

A Novel Tiny Object Recognition Algorithm Based on Unit Statistical Curvature Feature

Yimei Kang^(✉) and Xiang Li

College of Software, Beihang University, Beijing, China
{kangyimei, sy1421103}@buaa.edu.cn

Abstract. To recognize tiny objects whose sizes are in the range of 15×15 to 40×40 pixels, a novel image feature descriptor, unit statistical curvature feature (USCF), is proposed based on the statistics of unit curvature distribution. USCF can represent the local general invariant features of the image texture. Due to the curvature features are independent of image sizes, USCF algorithm had high recognition rate for object images in any size including tiny object images. USCF is invariant to rotation and linear illumination variation, and is partially invariant to viewpoint variation. Experimental results showed that the recognition rate of USCF algorithm was the highest for tiny object recognition compared to other nine typical object recognition algorithms under complex test conditions with simultaneous rotation, illumination, viewpoint variation and background interference.

Keywords: Object recognition · Tiny object · Feature descriptor · Unit Statistical Curvature Feature

1 Introduction

Recognition of tiny image objects taken by digital cameras is a key subject in machine vision. Recognizing an object accurately and quickly when it is very small and in a distance provides more time to take appropriate actions for a system that relies on machine vision, such as a robot, an Unmanned Aerial Vehicle (UAV), *etc.* However, automatic recognition gets more and more difficult when the objects are getting smaller, due to the tiny objects have very few pixels and texture information.

There are limited studies focused on this subject. Torralba *et al.* [1] implemented tiny objects classification whose sizes are 32×32 color pixels by using the nearest neighbor matching schemes and image indexing techniques. They showed that the 32×32 color pixel tiny images already seem to contain most of the relevant information needed to support reliable recognition. However, the approach is only used to classify objects but not to distinguish a tiny object from other objects in an image.

Multiple approaches can be used to recognize a big image object whose size is larger than 40×40 pixels by matching the image features, *e.g.* edge features [2–4],

invariant features [5–10], statistical features [11–13], *etc.* Tiny object whose size is smaller than 40×40 pixels has vague contours, and the algorithms based on edge features cannot work on tiny object recognition.

Some recognition algorithms based on invariant features are commonly used to recognize objects. SIFT [5] constructed feature descriptors based on histogram of magnitude and direction of gradients to characterize an object. To improve the calculation efficiency of SIFT, SURF [6] built feature descriptors based on sum of Haar wavelet responses. Rani *et al.* [7] found that the number of keypoints detected by using SIFT is more than that of SURF through a set of experiments. Rublee *et al.* prompted an efficient matching method called ORB [8] by combining FAST keypoint detector and BRIEF descriptor. Hauage *et al.* proposed another kind of invariant feature based on local symmetry feature [9].

The above mentioned invariant feature descriptors are invariant to uniform scaling and orientation variation. SIFT can precisely recognize the images by matching the keypoints which are extremes in a set of three DoG (Different of Gaussian) images. Similarly, SURF recognize the objects by matching keypoints derived from blob structure and ORB uses descriptor derived from corner keypoints in images. However, there are often no suitable keypoints when the image size is very small, such as in the range from 15×15 to 40×40 pixels.

Hu proposed a geometric feature descriptor based on invariant moments [10]. Hu's method is suitable for object recognition by using object shape. There is little information to construct the boundaries for tiny object images. Therefore, Hu's method cannot be used directly to recognize tiny objects. However, the invariant features of Hu's method can represent the tiny objects and can be used to recognize tiny objects.

Statistical features, such as histogram and entropy of an image, can also be used to characterize small objects because they do not rely on the size of the images. However, statistical features are too general and have no position information, therefore, they can hardly be applied to object recognition. In contrary to global entropy, unit entropy has information of position, and it can be used to recognize small objects. Fritz *et al.* [11] used unit entropy to build an entropy-based object model from discriminative local patterns for object representation and recognition.

HoG [12] and GIST [13] algorithms are also based on statistical features. HoG employs a histogram binning on the gradient orientation and extracts feature vector with a grid of overlapping blocks. GIST divides an image into 4×4 grids in which orientation histograms are extracted by using Gabor filters. When we tried to apply existing recognition algorithms to an industrial application to distinguish between tiny objects, unit entropy, HoG and GIST all performed well. Unfortunately, they didn't work when the object rotates.

Some nonlinear method based on machine learning can also be used to recognize objects, *e.g.* the methods using k -nearest neighbor method and semi-supervised learning [14], weight kernels over orientations [15], wavelet neural network [16], two-layer neural network [17], and convolutional neural network [18]. All these algorithms must train sample images before recognizing objects.

