

COMP-766
Shape Analysis in Computer Vision

Project Report
MATCHING SHAPES BY PROBABILISTIC GRAPH MATCHING

Frank Riggi[†]
ID: 260 070 301
frank.riggi@mail.mcgill.ca

[†]Centre for Intelligent Machines
McGill University

ABSTRACT

The ability to recognize shape is crucial in many vision applications. If a given shape could be accurately modelled and matched to a base shape with some degree of certainty, then a host of applications can be developed from computer vision to medical imaging. This project analyses some newly developed ideas which seek to extend well known notions of shape modelling by skeleton and probabilistic inexact graph matching. Further, issues will be discussed on how best to represent the shape as a graph by skeletal means and how a probabilistic Bayesian network can match these graphs.

1. INTRODUCTION

Since the dawn of computer vision, recognition of either object or pattern has been a major source of study and perplexity. Although great advances have come as of late, such as the ability to recognize faces through Principle Component Analysis as Turk and Pentland have shown [10], there are still much more perplexing issues that have yet to be dealt with successfully. In the medical imaging world, there is a vast potential for computer vision to continue to further develop and enhance new and existing tools for diagnoses.

Amongst these ideas, recognition and matching of shapes in reality is not in the least a trivial task. In essence, it is important to isolate the shapes and describe them in a meaningful manner. Further it is necessary to employ some type of matching algorithm that would allow for the matching of these shapes efficiently and as accurately as possible while having some type of confidence of the match. In addition, the graph matching process must be able to recognize shapes that are not exact yet similar enough.

This project examines past attempts at creating a meaningful way to represent shapes as graphs that would subsequently be an effective input to a graph matching solution. This problem has been studied for quite a while and as of late there have been new and worthwhile developments in the realm of generating skeletons from shape. Yet a question remains: How effective can these skeletons be and what information can be obtained from them that would be effective in a graph match? In addition, what type of graph matching solutions are effective in this regard and can in fact cope with non-exact or *inexact* graphs? In essence, the nature of this project and possible future work is examining how best to conduct probabilistic shape matching through graph matching means.

Since this task is non-trivial, positive results are difficult to obtain. However this project seeks a basic framework of knowledge and understanding that can hopefully be furthered in one form or another. In the following section, a background will be given on previous work and the basics of graph matching and the representation of shape as graphs. Later in Section 3, a discussion on the theory and approach used for this project will entail. Section 4 and 5 discuss implementation and results and possible future work.

2. BACKGROUND

Since this project describes aspects of graph matching and the representation of shapes as graphs, it is worthwhile to investigate in essence what a graph is and how it can be matched. Further it is important to comprehend a method of representing shape in terms of a graph. Therefore, basic concepts and definitions must be understood.

2.1 Definitions

In order to proceed and discuss graph matching, a dictionary of definitions are necessary to clarify any misconceptions about graphs and matches. Simply put, graphs are used in many circumstances to represent data in a graphical form. By definition in [11], a graph G is a set of *vertices* V that are connected by E *edges*, hence $G = \{V, E\}$ and can be either finite or infinite. In this paper, all graphs are assumed finite since they represent a particular finite shape. The *order* of a graph is the number vertices contained in it while its *size* is the number of edges. Two vertices are considered *adjacent* if the same edge connects them. A graph H can be called a *subgraph* of G if $V_H \subseteq V_G$ and $E_H \subseteq E_G$.

An *adjacency matrix* is a matrix that describes the interconnections of a graph's vertices. The adjacency matrix is purely a structural phenomenon. It is usually a sparse matrix with 0 diagonal (though not always) and 1s denoting the connections. An adjacency matrix contains two entries for each connection (since each graph vertex is a row and a column in the matrix) and is the basis for describing how a graph is constructed for many applications.

In order to better quantify a graph and associate real world data instead of simple relationships through just structure, attributes are included to describe particulars about each node (or vertex). Thus as described in [1], the *Attributed Related Graph* (ARG) is a triple, hence $G = \{V, E, A\}$ where V and E are the vertices and edges as before and A is the set of attributes where $A = \{x_i, \forall i \in V\}$. The set of attributes can in principle be any quantitative or qualitative descriptor of the vertices. Examples of attributes may be spatial coordinates, colour, slope, etc.

