Retrieving Objects Using Local Integral Invariants

Alaa Halawani[1] and Hashem Tamimi[2]

[1] Chair of Pattern Recognition and Image Processing, University of Freiburg, 79110 Freiburg, Germany
halawani@informatik.uni-freiburg.de
[2] Computer Science Dept., University of Tübingen, Sand 1, 72076 Tübingen, Germany
tamimi@informatik.uni-tuebingen.de

Abstract. The use of local features in computer vision has shown to be promising. Local features have several advantages including invariance to image transformations, independence of the background, and robustness in difficult situations like partial occlusions. In this paper we suggest using local integral invariants to extract local image descriptors around interest points and use them for the retrieval task. Integral invariants capture the local structure of the neighborhood around the points where they are computed. This makes them very well suited for constructing highly-discriminative local descriptors. We study two types of kernels used for extracting the feature vectors and compare the performance of both. The dimensionality of the feature vector to be used is investigated. We also compare our results with the SIFT features. Excellent results are obtained using a dataset that contains instances of objects that are viewed in difficult situations that include clutter and occlusion.

1 Introduction

One of the main difficulties in computer vision is to identify objects despite of the changes that may affect the appearance of the object. Possible changes include rotation, translation, changes in scale, changes in illumination conditions, and partial object occlusion. Moreover, objects are usually located in cluttered scenes. It is very important to take these facts into account when designing real-world applications. One of the well-known approaches that deal with object and image retrieval is the use of color histograms [1]. The use of color histograms is simple and fast but it works mainly for non-cluttered scenes or for pre-segmented objects. In [2], Schiele and Crowley proposed using multidimensional receptive field histograms for object recognition. The histograms are constructed from the responses of a vector of local linear neighborhood operators such as gradients, Laplacian, Gabor filters, and Gaussian derivatives. This method is also problematic when dealing with clutter and partial occlusion. Local invariant features computed around interest points have several interesting characteristics. First of all, they are invariant to image transformations like rotation and translation. They are also robust to partial occlusion, clutter,

H. Sundaram et al. (Eds.): CIVR 2006, LNCS 4071, pp. 251–260, 2006.

and changes in the background, as only corresponding local features in different scenes should match. This also eliminates the need for any prior segmentation. Schmid and Mohr [3] were among the first to use local image descriptors for the retrieval task. They have used rotation-invariant local feature vectors based on derivative-of-Gaussian local measurements. To identify the interest points, they have used the well-known Harris corner detector [4]. Gout and Boujemaa [5] extended this work to color images using color differential invariants. The feature vectors used in these methods have low dimensionality and are not highly distinctive. Lowe [6] has introduced the SIFT features that are invariant against similarity transformations. The feature vector used in SIFT consists of gradient orientation histograms summarizing the content of a 16×16 region around each point. In this paper we explore the use of local integral invariants for the purpose of retrieving objects located in complex scenes. Integral invariants capture the local structure of the neighborhood around the points where they are computed. Global versions of these features have proven to be robust to independent motion of objects, articulated objects, and topological deformations [7]. In addition to their invariance to Euclidean motion, we explain how it is possible to earn local similarity-invariant features. We study two types of kernels that are used to generate integral invariants. The effect of the dimensionality of the feature vectors is also investigated. A comparison with the SIFT features is also considered.

The paper is organized as follows. In section 2 a summary of the integral invariants is given. Extending the features to be scale invariant is described in section 3. Section 4 gives an overall look on the setup of the system. Results are summarized in section 5 and a conclusion is given in section 6.

2 Integral Invariants

Following is a brief description of the calculation of the rotation- and translation-invariant features based on integration. The idea of constructing invariant features is to apply a nonlinear kernel function, $f(\mathbf{I})$, to a gray-valued image, $\mathbf{I}$, and to integrate the result over all possible rotations and translations (Haar integral over the Euclidean motion):

$$\mathbf{T}[f](\mathbf{I}) = \frac{1}{PMN} \sum_{n_0=0}^{M-1} \sum_{n_1=0}^{N-1} \sum_{p=0}^{P-1} f(g(n_0, n_1, \varphi = p\frac{2\pi}{P})\mathbf{I}) \tag{1}$$

