IEEE MULTIMEDIA SYSTEMS '99
Dipartimento di Sistemi e Informatica
University of Florence
Via S. Marta 3
50139 Firenze,
ITALY

Dear Prof. Alberto del Bimbo,

Enclosed is the final version of our paper entitled, "IRUS: Image Retrieval Using Shape" for IEEE ICMCS'99.

best regards,
Michael S. Lew

Author Names and Affiliations
Meirav Adoram and Michael S. Lew
Leiden Institute for Advanced Computer Science
Niels Bohrweg 1
2333 CA Leiden
Netherlands

Contact Author: Michael S. Lew
Email: mlew@wi.leidenuniv.nl
Telephone: +31-71-527-7034

Abstract: *Finding shapes in image databases is a challenging topic in content based retrieval. In this paper the goal is to find database images which contain shapes similar to the query of the user. Unlike most solutions to this problem, the algorithm presented in this paper is meant to cope with changes in rotation, scale, translation, and lossy compression noise. A Java application was built which uses snakes and invariant moments. The GVF snake was used because it has two significant advantages over the traditional snake formulation. First, the GVF snake can fit into concavities, and second, the GVF snake can fit itself to objects using both expansion and contraction of the snake. The objects in the images were segmented with the active contours, and then invariant moments were calculated and compared with a minimum distance classifier. Retrieval quality of the system was measured with respect to original images, rotated images, scaled images, noisy images, and combinations of those distortions.*

Keywords: Content-based indexing/retrieval

IRUS: Image Retrieval Using Shape

Meirav Adoram and Michael S. Lew
Leiden Institute for Advanced Computer Science
2333 CA Leiden
Netherlands
{madoram, mlew}@cs.leidenuniv.nl

Abstract

Finding shapes in image databases is a challenging topic in content based retrieval. In this paper the goal is to find database images which contain shapes similar to the query of the user. Unlike most solutions to this problem, the algorithm presented in this paper is meant to cope with changes in rotation, scale, translation, and lossy compression noise. A Java application was built which uses snakes and invariant moments. The GVF snake was used because it has two significant advantages over the traditional snake formulation. First, the GVF snake can fit into concavities, and second, the GVF snake can fit itself to objects using both expansion and contraction of the snake. The objects in the images were segmented with the active contours, and then invariant moments were calculated and compared with a minimum distance classifier. Retrieval quality of the system was measured with respect to original images, rotated images, scaled images, noisy images, and combinations of those distortions.

1. Introduction

Locating and recognizing objects from images is a challenging goal. There is no general solution, and successful solutions are limited to specific domains. One of the obstacles that contribute to the fact that there is still no general solution is *segmentation*. How can we separate the objects from the background? Difficulties come from discretization, occlusions, poor contrast, viewing conditions, noise, complicated objects, complicated background etc. In the cases where the segmentation is less difficult and possible to overcome, the object shape is a characteristic which can contribute enormously in further analysis. If segmentation is not an option, a global search in the form of template matching is a possibility, where the template represents the desired object to be found. However, performing template matching over a dense structure of scales and rotations of an image is not an interactive solution regarding searches in large image databases.

In content-based image retrieval, the question of how to retrieve images of interest from a given image database is dealt with. Within this framework the shape of an object in an image is the primary retrieval characteristic in this paper.

Shape retrieval is an intuitive query method in any image retrieval system, either as stand-alone query or as a combined query of shape and other cues like color and texture. Object shape in the form of 2D boundary (contour) will be used to locate and retrieve images from image databases.

1.1 The Chosen Approach

In this work, the problem of image retrieval using shape was approached by active contours and invariant moments. Active contours were first introduced by Kass et al. [1], and were termed snakes by the nature of their movement. Active contours are a sophisticated approach to contour extraction and image interpretation. They are based on the idea of minimizing energy of a continuous spline contour subject to constraints on both its autonomous shape and external forces derived from a superposed image that pull the active contour toward image features such as lines and edges.