2.2 Approaches to Graph Matching

There are many approaches to graph matching and most usually lie within one of the following three categories: Graph Isomorphism; Association Graphs; and Subgraph Comparison. Graph Isomorphism is a mathematical technique used to match identical or (structurally identical) graphs. In particular, two graphs are isomorphic if there exists a one-to-one correspondence function f such that for each node of G , there corresponds a similar node in H . While isomorphism will identify if the two or more graphs being matched are identical, graph isomorphism does not allow for differences in the graphs, thus rendering it ineffective should graphs not correlate perfectly.

Association Graphs are graphs that represent the relationship between two graphs. Each potential two node match is represented by a node and each node is connected as it is in each graph. The main issue that arises using this form of matching is that as before, it is not as effective when dealing with two structurally differing graphs.

Lastly, subgraph comparison matches graphs by breaking them up into 'cliques' or unique groups and attempts to perform a matching algorithm on each. This form of matching works somewhat well with differing graphs as all errors in the subgraph match is added to a global error or cost for that specific match/alignment.

Many times, the above methods are referred to as *exact graph matching* as they tend to match graphs that are identical in structure. An *inexact graph matching* method however, attempts to match graphs that are not structurally alike. An inexact graph matching procedure will measure the similarity between two graphs (usually a model and sample), or further measure the similarity between two subgraphs. This ultimately results in aspects of a graph that cannot be matched correctly. Thus, a null vertex (or *dummy node*) ϕ is introduced. Therefore as defined in [1], a matching is represented as $f:V_1 \rightarrow V_2 \cup \phi$.

Wilson and Hancock's Structural Matching by Discrete Relaxation [1] article is one that is widely cited. In fact, it forms the basis for the inference procedures outlined in [6] and [2] which is the matching basis of this project.

2.3 Graphs from Shape

The issue that arises with attempting to model graphs from shape is a fundamental one: How should a graph represent a shape? Answering this question is complex since one must remember that a graph is only a set of vertices (or nodes) that are related by their connected edges. As a seemingly simple solution to this problem (but complex in practice), one could in fact model the shape as a skeleton. This could be done by drawing a skeleton along its medial axis (the exact middle of the shape). That will result in a graph with the edges representing contours while nodes will undoubtedly be represented by the junctions or endpoints of the edges. For instance, if one considers a dog, the legs, body and tail will all be represented by a skeletal segment formed from the medial axis. The main issue that arises with such a representation is what information can be obtained from the medial axis?

The concept of representing complex shapes as graphs is not new. There have been many in the past and present whom by some form or another, have modelled a shape into a skeletal form from the shape's medial axis. In [3], Blum and Nagel propose the Symmetric Axis Transform which seeks to quantify and qualify a shape (closed contour) as a graph. The shape is effectively broken down to its symmetric axis which is represented by a type of skeleton constructed of edges that connect certain defined points (such as branch, end and normal points). This symmetric (or medial) axis can be reconstructed by evolving a series of disks along the skeleton since radius data is known at all points. In addition, Blum and Nagel are further able to classify the edges and nodes by characteristics determined during the medial axis construction. For instance, they are able to classify an arc (edge) that connects to points as a by its curvature - be it a wedge, cup or worm.

Blum and Nagel present a fascinating concept. They are able to model a shape as a graph in the form of the symmetric axis which contains enough information to reconstruct it. This could potentially provide a sufficient amount of information to be able to classify a shape as a graph that could potentially be matched. Admittedly, Blum and Nagel state that the information presented is only to open the door for further analysis as there is no practical computational method of obtaining the symmetric axis. Nonetheless, the ideas presented

here have offered others the potential for further study this idea to present a computationally viable method of creating such a skeletal graph with good quantitative descriptors.

Continuing on a similar path in [5], Siddiqi *et al* use the concept of a curve evolution procedure to create a *shock graph* from a closed contour shape. They present an interesting model which develops a graph in which four types of *shocks* are defined which are dependent upon their ‘time’ and method of formation. For instance, a first-order shock occurs at an endpoint while a second-order shock is the point where two minimums in radius occurs and which results in a ‘pinch’ in the evolving curve, thus separating into two distinct curves. These shocks have a corresponding grammar and rules that dictate how they may be interconnected amongst each other. This would allow for the construction of a unique graph (or skeleton) that describes the shape in terms of their shocks and the connections between them. Based upon this, Siddiqi *et al* present a possible graph matching algorithm that matches shock graphs of objects between like classes and differing classes.

