Pace University
DigitalCommons@Pace

CSIS Technical Reports

Ivan G. Seidenberg School of Computer Science and Information Systems

7-1-2006

Analyzing Shape Context Using the Hamiltonian Cycle

Carl E. Abrams
Pace University

Charles C. Tappert
Pace University

Sung-Hyuk Cha
Pace University

SCHOOL OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

TECHNICAL REPORT

Number 228, July 2006

Analyzing Shape Context Using the Hamiltonian Cycle

Carl E. Abrams
Sung-Hyuk Cha
Charles C. Tappert

PACE UNIVERSITY
IVAN G. SEIDENBERG
SCHOOL OF COMPUTER SCIENCE
AND INFORMATION SYSTEMS

Carl E. Abrams holds a B.S. in Chemistry from the University of Connecticut, an M.S. in Computer Science from Stevens Institute of Technology, and a newly minted Doctorate in Computing from Pace University's D.P.S. Program. His career has taken him from chemist at Lederle Laboratories, to the position of Chief Information Officer for the Trading and Treasury Department at Chase, to the Director of Applications Development for the Swiss Bank in New York, to his current position at IBM Research in the T.J. Watson Research Center where he is Financial Services Sector Business Executive.

Sung-Hyuk Cha is Assistant Professor of Computer Science at Pace University. Dr. Cha holds baccalaureate and masters degrees from Rutgers University and a doctorate in Computer Science from the State University of New York at Buffalo.

Dr. Cha's research interests are in the areas of distance measure and pattern matching algorithms, pattern recognition, image analysis, and machine intelligence and data mining.

Charles C. Tappert is Professor of Computer Science and the Associate Program Chairperson for Pace University's Doctorate of Professional Studies in Computing. Before coming on-board to participate in the development of the D.P.S. program, Professor Tappert had served on the full-time faculty at the U.S. Military Academy at West Point for seven years. For the 26 years prior to that, Professor Tappert worked at IBM, mostly at the T.J. Watson Research Center, on speech recognition and pen computing. Counting journal articles, book chapters, conference papers, patents, and technical disclosures; Professor Tappert has well over a hundred publications. While at IBM he adjuncted at Pace, SUNY Purchase, and North Carolina State University.

Professor Tappert holds a B.S. in Engineering Sciences from Swarthmore College, an M.S. and Ph.D. in Electrical Engineering from Cornell University, and was a Fulbright Scholar. His research interests include pattern recognition, pen computing and voice applications, graphics, algorithms, artificial intelligence, human-computer interaction, and e-commerce.

Analyzing Shape Context Using the Hamiltonian Cycle

Carl E. Abrams, Sung-Hyuk Cha, and Charles C. Tappert

Computer Science Department, Pace University
861 Bedford Road, Pleasantville, New York, 10570 USA
carl.abrams@us.ibm.com, {scha, ctappert}@pace.edu

Abstract. Shape matching plays important roles in many fields such as object recognition, image retrieval etc. Belongie, et al. recently proposed a novel shape matching algorithm utilizing the *shape context* as a shape descriptor and the magnitude of the aligning two shape contexts as a distance measure. It was claimed to be an information rich descriptor that is invariant to translation, scale, and rotation. We examine the limitation of the algorithm using graph theory and present several geometrically different shapes that are considered identical by the shape context algorithm. Theoretical contributions pertain to linking shape context and the *Hamiltonian cycle*.

1 Introduction

Shape matching which is often used in the field of object recognition [7, 16] is usually addressed in two steps: shape representation (descriptors) and shape similarity. Shape descriptors can generally be categorized as either external or internal descriptors [11]. Internal descriptors abstract the shape using the points on the shape boundary, whereas external descriptors use the boundary points to create the shape abstraction. Numerous distance or similarity measures can be found in literature [4, 5, 9].

