A DIVIDE-AND-CONQUER APPROACH TO CONTOUR EXTRACTION AND INVARIANT FEATURES ANALYSIS IN SPATIAL IMAGE PROCESSING

Marina Gavrilova and Russel Apu

Department of Computer Science, University of Calgary
2500 University Drive NW, Calgary, AB +1 (403) 220-5105

Corresponding Author's E-mail: marina@cpsc.ucalgary.ca

Abstract— This paper presents a novel divide-and-conquer method to analyze spatial information, such as geometric shapes, contours and trajectories extracted as a discrete sequence of points (or pixels) from images or spatial sensors, including GPS or transponders. The method extracts contour point sequences and then uses a scale invariant analysis to extract invariant arc features. The arc feature is a generalization of scale invariant corners used in many object recognition and image matching methods. The method considers detection of corner like features in the presence of slow curvature and sharp noise and discretization (spatial quantization), typical for images obtained by aerial photography, digital map scanning or other GIS image acquisition techniques. The resulting feature vectors can be used for stable and robust object feature analysis and object detection. The developed method is found to be capable of ignoring local sharp noise and detecting globally prevailing sharp features. Experimental analysis confirms the efficiency and robustness of this method using several difficult shapes with considerable noise and ambiguity. The method allows not only stable feature detection but also general shape analysis such as convexity, linearity and curvature.

1. INTRODUCTION

Analysis of shapes and object boundaries for object recognition and classification is an important problem attracting more and more attention in Spatial Information Community (Canosa 2006). It is becoming increasingly important for GIS community to analyze spatial information, such as geometric shapes, contours and trajectories and derive information about spatial objects (such as their shape, contour, curvature etc). One of the ways to achieve this is to extract a discrete sequence of points (or pixels) from images or spatial sensors, including GPS or transponders, and to process these pixels further to identify object boundary. In the recent years, researchers started to utilize the unique footprint manifested by edges and invariant features (i.e. corners) for object recognition (Canosa 2006, Forsen and Moe 2006, Lowe 2004). These recognition methods are often applied to analyze the unique signature inherent due to the shape and color of objects which are spatially stable and scale-wise invariant (Lowe 2004). The analysis of such shapes and trajectories is very important in GIS. Analysis of GPS coordinates or transponder signals to estimate the navigation pattern of ships and vessels near the coastal region is a practical example where sophisticated shape analysis methods could be applied. Recognition of certain shapes or trajectories is crucial to such problems.

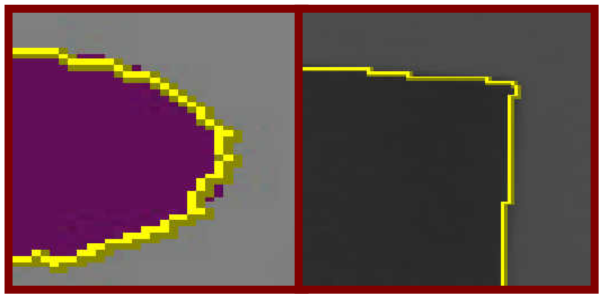

Figure 1. Noise and quantization (right) error in boundary extraction masks the local curvature

Regardless of whether shapes (or object boundaries) are extracted from images, a common artifact of the extraction process is the accumulation of noise and quantization patterns (Fig. 1) (Apu and Gavrilova 2007). It is often difficult to eliminate these noises without loss of detail. In shapes, the loss of detail is most profound around globally prevailing corners, which are important object features (Lowe 1999, 2004). Several methods have been proposed to deal with such problems. Popular approaches includes Kalman filter, Weiner filter, Fourier low pass filter, N-Cuts, Snakes (Active Contours) etc (Blum and Nagel, 1978, Denzler and Niemann 1995, Ferrie *et al*. 1993, Mardia and Jupp 2000). Some of these methods are computationally extensive and causes smoothing of corners (Canny 1986, Canosa 2006). Others have erratic convergence behavior (i.e. snakes) or require user interaction to setup key points and parameters for every shape (Aguado *et al*. 2000, Denzler and Niemann 1995, Ferrie *et al*. 1993). A new method to analyze shape curvature while preserving sharp discontinuities called Circular Augmented Rotational Trajectory (CART) was first proposed by Apu and Gavrilova (2007). The CART transformation presented prior to this paper was fast, robust and presented a number of opportunities in shape analysis. For example, CART representation of a shape can be used to test for linearity, discontinuity, curvature, convexity and other shape signatures (Apu and Gavrilova 2007).

