the following process:

- (Step 1) Use an edge detector [6] after preliminary smoothing to obtain edge points from the original gray level images.
- (Step 2) Link individual edge points to form edge curve contours.

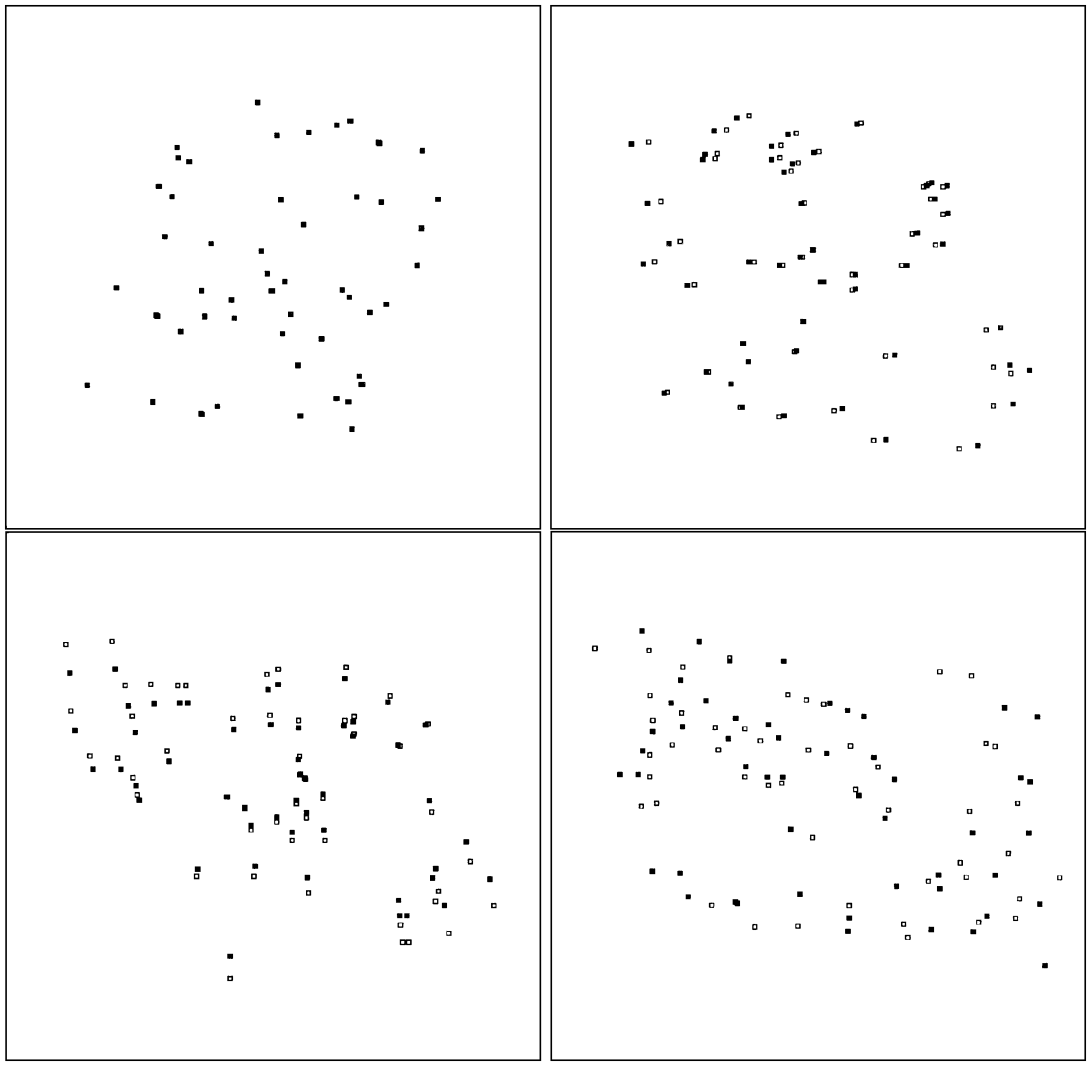

**FIG. 5.** Reconstructed data features by the recovered affine parameters. Reconstructed data features are superimposed on the data generated with no errors: with the error in recovering $L_{ij}$, upper left, 0.0027; upper right, 0.069; lower left, 0.11; lower right, 0.27. White boxes show the data features without errors while the black boxes show the reconstructed features.

• (Step 3) Using local curvatures along the contours, identify features as corners and inflection points, respectively, by detecting high curvature points and zero crossings based on the method described in [12]. Before actually detecting such features, we smoothed the curvatures along the curves [4].

The following first three examples test the proposed algorithm on objects with almost planar surfaces, then in the last two examples we also examine it on nonplanar surfaces.

Figure 8 show the results on pictures containing a Towelette container which has a planar front surface that we could exploit. The upper figures show the edge map of the pictures of it taken from two different views, with detected features out of the front surface superimposed on them as closed circles. The number of extracted features were respectively 141 and 125. Though the difference in the number of extracted features is 16, by our count about 30 features were missed between the two images. The middle figures show the respective normalized feature distributions using the proposed transformation. Despite the missing features and possible errors in locating the features, the normalized feature distributions look quite similar as expected: they appear to coincide by about 180° rotation around the centroid. In the lower figure, the reconstructed views from the right picture using the recovered affine transformation (from left to right) as described in Section 4.3 are superimposed with the right edge map. The error in the recovered affine parameters measured by the formula (10) was 0.0365.

Similarly, the results on the Beer-Box picture are shown in Fig. 9, which has again a planar front surface that we could use for the input to our algorithm. In the upper pictures, detected features are superimposed on the input edge maps. The number of features extracted from the two views were respectively 312

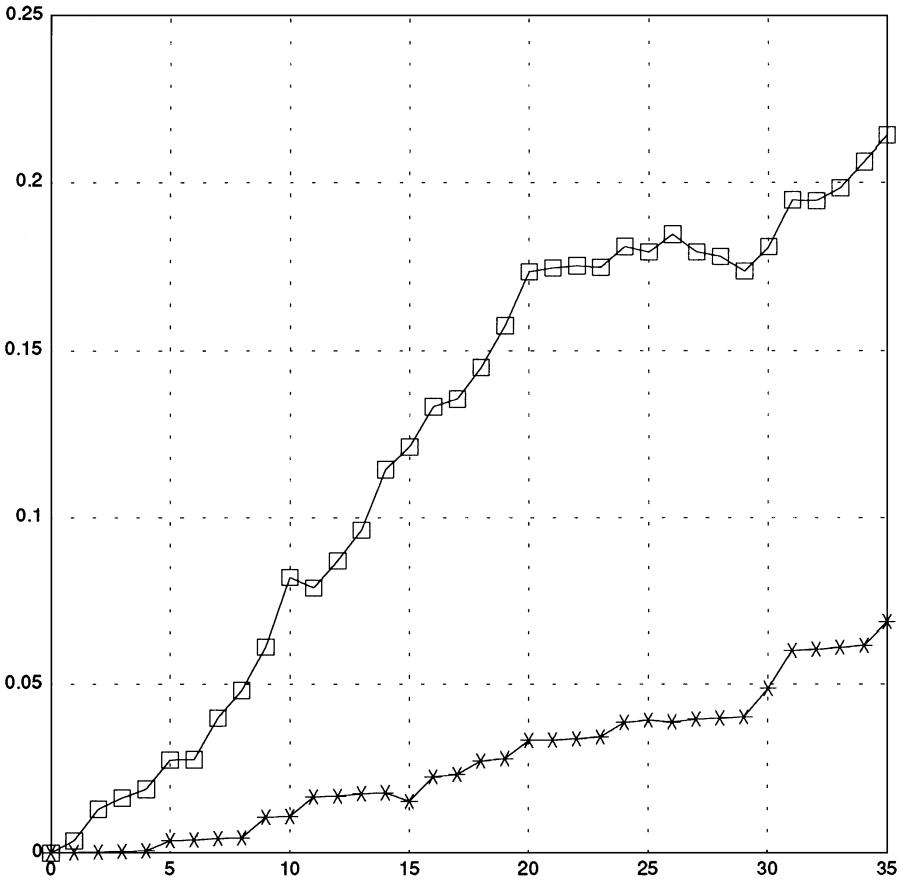


**FIG. 6.** Recovery error versus the rate of perturbation. Errors in recovering affine parameters $L_{ij}$ from data with depth perturbations. The horizontal axis shows the percentage of the Gaussian deviation to the average distance between closest features and the vertical axis shows the error in recovering $L_{ij}$. One hundred model and data pairs were used for each of the perturbation ratios, and 50 features were included in each model and data. The results for a single pair of initial clusters and those for multiple pairs are plotted respectively with the box and the star. For small depth perturbations, the recovered affine parameters can work as a good approximation estimate.

and 348. About 90 features were missed between the two images. Despite the unstable results of the feature extraction, both of the normalized feature distributions as shown in the middle figures are quite similar in shape, thus allowing a good alignment between the two views as we can see in the lower figure. The error in the recovered affine parameters measured by the formula (10) was 0.109.

