Spiral Object Recognition on Clusters

Xiangjian He, Qiang Wu and Tom Hintz
Department of Computer Systems
Faculty of Information Technology
University of Technology, Sydney
PO Box 123, Broadway 2007
NSW, Australia

Abstract *Object matching has many potential applications in industry, defense and medical science. Most matching methods introduced in recent years are based on the invariant representations. Main invariants applied in computer vision are algebraic, differential invariants and integral invariants. Our approach in this paper uses an affine integral invariant within a Spiral Architecture. The invariant representation is based on the extracted object contour. The parameter to be used for parameterizing an object contour is derived from the enclosed area. The Spiral Architecture posseses powerful computation features that are pertinent to the vision process. We present a parallel algorithm for object recognition on clusters. Image partitioning based on Spiral Architecture provides well-balanced load and absolutely uniform sub-images. The cluster-based object recognition greatly increases computation speed.*

Keywords: parallel algorithm, computer cluster, computer vision, object recognition

1 Introduction

A fundamental problem of object recognition is recognising objects within a given scene. Many recognition algorithms have been proposed. Typical of these are Chamfer recognition [2], Borgefors recognition [3] and the Hausdorff distance [10]. Remote motion detection is one of the areas requires the object recognition. One approach to the moving object recognition is to use the template matching technique [11] to convert an image sequence into a static shape pattern and then compare it to the prestored prototypes during the recognition process.

Contours extracted from images have proven quite useful in many object recognition algorithms. Based on the contours formed by the edge points, feature-based methods for the recognition of objects independent of their position, size, orientation and other variations have been the goal of recent research efforts (see, for example, [4, 7, 6, 9, 15]). Finding efficient invariant features is key to solving this problem.

The three main invariants applied in computer vision are algebraic, differential and integral invariants. Either type of invariant has advantages and disadvantages in computer vision applications as elaborated in [12] .

In this paper, a recognition method is presented. It is approached by the construction of an *integral invariant* [9] based on the *enclosed area* within a Spiral Architecture. The Spiral Architecture pocesses powerful computational features that are pertinent to the vision process.

In order to increase the computation speed, we propose a parallel algorithm for object recognition. We apply the high speed Spiral rotation [13] to uniformly partition images. This image partitioning not only divides an image into absolutely equal sizes of sub-images but also well guarantees the load balancing. Each sub-image is actually a contraction of the origional image with a rotation. The parallel algorithm assigns each slave node to work independently on the object recognition based on a sub-image. Edge detection and contour extraction of a 2D object play very important roles in this research. The algorithm is implemented on a cluster of Sun-workstaions.

The organisation of this paper is as follows. Section 2 shows an image partitioning method within the Spiral Architecture. We present the extraction of an invariant object representation

on each sub-image in Section 3. This is followed by the presentation of the parallel algorithm on clusters for object recognition in Section 4. The experimental results are demonstrated in Section 5. We conclude in Section 6.

2 Image Partitioning

The Spiral Architecture is used as an image data structure because of its inherent computation speed in basic image operation [14]. This section delas with image partitioning within the Spiral Arcitecture.

2.1 The Spiral Architecture

An image may be considered as the collection of pixels (picture elements). These elements correspond to the position of the photo receiving cells of the image capturing device. In the case of the human eye, these elements would represent the relative position of the rods and cones on the retina. The geometric arrangement of cones on the primate's retina can be described in terms of a hexagonal grid. This leads to the consideration of an image as an ordered collection of hexagonal cells. Each cell is assigned a number called 'Spiral Address' counted from 0 [14]. Along with the Spiral Addition and Spiral Multiplication operations on the Spiral Addresses, this collection forms the Spiral Architecture [14].

- Figure 1 shows a sample image (an arrow) which is represented by a collection of 7^5 hexagonal pixels.

Here, we use a set of four rectangular pixels to mimic a hexagonal pixel as shown in [5]. Figure 2 is a collection of seven hexagonal pixels constructed in this way [8].