Moments [2] describe shape in terms of its area, position, orientation, and other parameters. The set of invariant moments[3] makes a useful feature vector for the recognition of objects which must be detected regardless of position, size or orientation. Matching of the invariant moments feature vectors is computationally inexpensive and a promising candidate for interactive applications.

In our approach, the active contours are used in segmentation of objects offline, and invariant moments are used to match the objects online.

2. Active Contours

Active contours challenge the widely held view of bottom-up vision processes. The principal disadvantage with the bottom-up approach is its serial nature; errors generated at a low-level are passed on through the system without the possibility of correction. The principal advantage of active contours is that the image data, the initial estimate, desired contour properties and knowledge-based constraints are integrated into a single extraction process.

In the literature, A. D. Bimbo et al.[4] deform active contours over a shape in an image and measure the similarity between the two upon the degree of overlap and on how much energy the active contour had to spend in the deformation. A. K. Jain et al. [5] used a matching scheme

with deformable templates. Our work is different in that we use a GVF[8] based method to improve the automatic fit of the snakes to the object contours. Furthermore, we used invariant moments to find images which have been scaled and rotated.

Active contours are defined as energy-minimizing splines under the influence of internal and external forces. The internal forces of the active contour serve as a smoothness constraint designed to hold the active contour together (elasticity forces) and to keep it from bending too much (bending forces). The external forces guide the active contour towards image features such as high intensity gradients. The optimal contour position is computed such that the total energy is minimized. The contour can hence be viewed as a reasonable balance between geometrical smoothness properties and local correspondence with the intensity function of the reference image.

Let the active contour be given by a parametric representation $v(s) = (x(s), y(s))$, with s as the normalized arc length of the contour. The expression for the total energy can then be decomposed as follows:

$$E_{total} = \int_0^1 E(v(s))\,ds = \int_0^1 [E_{int}(v(s)) + E_{image}(v(s)) + E_{con}(v(s))]\,ds \quad (2.1)$$

where E_{int} represents the internal forces (or energy) which encourage smooth curves, E_{image} represents the local correspondence with the image function, and E_{con} represents a constraint force that can be included to attract the contour to specific points in the image plane. In the following discussions the E_{con} will be ignored. E_{image} is typically defined such that locations with high image gradients or short distances to image gradients are assigned low energy values.

2.1. Internal Energy

E_{int} is the internal energy term which controls the natural behavior of the active contour. It is designed to minimize the active contour's curvature and make it behave in an elastic manner. Regarding Kass et al. [1], the internal energy is defined as

$$E_{int}(v(s)) = \alpha(s)\left|\frac{dv(s)}{ds}\right|^2 + \beta(s)\left|\frac{d^2v(s)}{ds^2}\right|^2 \quad (2.2)$$

where $|\ |$ is the Euclidean norm. The first order continuity term, weighted by $\alpha(s)$, makes the contour behave elastically, whilst the second order curvature term, weighted by $\beta(s)$, makes it resistant to bending. Setting $\beta(s) = 0$ at a point s allows the active contour to become second order discontinuous at that point and develop a corner. Setting $\alpha(s) = \beta(s) = 0$ at a point s allows the active contour to become discontinuous. Active contours can interpolate gaps in edges phenomena known as subjective contours due to the use of the internal energy. It should be noted that $\alpha(s)$ and $\beta(s)$ are defined to be functions of the curve parameter s, and hence segments of the active contour may have different natural behavior. Minimizing the energy of the derivatives gives a smooth function.

2.2. Image Energy

E_{image} is the image energy term derived from the image data over which the active contour lies and is constructed to attract the active contour to desired feature points in the image, such as edges and lines. The edge based functional attracts the active contour to contours with large image gradients – that is, to locations of strong edges.

$$E_{edge} = -\left|\nabla I(x, y)\right| \quad (2.3)$$

2.3. Problems with Active Contours

There are a number of fundamental problems with the active contours, and solutions to those problems sometimes create problems in other components of the active contour model.

Initialization – The final extracted contour is highly dependent on the position and shape of the initial contour due to the presence of many local minima in the energy function. The initial contour must be placed near the required feature otherwise the contour can become obstructed by unwanted features like JPEG compression artifacts, closeness of a nearby object, and different other noises.

