# Foveated Object Recognition Using Corners

Thomas Arnow[1], *Member, IEEE,* and Alan C. Bovik[2], *Fellow, IEEE*

*Laboratory for Image and Video Engineering (LIVE)*
*Department of Electrical and Computer Engineering*
*The University of Texas at Austin, Austin, TX 78712-1084, USA*
*[2]Email: bovik@ece.utexas.edu Telephone: (512) 471-5370*

## Abstract

*We present a gray scale object recognition system that is based on foveated corner finding and that uses elements of Lowe's SIFT algorithm. The principles behind the algorithm are the use of high-information gray-scale corners as features, and an efficient corner-finding strategy to find them. The system is tested on a set of tool and airplane images and shown to perform well.*

## 1. Introduction

A major problem in computer and biological vision is object recognition. The elusive goal is to efficiently and accurately recognize objects in natural scenes. Our system combines foveation with corner finding and a modified implementation of Lowe's SIFT algorithm [1] to achieve recognition. Foveation [2]-[6] refers to a vision system with varying spatial resolution. The finest resolution occurs at the center of gaze (fovea) but falls off drastically with eccentricity, or distance from the center of the fovea in degrees.

Any foveated vision system incorporates visual search: pointing the fovea to regions of interest in an image. The primate eye, for example, moves about a scene via very fast movements called saccades, resulting in series of static fixations [2]. In machine vision, search can be accomplished by aiming a camera or in software.

Foveation acts as a powerful form of visual data compression - the amount of information flowing from the retina to the brain is far less than if the entire retina was sampled at foveal density. Arnow and Bovik [7] discuss foveated visual search and corner detection. Here we extend that work to demonstrate a corner based foveated object recognition system to possibly be used in a future robot vision system.

## 2. Related work

Lowe [1] describes an object recognition system that is invariant under rotation, translation, and scale called the The Scale Invariant Feature Transform (SIFT) transform, based on histograms of gradients around points of interest called *keypoints* or SIFT features. We describe SIFT below because a key part of our recognition system is based upon it.

Helmer and Lowe [8] describe an object class recognition system based on breaking an object down into its component parts. Using SIFT to locate keypoints and other features, they create feature vectors based on location, scale, and "appearance," then using maximum likelihood, determine whether each new vector belongs to a new or already discovered component.

Serre et al. [9] created a recognition system based on regions of the visual cortex. They use a network of four alternating layers, two representing simple cells, two, complex cells. The image is input to the first layer, S1, a bank of Gabor filters. The output of S1 in turn passes through a layer of complex cells, C1: local maxima over varying positions and scales. Next comes another layer of simple cells, S2, radial basis function, then a layer of complex cells, C2, maximum pooling. The outputs of the two complex cell layers are passed to a linear classifier.

Mutch and Lowe [10] developed a recognition based on that of Serre et al. [9]. They made several changes to Serre's model, including a pyramid scheme, inhibition of S1 and C2 outputs.

Kadir and Brady [11] developed a system based on saliency, scale, and image content. Saliency means something in an image which "seizes the viewer's attention." They use local image entropy as a definition of saliency and locate positions and scales of interest points, then follow this information from frame to frame for object tracking.

## 3. Lowe's SIFT transform

In the original version of the SIFT algorithm, detected features are invariant to object translation, scale, rotation, and partially invariant (i.e. robust) to changing viewpoints, and change in illumination.

### 3.1. Keypoint detection

This step calculates difference of Gaussians (DoG) at a range of scales and locates local extrema of the DoGs at each scale. Any given octave has scales set at a fixed number of intervals. Each local extremum is a candidate for a point of interest called a *keypoint.*

For each potential keypoint, an orientation is calculated by finding gradient maxima in a local window centered about the keypoint. The window is then rotated about the keypoint by the orientation. This window, when passed to the final step, results in a scale, rotation, and translation independent keypoint descriptor. Lowe used the following formulae to calculate gradient magnitude and orientation.

$$m(x,y) = \sqrt{\left(L(x+1,y)-L(x-1,y)\right)^2 + \left(L(x,y+1)-L(x,y-1)\right)^2} \quad (1)$$

$$\theta(x,y) = \tan^{-1}\left(L(x,y+1)-L(x,y-1)\right) / \left(L(x+1,y)-L(x-1,y)\right) \quad (2)$$

where $L$ is a Gaussian smoothed image at the closest scale to that of keypoint. Next, the window of magnitudes centered about the keypoint is weighted by a circular Gaussian,

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

centered about the keypoint with $\sigma$ set to 1.5 times the scale of the keypoint. This operation gives extra weight to gradients near the keypoint.