Subsequent to the Shock Graph, Siddiqi, Bouix *et al* [4] introduce the Hamilton-Jacobi Skeleton as a way of creating a skeleton from a shape using the eikonal equation. By introducing notions of Hamiltonian physics, Blum’s grassfire flow and the flux and divergence of the vector field, a method of computationally feasible method of creating a two or three dimensional skeleton is developed.

2.4 Applications to Graph Matching of Shape

The notion of graph matching has profound uses in the medical imaging world. Take the notion of attempting to classify the cortical sulci of the human brain for example. A sulcus (singular) is a cortical fold in the brain. Most adult humans more or less possess the same sulci. However each individual’s particular sulci may in fact differ slightly from most others’ due to any number of circumstances during maturity. These differences usually entail the addition or occlusion of branches, slight differences in orientation, buried gyri, etc. Therefore, if it were possible to model the sulci as graphs and attempt to match them using an inexact matching procedure with a model, it would be of great use to the medical world to identify key sulci in the brain.

Rivière, Mangin *et al* [7] present a method of classifying cortical sulci of the human brain as graphs. These graphs could then be matched to base brains. The approach is a fascinating one. From a magnetic resonance image (MRI), preprocessing of the images occur (i.e. bias correction, histogramming and segmentation) and subsequent graphs are created that represent the structure of the individual sulcus. However once the graph of the brain is constructed, the inference procedure that [7] employs is a trained neural network. Their neural network is trained with 16 segmented brains that doctors have pre-examined and identified the particular sulci. The trained neural network therefore analyses the inputted sulci graph and outputs an answer. Due to the nature of neural networks, there is no possible explanation offered for the choice that is output. This ‘black box’ solution may be of concern since not even confidence levels or probability of match is known.

3. THEORY AND APPROACH

The approach to graph matching that is examined in this project is an inexact matching method based largely upon [1] and refers to the work of [2] and [6]. In addition, the essence of the graph itself will be examined and how a constructed skeleton can best describe the particulars of an object.

3.1 A Probabilistic Approach to Graph Matching

The use of a probabilistic inference framework for graph matching is one of interest. In [1], Wilson and Hancock use a Bayesian method to iteratively re-label nodes to optimize the Maximum A Posteriori (MAP). In other words, the object is to optimize the following conditional probability through an iterative process.

$$P(f|A_1, A_2) = \frac{p(A_1, A_2|f)P(f)}{P(A_1, A_2)} \quad (1)$$

where $P(f)$ is the joint prior for the current configuration $p(A_1, A_2 / f)$ and $p(A_1, A_2)$ are the conditional measurement density and probability density functions respectively. Wilson and Hancock's match therefore will fulfil the following requirements (as summarized in [6]):

- The neighbours of u will be matched with the neighbours of v .
- Likewise, the neighbours of v will be matched with the neighbours of u .
- The matches to the null vertices will be kept to a minimum.

In [2], Farmer proposes several models for graph matching, many derived from Wilson and Hancock's work described above. A promising approach that was studied further in [6] is one where the global structure of the graph is analysed and matched with its neighbours and is treated as a Bayesian network.

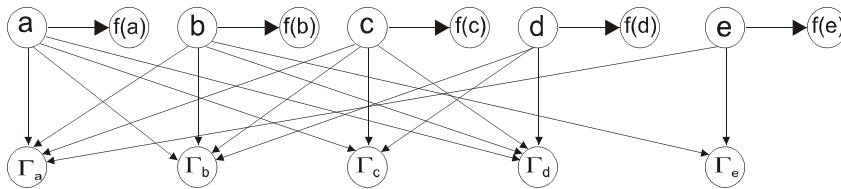


Figure 1 The belief network proposed by Farmer in [2] of a sample input graph (figure 2)

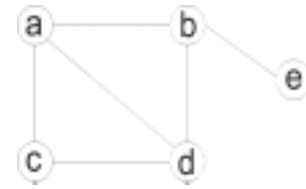


Figure 2 Sample input Graph

In figure 1, the belief network is displayed for a sample input graph (figure 2). Each node is labelled (a thru e). Attached to each node a function that represents the correctness of each match from the node's particular attributes (i.e. $f(a)$). Additionally, each node is further connected to their neighbourhood (Γ_i) node which is also connected to each node that connects to it in the sample input graph. This is to assure that process fulfils the first two requirements stated above.