Non-convex shapes – How do we extract non-convex shapes without compensating the importance of the internal forces, or without a corruption of the image data? For example pressure forces [9] (addition to the external force) can push an active contour into boundary concavities, but cannot be too strong or weak edges will be ignored. Pressure forces must also be initialized to push out or push in, a condition that mandates careful initialization.

The original method of Kass et al. [1] suffered from three main problems: dependence on the initial contour, numerical instability, and lack of guaranteed convergence to the global energy minimum. Amini et al. [6] improved the numerical instability by minimizing the energy functional using dynamic programming, which allows inclusion of hard constraints into the energy functional. However, memory requirements are large, being $O(nm^2)$, and the method is slow, being $O(nm^3)$. n is the number of contour points and m is the neighborhood size to which a contour point allowed to move in a single iteration. Seeing the difficulties with both

previous methods Williams and Shah [7] developed the greedy algorithm which combines speed, flexibility, and simplicity. The greedy algorithm is faster O(nm) than the dynamic programming and more stable and flexible for including constraints than the variational approach of Kass. During each iteration, a neighborhood of each point is examined and a point in the neighborhood with the smallest energy value provides the new location of the point. Iterations continue till the number of points in the active contour that moved to a new location in one iteration is below a specified threshold.

2.4. Gradient Vector Flow

Since the greedy algorithm easily accommodates new changes, there are three things we would like to add to it: the ability to inflate the contour as well as deflate it, deform to concavities, and to increase the capture range of the external forces. Those three additions reduce the sensitivity to initialization of the active contour and allow deformation inside concavities. This can be done by replacing the existing external force (image term) with the gradient vector flow (GVF) [8]. The GVF is an external force computed as a diffusion of the gradient vectors of an image, without blurring the edges. The idea of the diffusion equation is taken from physics. An example of the effect of the GVF external force can be seen in Figure 2.1. Figure 2.1 (a)-(c) shows the differences between the deformation with the gradient magnitude and the deformation with the gradient vector flow in the presence of a concavity. In Figure 2.2 we see the ability to inflate due to the use of the GVF as an external force. Figure 2.3 demonstrates the ability of the active contour with the GVF to cope with initialization across the shape.

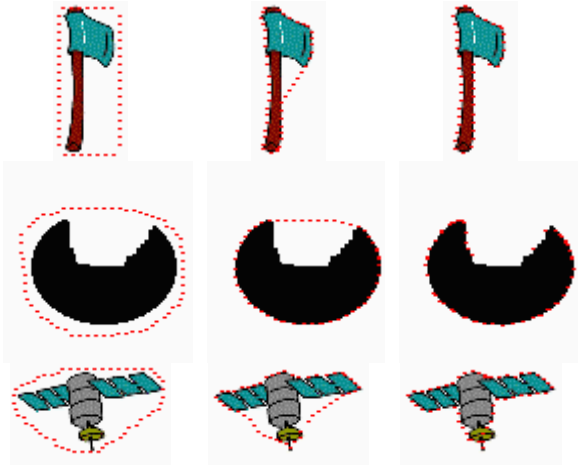
Xu and Prince in [8] define the gradient vector flow (GVF) field to be the vector field $\mathbf{v}(i,j) = (u(i,j), v(i,j))$ which is updated with every iteration of the diffusion equations, Equations (2.4) and (2.5).

$$u_{i,j}^{n+1} = (1 - b_{i,j})\,u_{i,j}^{n} + (u_{i+1,j}^{n} + u_{i,j+1}^{n} + u_{i-1,j}^{n} + u_{i,j-1}^{n} - 4\,u_{i,j}^{n}) + c_{i,j}^{1} \qquad (2.4)$$

$$v_{i,j}^{n+1} = (1 - b_{i,j})\,v_{i,j}^{n} + (v_{i+1,j}^{n} + v_{i,j+1}^{n} + v_{i-1,j}^{n} + v_{i,j-1}^{n} - 4\,v_{i,j}^{n}) + c_{i,j}^{2} \qquad (2.5)$$