In this paper, we present a new method based on computationally efficient divide-and-conquer technique that allows to extracts contour point sequences from images and then uses a scale invariant analysis to extract invariant arc features and significantly improves standard CART performance and applicability.

The rest of the paper is organized as follows. First we present a novel Divide-and-Conquer algorithm DQ-CART that is order of magnitude faster than previously presented algorithms. In our experimentation, the recorded speedup was between 10 times and 100 times depending on the complexity of the shape. The runtime of the new algorithm is highly adaptive and depends significantly on the complexity of the shape (simple shapes with low curvature converge much faster). Experimentation revealed that the new DQ-CART algorithm produces a more accurate Rotational Trajectory and corresponding R-Space representation. The new algorithm also enables the method to be efficiently applied to to a large number of shapes in real-time. We next present a new technique that allows the conversion of an R-Space shape representation to feature vectors called invariant arcs. Invariant arcs are generalization of invariant feature points as presented by Lowe (2004), Harris and Stephens (1988), Moreton and Sequin (1993) and others. This serves as a connection between other state of the art recognition methods such as Ada-Boost, Cascade etc. to DQ-CART representation (Canosa 2006). Therefore, the ideas presented in this paper are pivotal to the integration of the curvature analysis method with methods for object recognition, feature analysis and feature classification (using clustering of classifiers based on feature vectors).

2. LITERATURE SURVEY

In the area of Image Processing, the task of object boundary analysis is considered to be an important problem, which continues to expand and develop at present (Aguado *et al*. 2000, Canny 1986, Canosa 2006, Denzler and Niemann 1995, Ferrie *et al*. 1993). Early attempts show preferences towards edges and contours which soon turned out to be an insufficient representation for robust object recognition and analysis (Canny 1986, Canosa 2006). Application of methods such as the Canny Edge detector is often limited and hard to apply to natural images (or scenes) (Canosa 2006). Background Clutter, color ambiguity, intensity variation and other artifacts causes edge-detectors to fail. A common approach often found in machine vision utilizes Hough Transform (HT) for detection of certain geometric patterns (i.e. straight lines) to identify landmarks and man made objects (Aguado *et al*. 2000). A Generalized Hough Transform (GHT) can analyze arbitrary shape patterns but often found to be highly sensitive to noises. More sophisticated methods exist such as the Active Contours and snakes that use energy minimization technique to converge an object boundary (Blum and Nagel 1978, Denzler and Niemann 1995, Ferrie *et al.* 1993). In reality these methods are often not robust enough for stable and reliable object recognition and researchers continue to seek more sophisticated method that consider the multimodality, clutter and noise often present in natural images.

A significant advancement has been made recently in the development of methods relying on invariant object features (Harris and Stephens 1988). Invariant features are spatially and scale-wise robustly identifiable footprints of objects and shapes in images that contributes to the unique identification or recognition of an objects. Examples of invariant features in present literature include minutiae in fingerprint, Harris corner, Scale Invariant Feature Transform (SIFT), MSER etc (Canosa 2006, Forssen and Moe 2006, Lowe 1999, 2004, Moreton and Sequin 1993). A common approach at present is to identify such features from images and present them as feature vectors (Lowe 1999). A good number of techniques exist in pattern recognition that uses feature vectors as classifiers for robust object recognition (Canosa 2006). Weak classifiers such as the Ada-Boost and Haar Cascade are examples of such methods (Lowe 1999). The main difficulty of these methods lies in the robust detection of such features in the presence of clutter, occlusion, noise and other image artifacts. The Harris corner detector for example, often fails to detect obvious corners in natural images (Harris and Stepfens 1988). The SIFT method has proven to be a very effective technique that expands the idea of corners and considers a histogram based feature vector which are invariant and robust in the scale space (Lowe 2004). Scale invariant features have been proven to be effective and most useful for object recognition (Lowe 1999). However, methods such as SIFT are often found insufficient for the analysis of geometric shapes in which these features are ambiguously detected in different but similar shapes (Forssen and Moe 2006). For example, SIFT or Harris corner cannot be used efficiently to discriminate a hexagonal object from a pentagonal object. Forssen presented a method called MSER that addresses this limitation by utilizing region based features (Forssen and Moe 2006). These limitations are particularly amplified when considering applications such as street sign recognition in natural setting where occlusion, shadow, intensity variation, lighting variation and noise are present. Although invariant corners and features can be robustly detected using current methods, some crucial geometric information such as linearity, convexity, curvature etc. are missing from these representations.

