Extracting Salient Image Features Using Outlier Detection Techniques

Dimitri Lisin, Edward Riseman, and Allen Hanson

Department of Computer Science, UMASS - Amherst, Amherst, MA, USA

Abstract. This paper presents an efficient method for finding salient differential features in images. We argue that the problem of finding salient features among all the possible ones is equivalent to finding outliers in a high-dimensional data set. We apply outlier detection techniques used in data mining to devise a linear time algorithm to extract the salient features. This yields a definition of saliency which rests on a more principled basis and also produces more reliable feature correspondences between images than the more conventional ones.

1 Introduction

Extracting *salient* features is often a crucial preprocessing step for image analysis. Many vision tasks, such as object recognition, object detection, and stereo matching require establishing correspondences between point features from different images. Most often it is infeasible to consider potential features at every pixel in the image, so only a key subset, called *salient features*, are used.

In the worst case the salient features are identified by hand [2], but more often a general definition of saliency is used. Often corners, are considered salient [13], and also edges and blobs can be added to the set of salient features [8].

In this paper we define an image feature the same way as in [8], as a vector of Gaussian derivative responses at a pixel over a range of scales, In particular, we use the first and the second derivatives over three consecutive scales. Thus, a feature is defined at every pixel of every scale of the image, except for the top-most and the bottom-most scales. We use the normalized inner product between two feature vectors as a measure of similarity between the corresponding features. The normalization provides a degree of invariance to linear changes in intensity.

We argue that the space of all features in an image can be viewed as a large multi-dimensional data set, and the *salient points correspond to the outliers of the set.* We show that density-based outlier detection techniques [5], [1] used in data mining are applicable to the problem of finding salient features. We use these techniques together with the smoothness property of the feature space described in section 4 to devise a linear time algorithm to detect the salient features. Our definition of saliency rests on a more principled basis, and intuitively it seems to be more natural than those commonly used because it is derived from the structure of the feature space itself.

We present experiments in which we establish correspondences between features in pairs of images using only the feature similarity and the self-consistency constraint described in section 7. No other constraints are assumed. The experiments show, that features that are salient according to our definition produce more reliable correspondences than ones defined by more traditional means. Of course, many systems that require correspondences do use other constraints to successfully disambiguate the matches [13], [8]. However, extracting features that are salient according to our definition can be a useful preprocessing step, which may improve the overall accuracy of such systems and it may reduce the time required for disambiguating the matches.

2 Salient Features as Outliers

The problem of finding outliers in high-dimensional data sets arises in data mining. It is assumed that points in a data set form clusters, and outliers are points that do not belong to any of the clusters. In the context of data mining outliers represent unusual cases, such as fraud and other criminal activity in e-commerce.

Hawkins [4] defines an outlier as "an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism." On the other hand, the word *salient* is defined by the Webster's Dictionary as *prominent*. It is the opinion of the authors that the two definitions really point to the same concept. In particular, a multi-dimensional space of image features is really nothing more than a large data set. Intuitively we can see that features similar to many others would form clusters, while distinct features, which we might call *salient*, would be the outliers.

In contrast, one could also examine the clusters that exist in the feature space, as opposed to the outliers. Patterns formed by these clusters also carry information about the appearance of the scene in an image, and may be useful for recognition. Such an approach would be similar to the histogram-based technique described in [9], which uses the global distribution of differential features for recognition. However, this paper deals with point features, which may be used in applications other than object recognition, such as stereo matching and object detection. Point feature correspondences also offer ways to recover the three dimensional structure of scenes and objects, which is not possible with purely appearance-based approaches. Therefore, it is not the clustering, but the outlier detection algorithms that interest us.

In this paper we examine two such algorithms: the distance-based $DB(p, D)$ scheme [5], and the Local Outlier Factor approach [1]. We will show that with minor modifications both are applicable to our problem, and that the latter yields superior results.