where $\mathbf{T}[f](\mathbf{I})$ is the invariant feature of the image, M, N are the dimensions of the image, and g is an element in the transformation group, $\mathcal{G}$ (which consists here of rotations and translations). Bilinear interpolation is applied when the samples do not fall onto the image grid. The above equation suggests that invariant features are computed by applying a nonlinear function, f, to a circular neighborhood of each pixel in the image, then summing up all the results to get a single value representing the invariant feature. Using several different kernel functions builds up a feature space. To preserve more local information, one can

remove the summation over all translations. This results in a map $\mathbf{T}$ that has the same dimensions of $\mathbf{I}$:

$$(\mathbf{T}[f](\mathbf{I}))(n_0, n_1) = \frac{1}{P}\sum_{p=0}^{P-1} f\left(g\left(n_0, n_1, \varphi = p\frac{2\pi}{P}\right)\mathbf{I}\right). \tag{2}$$

Two types of non-linear kernel function, f, are considered. Invariant features can be computed by applying the monomial kernel, which has the form:

$$f(\mathbf{I}) = \left(\prod_{k=0}^{K-1} \mathbf{I}(\mathrm{x}_k, \mathrm{y}_k)\right)^{\frac{1}{K}}. \tag{3}$$

Integral invariants are computed using the monomial kernels by multiplying a constellation of pixels in the circular neighborhood of the center pixel and then averaging the local results. One disadvantage of this type of kernels is that it is sensitive to illumination changes. The work in [8] defines another kind of kernels that are robust to illumination changes. These kernels are called the relational kernel functions and have the form:

$$f(\mathbf{I}) = rel\left(\mathbf{I}(\mathrm{x}_0, \mathrm{y}_0) - \mathbf{I}(\mathrm{x}_1, \mathrm{y}_1)\right) \tag{4}$$

with the ramp function

$$rel(x) = \begin{cases} 1 & \text{if } x < -\epsilon \\ \frac{\epsilon - x}{2\epsilon} & \text{if } -\epsilon \leq x \leq \epsilon \\ 0 & \text{if } \epsilon < x \end{cases}. \tag{5}$$

Feature calculation using the relational kernels is similar to using the monomial kernels. In the case of the relational kernels, two pixels are compared using the rel-function instead of multiplying them. This kind of kernels is based on the Local Binary Pattern (LBP) texture features [9], which map the relation between a center pixel and its neighborhood pixels into a binary pattern. Equation 5 extends the LBP operator to give values that fall in $[0, 1]$. This is done in order to get rid of the discontinuity of the LBP operator which makes the features sensitive to noise. For more detailed theory about integral invariants, please refer to [7].

3 Scale Invariance

The integral invariants in the form described in Section 2 are invariant to Euclidean motion only and are therefore sensitive to scale changes. Extending the principle of integral invariants to be scale invariant is not easy due to the fact that compact groups are needed, whereas scaling is unbounded [10]. However, if the local scale of each interest point can be determined, we can establish local features that are scale invariant by adapting the local support (the patch size on which the kernel function acts) of the kernels used to the local scale of the point

around which they are evaluated. This way, the patch that is used for feature extraction covers always the same details of the image independent of the scale, which consequently means that the extracted feature vector is scale invariant. To achieve this, we use the difference-of-Gaussian (DoG)-based point detector introduced by Lowe in [6], which localizes points in scale and space.

4 Setup

4.1 Dataset

To run the experiments, 11 different objects were chosen. Model images for the objects were recorded. For the purpose of testing, each object was photographed 18 different times resulting in images that show each object in 18 different placements. The placements include rotation, scaling, illumination changes, partial occlusion, and combination of two or more of these situations. All shots were taken in different complex and cluttered backgrounds. The images have a resolution of 480×640 pixels and are stored in JPEG compressed format.