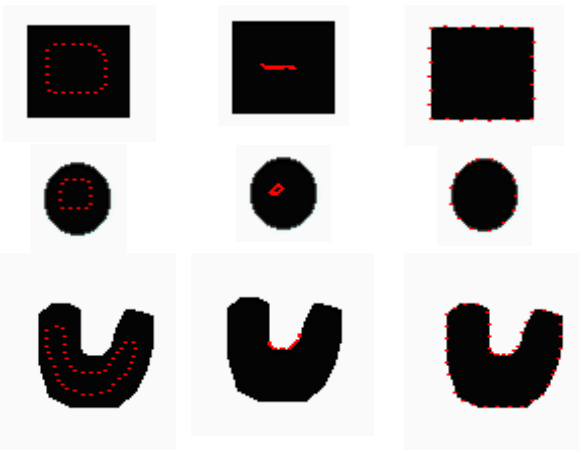
The initial values of u and v are the gradient values.

$$b_{i,j} = G_i(i,j)^2 + G_j(i,j)^2 \qquad (2.6)$$



(a) (b) (c)

Figure 2.1 The convergence of the active contour, (a) an initial position of the active contour in red dots for (b) and (c), (b) deformation with the gradient magnitude as the external force, (c) deformation with the GVF as the external force.



(a) (b) (c)

Figure 2.2 Inflation of the active contour, (a) initial position of the active contour, (b) final position of the active contour with the gradient magnitude, (c) final position of the active contour with the GVF.

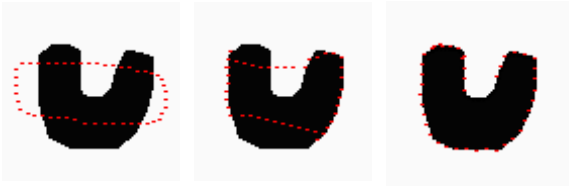
Where G_i is the first element of the gradient vector and G_j is the second element.

$$c^{1}(i,j) = b_{i,j} G_i(i,j) \qquad (2.7)$$

$$c^{2}(i,j) = b_{i,j} G_j(i,j) \qquad (2.8)$$

The second term in Equations (2.4) and (2.5) is the Laplacian operator. The intuition behind the diffusion equations is that in homogeneous regions, the first and third terms are zeros since the gradient is zero, and within those regions, u and v are each determined by Laplace's equation. This results in a type of "filling-in" of information taken from the boundaries

of the region. In regions of high gradient **v** is kept nearly equal to the gradient.



(a) (b) (c)

Figure 2.3. Initialization across the shape, (a) initial position, (b) deformation with the gradient magnitude, (c) deformation with the GVF.

Creating GVF field yields streamlines to a strong edges. In the presence of these streamlines, blobs and thin lines in the way to strong edges do not form any impediments to the movement of the active contour. It can be considered as an advantage if the blobs are in front of the shape, nevertheless it can be considered as a disadvantage if the active contour enters the shape's silhouette.

3. Invariant Moments

For a 2-D continuous function f(x, y), the moment of order (p + q) is defined for p,q =0,1,2,... as

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p\, y^q f(x, y)\, dxdy \qquad (3.1)$$

The infinite set of moments $\{m_{pq}, p, q = 0, 1, \ldots\}$ uniquely determine f(x, y), and vice-versa. In the case of a digital image, the moments are approximated by

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \qquad (3.2)$$

where the order of the moment is (p + q) as the above formulation, x and y are the pixel coordinates relative to some arbitrary standard origin, and f(x, y) represents the pixel brightness.

In a binary image m_{00} is the same as the area. The centroid of a shape can be expressed in terms of moments as $\bar{x} = \frac{m_{10}}{m_{00}}$ and $\bar{y} = \frac{m_{01}}{m_{00}}$.