3 Related Work

This work has been inspired by [8], which presents a system for learning differential features to recognize objects. In this paper, however, we only focus on the problem of establishing correspondences between point features. We propose methods for finding features that are likely to be matched correctly, and we defer the question of how to use them to the future work. We will present an overview of the existing definitions of salient point features, but we consider salient edges, curves, etc. to be outside the scope of this paper.

3.1 Saliency

Most often salient point features are defined as the local extrema of some function of the image. One example is using corners, or points of high curvature as salient [13]. Also, local maxima of "blobs" (the trace of the Hessian) and the gradient magnitude can be used [8]. Since such functions are combinations of image derivatives they can be computed very fast. Another interesting example is presented in [11], where a multiscale decomposition of an image is computed using a 1D wavelet at various orientations, and the local maxima of the sum of the wavelet responses are used as salient features.

A definition of saliency most similar to the one presented by this paper is given is [12]. In this work the feature components are the differential invariants at a pixel over a range of scales, and the Mahalanobis metric is used as feature distance. The saliency is defined in terms of the density of the feature space. Lower density regions correspond to higher saliency. This makes sense, because features at the low density regions of the space are unlike most others, and therefore are less likely to be mismatched.

In [12] a multivariate Gaussian mixture model of the feature space is used as a density function, whose local minima are considered salient. The main drawback of this approach is its time complexity, which is quadratic in the total number of features in the space. The problem can be alleviated by modeling a randomly sampled subset of the features instead of the entire space. This reduces the number of features that need to processed, but the time complexity is still $O(n^2)$.

The approach described in this paper is similar to that of [12] in that it uses the density of the feature space to define saliency. However, we use local outlier detection techniques which give us a more precise and well-founded definition of saliency. We also use the smoothness property of the feature space to reduce the time complexity to linear.

3.2 Multiscale Differential Features

The Gaussian and its derivatives are a family of kernels used to generate a linear isotropic scale-space of an image, which has been studied extensively under scale-space theory [7]. Image derivatives define the local behavior of the intensity surface, which makes them useful for describing the image features. Using Gaussian derivative filters at a range of σ's allows us to analyze the surface patches of varying sizes.

Our definition of a feature at a pixel and a particular scale σ_i is a vector of Gaussian derivative responses at three scales: $\sigma_{i-1}, \sigma_i, \sigma_{i+1}$. We use the first and the second derivatives [8]. Since we have 2 components of a first derivative, and

3 components of the second over 3 scales, our feature space has 15 dimensions. Using multiple scales increases the specificity of a feature.

This feature representation is not invariant to in-plane rotation. For the purpose of this paper we set up our experiments so that such invariance is not required. If, however, it is required the steerability property of the Gaussian derivative filters can be used as in [8], or, alternatively, the rotationally invariant combinations of derivatives can be used as in [12]. Our definition of saliency should still be applicable in these cases, but more experiments are needed to be certain.

4 Fast Density Estimation Using Smoothness

The outlier detection techniques that we consider in this paper use a notion of density of the data points in the space to find the outliers. We therefore need a way to compute the density of our feature space at every feature. One example of such a method is presented in [12], and it has been discussed in section 3.1. Recall that it takes $O(n^2)$ time, where n is the number of features.

A simpler way to estimate local density at a particular feature f is to compute the distance to the farthest of its k nearest neighbors, for some natural number k. We will denote the farthest neighbor as f_d, and the distance from f to f_d as $r(f)$, which is the radius of the smallest hyper-sphere containing the k nearest neighbors of f. Since we use a similarity measure rather than distance between features, in our case $r(f)$ is the similarity to the least similar of the neighbors. The problem with this approach is that first we need to determine what the k nearest neighbors of a feature are. This would require comparing each feature to every other feature, and would also take $O(n^2)$ time.

In this section we will show that our feature space is smooth, such that features that are neighbors in the image, also tend to be neighbors in the feature space. This will allow us to treat the nearest neighbors of a feature in the image as its nearest neighbors in the feature space, and reduce the time complexity of the density estimation to $O(n)$.