2.2 Uniform data partitioning

Sheridan in his PhD thesis [14] showed that multiplying an image represented in the Spiral Architecture by a Spiral Address represented a rotation with a scaling (i.e., a contraction). He also showed that the rotation implied a uniform partitioning of a image with a proper selection of the Spiral Address. In fact, each partition is

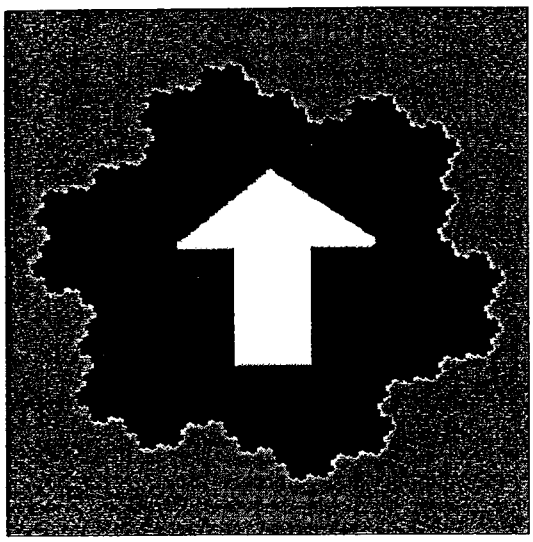

Figure 1: Sample image of "the arrow" in spiral space.

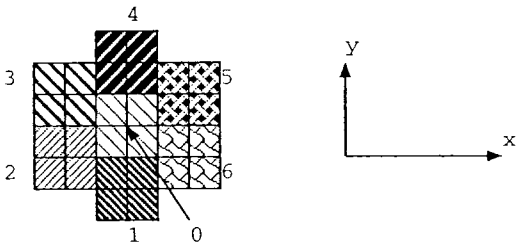

Figure 2: Distribution of 7 pixels constructed from rectangular pixels.

a contraction of the original image of the same size as shown in Figure 3 [1].

For example, multiplying an image by 10, separates the image into seven sub-images of the same size if the total number of hexagonal pixels for the image representation is 7^2. As all sub-images look almost the same, this partitioning scheme guarantees well load balancing for parallel processing. The importance of using this scheme is that the computation of the Spiral Multiplication is very fast. Each multiplication is implemented using an addition by taking a 'log' operation [13]. This significantly

[1] The occurrence of image distortion after partitioning is because the Spiral Structure in this paper is a mimic structure.

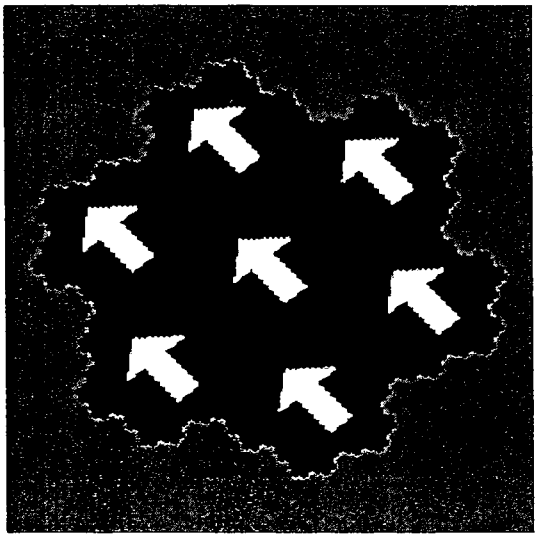

Figure 3: Seven partitions of "the arrow" in spiral space.

increases the computation speed.

3 Local Object Representation

In this section, we present object recognition on each sub-image. The recognition scheme is based on an affine invariant and the object contour extracted.

3.1 Affine invariant parameter

Let C be the pre-extracted contour on the plane represented by $(x(t), y(t))$ with parameter t. There is a well-known parameter which is linearly transformed under an affine transformation [12], and can be used for parameterising object contours. This is the enclosed area σ defined as in [1]

$$\sigma = \frac{1}{2}\int_{t_0}^{t} |x\dot{y} - y\dot{x}|dt. \tag{1}$$

Define a new parameter s as follows

$$s = \frac{2\sigma}{\int_C |x\dot{y} - y\dot{x}|dt} \tag{2}$$

where $\int_C$ denotes the line integral along C, then s is invariant under any affine transformation (See [9]), i.e., if $\tilde{s}$ is obtained from s undergoing an affine transformation, then $\tilde{s}$ is the parameter of $\tilde{C}$ and equal to s. Here, $\tilde{C}$ is the curve obtained from C through the affine transformation. In this paper, we use s as the parameter of the object contour. It is easy to see that the range of s is $[0, 1]$.