To have moments that are invariant to translation, scale, and rotation, first the central moments are calculated, μ,

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \qquad (3.3)$$

and then the normalized central moments are calculated, η, as

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{\lambda}} \qquad (3.4)$$

where $\lambda = \frac{(p+q)}{2} + 1$, and $(p+q) \geq 2$.

From these normalized parameters a set of invariant moments {ϕ} found by Hu[3], may then be calculated. The seven equations of the invariant moments contain terms up to order 3. The following is the derived list of the invariant moments:

$$\phi_1 = \eta_{20} + \eta_{02} \qquad (3.5)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi_4 = (\eta_{30} - \eta_{12})^2 + (\eta_{21} - \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\} + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\{3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\}$$

$$\phi_6 = (\eta_{20} - \eta_{02})\{(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\} + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{12} - \eta_{30})(\eta_{30} + \eta_{12})\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\} + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\{3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\}$$

Global (region) properties provide a firm common base for similarity measure between shapes silhouettes where gross structural features can be characterized by those moments. Since we don't deal with occlusion and the noise factor in the local search, the invariance to position, size, and orientation, and finally the low dimensionality as a vector of seven elements seems to be a good choice for using the invariant moments in matching shapes. The logarithm of the invariant moments is taken to reduce the dynamic range.

4. The Implementation of IRUS

In the off-line processing on every image in the database we place one active contour and deform it to the edges of the object in the image. From the resulting contour we calculate the invariant moments yielding a vector of seven elements. For every image we have two vectors, one for every algorithm. Those two vectors of every image in the database are written to a file called "features.res".

In the on-line processing, the active contour is placed around the user's shape, and deformed using the specified algorithm. Next, the invariant moments are calculated producing a vector of seven elements. The vector of the invariant moments is matched with the Euclidean distance to all the vectors from the chosen algorithm in the "features.res" file. The number of best matches to be shown can be adjusted interactively. Figure 4.1 shows the system flowchart.

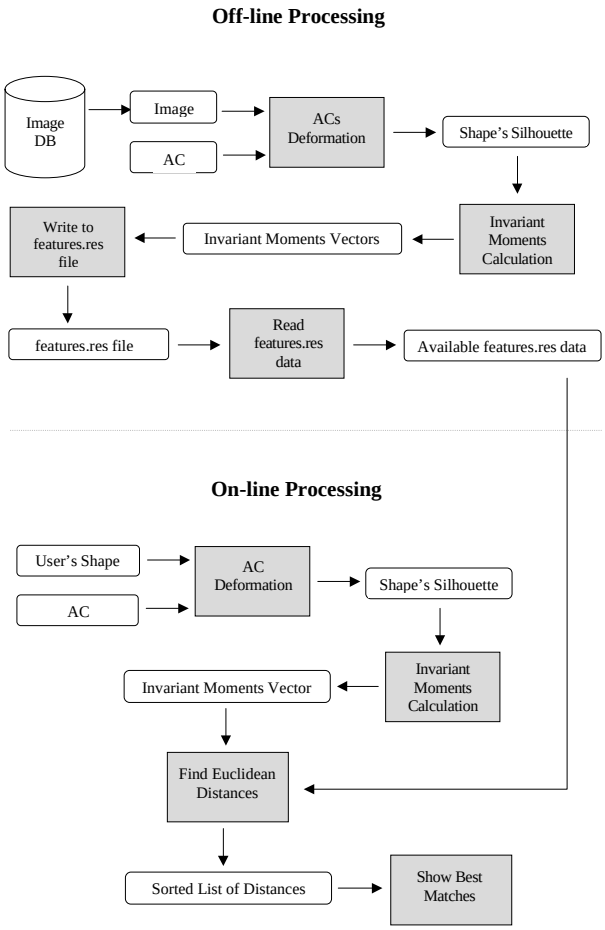


Figure 4.1 The Implementation of IRUS

5. Experiments

This section examines the retrieval behavior of IRUS. The test set preparation, the performance evaluation technique to measure retrieval accuracy, and the results including retrieval speed are all presented and discussed.