A histogram of the magnitudes weighted by orientation is created for the region surrounding the keypoint. The histogram is divided into 36 bins, each representing 10°. A parabola is fit to the 3 histogram values closest to each peak to interpolate the peak position for better accuracy. A window from the original image, centered about each keypoint is rotated by the maximum orientation of the histogram. A 16x16 sub-window is extracted after this rotation, which creates rotation invariance since similar image features at different angles will all be rotated to have similar orientations. The orientation process is illustrated below in Figs. 1 and 2.

Fig. 1. Sample image. Two corners on a rectangle.

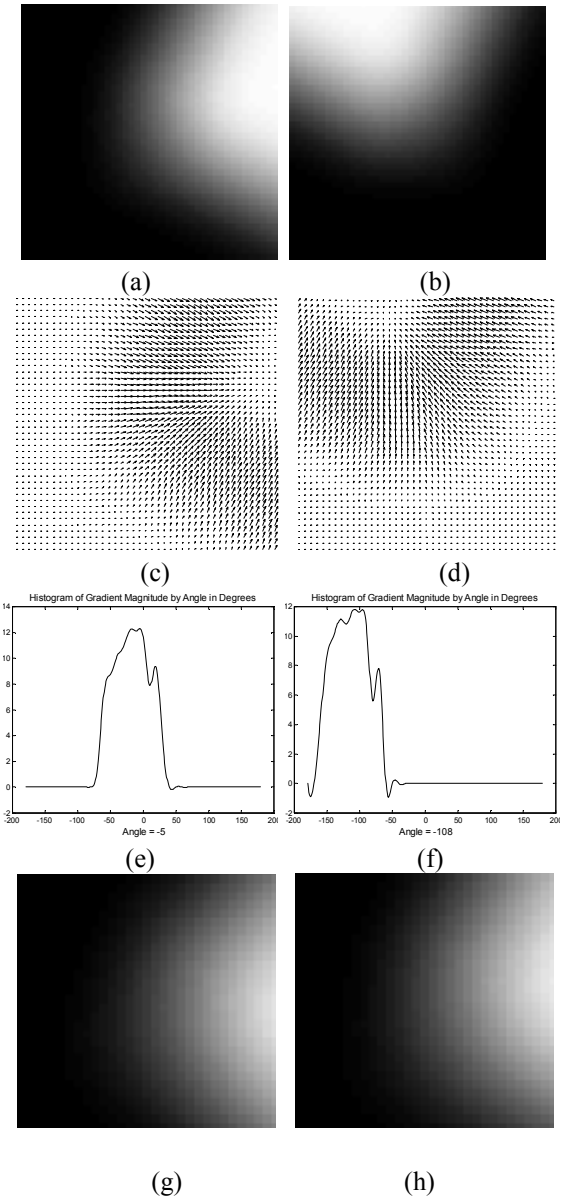


Fig. 2. Illustration of SIFT rotation invariance. (a), (b) Gaussian blurred corner images 1 and 2 from Fig. 1, respectively; (c), (d) corresponding gradients (needle diagrams); (e), (f) corresponding gradient histograms as a function of angle; (g), (h) the two corners after rotation.

### 3.2. Keypoint descriptor

The 16x16 rotated window is blurred by a Gaussian (3) with $\sigma$ set to the that of the scale of the keypoint. It is then weighted by another Gaussian centered about the keypoint, with a value of $\sigma$ set to half the window width or eight before converting the window to gradient magnitudes and phases. The Gaussian weighting makes the system more robust toward small mis-locations of the keypoint. The window is broken down into 4x4 sub-windows. For each sub-window a histogram of magnitudes into eight accumulators by phase angle. The result is a 4x4x8 or 128 pixel robust representation of the image about the keypoint.

In the implementation described here, keypoints are corners located by the foveated corner detection algorithm described in [7], which is based on the Canny edge detector [12]. Foveation is achieved by Gaussian filtering with a variable cutoff frequency, which increases with eccentricity [7]. Only two scales have been implemented to reduce complexity and execution time. The smaller scale uses the varyingcutoff frequency described in [7]. The other uses half that frequency.

## 4. Recognition Algorithm

The recognition system contains two algorithms. One creates a database, grouped by category, of SIFT features of known objects. The other identifies an unknown object by comparing its SIFT feature for each located corner with those stored in the database. Different objects have different numbers of corners.

The routine to create the database, collects a set of trial corners, and eliminates those with low curvature or gradient strength The remaining set of trial corners is sorted by decreasing curvature and placed into the database. Corners too near to one already in the database are eliminated.

During identification of an unknown object, corners are passed to the recognition algorithm "on the fly." Each corner of the unknown object is located, then filtered as described above, and if accepted, compared with the database. The correlation between it and each corner in the database is calculated:

$$C_{ab} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} b_{ij}}{\left(\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^{\ 2}\right)^{.5}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^{\ 2}\right)^{.5}} \qquad (4)$$

where $C_{ab}$ is the correlation between features $a$ and $b$. $C_{ab}$ is normalized to a maximum of 1.