In this paper, we proposed a novel image feature descriptor, unit statistical curvature feature (USCF) of the grayscale surface of an image, to characterize tiny objects with 15×15 to 40×40 pixels. The recognition algorithm based on USCF can recognize an object image in any size. Specifically, it had high recognition rate and computation efficiency for tiny objects. USCF is completely invariant to rotation and linear illumination variation. It is also partially invariant to viewpoint variation and background interference. The USCF algorithm is compared with other recognition algorithms including SIFT, SURF, ORB, gray histogram, entropy, unit entropy, GIST, HoG and Hu's moment invariants (Hu's MI) algorithms on the image datasets from ALOI-COL Database [19], COIL-100 Database [20], ETH-80 Database [21], ETHZ another 53 Objects Database and images from two videos, respectively. The experimental results showed that USCF algorithm had best performance in tiny object recognition under real complex environment against rotation, illumination, viewpoint variation and background interference.

2 The Principle of USCF Algorithm

An object image with less than 40×40 pixels often has vague contour and texture. It is difficult for existing recognition algorithms to extract enough features from such tiny objects. Hence, we tried a new way to recognize such tiny objects.

To recognize a tiny object, we need utilize the limited pixel information as much as possible. Firstly, we build a three-dimensional coordinate system $Oxyz$ with the positions and gray values of pixels in an image I . Let (x, y, z) represent a point in $Oxyz$, and let z be the gray value of pixel (x, y) in image I . Then we construct a fitting function $z = f(x, y)$ to convert the discrete points in $Oxyz$ to a curved surface, which outlines the gray value distribution tendency of image I . Figure 1 shows two objects with their fitted curved surfaces under different conditions. As shown in Fig. 1, different objects have different fitted curved surfaces while the shape of the fitted curved surfaces of an object is invariant to object rotation and illumination variation. Therefore, object recognition can be converted to compare the similarity of the fitted curved surfaces.

We constructed image features derived from the curvature to estimate the similarity of the fitted curved surface. Curvature can describe the gray value distribution of an image. Different from gradient, curvature is independent of the object orientation and keeps more object details because second derivative enhances the difference in details. Curvature is invariant to object rotation and linear illumination variation since it only relies on the shape of curved surface. Mean curvature reflects the local shape feature of the curved surface and Gaussian curvature reflects the convexity and concavity feature of the curved surface. Therefore, we combined Gaussian curvature $K(x, y, z)$ and mean curvature $H(x, y, z)$ of each point (x, y, z) in the fitted curved surface of an object to build the invariant feature of the object.

Map $K(x, y, z)$ and $H(x, y, z)$ to a two-dimensional coordinate O_{HK} to generate the curvature feature space of the object as shown in Fig. 2. We used

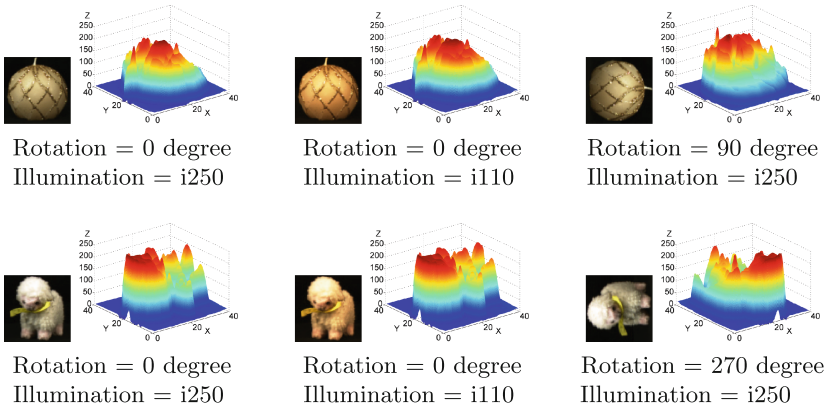


Fig. 1. Images (left) and fitted curved surfaces (right) of two objects in different conditions. The object images and their parameters of illumination condition were selected from ALOI-COL Database

coordinate (H, K) to represent a point in O_{HK} . The curvature feature space reflects the change of object image texture. If the color of a pixel is changed slightly or not changed at all compared to surrounding pixels, the absolute values of $K(x, y, z)$ and $H(x, y, z)$ should both be low. Hence, the smoother the image texture change is, the more points closer to the origin in O_{HK} are. Conversely, when image texture is changed dramatically, more points are away from the origin in O_{HK} . In general, majority of points in O_{HK} are close to origin because smooth area is the majority in an image.

The curvature of a pixel is calculated based on pixels surrounding that pixel, therefore it is sensitive to any changes of neighboring pixels. Any color variation of pixels will have an impact on the value of the curvature. This means the whole map in O_{HK} of an object is also sensitive to gray value fluctuation of each pixel. Hence, we partition the curvature feature space of an object to a number of unit areas and use the statistics of curvature features in each unit to generate a stable curvature feature matrix, *i.e.* unit statistical curvature feature. Then we recognize tiny objects by matching the similarity of their USCF matrices.