The Circular Augmented Rotational Trajectory (CART) method was originally conceived to address this problem and was shown to be highly effective for such analysis (Apu and Gavrilova 2007). A comprehensive description of CART is discussed in (Apu and Gavrilova 2007). In short, the CART method converts a sequence of points into a rotation invariant R-Space representation. IN this paper, we present a new method to convert invariant R-Space representation into feature vectors called invariant arcs, which are directly compatible to be used as weak classifiers. Powered by a highly efficient divide and conquer algorithm, the DQ-CART method becomes one of the most efficient and effective shape analysis tool.

3. METHOD DESCRIPTION

The transformation takes as input a sequence of n points (in $\mathbb{R}^2$) $x = x_1, x_2, \ldots, x_n$. A point sequence x either represents a contour/shape or a trajectory (i.e. GPS). The extraction of such contour points can be achieved in several ways. Several image segmentation algorithms (i.e. meanshift) can be applied to obtain region boundaries. An efficient working algorithm to extract contour points from simple images is called LR-Traversing (Apu and Gavrilova 2006). This method is fast enough for real-time application but does not work for textured images. One solution is to add a texture analysis method to preprocess the image. More reliable and robust methods are to extract contour points are currently under development.

The point sequence x is transformed into a representation called R-Space. A detailed description of the R-Space and its application is discussed by Apu and Gavrilova (2007). In this paper, we first provide a quick overview of R-Space, followed by the new DQ-CART algorithm and the invariant arc transform.

3.1. R-Space Shape representation and CART

The original CART method converts a sequence of points into R-Space which is a rotation invariant shape representation. The main idea is to estimate the local curvature at each point as the turn-rate due to a quadratic/circular arc. This process is continued along the shape estimating the turn-rate at each step and finally augmenting these circular arcs as an approximation of the shape. The result is a function $R_\lambda^\tau : \left[\mathbb{R}^2, \mathbb{R}^2, \ldots, \mathbb{R}^2\right] \rightarrow [\mathbb{R}, \mathbb{R}, \ldots, \mathbb{R}]$, that creates with a predefined arc-length λ and a tolerance τ. The tolerance τ is defined such that the curve resulting from the R-Space representation must not deviate from the original curve by more than τ. That is:

$$\forall p \exists q \left(\left(\left(p \in R_\lambda^\tau (x) \right) \wedge (q \in x) \right) \rightarrow |p - q| \leq \tau \right) \qquad (1)$$

The goal of the CART algorithm is to minimize the turn-rates while maintaining the constraint in (1). In essence, the method requires the estimation of the signed curvature per unit length $\omega_i = \nabla \omega(x_i)$ such that:

$$\nabla \omega(x, i) = \mathrm{CCW}\left(x_i, x_{i-1}, x_{i+1}\right) \times \theta(x, i) \qquad (2)$$

Here CCW is defined as followed:

$$\mathrm{CCW}(a,b,c) = \begin{cases} -1; & \text{if a,b and c are clockwise} \\ 0; & \text{if a, b and c are co-linear} \\ +1; & \text{if a,b and c are counter clockwise} \end{cases}$$

$\theta(x, i)$ is the local curvature based on quadratic approximation of the turn-rate (degrees per unit length of arc). The estimation of θ can be as simple as differentiating the curve. However, the approach would generate an R-Space representation that is highly sensitive to noise.

A robust approximation algorithm was presented by Apu and Gavrilova (2007). The algorithm was based on an approximation of the curvature by projecting various curves toward the shape from the current point x_i and selecting the path that allows traversing of maximum arc-length without violating the constraint in (1). Special care was taken to handle discontinuity by a novel backtracking mechanism. Although the method performs well, it is far from being the most efficient. The search for the best fitting curve (and corresponding turn-rate) requires projecting many curves for each step of the process each of which must be scanned to validate the constraint in (1). The Divide and Conquer CART algorithm (DQ-CART) presented in the following subsection is much more efficient and robust. It is also simpler to implement compared to the old method and requires less number of intrinsic parameters and special cases.