A Gaussian derivative response at some pixel (x, y), and some scale σ is obviously a function of x, y, and σ. Therefore, the features form a 3D manifold in a 15D space. Because we use Gaussian derivatives, the image at each scale σ is blurred which causes its derivatives to be smooth. As a result, the whole manifold has to be smooth, especially for the coarser scales. Figure 1 illustrates this idea. The top row shows the image of a mobile robot at 3 scales, and the bottom row shows a plot of the first x derivative (I_x) vs. the first y-derivative (I_y) of each corresponding scale plane. Each plot is a 2D projection of a plane's sub-manifold. We can see that the sub-manifolds, as expected, become progressively smoother as we move to coarser scales.

The smoothness property directly implies that features that are neighbors in the image (*image-neighbors*) also tend to be neighbors in the feature manifold (*actual-neighbors*). We make a stronger assumption:

Conjecture 1. 8 *immediate* neighbors of a feature in the image are also its 8 *nearest* neighbors in the feature manifold.

Conjecture 1 gives us 8-nearest actual-neighbors of a feature, “for free” without us having to search the entire feature space. This is what allows us to reduce the time complexity to linear. This assumption would not hold true for images with high contrast repetitive patterns. Because of this our approaches may not work well for the natural outdoor scenes, but it should be quite suitable for the indoor ones.

Table 1 shows the results of an experiment supporting our assumption. In this experiment, we took an image of size 106 x 85 and compared the 8 nearest image-neighbors of each feature in each scale plane to its actual-neighbors from the same scale plane. The first column shows the scale plane, and the second column shows the number of image-neighbors that also happen to be among the 8 nearest actual-neighbors of a feature, averaged over all 9010 features in the scale plane. We see that on average, over half of the 8 image-neighbors are also among the 8 nearest actual-neighbors.

The third column shows the actual rank of the 8th image-neighbor of a feature averaged over all features in a scale plane. Starting with scale 3 the 8th image-neighbor on average falls among the 50 nearest actual-neighbors. This is not bad, considering that each scale plane contains 9010 features.

The fourth column shows the relative error of our density estimate. It is actually the relative error of the similarity of a feature f to its 8th nearest image-neighbor with respect to the similarity of f to its 8th actual-neighbor. The relative error is averaged over all features in the scale plane. The last three columns provide a reference to see how significant the relative errors are. Column five shows the average relative error we would get if we always used the 100th actual-neighbor. Similarly the last two columns show the average relative errors for the 200th and 500th actual-neighbors. We can see that our relative errors are significantly less than those for the 100th, 200th and 500th actual-neighbors.

5 A Naive Approach to Outlier Detection

A simple approach to find outliers is to compute the density, $r(f)$, for each feature f at every pixel in the image at every scale to create a density map. The local minima of the density correspond to salient features. We call this approach naive, because we present a more sophisticated scheme in section 6.

This algorithm runs in linear time in the number of features. It takes $8N$ feature comparisons to compute $r(f)$ for every feature f and $26N$ comparisons of floating point numbers to find the local minima across scales, where N is the total number of features. The dimensionality of the feature space only affects the time it takes to compare two features, and this dependency is also linear.

This algorithm is related to the distance-based $DB(p, D)$ outlier detection scheme presented in [5]. In the scheme a data point o in a data set T is considered an outlier if at least fraction p of the points in T lie greater than distance D

from o. Essentially, a DB outlier minimizes the number of neighbors it has within a fixed hyper-sphere, and a naive salient feature maximizes the hyper-sphere containing its 8 nearest neighbors. In effect, they both minimize the ratio of the number of neighbors to the volume of the hyper-sphere containing them, i. e. the local density of the space.

[5] formally shows that the DB scheme is a generalization of statistical outlier tests for the normal and the Poisson distributions. This justification also applies to our naive approach, since we have shown its essential equivalence to DB.

6 Local Outlier Factor

Breunig [1] describes a more sophisticated algorithm for finding density-based local outliers using k-nearest neighbors, which, when applied to finding salient features, yields results superior to that of the naive approach from section 5. The algorithm computes a Local Outlier Factor (LOF) for each data point, which is a degree to which it is an outlier. $LOF(p)$, where p is a data point, is defined as the average of the ratios of densities at p's neighbors to the density at p. This scheme considers a point an outlier when its density is low relative to the densities at its neighbors, as opposed to the DB approach, which simply looks for low absolute density. It is more reasonable, because low density alone may not necessarily be characteristic of an outlier, e.g. in a case when the whole data set is very sparse. In section 7 we show empirical justification for preferring LOF over the naive approach.