We now proceed to find the s value at each point or pixel on a contour. Let us denote the enclosed area of a on the contour C by $E(a)$. The 'area' enclosed by the contour C, denoted by E, is given by

$$E = \sum_{a \in C} E(a). \tag{3}$$

Without loss of generality, we can choose any pixel on the contour and assign the parameter value $s = 0$ to this pixel. Then this pixel becomes the starting point of the contour. We denote the starting point a_0.

For any other point a on the contour, along the positive tangent direction (see [5]) starting from a_0, find all pixels between a_0 and a. Suppose that these pixels are $a_1, a_2, \cdots, a_n$, where $n \leq N$, a_{i+1} is the adjacent pixel of a_i in the positive tangent direction of a_i for $0 \leq i < n$ and $a_n = a$ (see Figure 4). Then the s value at a, denoted by s_a, is implemented by

$$s_a = \frac{1}{E}\sum_{i=0}^{n-1} E_{a_i}. \tag{4}$$

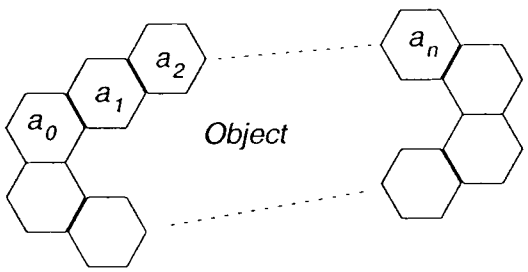

Figure 4: Distribution of contour points.

3.2 Representation of an affine integral invariant

Let the contour C be represented by

$$V(s) = \begin{pmatrix} x(s) \\ y(s) \end{pmatrix}$$

for $s \in [0,1]$. Let

$$I(s) = det[V(s+\Delta s) - V(s), V(s-\Delta s) - V(s)], \quad (5)$$

where $det[v_1, v_2]$ denotes the determinant of a matrix which consists of two column veotors, $v_1, v_2 \in R^2$. It easy to see that $I(s)$ represents the area made by two vectors, $V(s+\Delta s) - V(s)$ and $V(s-\Delta s) - V(s)$, as shown in Figure 5.

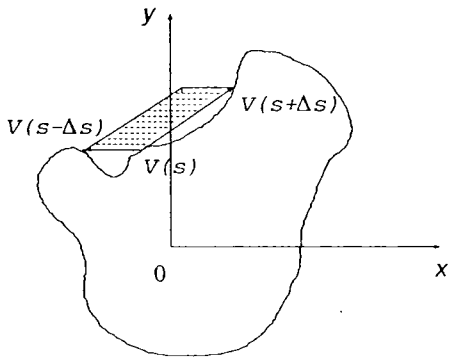

Figure 5: $I(s)$ is the area made by $V(s+\Delta s) - V(s)$ and $V(s-\Delta s) - V(s)$.

He et al. have proved in their paper [9] that $I(s)$ shown in Equation 5 is a relative invariant under the affine transformation, i.e., it relates to its image by a multiplication constant $det(A)$, where A is the affine transformation matrix. The multiplication constant can be removed then the representation is expressed in a ratio form as

$$M(s) = \frac{I(s)}{I(v)}, \quad (6)$$

where the position v is the location of the largest absolute magnitude of the relatively invariant representation $I(s)$.

Equation 6 is defined as the affine invariant representation at the contour point with parameter value s in this paper.

3.3 Local algorithm on object feature

The proposed affine integral invariant suggests below a procedure for finding the object feature in a sub-image. The area or the set of hexagonal pixels for the sub-image is called a 'local area'.