4.2 Feature Extraction

Using the DoG-based detector, the interest points in an image are identified. The detector gives the location and the scale of each candidate point in subpixel resolution. The number of extracted points is between 500 and 2000 depending on the image content. To extract the local integral invariants, we use a set of two-point kernel functions. For an interest point located at (n_0, n_1), each kernel function manipulates (multiplies in the case of monomials or subtracts and compares in the case of relationals) two points that lie on the circumferences of two circles of radii r_1, r_2, respectively, where $r_2 \geq r_1$. A phase shift, θ, between the two points is considered. The radii are adapted to the local scale of each interest point as described in section 3. An example of kernel application is shown in Figure 1. Having a set of V kernel functions, f_i, a V-dimensional feature vector around each interest point, (n_0, n_1), is constructed using:

$$\mathbf{F}(i)\,\big|_{(n_0,n_1)} = \frac{1}{P}\sum_{p=0}^{P-1} f_i\left(g\left(n_0, n_1, \varphi = p\frac{2\pi}{P}\right)\mathbf{I}\right),\ i = 1...V.$$

4.3 Retrieval Process

The feature vectors are saved in the database with a pointer to the model image to which they belong. In the retrieval process, a query image is presented to the system. The features of the image are extracted as described above. Each individual feature vector from the query image is compared using the Euclidean distance measure with all the other feature vectors stored in the database of the model images. A kd-tree structure [11] is used to speed up the search. Correspondences between the feature vectors are found based on the method described in [12] which allows for robust matching based on a confidence measure that depends on a uniqueness value assigned to each feature vector. A voting scheme

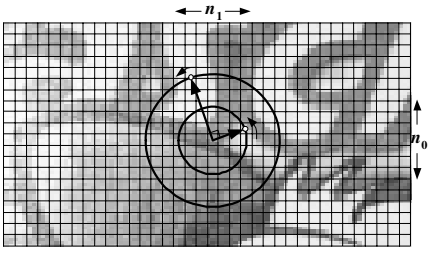


Fig. 1. Evaluation of the monomial kernel $f(\mathbf{I}) = \mathbf{I}(0,3).\mathbf{I}(6,0)$. The grayvalue of a point on a circle of radius $r = 3$ around (n_0, n_1) is multiplied by the grayvalue of a point on a circle of radius $r = 6$. Notice the phase shift of $\pi/2$ between the two points. Points that do not fall on the grid are interpolated. The same calculation strategy is also valid for the relational kernel function, $f(\mathbf{I}) = rel(\mathbf{I}(0,3) - \mathbf{I}(6,0))$, substituting multiplication with subtraction.

is applied to determine the object model that corresponds to the query image. If a match between a vector from the query image and another from a model image is fired, the model image gets a vote. Finally, the model image that gets the maximum number of matches wins the voting scheme and is returned as the best match to the query image.

4.4 Performance Measures

In addition to the recognition rate, we use two other measures to evaluate the quality of recognition: the *number of matches* and the *match precision*. The match precision gives an indication of how good the matching is, by giving the ratio of correct matches to the total number of matches in the image:

$$\text{Match Precision} = \frac{\text{Number of correct matches}}{\text{Total number of matches}} \times 100\%. \tag{6}$$

Given two images that contain the same object, a match between a point, $\mathbf{p_1}$, in the first image and a point, $\mathbf{p_2}$, in the second image is considered as a correct match if the error in the relative location of the two points is less that a threshold, τ:

$$\|\mathbf{p}_1 - \mathbf{H}\mathbf{p}_2\| < \tau, \tag{7}$$

where $\mathbf{H}$ refers to the homography. We set $\tau = 3$ pixels following [13].

5 Results

5.1 Monomial vs. Relational Features

The robustness of the features against intensity changes is an important aspect. Local characteristics should not change when lighting conditions change. Monomial type kernels are very sensitive to changes in the intensity since they work

directly on the raw grayvalue of the pixels. Consider the following equation that describes the affine change in intensity value, $\mathbf{I}(x, y)$:

$$\gamma(\mathbf{I}(x, y)) = a\mathbf{I}(x, y) + b. \tag{8}$$

Considering a simple monomial kernel that multiplies two grayvalues, $\mathbf{I}(x_1, y_1)$ and $\mathbf{I}(x_2, y_2)$, the result is simply $M = \mathbf{I}(x_1, y_1)\mathbf{I}(x_2, y_2)$. If the image was exposed to change in illumination conditions such that the equation above applies, then we have:

$$\begin{aligned} M' &= \gamma(\mathbf{I}(x_1, y_1))\gamma(\mathbf{I}(x_2, y_2)) \\ &= a^2\mathbf{I}(x_1, y_1)\mathbf{I}(x_2, y_2) + b[a(\mathbf{I}(x_1, y_1) + \mathbf{I}(x_2, y_2)) + b] \\ &= a^2 M + b\gamma(\mathbf{I}(x_1, y_1) + \mathbf{I}(x_2, y_2)). \end{aligned}$$