In Fig. 10, results on the Cocoa-Box pictures are shown, in which the right view was taken under much brighter light conditions, thus giving a quite noisy edge picture. In the upper pictures, detected features are superimposed on the input edge maps. The number of features extracted from the two views were respectively 282 and 262. About 90 features were missed between the two images. Despite this scene clutter in addition to the unstable feature extraction results, both of the normalized feature distributions as shown in the middle figures again have similar shape, bringing a good alignment between two views as shown in lower figure. The error in the recovered affine parameters was 0.078. On the other hand, Fig. 11 shows the results using a high thresholding value for edge detection on the right view. While a lot of meaningless intensity edges were removed this time, many useful features disappeared as well, resulting in only 205 features. Despite these conditions, the alignment of two views shows a fairly good match, though not perfect, as we can see in the figure. The error in the recovered affine parameters was 0.103.

In the following two experiments, we test the proposed method on surfaces with depth perturbation from planarity. In Fig. 12, results on Paper-Cup pictures are shown, where we used the texture drawn on the curved surface on the cup as the input to our method. The number of features extracted from the left and the right views, which are shown in the upper pictures, are respectively 160 and 196. About 50 features were missed between the two images. Despite the distortion of the curved surface from planarity, the normalized features still look similar, as found in the middle pictures. We obtain a good approximate alignment between the two views as we can see in the lower figure. Two error in the recovered affine parameters was 0.128.

A case of an object with a rough surface rather than a smooth curved surface is demonstrated in Fig. 13. The number of extracted features in the left and right views are respectively 151

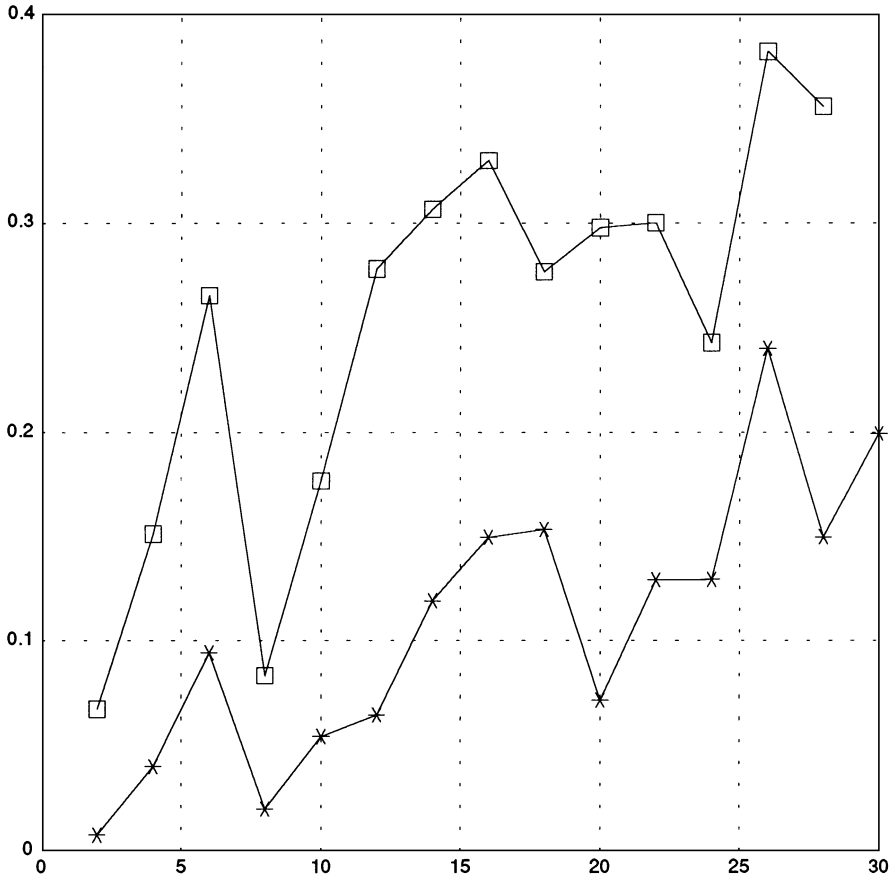


**FIG. 7.** Recovery error versus the rate of missing features. Errors in recovering affine parameters $L_{ij}$ in case with occlusion. The horizontal axis shows the percentages of the missing features and the vertical axis shows the error in recovering $L_{ij}$. The number of model features was 50. One hundred model and data pairs were used for each of the rate of missing features in the data. The results by a single pair of initial clusters and those by multiple pairs are plotted respectively with the box and the star.

and 167. In our count 60 features were missed between the two images. As will be noted in the upper figure, the regions of the telephone pad used for the input to our algorithm have keys and buttons, causing self-occlusions in their surrounding small area. However, we still can recognize strong similarity in their normalized feature distributions as shown in the middle pictures. Actually, a good approximate alignment was performed as shown in the lower picture. The error in the recovered affine parameters was 0.164.

In the examples presented above, for the number of features around 120–350, the running time for computing affine parameters ranged from 40 to 70 ms.

### *5.3. Tests on Images with Occlusions*

The tolerance of the proposed algorithm against occlusions of local parts caused in the region extraction process was also examined. Since the implementation of nearest-mean clustering used for the last two experiments was tuned for feature sets without such significant defects, (providing only 4 clusters), we needed a different method to deal with this case. The difference in the new implementation, which is based on the Kmean algorithm, is that it could produce more clusters than the previous one. Other steps of the procedure for recovering affine parameters described in Section 4.3 are the same. Our expectation is that by deriving more clusters than those of (nearly) minimum numbers we saw for the previous experiments, we can still obtain some correspondent clusters in the remaining regions in the given data which allows us to recover affine parameters even under drops of local parts.

As argued in the previous section, for the kind of data with considerable occlusion that we see below, the nearest-mean clustering approach that detects concentrations of features should still work well. This is because if the incomplete patches cut out from the image still have feature concentrations corresponding those of the model, there is still a good chance that those concentrations will be detected, enabling recovery of the parameters. Apart from this, we should note, however, that when the occlusion of the image data becomes large, we can no longer ignore the deviation of the correspondences of the centroids of the whole features between the model and the data, raising the problem of how to deal with the translational components of the