Therefore Wilson and Hancock propose that the probability of an individual node's match as being correct in their neighbourhood as

$$P(\Gamma_j | S_i) = \prod_{k=1}^{|S_j|} P(f(u_k) | v_k) \quad (2)$$

such that given assumptions made that structural errors can be modelled with a uniform distribution of P_ϕ . Therefore a distribution rule is as follows

$$P(f(u_k) | v_k) = \begin{cases} P_\phi & \text{if } f(u_k) = \phi \text{ or } v_k = \text{dummy} \\ (1 - P_\phi)(1 - P_e) & \text{if } f(u_k) = v_k \\ (1 - P_\phi)P_e & \text{if } f(u_k) \neq v_k \end{cases} \quad (3)$$

Furthermore, as demonstrated in [1] and simplified in [6], the above probability rules gives the following joint probability formula:

$$P(\Gamma_i | N_i = M_j) = e^{-w_{fw}H(N_i) - w_{bw}H(M_j) - w_\phi\phi} \quad (4)$$

where:

- $N_i = M_j$ is the set of assignments between the sample vertex clique N_i (neighbourhood) and the model vertices (M_j)
- w_{fw} is the weight factor for neighbours of u_i that are missing from the match.
- $H(N_i)$ is the *Hamming distance* or number of neighbours of u_i not matched to v_j
- w_{bw} is the weight factor for extra neighbours of v_j .
- $H(M_j)$ is the number of neighbours of v_j that are not matched to u_i 's neighbours.
- w_ϕ is the weight associated the null vertices (*dummy nodes*)
- ϕ is the count of the null vertices

The probability that each node matches with respect to their attributes is:

$$P(f(u_i) = \text{true} | u_i = v_j) = e^{w_{att} \|x_{u_i} - X_{v_j}\|} \quad (5)$$

where

- w_{att} is a weight parameter used for each observation (attribute)
- x_{u_i} are the observations at u_i and v_j respectively.
- X_{v_j} are the model observations.

Additionally, Brooks introduces the option of an Exclusive OR (XOR) node to assure that only one match is made per node. This node receives an input from all other nodes and its probability is:

$$P(XOR | V_{sample} \rightarrow (V_{model} \cup \phi)) = e^{w_{xor}X} \quad (6)$$

where

- $V_{sample} \rightarrow (V_{model} \cup \phi)$ is a complete mapping from sample to model vertices.
- w_{xor} is a weight parameter for the XOR node.
- X is the number of duplicated nodes being matched.

3.2 Graph Attributes

To increase the reliability of the graph matching process, attributes are introduced in an attempt to restrict matching of certain nodes. Without attributes, most graph matching algorithms are structural, meaning that only the interconnections between the nodes and edges are matched. As previously mentioned, there is the potential to obtain a wealth of information when skeletonizing a shape, so much so that the shape can be reconstructed quite satisfactorily. Therefore, it is to one's advantage that some of the more pertinent data is collected and input into the graph matching procedure to ensure a satisfactory match. It is also important to note that the data not be too restrictive. Therefore, it is a balance to determine what describes the characteristic as acceptable.

This project makes use of the implementation of Dimitrov *et al* [9] to construct Hamilton-Jacobi skeletons. Dimitrov's implementation is able to create a skeleton by using an extension of the divergence theorem.

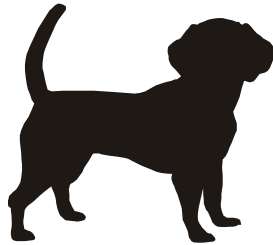


Figure 3 Closed contour shape of a dog.

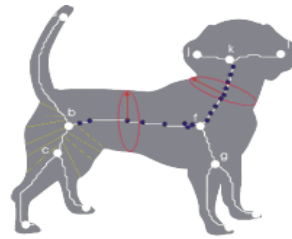


Figure 4 Example of the dog's skeletal fluxpoints.

From a practical standpoint, [9] is able to construct a skeleton and model it as a graph of nodes and edges. The nodes are defined as the endpoints and junction points of each skeletal segment which are defined as the edges. Further at each discreet point along the edge of the skeleton (called *flux points*), the given radius needed to reconstruct the original contour along with the skeletal tangent angle. Additionally, spatial coordinates at all nodes and flux points are known. This actually provides a wealth of information that not only allows for the shape's reconstruction but also to describe it in some meaningful way during a matching procedure.