The algorithm is built upon several key concepts. The first one is *k-distance* of a data point. Let D be the data set, let p, o be data points, $p, o \in D$, and let k be a positive integer. *k-distance*(o) is defined as the distance to the furthest of the k nearest neighbors, which makes it equal to $r(f)$ from section 5.

The reachability distance from p to o is defined as

$$reach\text{-}dist_k(p, o) = \max(k\text{-}distance(o), dist(p, o)) \tag{1}$$

In essence, this definition says that if point p is inside the *k-distance* neighborhood of o then *reach-dist*(p, o) is the *k-distance* of o. Otherwise, it is just the distance between p and o according to whatever distance metric is generally used for this data set. This is done to smooth out the statistical fluctuations of $dist(p, o)$ when p and o are close. It should also be noted that *reach-dist* is not reflexive, i. e. *reach-dist*$(p, o) \neq$ *reach-dist*(o, p).

The reachability distance is used to define local reachability density (lrd):

$$lrd_k(p) = 1 / \left(\frac{\sum_{o \in N(p)} reach\text{-}dist_k(p, o)}{|N(p)|} \right), \tag{2}$$

where $N(p)$ is the set of k nearest neighbors of p. In essence, $lrd_k(p)$ is the inverse of the average reachability distance of the neighbors of p. If we used plain distances instead of *reach-dist* then $lrd(p)$ would just be the average distance from p to its neighbors. But because of the smoothing effect of *reach-dist*, $lrd(p)$ is restricted to never be less than the average of the *k-distances* of its neighbors.

Using the local reachability density the local outlier factor (LOF) is defined:

$$LOF_k(p) = \left(\sum_{o \in N(p)} \frac{lrd_k(o)}{lrd_k(p)} \right) / (|N(p)|) \qquad (3)$$

LOF is the average ratio of the reachability densities of p's *k-distance* neighbors to that of p. The lower the density of p, relative to its neighbors, the higher the LOF(p), the degree of p being an outlier.

To be able to apply LOF to finding salient features all we need to do is set $k = 8$, and convert our feature similarity measure into distance by inverting it. Since LOF only assigns a saliency value to a feature, we need to impose some threshold to be able to say which features are salient enough and which ones are not. This is a question that deserves further exploration, but at this point we simply put the threshold 3 standard deviations above the mean of the LOF values at each scale plane.

The time complexity of LOF is still linear: $8N$ feature comparisons to compute the *k-distance* of each feature, plus $8N$ floating point comparisons to determine the *reach-distance* of each feature, plus $8N$ floating point operations to compute *lrd* of each feature, plus $8N$ floating point operations to compute LOF's. In practice it may be slightly slower than the naive approach.

7 Results

During testing we we try to establish correspondences between pairs of images using the self-consistency constraint, similar to the one in [6] and [10]. Let F_1 and F_2 be the sets of salient features from two images. We call $f \in F_1$ and $f' \in F_2$ a self-consistent match if features f and f' are mutually maximally similar. This ensures that the matches are bidirectional between the two images, and increases their reliability.

We tested our approach using synthetic images, where correctness of the matches could be verified automatically, and real images, where where the ground truth had to be supplied by a person. We wrote a simple GUI, which shows the reference image with a particular feature f, and the target image with matching feature f', and lets a user indicate whether or not the match is correct by pressing a key. We compute the percentage of correct matches, and use it as a measure of quality of each approach. No special preprocessing, such as histogram thresholding, is done on the test images before the salient features are extracted.