3.2. Curvature Estimation and Local Ambiguities

In this subsection, we establish the need for a sophisticated algorithm to estimate $\theta(x,i)$. Let us assume that the shape was extracted from an image by tracking border pixels. The resulting contour is quantized to the nearest integer pixel coordinate. Since there are only four directions (no diagonal) to travel from one pixel to its neighbor the resulting turn rate is approximately an integer multiple of 90 degrees (Fig. 1). These large magnitudes of pseudo-turns will mask any low curvature arcs (even straight lines). Unfortunately, shapes extracted from images (or GPS) is saturated by this discretized pattern which is almost impossible to separate from actual curvature by looking at a local subrange of the shape. For example, the shapes in Fig. 2 pose such a problem in the marked regions. It is apparent that there are several possible interpretations to these point sequences and even if the two local point sequences are identical (or similar) their interpretation may be dramatically different depending on the global shape context. The right hand side of Fig. 2 also shows several other possible interpretations to these point sequences which are equally viable. Therefore, it is often not possible to pick the right turn-rate without looking at the global shape.

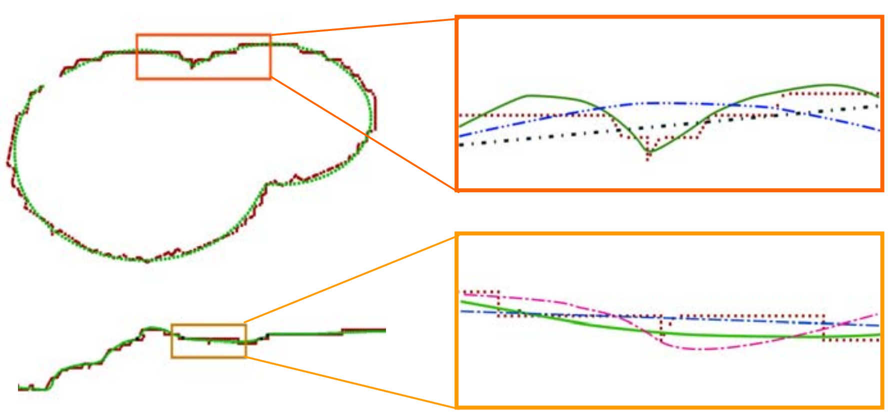

Figure 2. An example of ambiguity in local curvature. The two marked region comes from two different shapes that has similar configuration, but interpreted differently by CART due to its global context. The right magnifications show additional possible interpretation of the region.

3.3. The DQ-CART Algorithm

Surprisingly, the divide and conquer CART algorithm is based on a simple idea. We take a shape by its two end points x_0 and x_{n-1} and consider a base connecting these two end points (Fig. 3). In the case that the shape may be a loop ($x_0 \approx x_{n-1}$), we can initially divide the shape into two by taking the midpoint. For every segment $\left[x_i..x_j\right]$, we find a point x_k where $i \le k \le j$ such that x_k is the highest peak with respect to the line segment $\overline{x_i x_j}$ (maximum distance from the line segment). If an arc through the three points x_i, x_k, x_j satisfies the constraint in (1), then a stepping proportional to the arc-length is taken (adaptive stepping). However, a minimization algorithm is first applied to relax the curvature of this arc such that the constraint (1) is not broken (optimization). However, if the maximal arc breaks the threshold τ , the curve is split by pushing this point x_k into a stack and processing the segment $\left[x_i..x_k\right]$ (Fig. 3). At the beginning of each iteration, the right endpoint is retrieved from the stack and examined. The left (from array index point of view) endpoint is traversed and advanced accordingly.

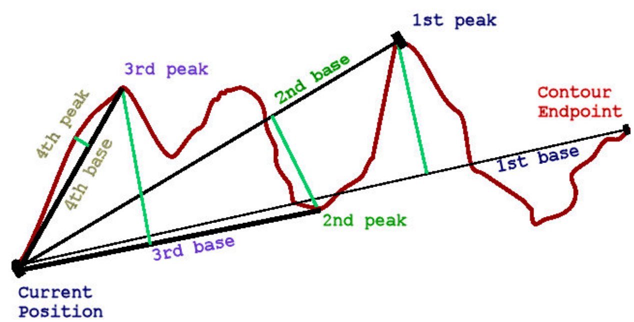

Figure 3. The progressive division of the shape (divide and conquer). For each base a peak is computed. The algorithms prepares to step along the 4th base since the 4th peak is small enough to allow a valid arc.