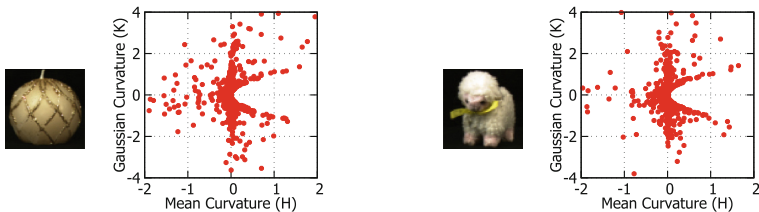


Fig. 2. Mapping of each pixel in two object images to O_{HK} using Gaussian curvature and mean curvature as coordinates

3 Proposed Algorithm

In this section, we present USCF recognition algorithm in details. Firstly, we employ least square method to fit curved surfaces of object images. Secondly, we calculate the Gaussian curvature and mean curvature of each pixel in object images according to the fitted curved surfaces. Thirdly, we build the curvature feature space in O_{HK} , and partition it into a number of unit areas according to the curvature distribution density. Then we count the number of points in each unit area to construct the USCF matrix. Finally, the similarity of the objects is obtained by matching the USCF matrices with Euclidean distance.

3.1 Generate Fitted Curved Surface

Least square method was used in this paper to generate the curved surface fitting function. The reason to use least square method is that it can optimize fitting function globally, and it is simple and converges quickly.

A function which consists of a number of primary functions and an unknown coefficient set is usually used to describe an unknown curved surface. We use polynomial functions as primary functions. Polynomial function can be differentiated arbitrary times, and it is easy to calculate. For X axis, we chose $P + 1$ polynomial functions of x denoted as $\varphi_r(x)$, where $r = 0, 1, \dots, P$. For Y axis, we chose $Q + 1$ polynomial functions of y denoted as $\phi_s(y)$, where $s = 0, 1, \dots, Q$. Let $\varphi_r(x)\phi_s(y)$ be the primary functions. Denote $\{c_{rs}\}$ as the unknown coefficient set. We can construct a function to represent an unknown curved surface as follows:

$$f(x, y) = \sum_{s=0}^Q \sum_{r=0}^P c_{rs} \varphi_r(x) \phi_s(y) \tag{1}$$

Assume there are $(m + 1) \times (n + 1)$ points in $Oxyz$ denoted as $S = \{(x_i, y_j, z_{ij})\}$, where $i = 0, 1, \dots, m$ and $j = 0, 1, \dots, n$. Fitted curved surface is an approximation to the actual curved surface, therefore, there exist errors between the calculated values from the fitted curved surface function and the actual values. We define the error square as follows:

$$\begin{aligned} I &= \sum_{j=0}^n \sum_{i=0}^m [f(x_i, y_j) - z_{ij}]^2 \\ &= \sum_{j=0}^n \sum_{i=0}^m \left[\sum_{s=0}^Q \sum_{r=0}^P c_{rs} \varphi_r(x) \phi_s(y) - z_{ij} \right]^2 \end{aligned} \tag{2}$$

If there is a coefficient set $\{c_{rs}^*\}$ which minimize I , then $f^*(x, y)$ based on $\{c_{rs}^*\}$ is the fitted curved surface of point set S by using least square fitting method. In this situation, the following equation set must be true:

$$\begin{aligned} \frac{\partial I}{\partial c_{rs}^*} &= 2 \sum_{j=0}^n \sum_{i=0}^m [(f^*(x_i, y_j) - z_{ij}) \varphi_r(x_i) \phi_s(y_j)] \\ &= 0 \quad (r = 0, 1, \dots, P; s = 0, 1, \dots, Q) \end{aligned} \tag{3}$$

Denoting matrices

$$\begin{aligned}
 A &= [\varphi_r(x_i)]_{(m+1) \times (P+1)} \\
 B &= [\phi_s(y_j)]_{(n+1) \times (Q+1)} \\
 Z &= [z_{ij}]_{(m+1) \times (n+1)} \\
 C &= [c_{ij}^*]_{(P+1) \times (Q+1)}
 \end{aligned} \tag{4}$$

We can obtain coefficient values by simplifying Eq. (3) and substituting the matrices in Eq. (4) into the simplified equation.

$$C = (A^T A)^{-1} A^T Z B (B^T B)^{-1} \tag{5}$$

Correspondingly, we get the fitted curved surface $f^*(x, y)$ of point set S .

3.2 Calculate Curvatures

Once the fitted curved surface is obtained, we can calculate the curvatures of the curved surface. The fitted curved surface function can be rewritten as a vector equation as follows:

$$\vec{t} = (x, y, f^*(x, y)) \tag{6}$$

Denoting f^* as $f^*(x, y)$, we can obtain the first order differential and the second order differential about x and y of \vec{t} as follows:

$$\begin{aligned}
 \vec{t}_x &= (1, 0, f_x^*), \quad \vec{t}_y = (0, 1, f_y^*) \\
 \vec{t}_{xx} &= (0, 0, f_{xx}^*), \quad \vec{t}_{yy} = (0, 0, f_{yy}^*), \quad \vec{t}_{xy} = (0, 0, f_{xy}^*) \\
 f_x^* &= \frac{\partial f^*}{\partial x}, \quad f_y^* = \frac{\partial f^*}{\partial y}, \quad f_{xx}^* = \frac{\partial^2 f^*}{\partial x^2}, \quad f_{yy}^* = \frac{\partial^2 f^*}{\partial y^2}, \quad f_{xy}^* = \frac{\partial^2 f^*}{\partial x \partial y}
 \end{aligned} \tag{7}$$