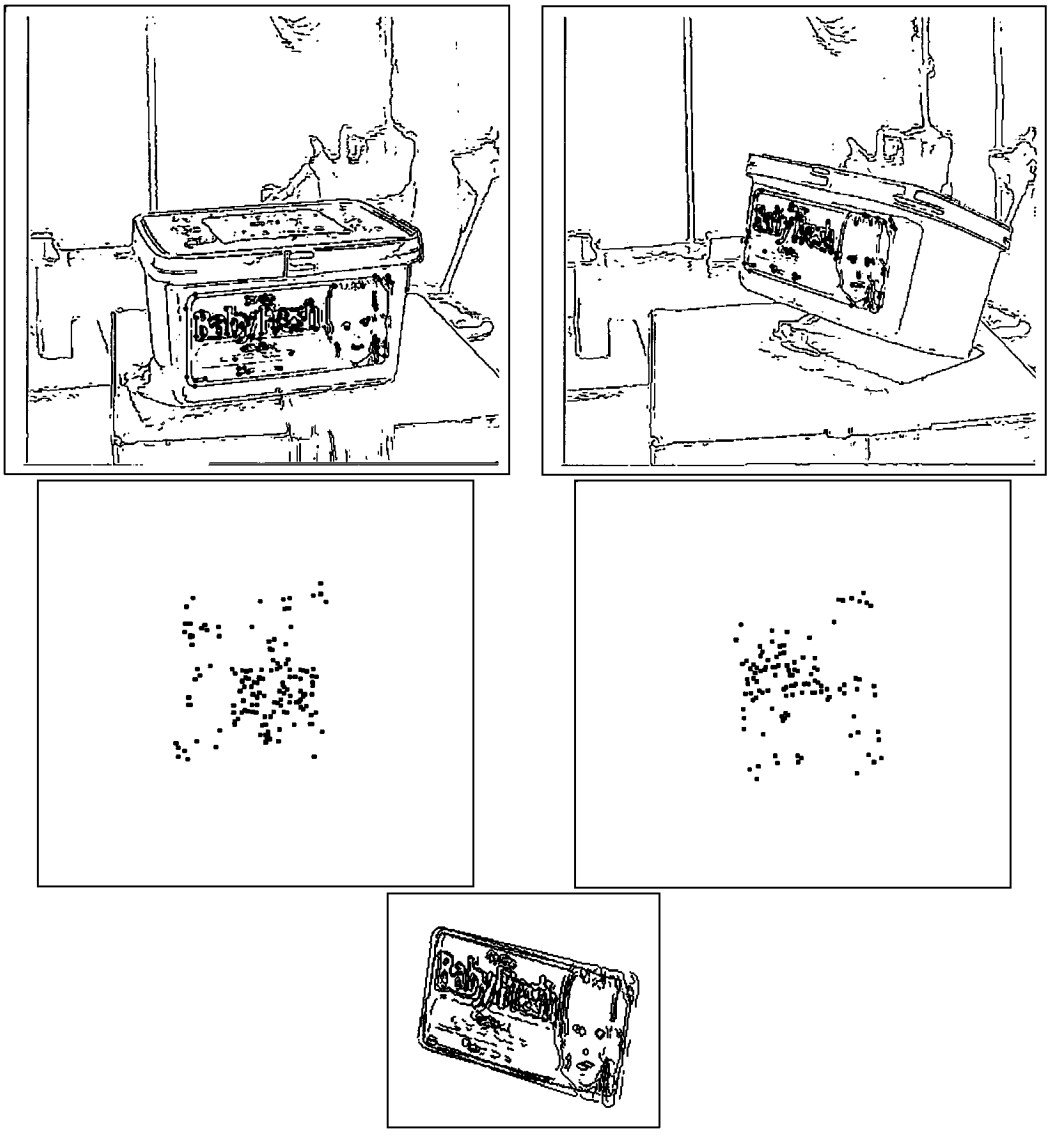

**FIG. 8.** Results on the Towelette pictures. Upper pictures show the edge map of the front part of the Towelette container taken from two different views, with detected features superimposed on it with closed circles. The middle figures show the respective normalized feature distributions obtained using the proposed transformation. Despite the missing features and possible errors in locating the features, the normalized feature distributions appear to coincide by 180° rotation. In the lower figure, the reconstructed view from the right picture using the recovered affine transformation (from left to right) as described in Section 4.3 is superimposed with the right edge map. The error in the recovered affine parameters measured by the formula (10) was 0.0365.

transformation. In this case, however, we can use other detected centroids to remove or to recover translational terms. In the first example that follows, we do not consider this in aligning the model with the image, and thus we actually see some translational deviation in the results when the area dropped occupied a large portion of the object. In the second example, we deal with this by using one of the cluster centroids generated. To test this effect of occlusion in our experiments, we removed nearly one quarter or one half of the planar regions manually from the almost complete patches in the images we used earlier. As the initial seed points (cluster centroids) for Kmean clustering, we used 5% of the total number of features, and we picked centroids of clusters containing more than 3% of the total number of features (these ratios were fixed throughout the following experiments).

Figure 14 shows the results of a test using the same Towelette container pictures as those used in the last experiments. The top left picture is the planar patch for the model, and top right is the corresponding almost complete image patch to be recognized, where produced cluster centroids are depicted by large crosses: 10 clusters for the model, 6 for the data. The left picture in the second row shows the results on data for which the lower left quarter is dropped, while in the right of second row almost one half of the region is removed: 6 clusters for the left, 7 clusters

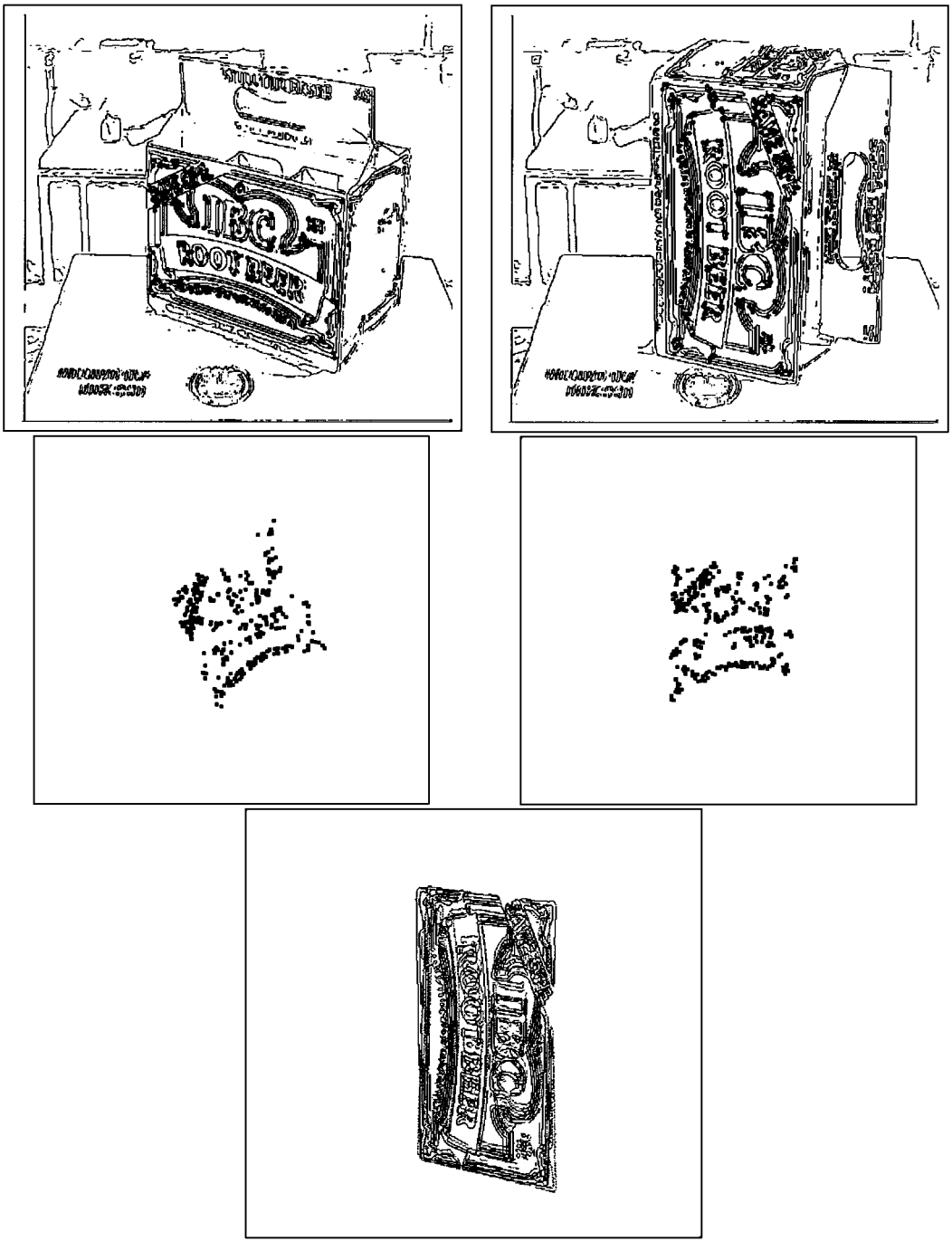


**FIG. 9.** Results on the Beer-Box pictures. In the upper pictures, detected features are superimposed on the input edge maps. Despite the unstable results of the feature extraction, both of the normalized feature distributions as shown in the middle figures are similar in shape, thus allowing a good alignment between the two views as we can see in the lower figure. The error in the recovered affine parameters measured by the formula (10) was 0.109.

for the right were produced. Looking at those pictures carefully, we note that some cluster centroids are still correspondent. In the third row, results superimposing the reconstracted data from the model with the image data are shown, where for the middle picture, nearly one quarter of the region was dropped which lost nearly 10% of the features, while for the right figure almost half was dropped which lost 35% of the features. The left image shows the results on almost complete data; thus we still obtain fairly good alignment, except for the translational gap due to the move of the centroid of the whole features which became unignorable when we dropped half of the patch. The errors in the recovered affine parameters measured by the formula (10)