For ease of understanding we break the algorithm into methods in a top-down manner. Due to space constraint some trivial methods are only described.

3.3.1. System assumptions and parameters

There are a few global variables that retains the original shape and the CART representation. The array Path[] of size n contains the n original contour points. The arrays cPos[], cDir[], cTurn[] and cNear[] contains the position, tangent, turn-rate and nearest point (index of the nearest point in Path[]) respectively that represents the R-Space of the shape. The size of these arrays is indicated by m. The method uses a number of constants, all of which is related to the precision of the algorithm. No parameters require adjusting while scanning different shapes. The following parameters presented in Table 1 are defined to process images (and pixel coordinates).

Table 1. System Constants for DQ-CART

Constant	Description	**Default**
Granularity, λ	Length of an unit arc step	1.0
Tolerance, τ	Maximum distance from original curve	1.75
PATH_RATIO	Adaptive stepping ratio	40%
MIN_LOOP	Precision of the minimization loop, a value of allows 1/2^10 resolution	10
DISC_STEP	Backtracking steps when discontinuity is detected	6

3.3.2. The DQ-CART Transformation

The main DQ-CART algorithm utilizes a stack to subsequently divide the curve and expand the R-Space contour along the original contour without violating constraint (1). The algorithm maintains a partially complete R-Space array set, left and right index, index of the maximal peak maxi, current position pos, current direction dir and index to the nearest point. Readers should pay attention to some details pertaining to border conditions in order to prevent infinite loops in some cases. The algorithm is somewhat tricky due to these border cases (i.e. what if maxi==right), but once they are handled properly the algorithm works perfectly.

The method starts with initialization, and introduction of the subdivision stack that stores curve indices. Next, the main loop proceeds in traversing the curvature until the end of the contour is reached. First, the right index is retrieved from the stack. The current extrema is computed. A possible discontinuity due to the steep turn-rate is handled in the process (note that definition of a steep turn-rate is discussed later). If the calculated arc appears to satisfy specified constraint (1), a proportional stepping is performed.

Otherwise, the maximal peak is pushed into the stack the entire process is repeated.

3.3.3 Discontinuity Handling

Readers should be aware that the CART algorithm uses a tolerance τ that limits the maximum turn-rate (curvature) of a standard stepping. If this condition is violated, the algorithm may take a turn so steep that it forms a complete cycle without breaking the tolerance condition. Thus, the method may enter an infinite loop! To prevent this from happening we must limit the maximum allowable turn-rate $t_{\max}$. A good (and safe) constraint is given below:

$$t_{\max} = {}^{270}\!/_{\pi\tau} \qquad (3)$$

The detection mechanism utilized in the algorithm is not intuitive, but the most efficient way. When a discontinuity is present, the effective counter-measure is handled by Resolve_Discontinuity() method.

3.3.4. Optimizing the turn-rate

The optimization of the turn-rate is a process of relaxation that finds the minimum turn-rate that satisfies the constraint (1). The method starts with a tolerance passing arc and tries to converge to a less steep arc. As part of processing, we assert the latest turnrate into R-Space state as Valid_Path method also inserts R-Space points into the array set.

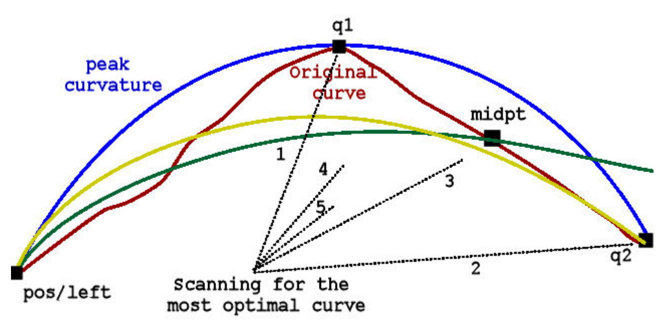

Figure 4. The subsequent turn-rate optimization. A Binary search is performed along the range [mid,right]→[q1..q2] and the best matching curve is returned.

3.3.5 Computing Subsequent Optimal Turn-Rate

The final step of the processing is an optimal turn-rate computation. This method employs a binary search for the most optimal curvature (Fig. 4). There are several special cases that must be considered, omitted here for briefness. Last but not the least, some constants in addition to the ones listed in Table 1 is used deep in the code which are all related to precision. One example would be a neighborhood span constant specifying the number of points scanned to find the nearest point pairs (utilized in Constant_Turn() method).