A voting system is used to recognize an unknown object. First, a corner in the object is located. If it is of sufficient curvature and gradient strength, the correlation $C_{ij}$ between it and each corner in the database is calculated, where $i$ represents the corner in the unknown object and $j$ represents a corner in the database. The maximum of all the correlations of corner $i$ over the database is determined:

$$maxvalue_i = \arg\max_i C_{ij}, \qquad (5)$$

For each corner $i$, the category of the item in the database corresponding to $maxvalue_i$ receives a vote. The tentative choice, which may change as more corners are processed, is the category with the maximum number of corners. A stopping point is reached when either a sum of twelve votes has been reached or when at least 70% of the votes are for the same category.

### VII. Results

Eight categories of objects where used to test the algorithm. These include hand tools from Sclaroff [13] and the Rutgers tool database [14], and images of fighter and passenger jets. Figure 3 depicts a number of these images. Table I gives a description of each category and its two letter code.

Table I. Object types and codes

| Object Type | Total Num | Training Set | Num Correct | % Correct |
|---|---|---|---|---|
| Brush | 5 | 3 | 3 | 60.0 |
| Fighter Jet | 7 | 3 | 5 | 71.4 |
| Hammer | 5 | 3 | 4 | 80.0 |
| Passngr. Jet | 8 | 4 | 8 | 100 |
| Pliers | 8 | 4 | 8 | 100 |
| Screw Drvr. | 10 | 5 | 9 | 90.0 |
| Sledge Ham. | 10 | 5 | 6 | 60.0 |
| Wrench | 25 | 9 | 23 | 92.0 |
| Total Obj. | 78 | 36 | 66 | 84.6 |

Fig. 3. Sample images of each category.

**VIII. Conclusions**

We have proposed and developed a gray-level object recognition system based on the sequential foveated detection of high-information corners to be used as features in a SIFT-like recognition system. Similar foveated search processes in biological vision systems served as an inspiration to this approach. Results show promise, but significant increases in accuracy might require global information on the shape of the unknown object.

## 10. References


[1] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l J. Comput. Vision,* 60, 2, pp. 91-110, 2004.

[2] W.S. Geisler & M.S. Banks, "Visual performance," in *Handbook of Optics, Vol. 1: Fundamentals, Techniques, & Design* 2nd ed., M. Bass, Ed. New York: McGraw-Hill, 1995.

[3] A.L. Yarbus, *Eye Movements and Vision.* New York: Plenum Press, 1967.

[4] P.T. Kortum and W.S. Geisler, "Implementation of a foveated image coding system for image bandwidth reduction," *SPIE 2657*, pp. 350-360, 1996.

[5] W.N. Klarquist and A.C. Bovik, "FOVEA: A foveated vergent active stereo vision system for dynamic three-dimensional scene recovery," *IEEE Trans. Robotics and Automation.,* vol. 14, no. 2, pp. 755-770, 1998.

[6] W.S Geisler and D.B. Hamilton, "Sampling-theory analysis of spatial vision," *Journal of the Optical Society of America A*, vol. 3, no. 1, pp. 62-70, 1986.

[7] T. Arnow and A.C. Bovik, "Foveated visual search for corners," *IEEE Trans Image Process.*, vol. 16, pp. 813-823, March 2007.

[8] S. Helmer and D.G. Lowe, "Object recognition with many local features," *Workshop on Generative Model Based Vision 2004 (GMBV),* Washington, D.C. (July 2004).

[9] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber and T. Poggio, "Object recognition with cortex-like mechanisms," *IEEE Trans Pattern Anal Machine Intell*, 29 (3), pp. 411-426, 2007.

[10] J. Mutch and D.G. Lowe, "Multiclass object recognition with sparse, localized features," *IEEE Conf Computer Vision and Pattern Recognition,* New York, NY, pp. 11-18, 2006.

[11] T. Kadir and M. Brady. Scale, saliency and image description. *Int'l J. Comput Vision*, 45 (2) pp. 83–105, 2001.

[12] J.F. Canny, "Finding Edges and Lines in Images," *AI Lab MIT Technical Report AD-A130824*, 1983.

[13] S. Sclaroff, "Deformable Prototypes for encoding shape categories in image databases." *Pattern Recognition* vol 30 no 4, pp 627-642, 1997.

[14] K. Siddiqi, A. Shokoufandeh, S. Dickinson and S. Zucker, "ShockGraphs and Shape Matching," *IEEE Int'l Conf. Comput. Vision*, Bombay, January 4-7, 1998.