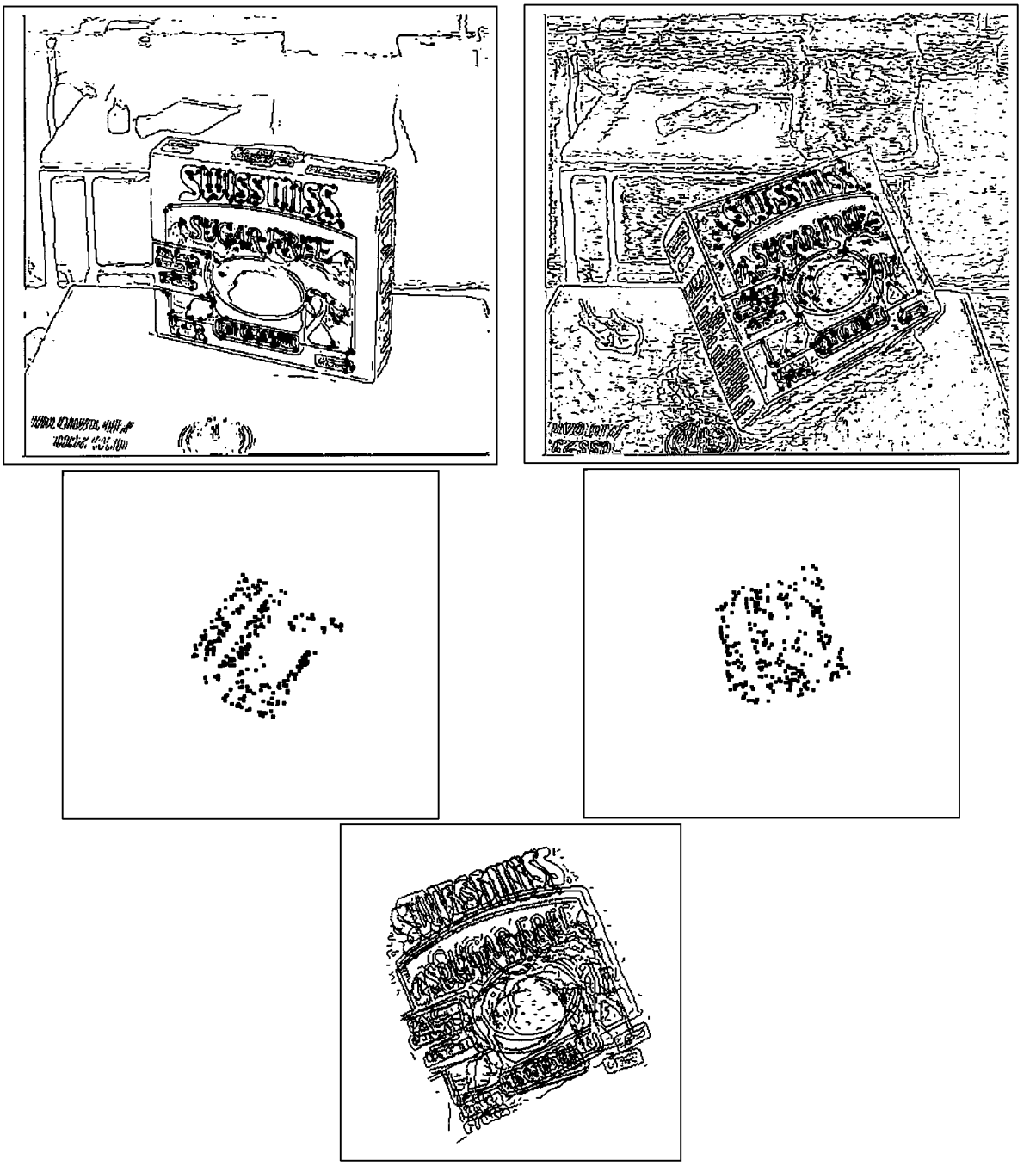


**FIG. 10.** Results on the Cocoa-Box pictures under different lighting conditions. In the upper pictures, detected features are superimposed on the input edge maps. The right picture was taken under a extremely bright lighting conditions, bringing a quite noisy edge map. Despite this scene clutter in addition to the unstable feature extraction results, both of the normalized feature distributions as shown in the middle figures have again pretty similar shape, bringing a good alignment between two views as shown in the lower figure. The error in the recovered affine parameters was 0.078.

were for the figure in the third row 0.124, for the bottom left 0.188, and for the bottom right 0.124. As demonstrated in the next example, this problem can be fixed using a centroid of one of the produced clusters.

In Fig. 15, results of clustering using the Kmean procedure are presented in the normalized coordinates, where top left is for the model, top right is for almost complete corresponding data patch, bottom left is for the data with a drop of the quarter of the patch, and bottom right is for the data with half of patch dropped. It shows how the nearest-mean procedure on the normalized coordinates finds corresponding concentrations of the features between the model and the data even with considerable

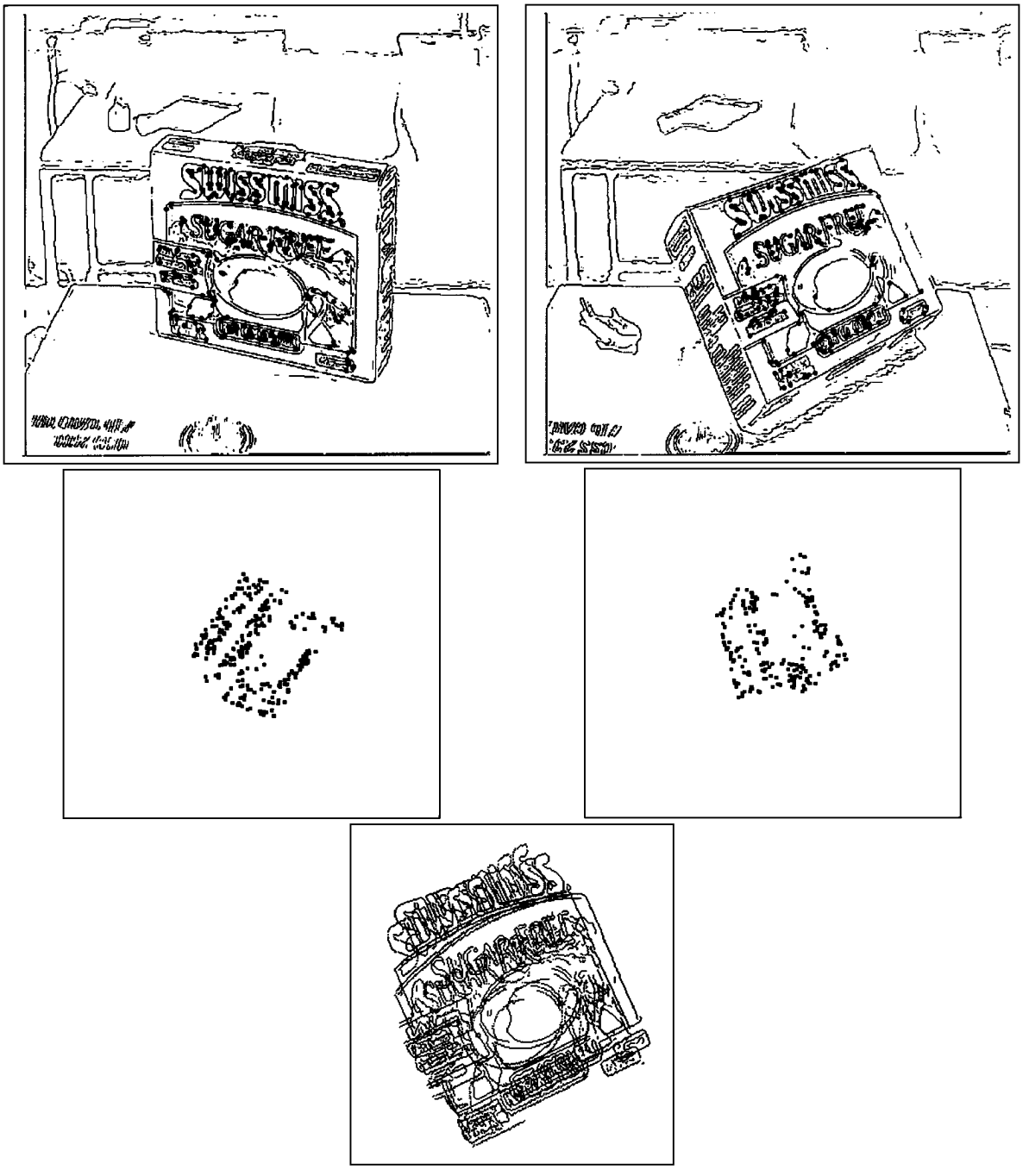


**FIG. 11.** Results on the Cocoa-Box pictures with different thresholds for edge detection. A different thresholding value was used for detecting edges in the right picture. Despite of these conditions, the alignment of two views resulted in a fairly good match, though not perfect. The error in the recovered affine parameters was 0.103.

occlusions of the patches, providing a fairly good recovery of affine parameters. Note that in case of the larger occlusion of the patch, the distortion of the normalized distributions becomes serious, such that individual corresponding features never come close by any rotation.