So the result will be scaled by a factor of a^2 and shifted by a factor that is equal to the scaled affine change of the sum of the two grayvalues. The behavior of relational kernels, on the other hand, is different. They are robust against intensity changes as they consider relations between grayvalues of the pixels rather than the raw values themselves. A relational kernel function applied to $\mathbf{I}(x_1, y_1)$ and $\mathbf{I}(x_2, y_2)$ will give a result of $R = rel(\delta)$, where $\delta = \mathbf{I}(x_1, y_1) - \mathbf{I}(x_2, y_2)$ and rel is the fuzzy relation defined in Equation 5. When the intensity changes according to Equation 8, then:

$$\begin{aligned} R' &= rel(\gamma(\mathbf{I}(x_1, y_1)) - \gamma(\mathbf{I}(x_2, y_2))) \\ &= rel(a\delta). \\ &= \begin{cases} 1 & \text{if } \delta < -\epsilon/a \\ \frac{\epsilon - a\delta}{2\epsilon} & \text{if } -\epsilon/a \leq \delta \leq \epsilon/a \\ 0 & \text{if } \epsilon/a < \delta \end{cases} \end{aligned}$$

Clearly, the result is invariant against the shift in the mean luminance. The effect of the scaling parameter, a, depends on the threshold value, ϵ, used in the ramp function, rel. As ϵ approaches zero, a smaller range of δ values will be affected by the scaling parameter a. If ϵ is set equal to zero, then we return to the LBP step operator [9] which is totally invariant to exact monotonic gray scale transformations but sensitive to disturbances. So ϵ is chosen to achieve a good compromise between the robustness against noise and the robustness against gray scale changes. In our experiments, $\epsilon = 0.098$ ($\mathbf{I}(x, y) \in [0, 1]$) was found to give very good results. Two tests were carried out on the database. Once with monomial type features and another with relational type features. In each case, 43 kernels were used to extract a feature vector around each interest point. The results shown in Table 1 depict the tremendous difference in the performance between both types of features. Not only the recognition rate of the monomial features is very low, but also the number of matches per image is small and the percentage of correct matches is low.

Trying to enhance the performance of the monomial-based features, the feature vectors were normalized such that they have a zero mean and a unit variance. A remarkable enhancement in the performance was observed after the

Table 1. Comparison between the results of the tests based on monomial and relational features

	Recognition rate	Avg. # of matches	Avg. match precision
Monomial Features	30.15%	9.94	60.31%
Relational Features	98.99%	72.34	91%

Table 2. Results after normalizing the monomial feature vectors to zero mean and unit variance

	Recog. rate	Avg. # of matches	Avg. match precision
Monomial Features (norm.)	89.45%	36.56	82.1%
Relational Features	98.99%	72.34	91%