Belongie, et al. recently proposed a novel shape matching algorithm utilizing the shape context as a shape descriptor and the magnitude of the aligning two shape contexts as a distance measure [1-3]. The notion of a shape context is to represent the relationship of each point on the boundary of a shape to all the other boundary points and then convert that representation to polar histograms. With each point represented as a polar histogram, matching one shape to another is simplified to comparing histograms and finding the best overall fit as the least cost assignment of points of one shape to the points of the other. The matching of the points is a form of the classic *bi-partite* graph-matching assignment problem which can be efficiently solved by the *Hungarian* method [6, 10]. We shall refer to this algorithm as the *SC algorithm*.

Every comparison-based shape-matching algorithm is subject to special exceptional cases where two shapes look similar to a human observer while the shape matching algorithm suggests otherwise [9, 14]. A well-known case involves the shapes of the numerals '3', '8', and '*8*'. When these shapes are represented by pixel values only, a simple *Hamming* distance suggests that the distance between '3' and '8' is smaller than that between '8' and '*8*' [9, 14]. Finding and understanding these special exceptional cases can be the key to discovering a better shape matching method. The *SC algorithm* is not an exception. Here, we present several geometrically different shapes that are considered identical by the shape context algorithm. Utilizing the *Hamiltonian cycle* from graph theory, we generated numerous exceptional cases for the *SC algorithm*.

The rest of the paper is organized as follows. In section 2 we review the shape context and shape matching method used in the *SC algorithm*. Section 3 presents several exceptional shapes and theoretical results based on graph theory. Section 4 concludes this work.

2 Shape Context Algorithm

In this section, we summarize the *SC algorithm* developed by Belongie, et al. [1-3]. There are two steps in the algorithm: shape representation and shape matching.

2.1 Shape Representation

The shape of an image is represented by a shape context which we denote as S. Inputs for this step include an image I containing a single shape and n, the number of points that define the boundary. We assume that the shapes are perfectly segmented and the boundary edge has been detected. The n points are placed on the boundary of the shape such that the boundary distances between adjacent points are the same and are ordered $(p_1, p_2, \dots p_n)$ where p_1 is the starting point. Let the distance between two adjacent points be d. Figure 1 (b) shows the boundary of a shape and six points are placed.

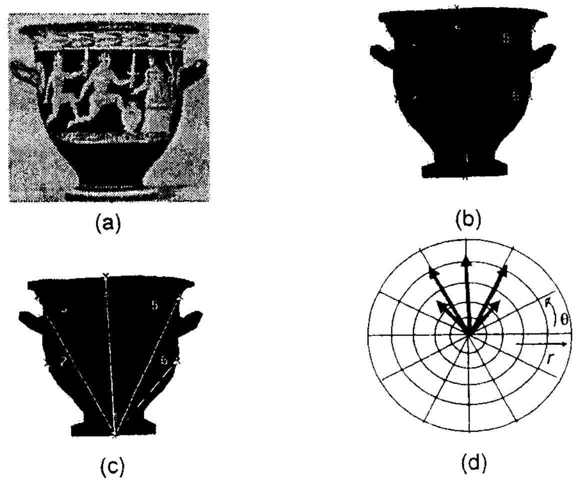


Fig. 1. The Shape and shape context point.

Next, vectors from each point on the boundary to every other point on the boundary are drawn. For example, p_1 shown in Figure 1 (c) has five vectors to all of the remaining points on the boundary. The same procedure is applied for the remaining *n-1* points as shown in Figure 2.