1. By applying a method for contour extraction to sub-image, the contour of the object in the sub-image is extracted.

2. For the contour,

   - the parameter value s at any contour point is computed by Equation 4;
   - determine contour points which have parameter values approximate to $s + \Delta s$ and $s - \Delta s$ for any given contour point with parameter s and a fixed Δs (which is set to be the same for all s, i.e., it is independent of s);
   - $I(s)$ and hence $M(s)$ defined in Equation 6 for any point with parameter s is computed;
   - compute the average $M(s)$ on the contour and denote it by $\bar{M}$, i.e.,

$$\bar{M} = \frac{1}{N} \sum \{M(s) | V(s) \in C\} \quad (7)$$

   where C is the contour, $V(s)$ is the point on C with parameter s and N is the number of points on the contour.

$\bar{M}$ obtained in this section is the feature of the object in the sub-image. This value is invariant under general affine transformation.

4 Parallel Algorithm on Clusters

One important application of the recognition of an unknown object is to recognise the object in an image as one of a number of model objects. This process requires the representation (or the feature) of the object be matched with one of those in the database. An object in a scene is considered to belong to a specific class if the degree of dissimilarity between the unknown object and models of that class is the smallest in comparision to the others. In this section, we present a parallel algorithm for object recognition based on a client-server model. The algorithm applies the uniform partitioning technique as described in Section 2.

1. Each image is represented by a collection of 7^5 hexagonal pixels.

2. The master node partitions the images into sub-images. This is done by multiplying the images by Spiral Address 10^4. The images include those containing model objects and the one containing the unknown object.

3. The master assign each sub-image to a slave node.

4. Each slave node work independently on the sub-image for contour and feature extraction of the object within the sub-image. The same Δs value is used by all slave nodes.

5. The master node collects the results from all slave nodes. These results include the features ($\bar{M}$s) of the model objects and the unknown object. Note that each of the objects has seven feature values, of which each is calculated by a slave node.

6. The master node compares the $\bar{M}$ value of the contour of the unknown object with the value of each model object within the same local area. The difference between the two values should be small if the two contours represent the same object, of which one can be obtained from another using an affine transformation. This difference defines the *degree of dissimilarity* between the two objects.

7. The master node take the average of the degrees of dissimilarity for each pair of unknown object and model object over the seven local areas. This is the square root of the sum of the degrees square.

8. The smaller average value indicates a better match between the unknown object and the model object.

5 Experimental results

As a simplified illustration of our parallel matching algorithm, three images represented in the Spiral Architecture are considered. One is the image containing the arrow as shown in Figure 1. The other two are shown in Figure 6. The ellipsoid, disk and arrow are the objects in these images. We are comparing the ellipsoid with the disk and the arrow, i.e., we are going

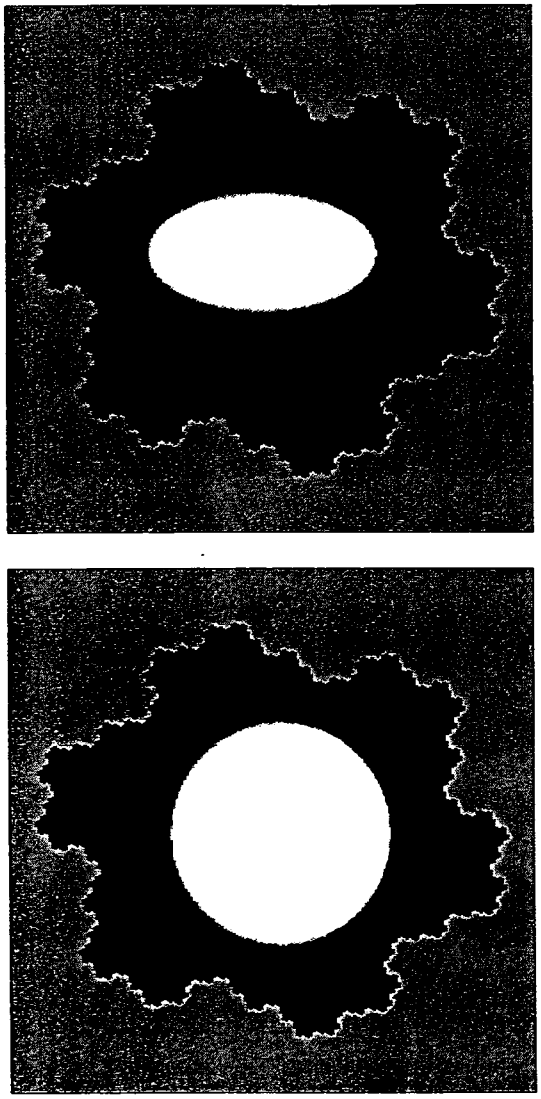

Figure 6: (upper) an ellipsoid and (lower) a disk.

to find out which of the disk or the arrow the ellipsoid matches.