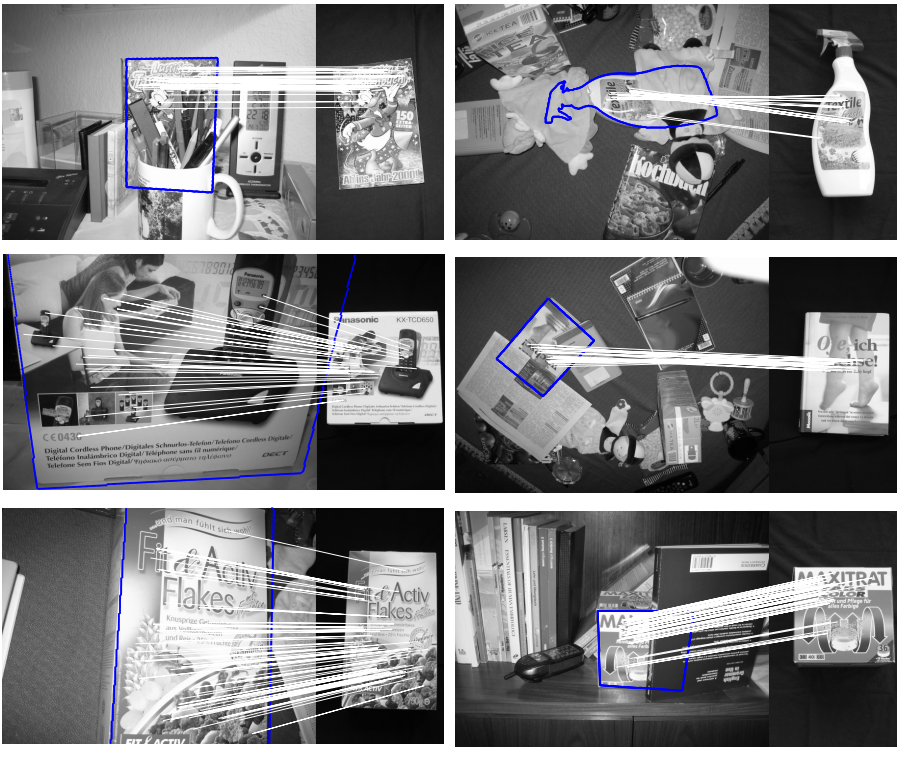
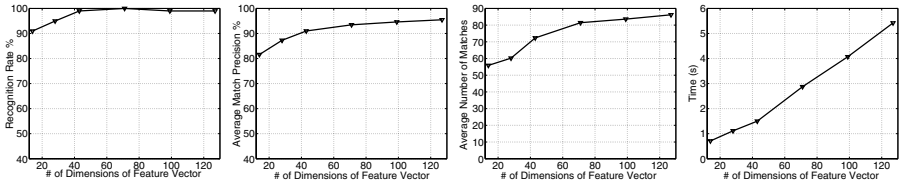


Fig. 2. Some query results using relational features

normalization as can be seen in Table 2. Nevertheless there is still a big gap between the performance of the relational-based features and the monomial-based features. Figure 2 shows different example queries for different instances of some objects in the database using relational features. The left part of each example shows the query image of a complex scene that contains an instance of one



(a) Recognition rate (b) Avg. match precision (c) Avg. number of matches (d) Avg. recognition time

Fig. 3. Performance as a function of vector dimensionality

object. Different situations like object occlusion, rotation, scaling, and intensity change are shown. The result, expressed by the model image that won the voting scheme, is shown to the right. The outline of the detected object is also shown on the query image. For the sake of clarity, not all matches between the query and the model are displayed. Only randomly selected matches that do not overlap are shown. In the rest of the experiments only relational-based feature vectors are used.

5.2 Dimensionality of the Feature Vectors

Usually, high-dimensional feature vectors convey more information about the local characteristics of a point and lead to better results but at the expense of the storage requirements and the computation and matching complexities. Several experiments were run using feature vectors with dimensionality that ranged between 14 and 127 dimensions. In each case, the recognition rate, the average number of matches and the average match precision were observed. Figure 3 shows the results. The recognition rate and quality (number of matches and match precision) go up as the dimensionality increases. The average match precision shows that, in most cases, most of the matches are correct matches. Only for low dimensionality the average match precision is under 90% (about 82% for dimensionality = 14). It reaches about 95% for 127-dimensional vectors. The time needed for recognition depends on the dimensionality of the feature vector in addition to the number of interest points (the number of feature vectors consequently) found in each image. The average recognition time as a function of the vector dimensionality is shown in Figure 3(d). Utilizing feature vectors with dimensionality of about 40 gives a good compromise between recognition rate and quality from one side and complexity and storage from the other side.

5.3 Comparison with SIFT

We report here the performance achieved when using the well-known SIFT features. Table 3 summarizes the results. More matches and higher match precision are obtained using SIFT. However, the recognition rate achieved using SIFT is the same as that achieved using the relational integral invariants.