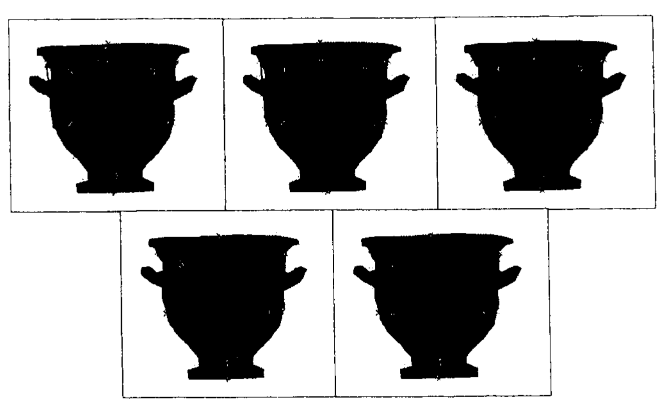


Fig. 2. Shape Context Vectors for Pottery Shape

The set of n sets of vectors for each point represent the shape context for the boundary. The vectors are then transformed into a form that is useful for the computation of the comparison of shapes: compact, comparable and computable. The transformation is accomplished by converting the vectors at each point into a polar histogram. A representative polar histogram is shown in Figure 1 (d). That is, since a vector is defined by its angle with respect to a reference point and its distance, quantizing both dimensions will yield a set of six histograms, one at each point. The bins spacing selected for the histogram is 30 degree increments for the angles yielding 12 angle bins. The radius r is divided into 5 increments. The log r is used to make the shape histogram more sensitive to points closer in to the reference point. The total number of bins in a polar histogram is 60.

Rotational invariance is achieved by the proper selection of the reference point for the vector angles. By using the turning angle at a point as the reference angle from which the angles to all the other points on the shape are compared, shape contexts that are insensitive to rotational changes in the shape are created.

Building polar histograms at all n points results in a compact representation of the boundary as a set of n shape-context histograms. The shape, S, is represented by n polar histograms.

$$S = \bigcup_{i=p_1 \cdots p_n} h(i) \qquad (1)$$

2.2 Shape Matching

Shape matching step takes two shapes, S and S' represented by n polar histograms as inputs. The output is the scalar distance value indicating the similarity between two shapes. The lower the value, the more similar the two shapes. The quantitative comparison of the two shapes is easily obtained using the simple χ^2 histogram comparison as shown in eqn (2)

$$C_{i,j} = 1/2\sum_{k=1}^{b}[h(k) - h'(k)]^2 /(h(k) + h'(k)) \text{ where } i, j = p_1 \cdots p_n \qquad (2)$$

$C_{i,j}$ is the cost of the p_i point on the shape S and the p_j point on the other shape S'. b is the number of bins in the polar histogram.

The resulting cost matrix derived from Equation 2 is an $n \times n$ cost matrix. The comparison of the two shapes is now performed by matching a point on the first shape to one and only one point on the second shape such that the sum of the costs of all points so matched is minimized. The matching of the points in this manner is a form of the classic bi-partite graph matching problem known as the "Assignment Problem". The assignment problem is solved by the Hungarian method [6, 10] which runs in $O(n^3)$ time complexity. The *SC algorithm* returns a single scalar distance value, denoted by $SCdistance(S, S')$ given in Equation 3.

$$SCdistance(S, S') = \min_{\pi(i)} \sum_{i=p_1}^{p_n} C_{i,\pi(i)} \text{ where } \pi(i) \text{ is a permutation of } i \qquad (3)$$

3 Hamiltonian cycle and shape context

In this section, we utilize graph theory to generate exceptional cases for the *SC algorithm*. Let $G = (V, E)$ be a simple unit graph with n vertices $V = \{v_1 \ldots v_n\}$ such that G contains a *Hamiltonian cycle* and all edges have the same unit length d. A Hamiltonian cycle is a closed path graph, H such that each vertex of G is visited exactly once except for v_1 which is the starting and ending vertex.

There are twelve possible Hamiltonian cycles for G in Figure 3. Note that a clockwise Hamiltonian cycle is identical to the counterclockwise one in shape. If a unit graph G has $n = |V|$ vertices, then every $H = (V, E')$ has exactly n vertices and $n = |E'|$ edges where $E' \subseteq E$.

Let a shape S on a binary image I be a connected component where every pixel whose value is 1 is an element of S. A shape S partitions the two dimensional plane into two regions: the interior (union of pixels whose value is 1) and exterior (union of pixels whose value is 0).