In Fig. 16, results on similar tests using the Beer-Box pictures are presented. The top figure shows the model with extracted cluster centroids superimposed with large crosses; the first from the left in the middle row is the results of almost complete data to be recognized; the second is the data for which one quarter of the patch in lower left corner is dropped that included 22% of the original whole features; the third is the data where the upper right corner part is dropped, leading to 35% loss of features; in the fourth the results on the data for which upper half of the patch was

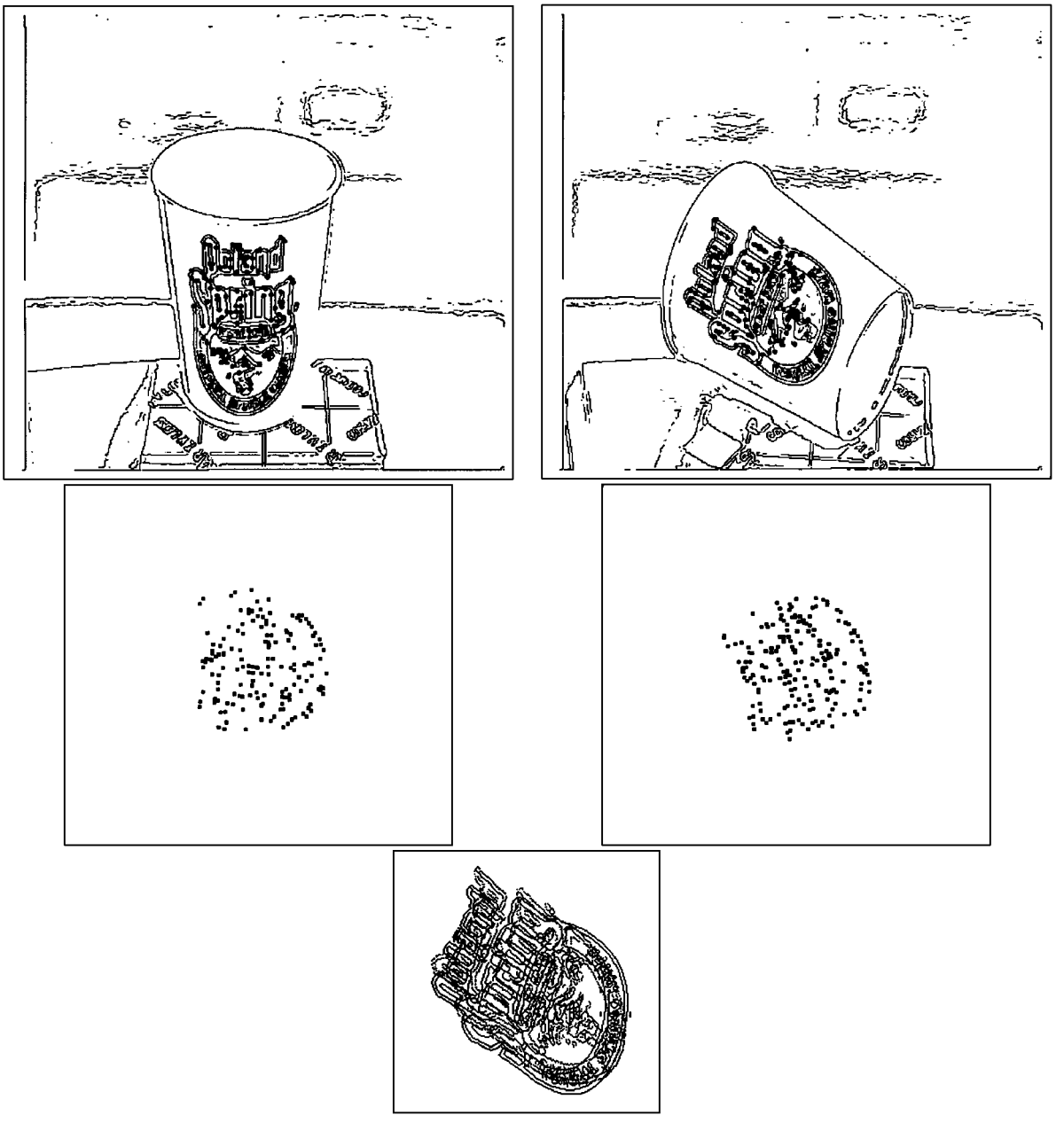

**FIG. 12.** Results on the Paper-Cup pictures. In the upper pictures, detected features are superimposed on the input edge maps. Despite the distortion of the curved surface from planarity, both of the normalized feature distributions as found in the middle figures are similar in shape, thus yielding a good approximate alignment between two views, in the lower figure. The error in the recovered affine parameters was 0.128.

totally lost is given, which amounted to 56% loss of the features. The number of generated clusters were: top (model) 12, in the second row (data), first from the left 13, second 12, third 11, fourth 6. It should be noted that even if the rate of the dropped area increases up to more than 50%, the clustering procedure still generates correspondent clusters. In the reconstructed data in the third row (ordered in the same order as the second row), we see quite accurate alignment between the model and the data. The errors in the recovered affine parameters were: for the first from the left 0.048, second 0.059, third 0.092, and for the fourth 0.191. Here, the translational component was also considered as the parameters of affine matching and was handled using one of produced cluster centroids.

Figure 17 shows the results of the clustering, wherein we confirm that aligning the model features with the data features using the rotation angle would never bring satisfactory results for recovering affine parameters, as the area of the dropped parts increases.

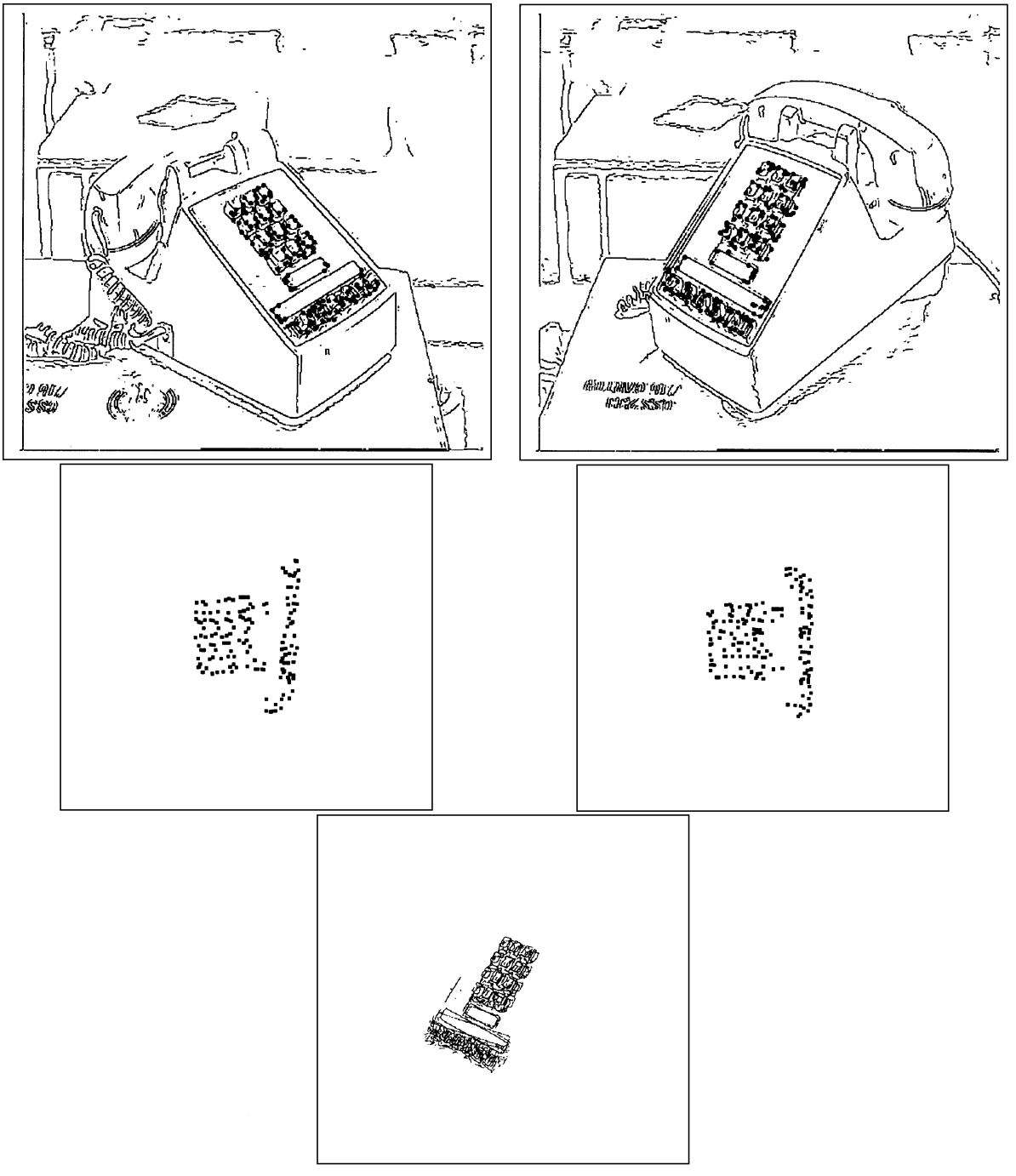

**FIG. 13.** Results on the Phone pictures. In the upper figure, features extracted out of the telephone pad part are superimposed on their edge pictures. As the telephone pad has keys and buttons, the pictures are self occluded in their surrounding small area. However, in the middle pictures we can still recognize some similarity in their normalized feature distributions. The alignment resulted in a good approximation, though not so accurate as those presented above. The error in the recovered affine parameters was 0.164.

### *5.4. Discussions*

Through the experimental results obtained by the computer simulation and tests on the natural pictures, the following are noted.

- The proposed algorithm is quite stable against errors in locating features, perturbations of the surface from planarity, and missing features happening randomly in natural pictures, as demonstrated using point features extracted by standard edge detection plus feature extraction algorithms.