To evaluate our approaches we compare the percentages they yield to those produced by using corners and edges, which are common ways to extract salient features. We define corners the same way as [8] as local maxima of

$$s_{corner}(\sigma) = \sigma^4 |I_{yy}^2 I_{xx} - 2 I_x I_y I_{xy} + I_{xx}^2 I_{yy}|, \qquad (4)$$

where image derivatives are computed by convolving the image I with a Gaussian derivative filter of the appropriate σ. Edges are defined as local maxima of the gradient magnitude. Unlike [8] we compute the local maxima in each scale plane, as opposed to over the entire scale volume. This reduces the amount of computation required and seems to yield greater matching accuracy.

The LOF approach takes into account relative feature space density at a particular feature, so we see no need to use local maxima. Instead we use a threshold .3 times the standard deviation above the mean. This threshold was determined empirically, and further investigation is required to find a more principled way of setting it.

For each saliency definition, we extracted the sets of salient points from the reference and the target images, and sorted them by their saliency. Then, proceding down the reference set in order, we found the first 100 self-consistent matches, and sorted those by the similarity between the matched pairs of features. We took the top 25 of those to be tested for correctness.

7.1 Synthetic data

To generate synthetic data we used an image of a lab, containing a mobile robot (the target from figure 3), whcih is 320x240 pixels. We cropped out 10 randomly chosen sub-images of size 100x100 pixels from it, and generated a range of scaled images from them by subsampling and interpolation. The scaled images range in size from 50% to 200% of the original sub-image, with the increment of 10%. The matching accuracy is averaged for each scale over the 10 randomly chosen sub-images.

The results of this experiment are summarized in a graph in figure 2. The x-axis represents the scale factor of the sub-image, and the y-axis shows the average matching accuracy as a percentage. The corners performed the worst. Our naive approach was better then the edges when the sub-image was unchanged (scale factor 1), but degraded much less gracefully when the scale changed. The LOF approach performed the best. In fact its accuracy was 50% or higer for the scale factors ranging from .8 to 1.4.

It took 124 seconds to test the corners, on average .8 seconds per image. Edges took approximately the same time, and LOF took 417 s., or 2.6 s. per image. The computation was done on a 2GHz Pentum 4 running Linux. The system can be further optimized, and the computation is naturally parallelizable, since each scale plane can be processed independently.

7.2 Real Data: Task 1

In this series of experiments we matched features from multiple reference images of a mobile robot into a single target, a cluttered scene containing the robot (Figure 3). The reference images 0 - 4 vary in scale with respect to the target. References 5 and 6, while having roughly the same scale, exibit out-of-plane rotation and a background change.

The experimental results for this task are summarized in table 1. It shows that the naive approach performed significantly better then the corner but not as well as the edges. The LOF, however, did as well as the edges in cases 0, 1, and 2, somewhat worse in case 3, and significantly better in cases 4, 5, and 6. Reference 4 has the largest difference in scale from the target, as well as partial occlusion, and references 5 and 6 have out-of-plane rotation, as we mentioned above. Thus the LOF approach appears to be the most robust.

7.3 Real Data: Task 2

In this task we took three pairs of images produced by cameras mounted on a stereo head and tried to match features from one image to another in each pair. The camera's optical axes are not parallel, so the images in each pair are quite different. No calibration information or any other constraints associated with stereo were used.

The results for the three pairs of images, which we called "coke", "drill", and "blocks" (Figure 4), are summarized in table 2. Here again the LOF approach performed the best.

8 Conclusions and Future Work

In this paper we have presented a novel idea that salient differential features can be viewed as outliers in a high-dimensional space, and therefore outlier detection techniques used in data mining are applicable for their extraction. We have presented two algorithms based on two different outlier detection schemes: the distance-based approach and the LOF. The latter one is a more sophisticated approach, which better fits an intuitive notion of an outlier, and produces more reliable features.

Further investigation is needed to determine if our concept of saliency can be applied to image features defined by using other means, such as the differential invariants, and using other distance metrics, such as the Mahalanobis distance.

This work is a part of an ongoing effort to build a learning system for object recognition and detection capable of handling an unconstrained environment. The LOF-based salient feature extraction may become an important component of such a system.