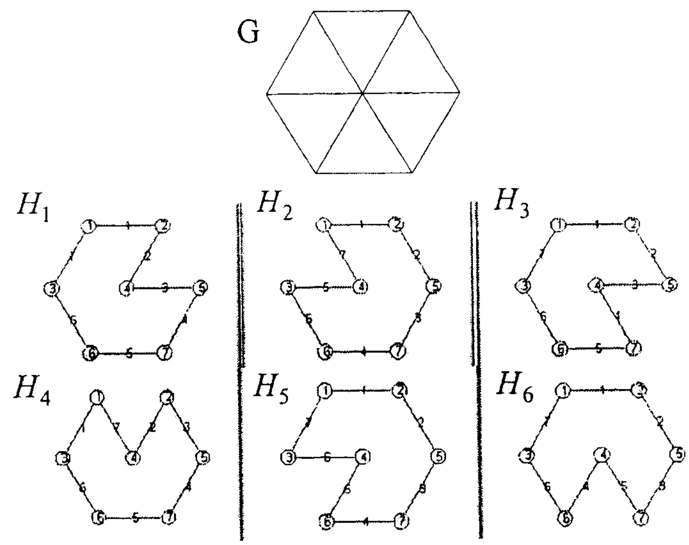


Fig. 3. A unit planar graph G and its Hamiltonian cycles.

Lemma 1: Every H_i for a unit graph G forms a certain shape S_i

Proof: H_i is a closed cycle graph by definition or an *n-gon* if G is planar. *By Jordan curve theorem* [12, 15], H_i partitions the plane into two regions, the interior and exterior. Painting the interior and exterior regions with black and white colors, respectively, produces a connected component shape. Thus, every H_i for a unit graph G forms a certain shape S_i. ■

Note that if one travels the boundary of H_i in clockwise, the object is always on the right hand side if G is planar. All examples given in this paper are unit planar graphs. If G is not a planar graph, one can easily find an example where the object side is changed from right hand to left hand.

Figure 4 shows another example of a unit planar graph and some of its Hamiltonian cycles. Now consider shapes S_1 and S_3 in Figure 5 (a) produced by H_6 and H_4 in Figure 4, respectively. S_2 was generated by slightly changing S_1 while maintaining the boundary distance of S_2 equal to that of $S_1 = nd$. S_1 is so similar to S_2 that the human visual system can hardly distinguish the two, while S_3 is very different from both S_1 and S_2. However, the *SC algorithm* calculates the $SCdistance(S_1, S_3) = 0$, i.e., S_1 and S_3 are identical, while $SCdistance(S_1, S_2) > 0$ as presented in the distance matrix of Figure 5 (b).

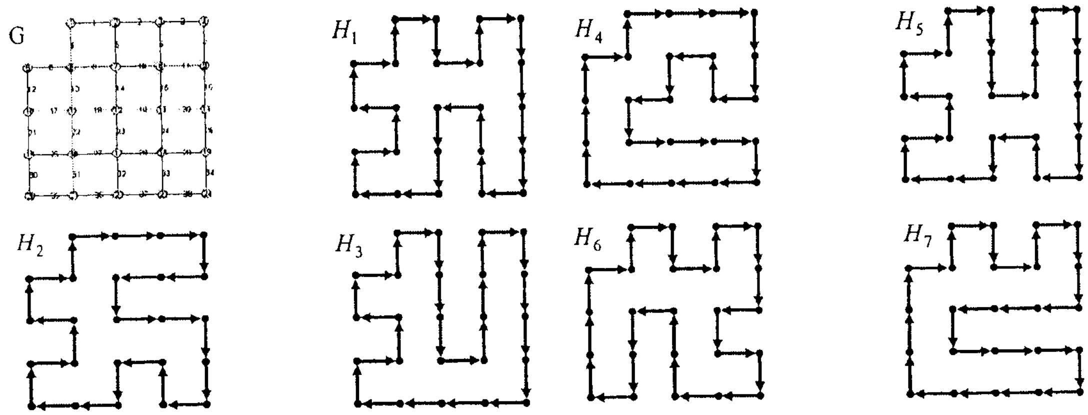
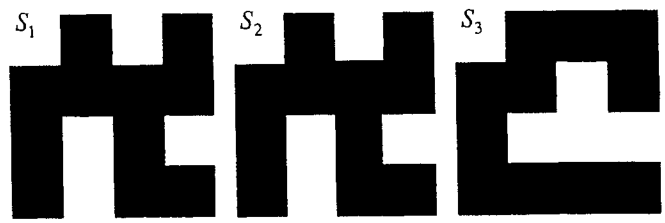