3.4. The Invariant Arc Analysis

Once the R-Space representation has been computed, the resulting array cTurn[] can be used to compute the scale invariant arcs. The scale invariant arcs are stable subrange of the R-Space (corner like features) that is spatially stable. Since method deals with arbitrary curvature the interaction of left hand side and right hand side arcs causes the feature point to shift spatially when moving along the scale space. In short, not all regions of a shape is stable.

We present a very easy mechanism to find these stable arcs that is tested to be robust and accurate. Intuitively, we apply a Haar Wavelet like feature called the radial integral R-Space $\tilde{\omega}$:

$$\tilde{\omega}_i^r = \sum_{j=i-r}^{i+r} \nabla\omega(x, j) \tag{4}$$

For efficient processing the cumulative sum of $\nabla\omega$ is precomputed and requires constant operation for each $\tilde{\omega}$ evaluation. A scale influence function μ^r is computed as followed:

$$\mu_i^r = \prod_{k=1}^{r} \tilde{\omega}_i^k \big/ k \tag{5}$$

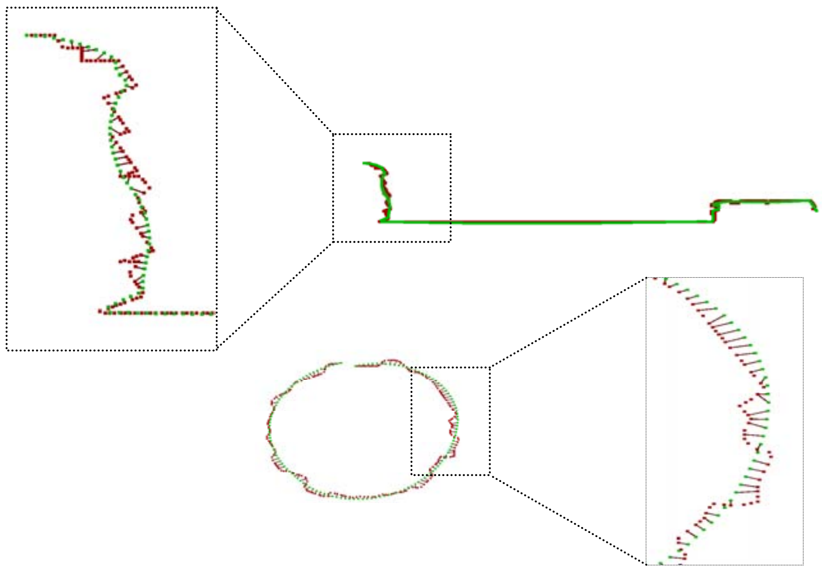

Figure 5. Using the CART method to extract global shapes

The peaks (local maxima) in the resulting function define the presence of these invariant arc features. Each peak is expanded to find the bell shaped regions. This is the span of the arc feature. To find the exact location of the feature an algorithm similar to Comp_Extrema() is applied. The Feature Vector is identified at this peak. The direction of the vector is towards the center of the arc. The length of the vector is equal to the sharpness ρ:

$$\rho = \frac{\text{Peak turn-rate}}{\text{total turn}} \qquad (6)$$

The feature vector can be extended further to include additional information such as arc length, color regions, total turn, turn strength, scale influence etc.

4. EXPERIMENTAL RESULTS AND ANALYSIS

The new method was tested against a number difficult shapes used to analyze the previous algorithm. The convergence behavior was mostly identical Fig 5). A noticeable improvement in the R-Space resolution (especially near steep curves) is observed in the new DQ-CART. The most dramatic improvement was observed in recorded runtime. The range of speedup was observed between 10 times to up to 100 times. As the complexity and length of the shape is essential for large aerial photo processing or digitized maps, the importance of increase in run-time becomes much more profound. For comparison, a summary of few test cases is provided in Table 2.