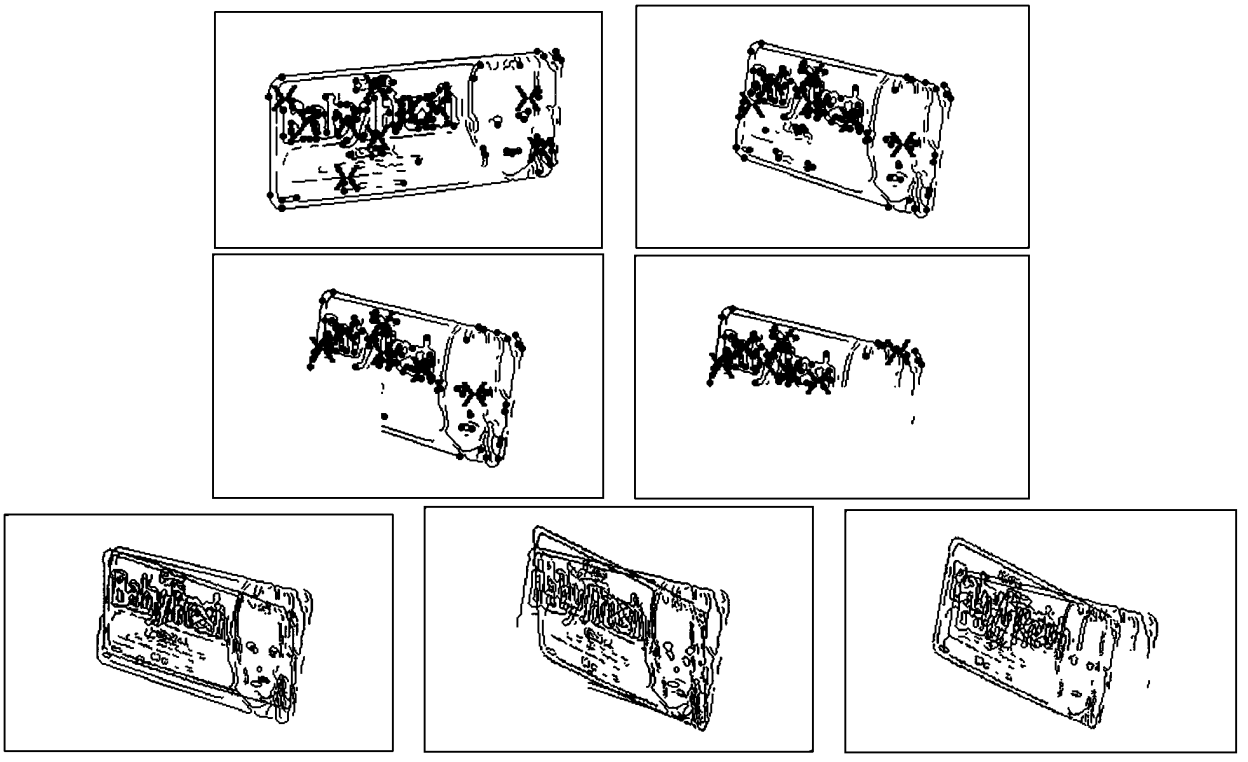

**FIG. 14.** Tests on occluded image of the Towelette. The top left picture is the planar patch for the model, and top right is the corresponding almost complete image patch to be recognized, where produced cluster centroids are depicted by large crosses. The left picture in the second row shows the results when the lower left quarter is dropped, while in the right of second row almost half part of the region is removed. Note that some the cluster centroids are still correspondent. The third row shows results superimposing the reconstructed data from the model with the image data, where for the middle picture, nearly one quarter of the region was dropped which lost nearly 10% of the features, and for the right figure almost half was dropped which lost 35% of the features. The first image of the third row shows results on almost complete data. The errors in the recovered affine parameters measured by the formula (10) were for the figure in the third row 0.124, for the bottom left 0.188, and for the bottom right 0.124.

- It can also tolerate drops/occlusions of local parts of the patch to some extent. As far as we have examined, it can cope with natural pictures sufficiently accurately which have considerable collapses of image data: even in the case of more than 50% of the area being dropped, it could perform still usable accuracy of recovering parameters.
- The proposed algorithm is computationally extremely fast as compared with conventional algorithms.
- For the cases where no significant drops of patch happened, as we increase the number of different initial clusterings, the accuracy of recovering affine parameters is considerably improved. This trend will also hold true for the case of using Kmean implementation, although it was not included in the experiments. However, this imposes the greater computational cost. Clearly there is a trade-off between the accuracy and the computational cost.
- In the results on the natural pictures, readers might notice small slide-offs in the point to point matches between the original data and the reconstructed data. Note that, however, as the output of an alignment operation we do not need the complete coincidence between those feature positions. Once we have obtained a good alignment, if not perfect, and found the correspondences of each feature in the model and the data, we can employ other methods such as least square errors to minimize the errors in recovering affine parameters, which is performed by a direct computation. We should also point out that not every application of object recognition does require a perfect alignment. There will be cases in which the accuracy demonstrated in this section suffice.
- In case the feature extractions result in a quite unstable output, either due to some significant change of the imaging condition, or, simply because the surface is occluded, since the computational cost of our algorithm is negligibly small, it can still be used to realize a coarse-to-fine approach: first by applying our algorithm to the planar portions on the objects to obtain an approximate alignment, thereby trimming needless combinatory spaces of the search, and then by using conventional exhaustive search method within the limited search space to recover the precise parameters.

## 6. CONCLUSION

We have proposed a quite stable and efficient algorithm for recognizing 3D objects by combining an affine invariant property of the planar surface with the centroid alignment approach. The basic strategy is the following: By decorrelating the given feature distributions we obtain normalized shape of the

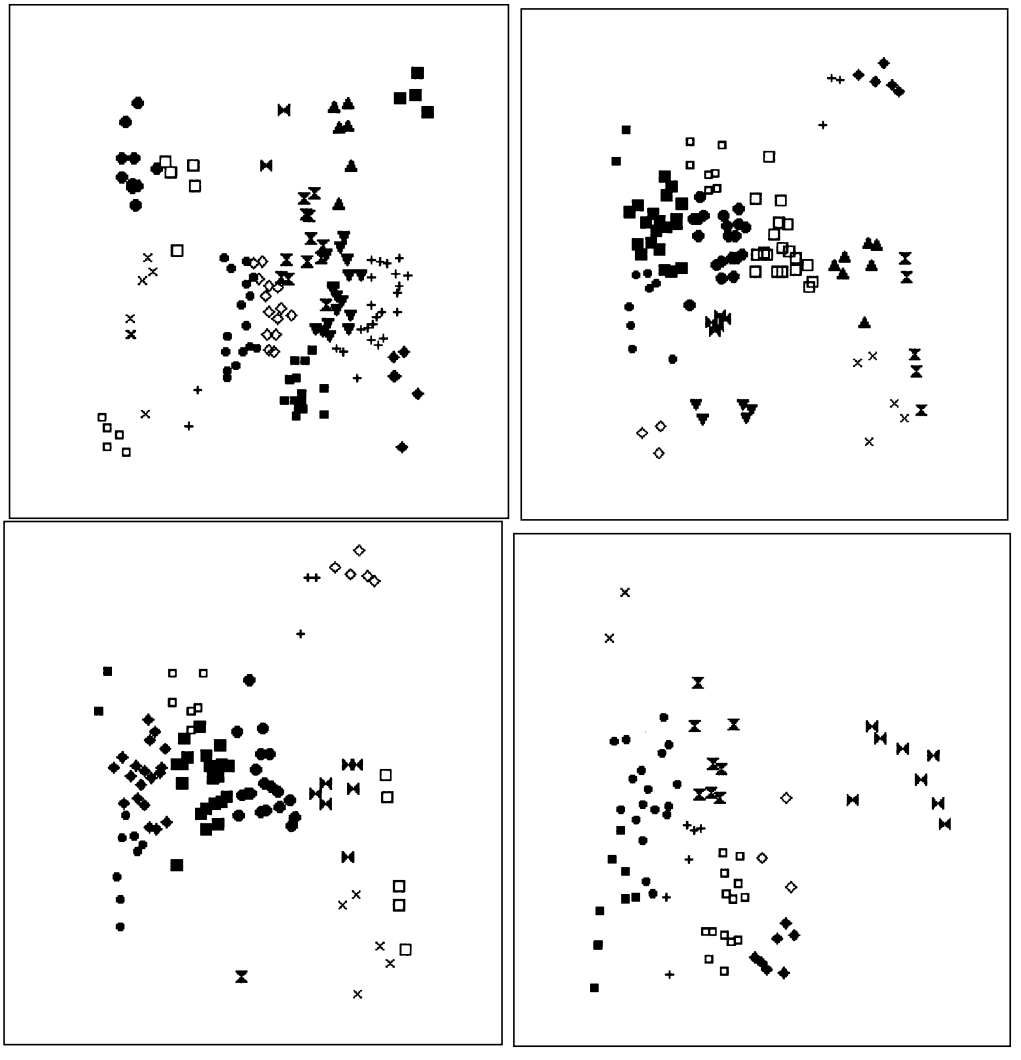

**FIG. 15.** Results of clustering by Kmean on the Towelette pictures. Top left is for the model, top right is for almost complete corresponding data patch, bottom left is for the data with a drop of the quarter of the patch, and bottom right is for the data with half of patch dropped. It shows how the nearest-mean procedure on the normalized coordinate find corresponding concentrations of the features between the model and the data even with considerable drops/occlusions of the patches, providing a fairly good recovery of affine parameters. Note that in case of the larger drop of the patch, the distortion of the normalized distributions becomes serious, so that individual corresponding features never come close by any rotation.

distributions up to rotations for the model and the data, regardless of the pose of the surface in 3D space. Then, we produced potentially correspondent clusters of the features, via clustering minimizing the size of each cluster in the normalized coordinate. Instead of using the rotation angle between thus normalized model and data, we directly used the coordinate of the cluster centroids in the original (image) coordinate space to recover affine parameters that produced the data from the model, through the alignment framework. This brought a quite robust recognition of planar surfaces. We demonstrated that even under significant damage of the given image data, the proposed algorithm could perform a fairly good recovery of the parameters. Also, the algorithm was found to be quite efficient: it took at most only 100 ms for matching an object with more than 300 features on SPARCstationIPX.