The values of H and K can be obtained through the fundamental form definition of the curved surface as follows:

$$\begin{aligned}
 H &= \frac{LG - 2MF + NE}{2(EG - F^2)} \\
 K &= \frac{LN - M^2}{EG - F^2}
 \end{aligned} \tag{8}$$

where E, F, G, L, M, N are parameters of the first fundamental form and the second fundamental form of the curved surface and depend on Eq. (7).

$$\begin{aligned}
 E &= \vec{t}_x \cdot \vec{t}_x, \quad L = \vec{t}_{xx} \cdot \frac{\vec{t}_x \times \vec{t}_y}{|\vec{t}_x \times \vec{t}_y|} \\
 F &= \vec{t}_x \cdot \vec{t}_y, \quad M = \vec{t}_{xy} \cdot \frac{\vec{t}_x \times \vec{t}_y}{|\vec{t}_x \times \vec{t}_y|} \\
 G &= \vec{t}_y \cdot \vec{t}_y, \quad N = \vec{t}_{yy} \cdot \frac{\vec{t}_x \times \vec{t}_y}{|\vec{t}_x \times \vec{t}_y|}
 \end{aligned} \tag{9}$$

Replacing parameters in Eq. (8) with the parameters in Eq. (9) and combining with Eq. (7), we can get H and K represented by the differential forms of f^* shown as following:

$$\begin{aligned}
 H &= \frac{(1 + f_y^{*2})f_{xx}^* + (1 + f_x^{*2})f_{yy}^* - 2f_x^*f_y^*f_{xy}^*}{2(1 + f_x^{*2} + f_y^{*2})^{3/2}} \\
 K &= \frac{f_{xx}^*f_{yy}^* - f_{xy}^{*2}}{(1 + f_x^{*2} + f_y^{*2})^2}
 \end{aligned}
 \tag{10}$$

Using Eq. (10), we can get H and K of pixels in an image based on the fitted function of the image.

3.3 Generate USCF Matrix

Once the curvature feature space in O_{HK} is obtained, we partition the curvature feature space into $w \times v$ units. Then we count the number of pixels in each unit to generate the USCF matrix of an image.

The non-uniform distribution of pixels in O_{HK} made it ineffective to uniformly partition the curvature distribution area. Points in O_{HK} are distributed widely but most points are close to the origin. Uniform partition of the curvature feature space will result in non-uniform statistical features, *e.g.* a small number of units will contain most points and most units are empty, which will result in USCF matrices of different objects with no significant difference. Hence, we partition the curvature feature space non-uniformly according to the density to generate distinct USCF matrix with uniform statistical features.

Assume the curvature feature space of the image is $Area$, which is defined as $Area = \{(H, K) | a < H < b, c < K < d\}$. We use delimiters $H_i (i = 0, 1, \dots, w)$ and $K_j (j = 0, 1, \dots, v)$ to divide curvature feature space $Area$ into $w \times v$ parts, where $H_{i-1} < H_i, H_0 = a, H_w = b$ and $K_{j-1} < K_j, K_0 = c, K_v = d$. We denote a part as $Area_{ji}$, which is defined as follows:

$$Area_{ji} = \{(H, K) | H_{i-1} < H \leq H_i, K_{j-1} < K \leq K_j\}
 \tag{11}$$

where $i = 1, 2, \dots, w$ and $j = 1, 2, \dots, v$.

We use $count(Area_{ji})$ to represent the number of pixels whose curvature coordinates (H, K) are located in $Area_{ji}$. Subsequently, we can define the USCF matrix as follows:

$$D = [count(Area_{ji})]_{v \times w}
 \tag{12}$$

The USCF matrix D reflects the curvature feature of a curved surface and we use it to represent the image that the curved surface is fitted from.

The recognition ability of USCF algorithm depends on the partition of curvature feature space. The more units curvature feature space is divided into, the more detailed textures of an object are kept. Meanwhile, the recognition result can be easily affected by image change because smaller unit partition weakens general feature and is more sensitive to local feature variation. On the other side,

the less parts curvature feature space is divided into, the better image change immunity of USCF algorithm is, with the cost of reducing the recognition precision of USCF algorithm.