References

1. Breunig M. M., Kriegel, H.-P., Ng, R., Sander, J. "LOF: Identifying Density-Based Local Outliers," *Proc. ACM SIGMOD Int. Conf. on Management of Data*, (2000).
2. Delaert, F., Seitz, S. M, Thorpe, C. E., Thrun, S., "Structure from Motion Without Correspondences," *Proc. Computer Vision and Pattern Recognition Conf.*, (2000).
3. Freeman, W. T., and Adelson, E. H. "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 9 (1991).
4. Hawkins, D., *Identification of Outliers*, Chapman and Hall, London, (1980).

5. Knorr, E., Ng, R. "Algorithms for Mining Distance-based Outliers in Large Datasets," *Proc. of 24th Int. Conf. On Very Large Data Bases*, (1998).
6. Leclerc, Y.G., Q.T. Luong, et al., "Self-consistency: A novel approach to characterizing the accuracy and reliability of point correspondence algorithms," DARPA Image Understanding Workshop, (1998)
7. Lindeberg, T. "Feature detection with automatic scale selection," *Intl. Journal of Computer Vision* 30, 2 (1998).
8. Piater, J. *Visual Feature Learning.* Ph.D. dissertation, Dept. of Computer Science, University of Massachusetts Amherst, (2001).
9. Ravela, S. and A. Hanson, "On Multi-Scale Differential Features for Face Recognition," Vision Interface, Ottawa, (2001).
10. Schultz, H., A. R. Hanson, E. M. Riseman, F. R. Stolle, D. Woo, Z. Zhu, "A Self-consistency Technique for Fusing 3D Information," Invited talk presented at the IEEE 5th Intl. Conf. on Information Fusion, Annapolis, Maryland, (2002).
11. Shokoufandeh, A., Marsic, I., Dickinson, S.J. "View-Based Object Recognition Using Saliency Maps," TR DCS-TR-339, Dept. of Computer Science, Rutgers University, (1998).
12. Walker, K. N., Cootes, T. F., Taylor, C. J. "Locating Salient Object Features," Proc. of BMVC, (1998).
13. Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q-T. "A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry," *Artificial Intelligence* 78, 1-2, (1995).

Table 1. Empirical Evidence for Conjecture 1

scale	8-nearest	max rank	% relative error	% error 100	% error 200	% error 500
1	3.8	167.9	4.39	4.46	6.75	11.25
2	4.5	72.87	1.86	2.99	4.58	7.92
3	4.6	46.70	1.09	2.12	3.30	5.79
4	4.5	39.28	0.77	1.64	2.58	4.64
5	4.5	33.86	0.58	1.48	2.27	4.08
6	4.4	31.21	0.48	1.38	2.09	3.76
7	4.3	32.76	0.40	1.32	2.04	3.68
8	4.3	27.66	0.27	1.21	2.02	3.79
9	4.5	23.46	0.15	0.99	1.77	3.69
10	4.7	21.71	0.09	0.67	1.23	2.72

Table 2. Matching accuracy for task 1

Ref.	Corners	Edges	Naive	LOF
0	76%	100%	100%	100%
1	52%	96%	92%	96%
2	40%	92%	64%	92%
3	48%	84%	48%	76%
4	0%	28%	8%	44%
5	12%	36%	44%	80%
6	4%	28%	40%	52%

Table 3. Matching accuracy for task 2

	Corners	Edges	Naive	LOF
coke	28%	24%	68%	72%
drill	12%	60%	48%	60%
blocks	12%	24%	20%	48%

Fig. 1. Scale planes 1, 5, and 9, with $\sigma = \sqrt{2}, 4\sqrt{2}, 16\sqrt{2}$ respectively, and their corresponding feature sub-manifolds