Fig. 4. A unit planar graph G and some of its Hamiltonian cycles



(a)

$$\begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \left(\begin{array}{ccc} 0 & 105 & 0 \\ 105 & 0 & 106 \\ 0 & 106 & 0 \end{array} \right) \end{array}$$

(b)

Fig. 5. *SC algorithm* performance on three shapes

Theorem 1: All shapes S_i formed by H_i for a unit planar graph G are considered identical shapes by *SC algorithm* if $p_1 \in V$.

Proof: Let S and S' be two distinctive shapes formed by two Hamiltonian cycles for a unit graph G where $V = \{v_1 \ldots v_n\}$. The *SC algorithm* will place p_is exactly on vertices on G since all edges are unit distance d in length. Let h and h' be polar histograms for S and S', respectively. Since locations of vertices are the same in two shapes, $h = h'$. Hence, we can derive the following from the Equation 2.

$$C_{i,i} = 1/2 \sum_{k=1}^{b} [h(k) - h'(k)]^2 / (h(k) + h'(k)) = 0 \quad \text{where } i = v_1 \ldots v_n$$

$$SCdistance(S, S') = \sum_{i=v_1}^{v_n} C_{i,i} = 0$$

Therefore, all shapes S_i formed by H_i for a unit planar graph G are considered identical shapes by the *SC algorithm* if $p_1 \in V$. ■

Corollary 1: The number of shapes that are considered identical by the *SC algorithm* but geometrically different $\geq \frac{H_i \text{ for a unit planar graph } G}{2}$ where $v_1 = v_s = v_e$.

Proof: Each H_i produces a shape S_i. S_i can be formed by H_i or H_i^r where H_i^r is the reverse order of H_i. Hence, $|S| \geq \frac{H_i \text{ for a unit planar graph } G}{2}$. ■

The maximum number of Hamilton cycles can be found at [13].

5 Discussions

In this paper, we reviewed the shape context and matching algorithm developed by Belongie et al. We then used the concept of Hamiltonian cycles to generate several shapes that appear different to human observers but that are considered identical by the *SC algorithm*.

Another way to visualize the *SC algorithm* is using *n* pins and a bendable loop of size *nd*. As shown in Figure 6, given a shape *S* whose boundary length is *nd*, one can place a bendable loop on the boundary of *S* and fix the loop with pins such that the distance between two pins is exactly *d*. Albeit the loop is fixed by pins, the loop can deform between pins. This gives great deformable invariant ability to the *SC algorithm*.

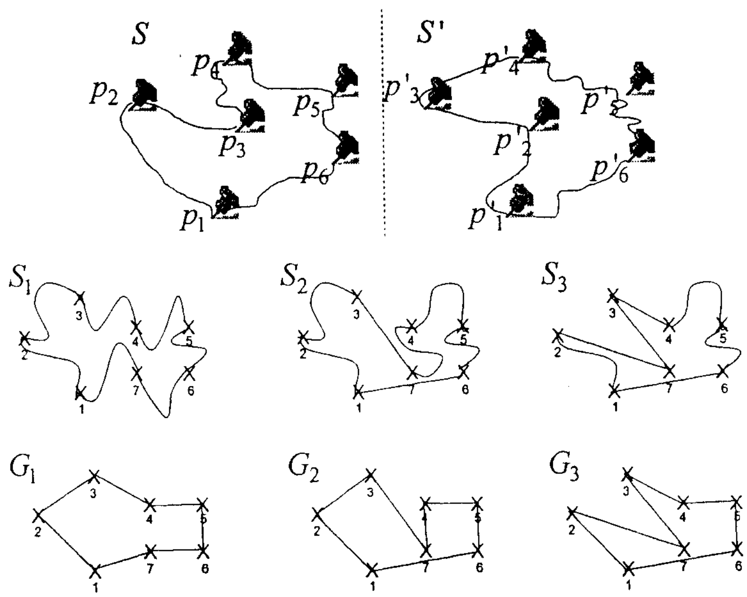


Fig. 6. Shape descriptors and graph representation.