3.4 Compare the Similarity of USCF Matrices

We can obtain the similarity of template object image and candidate object image by comparing the similarity of their USCF matrices. Let $D[i, j]$ represent the element located in the i th row and the j th column of matrix D . Let D_T and D_M represent the USCF matrices of template object image and candidate object image, respectively. We define $dist$ as the metric to measure the similarity between D_T and D_M by calculating their Euclidean distance. To normalize $dist$ so that all the values are in $[0, 1]$, every element in the matrix will be divided by the sum of all elements in the matrix.

$$dist = \sum_{i=1}^v \sum_{j=1}^w \left(D_T[i, j] / \sum_{p=1}^v \sum_{q=1}^w D_T[p, q] - D_M[i, j] / \sum_{p=1}^v \sum_{q=1}^w D_M[p, q] \right)^2 \quad (13)$$

The value of $dist$ represents the degree of similarity between template and candidate images. If $D_T = D_M$, the value of $dist$ is equal to zero.

4 Experimental Results

All experiments in this section were carried out on a desktop PC with Intel(R) Core(TM) i5-3470 CPU and 8 GB memory space. USCF algorithm was compared with SIFT, SURF, ORB, gray histogram, entropy, unit entropy, GIST, HoG and Hu's moment invariants algorithms on ALOI-COL Database, COIL-100 Database, ETH-80 Database and ETHZ another 53 Objects Database, respectively. SIFT, SURF, ORB, HoG and Hu's moment invariants algorithms were provided by OpenCV 3.0. Other algorithms were programmed in C++ language.

Firstly, we compared the recognition rates of the ten test algorithms on the images against one of the variables, *i.e.* rotation, illumination, or viewpoint variation. Then we compared USCF algorithm with the other nine algorithms under a real complex environment with simultaneous variation of multiple variables including rotation, illumination and viewpoint. These four types of experiments were performed on images with sizes of 15×15 , 20×20 , 25×25 , 30×30 , 35×35 and 40×40 pixels, which were shrunk from the images in above image database by using bicubic interpolation, respectively. Finally, we gave comparison of USCF and other nine algorithms on images from two videos.

4.1 Parameter Selection

The selection of fitting parameters affects the results of USCF algorithm. To avoid complex fitting function and get accurate curvature values, we calculated the curvatures of every pixel by using the pixel with 8 surrounding pixels to fit

each local small curved surface respectively. To simplify calculation, the primary functions $\varphi_r(x)$ and $\phi_s(y)$ were chosen as follows:

$$\begin{aligned}\varphi_r(x) &= x^r \quad (r = 0, 1, 2) \\ \phi_s(y) &= y^s \quad (s = 0, 1, 2)\end{aligned}\tag{14}$$

After performing a large number of experiments, we found that the distribution area of K and H of most pixels are in the range of $(-1000, 1000)$ and partitioning the distribution area into 17 parts and 11 parts can get a good recognition performance. In this situation, most units are not empty and contain enough points to distinguish tiny objects and eliminate interference. The demarcation points was generated as follows:

$$\begin{aligned}H_i &= [(i - 5.5)/|i - 5.5|] \times 10^{|i-5.5|-2.5} \quad (i = 0, 1, \dots, 11) \\ K_j &= [(j - 8.5)/|j - 8.5|] \times 10^{|j-8.5|-5.5} \quad (j = 0, 1, \dots, 17)\end{aligned}\tag{15}$$

In the experiments, FlannBasedMatcher was used as the matching strategy and at least 3 keypoints were required to correctly match the candidate and the template images for SIFT, SURF and ORB algorithms. For entropy algorithm, the absolute value of entropy difference between the template and candidate object images was the matching criterion. For unit entropy and HoG algorithms, we used 5×5 pixels as the size of each unit. We directly applied Hu's moment invariants of the whole image to match the object but did not extract the contour of the object, and the cosine value of its feature vector was the metric to measure the similarity of the template object image and candidate object image. For unit entropy, GIST, HoG and gray histogram, we compared their feature vector similarity by using Euclidean distance.

4.2 Robustness Against Rotation

COIL-100 Database was used to evaluate the anti-rotation performance of USCF and other nine algorithms. The candidate object images were rotated clockwise by 15, 60, 90 and 175°, respectively. Some of the selected experimental images are shown in Fig. 3.

As shown in Fig. 4, the recognition rate of USCF algorithm was from 70% up to 90% with the size of the objects from 15×15 to 40×40 pixels. USCF is robust to rotation because USCF is based on curvature, which is independent of the object orientation, and USCF non-uniformly partitions the curvature distribution map according to the curvature distribution density but not directly partitions the original images. The recognition rate of Hu's moment invariants was from 80% to 90%. The performance of USCF algorithm was as good as that of Hu's moment invariants algorithm when the size of the object was larger than 35×35 pixels.



Fig. 3. Examples of selected experimental images. From left to right, the rotation angles of the objects are 0, 15, 60, 90 and 175°, respectively

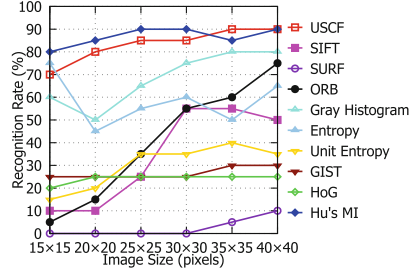


Fig. 4. Anti-rotation experimental results of USCF and other nine algorithms on the test images with different sizes

Entropy and gray histogram had worse performance than USCF and Hu’s moment invariants but had better performance than other algorithms, because they are based on general statistical feature of the images which is insensitive to image rotation. Unit entropy, HoG and GIST performed poorly because they directly partition the original images. SIFT, SURF and ORB are robust to rotation, but they cannot obtain enough keypoints for tiny object recognition. The best recognition rate of SURF was only 10% in the experiments due to its method used to detect keypoints. SIFT and ORB had better performance than SURF. However, the recognition rates for those algorithms increased as the object sizes increased.