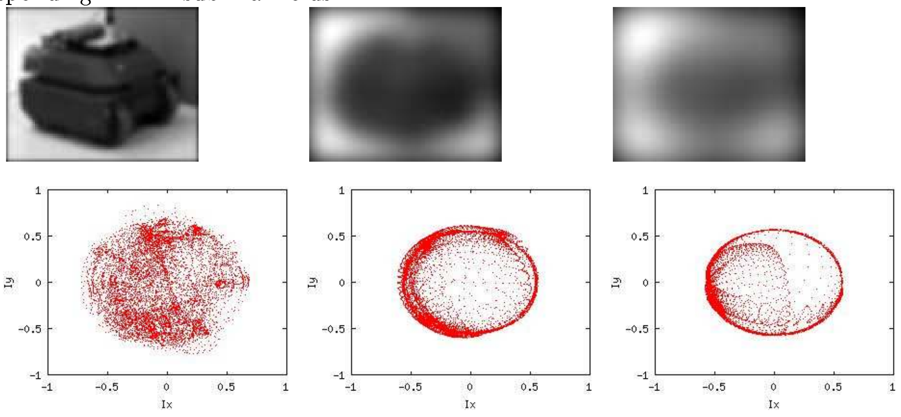

Fig. 2. Accuracy for synthetic data

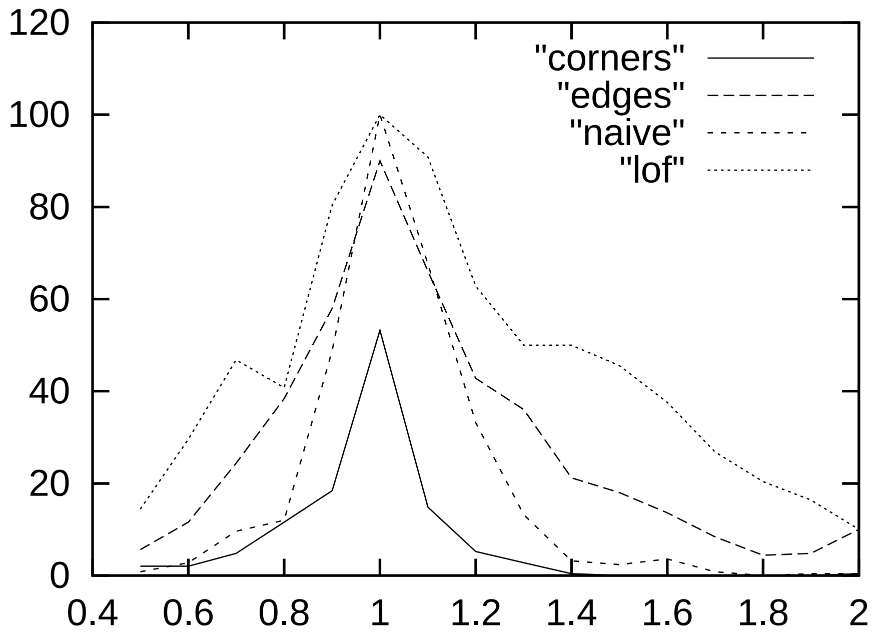

Fig. 3. The reference and target images for task 1.

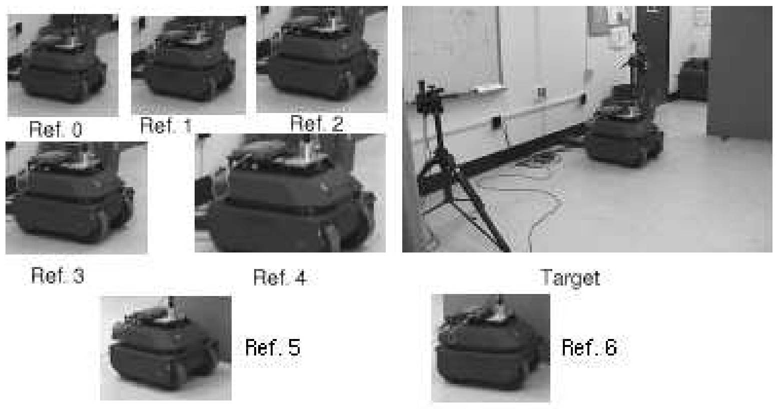

Fig. 4. The image pairs ("coke", "drill", and "blocks") with 25 self-consistent salient features extracted using LOF. The circles' sizes correspond to features' scales.