5.1. Database and Test Set

The image database used in this work consisted of 3,500 graphics and trademarks images. These images were chosen because they have precisely one object per image. Nevertheless, they are useful to graphics professionals and often used in presentations, flyers, brochures, and websites. Samples of images from the database are shown below in Figure 5.1. To be able to measure the performance with respect to scale, rotation, and noise, we took the following steps: 1500 images were randomly selected from the database to serve as the queries group.

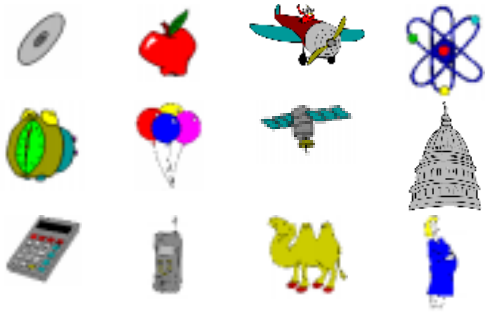


Figure 5.1 Samples of images from the database.

For each image from the queries group we applied the following operations, and the resulting images were added to the database.

(a) Random rotation between: 0 to 360 degrees.
(b) Random scale between 50% - 200%.
(c) JPEG compression noise from Quality = 75%.
(d) Combination of (a) and (b) with the same values as in (a) and (b).
(e) Combination of (a), (b), and (c), with the same values used in (a), (b), and (c).

5.2. The Performance Evaluation Technique

The performance evaluation technique that we use are based on ideas proposed for the Philips Research benchmarking investigation [10]. In their study, the "visible window" is defined to be the top $\log_2 n$ (n is the database size) matches. The premise behind this criterion is that a user typically only looks at the first page of results. As the database size, n, grows, a visible window of size L = log n stays user friendly. The visible fraction is defined as the percentage of correct results that appear in the visible window.

The second important question is where the correct matches are placed within the visible window. The visible position, P_v, is defined as the ranking error divided by the size of the visible window, L.

5.3. Results

The weights used for the testing were 0.3 continuity weight, 0.6 curvature weight, and 1.4 image weight. Since the distance to the object in the database image is not known, we need relatively high image weight (1.4). We also would like to maintain some structure in the active contour's deformation (0.6), and have equal distances between active contour's points (0.3). With the greedy + GVF algorithm, the weights values were: 0.4 continuity weight, 0.3 curvature weight, and 0.65 image weight. Since the GVF field provides extended capture range, the image weight (0.65) is lower compared to the greedy case.

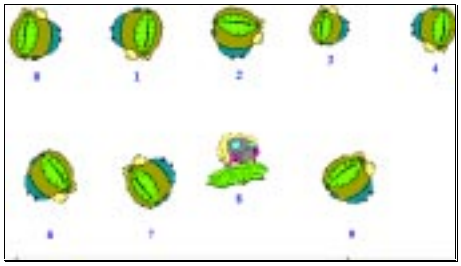


Figure 5.2. Results from an image query of the clock in position 0.

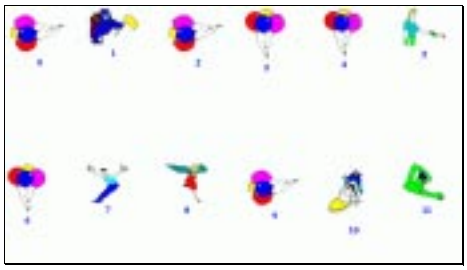


Figure 5.3. Results from a query of the balloons in position 0.

Figures 5.2 and 5.3 display results for images of a clock and ballons. Note that in Figure 5.3., results 7 and 8 were found before 9. This occurred because the active contour for result 9 went through one of the strings. For these tests, the GVF based method had a consistently greater visible fraction and visible position as shown in Figure 5.4. The most difficult situation was rotation for the Greedy method and JPEG compression noise for GVF. The average retrieval time was 1.6 seconds on a Pentium II at 266 MHZ with 128MB RAM.