It should be noticed that the dimensionality of the used integral invariant vectors (43 dimensions) is much less than that of SIFT (128 dimensions). Moreover,

Table 3. Results using SIFT features

	Recognition rate	Average # of matches	Average match precision
SIFT	98.99%	151.8	96%
Rel. Integral Invariants	98.99%	72.34	91%

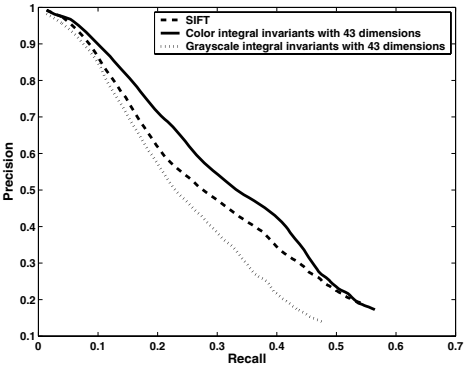


Fig. 4. Precision-Recall graph using 400 queries of the COIL-100 database

in SIFT, about 15% of the points have multiple feature vectors corresponding to multiple orientations assigned to these points. This leads to more storage and matching requirements. This is not the case for the integral invariants. An important advantage of the integral invariants over SIFT is their capability of exploiting color information when needed. SIFT is extracted using grayscale information only. The integral invariants can be extracted from color information by simply applying the kernel functions to the different channels of the color space. To demonstrate this, we carried out some experiments on the COIL-100 database of color objects. In this database, images of 100 different objects were taken at pose intervals of 5 degrees resulting in 72 poses per object. Exploiting color information in such a database is a big plus as the number of points detected in the images of the database is very small (not more than 10 for some objects). Moreover, some objects in the database are similar but with different colors. We tested the performance of the different methods in retrieving objects using 400 query images (4 per object). Figure 4 shows the results in terms of the precision-recall performance measures. The best results were achieved using the color-based integral invariants. SIFT comes in the second place and the grayscale-based integral invariants come third.

6 Conclusion

In this paper we have investigated the use of integral invariants to extract local feature vectors around interest points for the purpose of retrieving objects in

complex scenes. The features are by definition invariant to Euclidean motion and can be made similarity invariant by adapting the support of the kernels used to the local support of the interest points. Two types of features were discussed; relational type and monomial type features. Much better results in terms of recognition rate and quality were achieved using the relational type features as they are robust against intensity changes, which is not the case with the monomial type features. Tests were conducted using test images of objects against cluttered backgrounds and in difficult situations like partial object occlusion and intensity changes. Several experiments considering the feature vector length were conducted and it was found that vectors with about 40 dimensions give a good compromise between performance and complexity. We compared our work with the SIFT features. The main advantages of the integral invariants over SIFT is the lower dimensionality and the ability to use color information if needed.

References

1. Swain, M.J., Ballard, D.H.: Color indexing. IJCV **7** (1991) 11–32
2. Schiele, B., Crowley, J.L.: Object Recognition Using Multidimensional Receptive Field Histograms. In: ECCV. Volume 1. (1996) 610–619
3. Schmid, C., Mohr, R.: Local Grayvalue Invariants for Image Retrieval. PAMI **19** (1997) 530–535
4. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: Proceedings of The Fourth Alvey Vision Conference. (1988) 147–151
5. Gouet, V., Boujemaa, N.: Object-based Queries using Color Points of Interest. In: CBAIVL, Kauai, Hawaii, USA (2001) 30–36
6. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. IJCV **60** (2004) 91–110
7. Schulz-Mirbach, H.: Invariant Features for Gray Scale Images. In: 17th DAGM, Bielefeld (1995) 1–14
8. Schael, M.: Invariant Grey Scale Features for Texture Analysis Based on Group Averaging with Relational Kernel Functions. Technical Report 1/01, University of Freiburg (2001)
9. Ojala, T., Pietikäinen, M., Mäenpää, T.: Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns. In: ECCV. (2000) 404–420
10. Siggelkow, S.: Feature Historgrams for Content-Based Image Retrieval. PhD thesis, Albert-Ludwigs-Universität, Freiburg (2002)
11. Freidman, J.H., Bentley, J.L., Finkel, R.A.: An Algorithm for Finding Best Matches in Logarithmic Expected Time. ACM Transactions on Mathematical Software **3** (1977) 209–226
12. Biber, P., Straßer, W.: Solving the Correspondence Problem by Finding Unique Features. In: 16th International Conference on Vision Interface. (2003)
13. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. In: CVPR. Volume 2. (2003) 257–263