Fig. 5. Examples of selected experimental images from ALOI-COL Database. From left to right, the illumination conditions are i110, i140, i170, i210 and i250, respectively

4.3 Robustness Against Illumination Variation

ALOI-COL Database was used to evaluate the robustness against illumination variation of the algorithms. We used the object images in the dataset illuminated under condition i250 as template object image and the object images illuminated under conditions i110, i140, i170 and i210 as candidate object images, respectively. Some of the selected test images are shown in Fig. 5.

As shown in Fig. 6, unit entropy, HoG and GIST algorithm recognized tiny object very well under varying illumination because their local statistical features are insensitive to illumination change. The recognition rate of the three algorithms were 100% almost in all cases except it was 80% for unit entropy when the object size was 15x15 pixels. USCF algorithm performed better than other six algorithms because USCF is invariant to linear illumination change. When the object size became 40x40 pixels, the recognition rate of USCF algorithm was 100%. Gray histogram, entropy and Hu’s moment invariants algorithms are

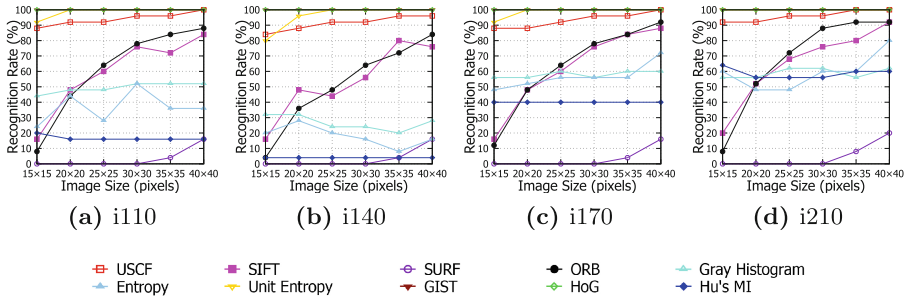


Fig. 6. Experimental results of USCF and other nine algorithms under different illumination conditions for different sizes of the images

based on whole gray values which change obviously with illumination varying. Their recognition rates were worse than those of SIFT and ORB. SIFT, SURF and ORB are also invariant to linear illumination change. Their poor performance is mainly due to their limitation to tiny object and the contrast change produced by non-uniform illumination change. However, the recognition rates increased with the object sizes increased. The best recognition rate of SURF was only 20% while those of SIFT and ORB were 90%.

4.4 Robustness Against Slight Camera Viewpoint Variation

The COIL-100 Database was used to evaluate the performance of the ten test algorithms under varying viewpoint. The images with the view of 0 degree were used as the template images while the images with views of 5, 10, 15 and 20° were candidate images. Some of the selected test images are shown in Fig. 7.

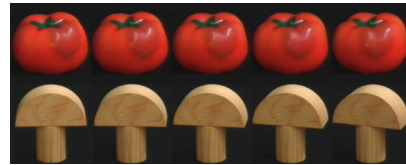


Fig. 7. Examples of the selected test images from COIL-100 Database. From left to right, the view of camera are 0, 5, 10, 15 and 20° respectively

As shown in Fig. 8, the algorithms based on local statistical information, including HoG, GIST, unit entropy, and USCF, performed much better than other algorithms. Their recognition rates were close to 100% when viewpoint change was no more than 10°, while the performance reduced with the increasing of viewpoint change. This indicates that HoG, GIST, unit entropy and USCF algorithms are partially invariant to viewpoint variation. Curvature is derived from calculating second derivative of the fitted curved surface, which keeps more image details and is more sensitive to non-uniform change of local gray value than gradient, energy spectra and entropy. Therefore, the recognition rates of USCF were close to those of HoG, GIST, and unit entropy algorithms but lower than them.