However, one can change the loop while satisfying the distance between pins, e.g., S'. Note that the *SC algorithm* is indifferent to the order of the points [8]. Thus, S_1, S_2, and S_3 are equivalent shapes by the *SC algorithm* while their graph representations, G_1, G_2, and G_3, are different. In this paper, we used graph theory concepts to generate what can be considered counterexamples for the *SC* algorithm. We plan to explore the combination of the shape context representation and matching with graph invariance to obtain improved object recognition algorithms.

References

1. Belongie, S., "Image segmentation and shape matching for object recognition," PhD thesis, 2000
2. Belongie, S., Malik, J., and Puzicha, J., "Matching Shapes," in the Proceedings of International Conference on Computer Vision, Volume 1 2001

3. Belongie, S., Malik, J., and Puzicha, J., "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 24, No 24, pp 509-522, 2002
4. Duda, R. O., Hart. D. G. and Stork, D. G., *Pattern Classification*, Wiley New York, 2nd ed., 1999
5. Dunn, G. and Everitt, B. S., An introduction to mathematical taxonomy, Cambridge University Press 1982
6. Kuhn, H. W., "The Hungarian Method for the assignment problem," *Naval Research Logistic Quarterly*, vol. 2, pp. 83-97, 1955
7. Loncaric, S., "A survey of shape analysis techniques," *Pattern Recognition*, vol. 73, pp. 983-1001, 1999
8. Mikolajczyk, K. and Schmid, C., "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1615-1630, 2005
9. Nadler, M. and Smith, E. P., *Pattern Recognition Engineering*. John Wiley & Sons, Inc., 1993
10. Papadimitriou, C. and Steiglitz, K., *Combinatorial Optimization*. Mineola, NY: Dover Publications, 1988
11. Pavlidis, T., "A review of algorithms for shape analysis," *Computer Graphics and Image Processing*, vol. 7, pp. 243-278, 1978.
12. Preparata, F. P. and Shamos, M. I., *Computational Geometry. an Introduction* Springer-Verlag New York Inc. 1985
13. Teunter, R. H. and Poor, E. S., "The maximum number of Hamiltonian cycles in graphs with a fixed number of vertices and edges," *Operations Research Letters*, vol., vol 26, pp. 91098, 2000
14. Ullmann, J. R., *Pattern Recognition Techniques*, Crane, Russak, New York, 1973
15. Veblen, O., "Theory on plane curves in non-metrical analysis situs," *Transactions of the American Mathematical Society* vol. 6, pp. 83-98, 1905
16. Veltkamp, R. C. and Hagedoorn, M., "State of the Art in Shape Matching," Univeristy Utrecht UU-CS-1999-27, 1999

The Ivan G. Seidenberg
School of Computer Science and Information Systems
Pace University

Technical Report Series

The School of Computer Science and Information Systems, through the Technical Report Series, provides members of the community an opportunity to disseminate the results of their research by publishing monographs, working papers, and tutorials. *Technical Reports* is a place where scholarly striving is respected.

All preprints and recent reprints are requested and accepted. New manuscripts are read by two members of the editorial board; the editor decides upon publication. Authors, please note that production is Xerographic from your submission. Statements of policy and mission may be found in issues #29 (April 1990) and #34 (September 1990).

Please direct submissions as well as requests for single copies to:

Allen Stix
The Ivan G. Seidenberg School of Computer Science and Information Systems
Goldstein Academic Center
Pace University
861 Bedford Road
Pleasantville, NY 10570-2799