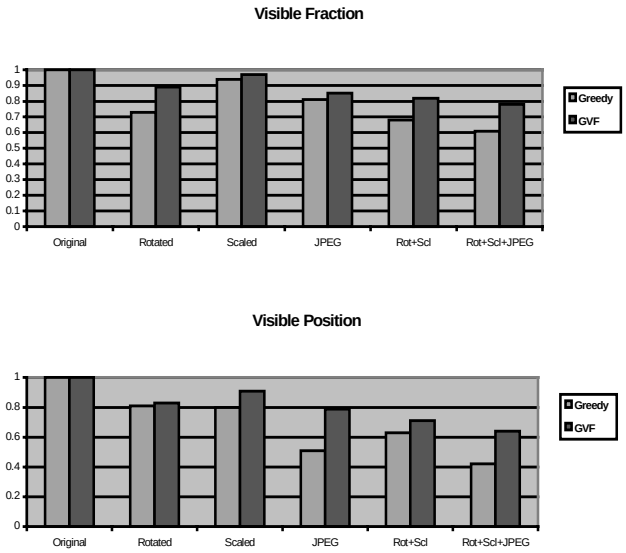


Figure 5.4 The retrieval performance results for L = 11

5.4. Limitations

We assume that there is only one object in each image. Complex scenes with many objects will not be segmented correctly, which will result in incorrect moments and low retrieval accuracy. As mentioned earlier, the general problem of segmentation is difficult and unlikely to be completely solved in the near future. Furthermore, many useful contexts only have one object per image such as clipart and trademark databases. The snake method has the advantage that a smoothness parameter can be adjusted for applications where the discretization noise or lossy compression artifacts are known to be large.

6. Conclusions

In this work, we showed that the GVF based snakes give better retrieval results than the traditional snakes. In particular, the GVF snakes have the advantage over traditional snakes in that it is not necessary to know apriori whether the snake must be expanded or contracted to fit the object contour. Furthermore, the GVF snakes have the ability to fit into concavities in the object, which traditional snakes typically can not do. Both of these factors resulted in significant retrieval accuracy improvements in the context of finding rotated, scaled, and JPEG compressed images.

Future work will focus on comparing different shape features regarding retrieval accuracy, and investigating automatic methods for segmenting multiple objects from the database images.

Acknowledgements

This research was funded by a grant from Philips Research, Eindhoven.

References

[1] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active Contour Models”, International Journal of Computer Vision, pp. 321-331, 1988.

[2] R. C. Gonzalez and R. E. Woods, “Digital Image Processing”, Addison Wesley, 1993.

[3] M.K. Hu, “Visual Pattern Recognition by Moment Invariants”, IRA Trans. on Information Theory, vol. 17-8, no. 2, pp. 179-187, Feb. 1962.

[4] A. Del Bimbo and P. Pala, “Visual Image Retrieval by Elastic Matching of User Sketches”, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 19, No. 2, pp. 121-132, February 1997.

[5] A. K. Jain, Y. Zhong, and S. Lakshmanan, “Object Matching Using Deformable Template”, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 18, No. 3, pp. 267-278, March 1996.

[6] A. A. Amini, S. Tehrani, and T. E. Weymouth. “Using dynamic programming for minimizing the energy of active contours in the presence of hard constraints”, in Proceedings, Second International Conference on Computer Vision, pp. 95 – 99, 1988.

[7] D. J. Williams and M. Shah, “A fast algorithm for active contours and curvature estimation”, CVGIP: Image Understanding, pp. 14 – 26, January 1992.

[8] C. Xu and J. L. Prince, “ Gradient Vector Flow: A New External Force for Snakes”, IEEE Proc. Conf. On Comp. Vis. Patt. Recog. (CVPR’ 97), pp. 66-71.

[9] L. Cohen, “On Active Contour Models and Balloons”, CVGIP: Image Understanding, Vol. 53, No. 2, March, pp. 211-218, 1991.

[10] D. P. Huijsmans, M. S. Lew, D. Denteneer, “Quality measures for interactive image retrieval with a performance evaluation of two 3x3 texel-based methods”, International Conference on Image Analysis and Processing, pp. 22-29, Sept., 1997.