Twenty two Sun-Workstations (Ultra 5, RAM 128MB, CPU 330Mbps) are used for the computation. One is used as the master and the remaining twenty one are used as slaves. Applying a Spiral Multiplication to these images, each image is separated into seven sub-images (see, for example, Figure 3 for the arrow sub-images). The contours of these objects in the seven local areas are shown in Figure 7.

For this algorithm, Δs set to be the value such that the pixel with $s \pm \Delta s$ is about two

Table 1: **Object Features**

	Ellipsoid	Disk	Arrow
Area 1	0.360000	0.410526	0.106383
Area 2	0.147260	0.258681	0.267025
Area 3	0.136986	0.202128	0.214286
Area 4	0.195946	0.225263	0.301075
Area 5	0.138514	0.144737	0.159722
Area 6	0.136905	0.214286	0.245520
Area 7	0.199324	0.299479	0.141053

pixels away from the pixel with s. It is calculated that $\Delta s = 0.023241$. The values of $\bar{M}$ of these contours with the Δs values are listed in Table 1. This table shows the following.

1. The feature values ($\bar{M}$s) of each object in each local area.

2. The degrees of dissimilarity on local areas 1-6 indicate that the Ellipsoid match the Disk better than the Arrow. For example, in local area 1, the degree of dissimilarity between the ellipsoid and the disk is $0.410526 - 0.360000 = 0.05526$ and is much smaller than the degree of between the ellipsoid and the arrow which is $0.360000 - 0.106383 = 0.253617$.

3. However, in area 7, the degree of dissimilarity between the ellipsoid and the disk is bigger. Hence, results at this area tells a wrong match. This is caused by the image noise. Hence, it is necessary to take an average of the results in all seven areas.

4. The average of the degrees of dissimilarity (see Step 7 in Section 4) between the ellipsoid and the disk is 0.027153, and the average between the ellipsoid and the arrow is 0.047667. Hence, we conclude that the Ellipsoid match the Disk better than the Arrow.

5. If there is no noise, the $\bar{M}$ values of the disk and ellipsoid must be 1.

6 Conclusion

In this paper, a parallel object recognition algorithm was proposed within the Spiral Architecture. A uniform partitioning method for parallel algorithm was applied within the Spiral Architecture based on 'Spiral Multiplication'. This proposed partitioning scheme guarantees well load balancing and fast approach.

This is implemented on a cluster of Sun-Workstations. The proposed algorithm greatly increases the computation speed. The time for the object matching using the algorithm presented in this paper requires less than two minutes to complete in contrast with about three hours using sequential approach.

References

[1] K. Arbter, W.E. Snyder, H. Burkhardt, and G. Hirzinger, *Application of affine-invariant Fourier descriptors to recognition of 3D objects*, IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990), 640–647.

[2] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf, *Parametric correspondence and chamfer matching: Two new techniques for image matching*, Proc. 5th Int. Joint Conf. Artificial Intelligence (Cambridge, MA), 1977, pp. 659–663.

[3] G. Borgefors, *Hierarchical chamfer matching: a parametric edge matching algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence **10** (1988), 849–865.

[4] D.L. Borges and R.B. Fisher, *Class-based recognition of 3D objects represented by columetric primitives*, Image and Vision Computing **15** (1997), 655–664.

[5] Xiangjian He and Tom Hintz, *Object contour extraction in spiral space*, Australian Computer Science Communications **21** (1999), 182–192.

[6] ———, *Refining occluded object recognition within the spiral architecture*, 2000 International Conference on Image Science, Systems and Applications (Las Vegas), 2000, pp. 701–708.

[7] Xiangjian He, Tom Hintz, and Ury Szewcow, *Affine integral invariants and object recognition*, Proc. 3rd High Performance Computing Asia Conference and Exhibition (Singapore), 1998, pp. 419–423.