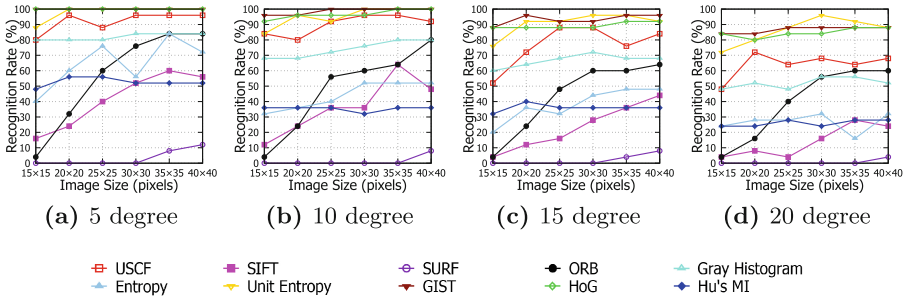


Fig. 8. Experimental results of USCf and other nine algorithms under different viewpoints for different sizes of images

Slight viewpoint changes have relatively little impact on statistical features. As shown in Fig. 8, the best recognition rates of gray histogram and entropy algorithms were 84 % when the camera view changed by 5°. However, entropy algorithm performed worse than gray histogram. The recognition rates of Hu’s moment invariants algorithm were from 50 % down to about 25 % with the viewpoint varied from 5° to 20°. The results shows that Hu’s moment invariants algorithm is sensitive to viewpoint variation, because Hu’s moment invariants are based on the image centroid which shift as viewpoint changes. ORB performed much better than SIFT, SURF algorithms. The best recognition rate of ORB reached 80 % when the size of the object was 40×40 pixels and the viewpoint changed by 10°. The best performance of SURF was only 12 % when the size of the object was 40×40 pixels and the viewpoint changed by 5°.

4.5 Recognition Rate in Complicated Conditions

In a real-world environment, object recognition is usually carried under a complicated condition with simultaneous variation of multiple variables including variation of rotation, illumination and viewpoint. ETH-80 Database and ETHZ another 53 Objects Database were used to test the recognition ability of the algorithms in complicated conditions. In the two datasets, each object is represented by multiple images with different status such as upside down, rotation, different viewpoint or illumination. Some of the selected test images are shown in Fig. 9.

As shown in Fig. 10, USCf algorithm had the best performance among the ten algorithms in the experiments. The recognition rate of USCf was 90 % when the sizes of tiny objects were from 25×25 to 35×35 pixels. The best recognition rate of USCf algorithm reached 95 % when the object size was 40×40 pixels. Since unit entropy, HoG and GIST algorithms are sensitive to rotation, the best performance of unit entropy algorithm was 75 % when the sizes of tiny object was 35×35 pixels and the recognition rates of HoG and GIST algorithms were no more than 60 % in all sizes. The recognition rates of gray histogram was from 50 % to 63 % with the object sizes from 15×15 to 40×40 pixels. The best

recognition rate of Hu’s moment invariants was 70 % when the object size was 30×30 pixels. In such a complex environment, entropy, ORB, SIFT and SURF algorithms had poor performance. The best recognition rate of entropy algorithm was 35 % when the object size was 20×20 pixels. The best performance of SIFT and ORB was 40 % and 42 %, respectively, and SURF algorithm could hardly have effective recognition to any sizes of tiny objects used in the experiments.

The experimental results showed that USCF algorithm had best performance among all algorithms tested in tiny object recognition under real complex environment with simultaneous variation of rotation, illumination and viewpoint.



Fig. 9. Examples of selected test images from ETH-80 Database and ETHZ another 53 Objects Database

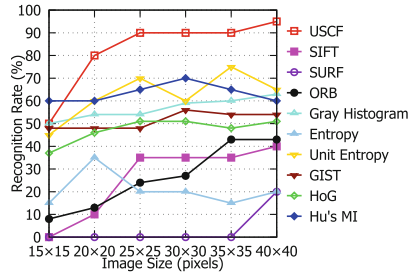


Fig. 10. Experimental results of USCF and other nine algorithms on the test images with different sizes under complex conditions

4.6 Comparison of the Test Algorithms on Images from Videos

In this section, the ten algorithms were applied on the images from two videos, a toy car video taken by ourselves and a jet flight video taken in an air show, with background interference and random varying in rotation, viewpoint and illumination as shown in Fig. 11. The jet of the flight was great background interference because it is similar to the object. The information of the test images is shown in Table 1. We cut the object image from a frame as the template and used the object image with similar scale in other frames as candidate images. All test images were directly cut from the videos without preprocessing.

From Table. 2, we can see that USCF had the best performance among the test algorithms. The recognition rates of USCF were from 65 % to 100 % while those of HoG and GIST, which performed best among the nine compared algorithms, were from 2 % to 100 % and 4 % to 100 %, respectively. HoG and GIST performed quite poorly when object rotates violently. Algorithms based on simple statistic of gray values, such as entropy, unit entropy and gray histogram, were badly affected by the background whose color is similar to the object color. Such background interference greatly affected Hu’s moment invariants as well. SURF could hardly recognize the tiny objects. SIFT and ORB were badly affected by comprehensive condition changes but had relative good performance to the jet background.