## APPENDIX

In the Appendix, we show the proof of the Proposition 1.

LEMMA 4. *A necessary and sufficient condition for the linear transformations presented in (2)–(5) to commute (i.e., to arrive at the same values for $Y'$) for all $X$, $X'$, where $T$ is a general matrix (see Fig. 1) is*

$$H'^{\frac{1}{2}} U H^{-\frac{1}{2}} = T \tag{12}$$

*for some orthogonal matrix $U$, where,*

$$H' = A' \Sigma_{X'} A'^{T} \tag{13}$$

$$H = A \Sigma_X A^T, \tag{14}$$

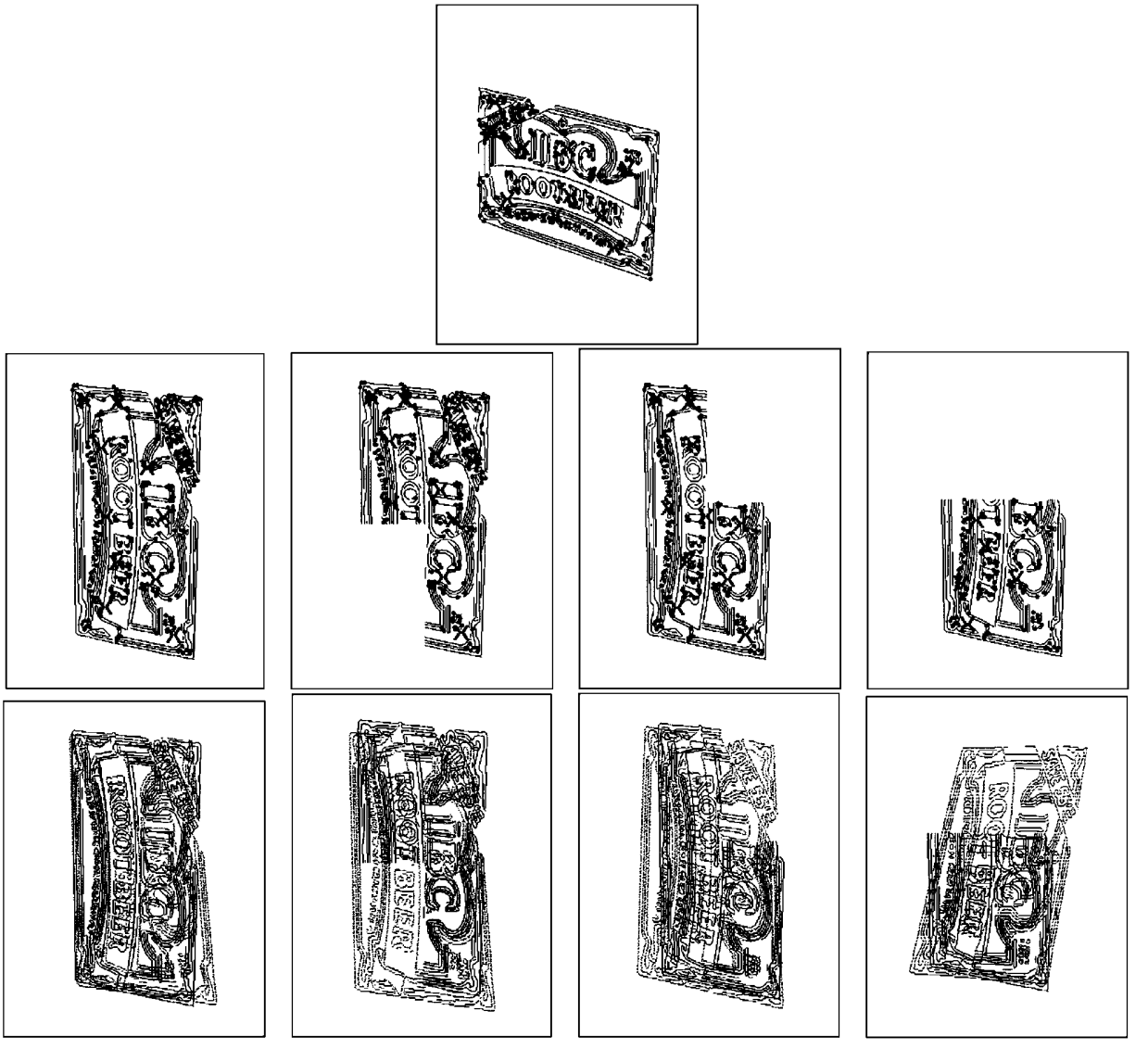

**FIG. 16.** Tests on occluded image of the Beer-Box. Top figure shows the model with extracted cluster centroids superimposed with large crosses. Middle row shows results using complete data, data for which one quarter of the lower left corner is dropped (22% of the original features), the data where the upper right corner part is dropped (35% loss of features), and data for which the upper half of the patch was totally lost (56% loss of the features). The third row shows the reconstructed data, ordered in the same order as the second row. The errors in the recovered affine parameters were: for the first from the left 0.048, second 0.059, third 0.092, and for the fourth 0.191. Here, the translational component was also considered as the parameters of affine matching, and was handled using one of produced cluster centroids.

*where $\Sigma_X$ and $\Sigma_{X'}$ represent the covariance matrices of $X$ and $X'$, respectively.*

*Proof.*

$$\Sigma_{X'} = L\,\Sigma_X L^T. \tag{15}$$

Substituting (15) into (13) yields

$$A' L\,\Sigma_X L^T A'^T = H'. \tag{16}$$

On the other hand from (14) we have

$$\Sigma_X = A^{-1}H(A^T)^{-1} = A^{-1}H(A^{-1})^T. \tag{17}$$

Then, substituting (17) into (16) yields

$$(A' L A^{-1})H(A' L A^{-1})^T = H'. \tag{18}$$

Since $H$ and $H'$ are positive definite symmetric matrices, (18) can be rewritten as

$$(A' L A^{-1} H^{\frac{1}{2}})(A' L A^{-1} H^{\frac{1}{2}})^T = H'^{\frac{1}{2}}(H'^{\frac{1}{2}})^T, \tag{19}$$

where $H^{\frac{1}{2}}$, $H'^{\frac{1}{2}}$ are again positive definite symmetric matrices.