[8] ———, *Parallel algorithm on edge detection with spiral architecture*, Proc. 3rd High Performance Computing Asia Conference and Exhibition (Singapore), 1998, pp. 460–467.

[9] ———, *Object recognition by an integral invariant within spiral architecture*, Proc. 4th Asian Conference on Computer Vision (Taipei, Taiwan), 2000, pp. 294–299.

[10] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidke, *Comparing images using the Hausdorff distance*, IEEE Transactions on Pattern Analysis and Machine Intelligence **15** (1993), 850–863.

[11] J.E.Boyd and J.J.Little, *Global versus structured interpretation of motion: Moving light displays*, Proc. IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated objects (Puerto Rico), 1997, pp. 18–25.

[12] J. Sato and R. Cipolla, *Affine integral invariants for extracting symmetry axes*, Image and Vision Computing **15** (1997), 627–635.

[13] P. Sheridan, T. Hintz, and D. Alexander, *Pseudo-invariant image transformations on a hexagonal lattice*, Image and Vision Computing **18** (2000), 907–917.

[14] Phil Sheridan, *Sprial architecture for machine vision*, Ph.D. thesis, University of Technology, Sydney, 1996.

[15] Q.M. Tieng and W.W. Boles, *Complex Daubechies wavelet based affine invariant representation for object recognition*, Proc. IEEE Inter. Conf. Image Processing (Austin, Texas), 1994, pp. 198–202.

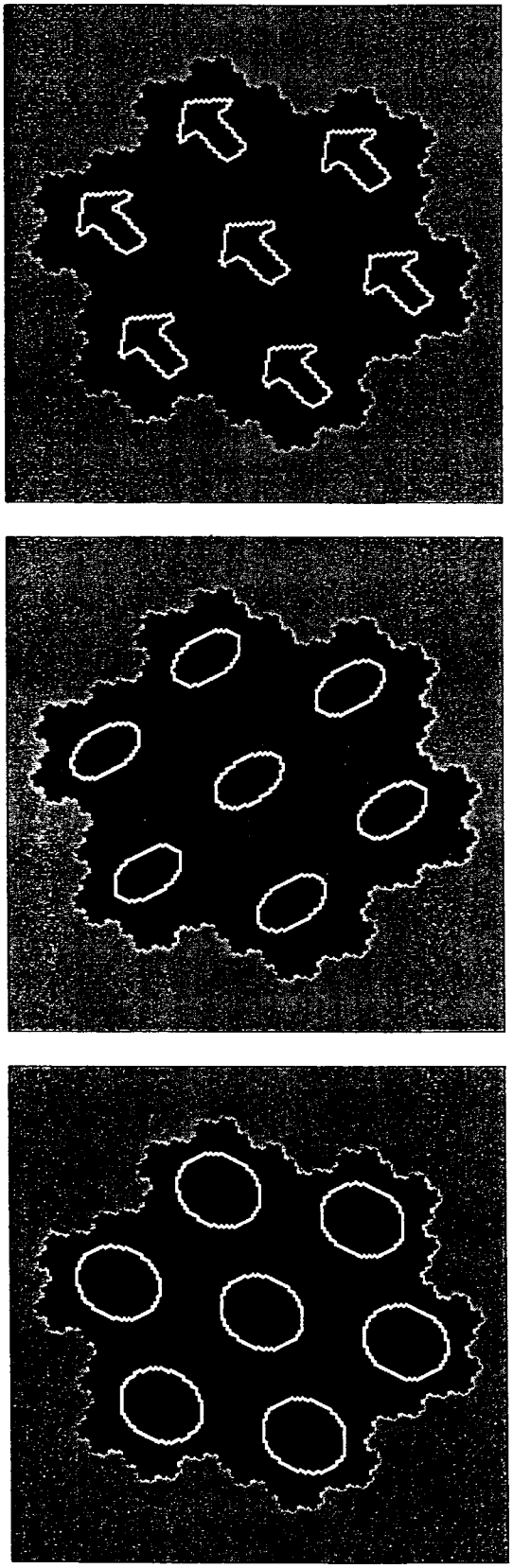

Figure 7: (upper) contour of the arrow, (middle) contour of the ellipse and (lower contour of the circle.