Table 1. The information of the test images used in the experiment

Video	Images set (number)	Template size	Range of candidate image size	Rotation	Viewpoint change	Illumination chang
Toy Car	Data1 (60)	40×29	38×27 to 44×33	Slight	Mean	Slight
	Data2 (110)	36×26	32×22 to 38×28	Great	Mean	Slight
	Data3 (26)	26×17	25×14 to 27×18	Slight	Mean	Mean
Jet Flight	Data4 (51)	35×35	23×23 to 38×38	Slight	Slight	Slight
	Data5 (100)	23×23	18×18 to 29×29	Slight	Slight	Slight

Table 2. The performance of the test algorithms on images from the video datasets

Test algorithm	Recognition rate				
	Data1	Data2	Data3	Data4	Data5
USCF	100 %	100 %	65 %	90 %	97 %
SIFT	25 %	0	0	90 %	42 %
SURF	0	0	0	4 %	0
ORB	17 %	5 %	7 %	60 %	23 %
Gray Histogram	50 %	52 %	57 %	16 %	29 %
Entropy	13 %	18 %	4 %	17 %	2 %
Unit Entropy	98 %	77 %	65 %	53 %	13 %
GIST	96 %	4 %	100 %	100 %	100 %
HoG	100 %	2 %	100 %	100 %	100 %
Hu's MI	27 %	26 %	11 %	16 %	5 %



Fig. 11. Some recognition results of USCF applied in recognizing object images from two videos. The template images are on the left top corner. The red boxes show the recognition results (Color figure online)

5 Conclusions

In this paper, we proposed a novel object recognition algorithm, USCF algorithm, based on unit statistical curvature feature. USCF algorithm calculates mean curvature and Gaussian curvature of each pixel in the fitted curved surface of an object image to generate a unit statistical curvature feature matrix to characterize the tiny object. The experimental results showed that USCF algorithm is robust to rotation and illumination variation, and can tolerate slight viewpoint variation. Under complex test conditions with simultaneous rotation, illumination, viewpoint variation and background interference, the recognition rate of USCF was the highest among all ten tested algorithms. USCF cost less than 40 ms on a desktop PC with Intel(R) Core(TM) i5-3470 CPU when the image sizes were smaller than 40×40 pixels, which indicates that USCF can be applied in a real time application for tiny object recognition.

References

1. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(11), 1958–1970 (2008)
2. Nguyen, D.T.: A novel chamfer template matching method using variational mean field. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2425–2432. IEEE (2014)
3. Satpathy, A., Jiang, X., Eng, H.L.: LBP-based edge-texture features for object recognition. *IEEE Trans. Image Process.* **23**(5), 1953–1964 (2014)
4. Xu, Y., Quan, Y., Zhang, Z., Ji, H., Fermüller, C., Nishigaki, M., Dementhon, D.: Contour-based recognition. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3402–3409. IEEE (2012)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
6. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **110**(3), 346–359 (2008)
7. Rani, R., Grewal, S.K., Panwar, K.: Object recognition: performance evaluation using SIFT and SURF. *Int. J. Comput. Appl.* **75**(3), 39–47 (2013)
8. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: 2011 International Conference on Computer Vision, pp. 2564–2571. IEEE (2011)
9. Hauage, D.C., Snavely, N.: Image matching using local symmetry features. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 206–213. IEEE (2012)
10. Hu, M.K.: Visual pattern recognition by moment invariants. *IRE Trans. Inf. Theor.* **8**(2), 179–187 (1962)
11. Fritz, G., Paletta, L., Bischof, H.: Object recognition using local information content. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 2, pp. 15–18. IEEE (2004)
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 886–893. IEEE (2005)
13. Douze, M., Jégou, H., Sandhawalia, H., Amsaleg, L., Schmid, C.: Evaluation of GIST descriptors for web-scale image search. In: Proceedings of the ACM International Conference on Image and Video Retrieval, p. 19. ACM (2009)
14. Ebert, S., Larlus, D., Schiele, B.: Extracting structures in image collections for object recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 720–733. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-15549-9_52](https://doi.org/10.1007/978-3-642-15549-9_52)
15. Fasel, B., Gatica-Perez, D.: Rotation-invariant neoperceptron. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 3, pp. 336–339. IEEE (2006)
16. Pan, H., Xia, L.Z.: Efficient object recognition using boundary representation and wavelet neural network. *IEEE Trans. Neural Netw.* **19**(12), 2132–2149 (2008)
17. Wang, B., Bai, X., Wang, X., Liu, W., Tu, Z.: Object recognition using junctions. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Computer Vision – ECCV 2010. LNCS, vol. 6315, pp. 15–28. Springer, Heidelberg (2010)

18. Agrawal, P., Girshick, R., Malik, J.: Analyzing the performance of multilayer neural networks for object recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 329–344. Springer, Heidelberg (2014). doi:[10.1007/978-3-319-10584-0_22](https://doi.org/10.1007/978-3-319-10584-0_22)
19. Nene, S.A., Nayar, S.K., Murase, H., et al.: Columbia object image library (coil-20). Technical report, Technical report CUCS-005-96 (1996)
20. Geusebroek, J.M., Burghouts, G.J., Smeulders, A.W.: The amsterdam library of object images. *Int. J. Comput. Vis.* **61**(1), 103–112 (2005)
21. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. II-409. IEEE (2003)