Table 2. System Constants for DQ-CART

Shape #	CART (MS)	DQ-CART (MS)	Speedup
1	94.7	9.2	10.3
2	61.4	4.0	15.4
3	692.6	25.9	26.7
4	501.2	11.2	44.8
5	213.9	8.1	26.4
6	107.1	5.8	18.5
7	98.4	9.2	10.7
8	97.6	4.4	22.2
9	37.5	3.1	12.1

A good number of tests have been performed with the invariant arc analysis. To test the repeatability and robustness we carried out an experiment with 5 test shapes. Each test shape was manipulated to generate a number of rotated and scaled (also reduced/increased number of contour points) shapes. Because the process was done manually, we limited the number of transforms to 10 (thus a total of 30). The master shape is used as a template and the feature vectors are recorded. The feature vectors from each subsequent template are then matched with the master template to compute a robustness score. A 100% score means that all features were matched across all transformations. In most cases, the number of features that are missed were weak features that are barely detectable. As a result, their presence was intermittent in different scaling and rotation of the shapes. A weighted scoring (weighted according to scale influence μ^r) would reveal a much better picture since during the experiment the stable corners were never missed.

5. Conclusion

In this paper we presented a novel divide and conquer algorithm to compute the representation of shapes and offered a novel technique to transform this representation to feature vectors. The method allows to extract shape contour even in the presence of noise and sharp features, and provides a convenient representation for invariant arcs. The experimentation supports the claims that the new method is significantly faster than the one presented in earlier literature. The runtime analysis indicates that the new method can be applied to a large number of shapes off-line or in real-time, thus opening a large number of possible applications. The conversion to robust invariant arc allows pattern recognition using state of the art weak classifiers such as a Ada-Boost and Cascades. Further directions of research will involve extensive case study of method performance with variety of spatial images, and concentrate on the invariance analysis and robust feature vectors.

ACKNOWLEDGMENTS

Authors would like to thank NSERC and GEOIDE agencies for their support of this project.

REFERENCES.

1. Aguado, A. S., Montiel, E., and Nixon, M. S. Bias Error Analysis of the Generalized Hough Transform. J. Math. Imaging Vis. 12, 1, pp. 25-42, 2000.
2. Apu, R. A. and Gavrilova, M. "CART: Shape recognition and curvature estimation using circular augmented rotational trajectory", 3IA 2007, Ninth International Conference on Computer Graphics and Artificial Intelligence, Athens (GREECE), pp. 75-85, 2007.
3. Apu, R. A. and Gavrilova, M. L. "Flexible Tracking of Object Contours Using LR-Traversing Algorithm". CGIV 2006, IEEE-CS, Computer Graphics, Imaging and Visualization: Techniques and Applications, Australia, pp. 503-513, 2006.
4. Blum, H. and Nagel, R. N. Shape description using weighted symmetric axis features. Pattern Recognition, 10:167-180, 1978.
5. Canny, J. A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, pp. 679-698, Nov 1986.
6. Canosa, R. L. Image understanding as a second course in AI: preparing students for research. 37th SIGCSE Technical Symposium on Computer Science Education (Houston, Texas, USA, SIGCSE '06. ACM Press, New York, NY, pp. 543-547, 2006.
7. Denzler, J. and Niemann, H. Evaluating the performance of active contour models for real-- time object tracking. 2nd Asian Conference on Computer Vision, pages II/341-II/345, Singapore, 1995.
8. Ferrie, F. P., Lagarde, J. and Whaite, P. Darboux frames, snakes, and super-quadrics: Geometry from the bottom up. IEEE Trans. on Pattern Analysis and Machine Intelligence, 15(8):771-784, August 1993.
9. Forssen, P. and Moe, A. 2006. Autonomous Learning of Object Appearances using Colour Contour Frames. 3rd Canadian Conference on Computer and Robot Vision (Crv'06) IEEE Computer Society, Washington, DC, vol. 00, p.3, 2006.
10. Harris, C. and Stephens, M. A Combined Corner and Edge Detector. Alvey Vision Conf., Univ. Manchester, pp. 147-151, 1988.
11. Khabou, M. A., Hermi, L., and Rhouma, M. B. Shape recognition using eigenvalues of the Dirichlet Laplacian. Pattern Recogn. 40, 1, pp. 141-153, 2007.
12. Lowe, D. G. Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.
13. Lowe, D. G. Object recognition from local scale-invariant features, International Conference on Computer Vision, Corfu, Greece, pp. 1150-1157, 1999.
14. Mardia, K. V. and Jupp, P. E. Directional Statistics, Wiley Series in Probability and Statistics, John Wiley & Sons Inc. NY, USA, 2000.
15. Moreton, H. P. and Séquin, C. H. Scale-Invariant Minimum-Cost Curves: Fair and Robust Design Implements Computer Graphics Forum 12 (3), 473–484, 1993.