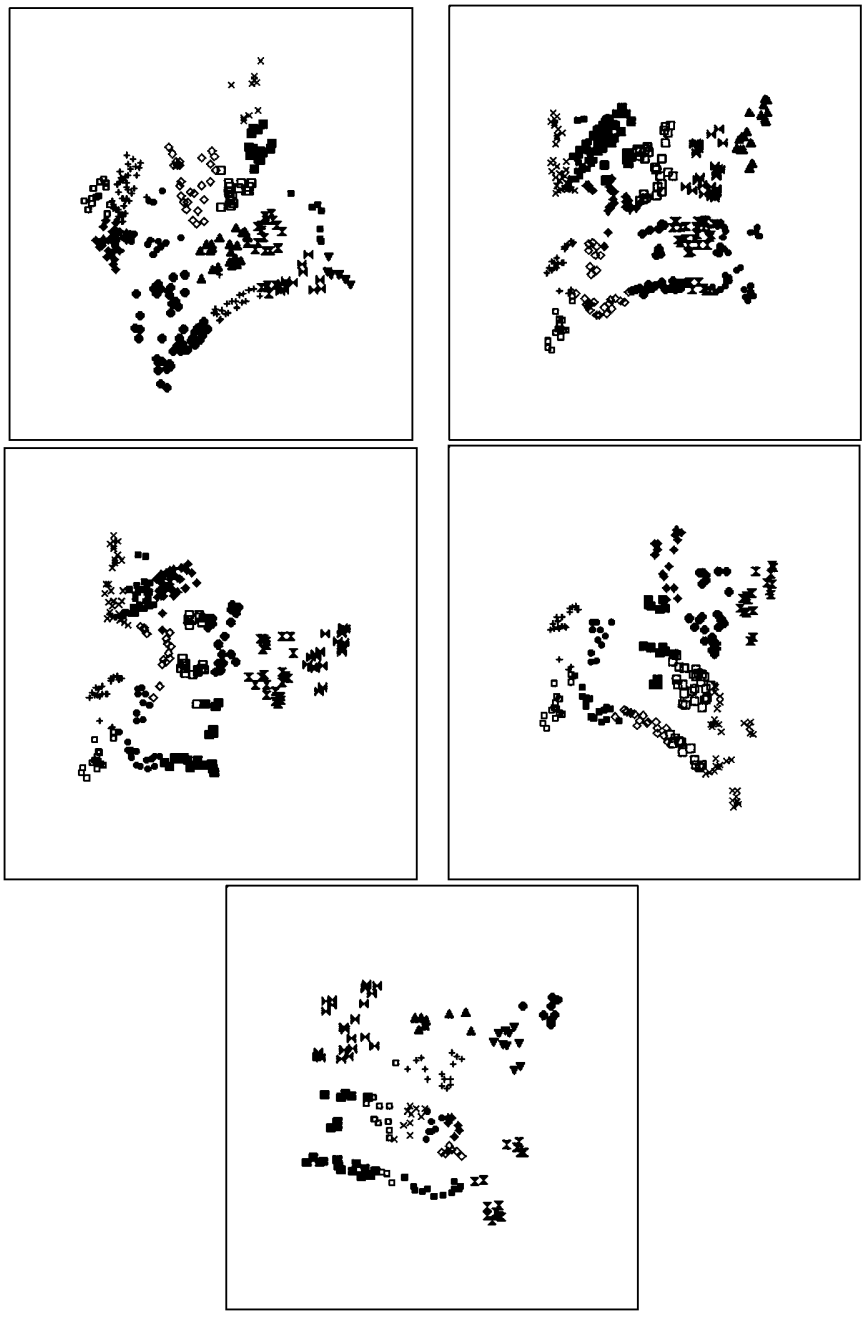

**FIG. 17.** Results of clustering by Kmean on the Beer-Box pictures. Top left is for the model, top right is for almost complete corresponding data patch, middle left is for the data with a drop of the quarter of the patch in its lower left corner, middle right is for the data for which the upper right corner is dropped, and bottom is for the data with almost upper half being dropped. It shows how the nearest-mean procedure on the normalized coordinate find corresponding concentrations of the features between the model and the data even with considerable drops/occlusions of the patches, providing a quite accurate recovery of affine parameters.

Then, from (19)

$$A'LA^{-1}H^{\frac{1}{2}} = H'^{\frac{1}{2}}U, \tag{20}$$

for some orthogonal $U$. Thus, we get

$$A'LA^{-1} = H'^{\frac{1}{2}}UH^{-\frac{1}{2}}, \tag{21}$$

where $U$ is an orthogonal matrix.

Then, combining (21) with $A'L = TA$, finally we reach (12). Clearly, (12) is also a sufficient condition. ■

Now, if we limit $T$ to orthogonal transformations, the following Lemma completes the proof of the proposition.

LEMMA 5. *As long as we are allowed to use only up to second-order statistics of the feature set, a necessary and sufficient*

*condition that $T$ in (12) is an orthogonal matrix is*

$$H' = H = c^2 I, \tag{22}$$

*where $I$ is the identity matrix and $c$ is an arbitrary scalar constant.*

*Proof.* Using the assumption that $T$ is an orthogonal matrix, from (12), we have

$$I = TT^T \tag{23}$$

$$= \{H'^{\frac{1}{2}} U H^{-\frac{1}{2}}\}\{H'^{\frac{1}{2}} U H^{-\frac{1}{2}}\}^T \tag{24}$$

$$= H'^{\frac{1}{2}} U H^{-1} U^T H'^{\frac{1}{2}}. \tag{25}$$

Rearranging this we get

$$U^T H' = H U^T. \tag{26}$$

Since as described we have no way to determine the matrix $U$, Eq. (26) must hold true for any orthogonal matrix $U$. Then, as $H$ and $H'$ are positive definite,

$$H = H' = c^2 I, \tag{27}$$

where $c$ is an arbitrary scalar constant. ∎

## REFERENCES


1. T. D. Alter and W. E. L. Grimson, Fast and robust 3D recognition by alignment, in *Fourth International Conference on Computer Vision, Berlin, Germany, 1993.*
2. S. Ando, Gradient-based feature extraction operators for the classification of dynamical images, in *Transactions of Society of Instrument and Control Engineers in Japan*, Vol. 25, No. 4, pp. 496–503, 1989. [In Japanese]
3. S. Ando and K. Nagao, Gradient-based feature extraction operators for the segmentation of image curves, in *Transactions of Society of Instrument and Control Engineers in Japan*, Vol. 26, No. 7, pp. 826–832, 1990. [In Japanese]
4. H. Asada and M. Brady, Curvature primal sketch, *IEEE Trans. Patt. Anal. Machine Intell.* **PAMI-8**, 1986, 2–14.
5. Kalle Åstroöm, Affine invariants of planar sets, in *Proc. IAPR SCIA'93, 1993*, pp. 769–776.
6. J. F. Canny, A computational approach to edge detection, *IEEE Trans. Patt. Anal. Machine Intell.* **PAMI-8**, 1986, 34–43.
7. D. Cyganski and J. A. Orr, Applications of tensor theory of object recognition and orientation determination, *IEEE Trans. Patt. Anal. Machine Intell.* **PAMI-7**(6), November 1985.
8. D. Cyganski and J. A. Orr, The applications of image tensors and a new decomposition, *Pattern Recognition Theory and Applications* (P. A. Devijver and J. Kitter, Eds.), NATO ASI Series, Vol. F30, pp. 481–491, Springer-Verlag, Berlin/Heiderberg, 1987.
9. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, 1972.
10. K. Fukunaga and W. L. G. Koontz, A criterion and an algorithm for grouping data, *IEEE Trans. Comput.* **C-19**, October 1970, 917–923.
11. W. E. L. Grimson, *Object Recognition by Computer*, MIT Press, Cambridge, MA, 1991.
12. D. P. Huttenlocher and S. Ullman, Recognizing solid objects by alignment with an image, *Int. J. Comp. Vision* **5**(2), 1990, 195–212.
13. M. Iri and T. Kan, *Linear Algebra*, pp. 120–147, Kyouiku-Syuppan, 1985. [In Japanese]
14. M. Iri and T. Kan, *Introduction to Tensor Analysis*, pp. 207–261, Kyouiku-Syuppan, 1973. [In Japanese]
15. J. J. Koenderink and A. J. Van Doorn, Affine structure form motion, *J. Opt. Soc. Am.* **8**, 1991, 377–385.
16. Y. Lamdan, J. T. Schwartz, and H. J. Wolfson, Affine invariant model based object recognition, *IEEE Trans. Robotics Automation* **6**, 1988, 238–249.
17. D. Lovelock and H. Rund, *Tensors, Differential Forms, and Variational Principles*, pp. 1–53, Dover, 1975.
18. J. MacQueen, Some methods for classification and analysis of multivariate observations, in *Proc. 5th Berkeley Symp. on Probability and Statistics, 1967*, pp. 281–297.
19. K. Nagao, M. Sohma, K. Kawakami, and S. Ando, Detecting contours in image sequences, in *Transactions of the Institute of Electronics, Information and Communication Engineers in Japan on Information and Systems, 1993*, Vol. E76-D, No. 10, pp. 1162–1173. [In English]
20. W. H. Press *et al.*, *Numerical Recipes in C*, pp. 610–614, Cambridge Univ. Press, Cambridge, UK, 1985.
21. A. Shashua, *Correspondence and Affine Shape from two Orthographic Views: Motion and Recognition*, A. I. Memo No. 1327, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, December 1991.
22. M. J. Swain, *Color Indexing*, Ph.D. thesis, Chap. 3, University of Rochester Technical Report No. 360, November 1990.
23. S. K. Nayar and R. M. Bolle, Reflectance ratio: A photometric invariant for object recognition, in *Proc. Fourth ICCV Conf., 1993*, pp. 280–285.
24. T. F. Syeda-Mahmood, Data and model-driven selection using color regions, in *Proc. ECCV Conf., 1992*, pp. 321–327.
25. W. B. Thompson, K. M. Mutch, and V. A. Berzins, Dynamic occlusion analysis in optical flow fields, *IEEE Trans. Patt. Anal. Machine Intell.* **PAMI-7**, 1985, 374–383.
26. S. Ullman and R. Basri, Recognition by linear combinations of models, *IEEE Trans. Patt. Anal. Machine Intell.* **PAMI-13**, 1991, 992–1006.