The Hamilton-Jacobi skeleton from [4] and [9] offers a quick method of achieving a graph from a shape. Nonetheless, the graph representation that is usually outputted may contain surprising edges that from a curve evolution or flux-density standpoint may seem practical, but may not be representative of the entire shape. Take the panther for example in figure 5. On the whole, it's skeleton is generally representative of the animal except for the edge that lies behind the head that will reconstruct the slight hump on the panther's back. It is therefore important to realize that on occasion even a slight change (such as in pose or orientation) can have a significant impact on it's skeletal representation. Therefore prior to employing the skeletal graph to a matching algorithm, considerations for such segments should be made. This is further discussed in the subsequent section.

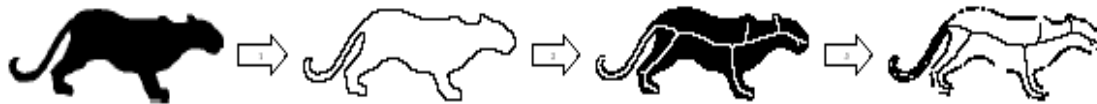


Figure 5 The panther from [9]. Notice that the small hump behind it's head contributes a large arc to the skeleton yet it only reconstructs a small portion of the shape.

3.2.1 Attributes from the Hamilton-Jacobi Skeleton

As stated above, there are many possible attributes that can be retrieved from the Hamilton-Jacobi skeleton. Firstly, since the nodes are defined as the endpoints or junctions of the arcs, the spatial coordinates of these vertices are attainable. Therefore from that knowledge alone along with the node's relationships amongst each other (interconnections), it would be possible to limit the nodes' possible matches to those which are spatially similar. However, spatial locations of these nodes alone do not seem sufficient. Take for example the comparison of a dog and cat (or any four legged animal for example). From a standpoint of knowing only the locations of nodes, how would one distinguish it fully if there are similar nodes for just each extremity and connections?

If it were possible to know further information about the structure of each animal, it would in fact assist in the matching process. It had been previously stated that each skeletal segment is created along the shape's medial axis. Thus it is each arc (or the edge segment) that provides more characteristic data. For instance, knowing that a cat's torso differs from a dog's which further differs from a horse provides a possible attribute of interest. Much of this information can be obtained in what [9] refers to as flux points - the discrete points that make up each edge (or arc) between two nodes. Simply assigning each flux point as a node could provide much more information in terms of attributes, but would also entail having a graph of hundreds (or even thousands) of nodes. This obviously would provide too much information, so much so that a match probably would never occur and would be too complex for any algorithm to handle.

Graph Duality

A useful concept of graph theory is that of *duality*. More specifically, a *dual* of a graph is a transformation which essentially entails assigning the original edges as vertices and their original connections at the vertices as edges. Of course this is not a one-to-one transformation since the size of the former graph becomes the new graph's order but the opposite is not necessarily true.

Since edge segments are now considered nodes, a variety of attributes attainable through the flux points can be used to describe these arcs such that they can in fact be considered characteristic of the originating shape.

Arc Length

Perhaps the first attribute under consideration should be that of arc length. The arc length of a contour could simply be calculated from their discrete (x,y) coordinates of the flux points for each segment. Hence, each segment's length could be determined by an iterative process where the arc length L is simply:

$$L = \sqrt{\Delta x^2 + \Delta y^2} \quad (7)$$

Contour Slope

A measure that may be of use is the slope of each skeletal segment. Since there is the possibility that the segment's slope should change from one part to another, two or more slope values could be taken. However, it is obvious that this may not be an attribute of choice since it is extremely pose dependent. If a constraint was put upon pose, this measure may be of some use. Nonetheless, it is still considered a potential attribute since it is a measure that can be obtained through the skeletonization.

Radius

Perhaps the most descriptive attribute of each segment is its radius values at each flux point. More specifically, relationships between segments and their change in radius values can be obtained and utilized. By examining skeletal segments of the dog and plotting their corresponding flux point radius, patterns emerge that can undoubtedly be considered attributes. For example, the radius of some segments do not change significantly, some seem to increase or decrease, while others seem to dip or arch in the middle. In essence, this is a concept that is examined in [5] where the *shocks* are labelled by their method of construction through radii. Hence a dip in radius along a segment would in fact correspond to a second-order *shock*. If maxima and minima in radius are located along each segment, a relationship between them can be calculated.

In taking the slope of the radius plot (i.e. the rate or change or derivative) along each segment, the characteristic of knowing whether the radius is increasing, decreasing or both (dip or arc). Hence, from observations conducted from simple images, calculating two slope values (from origin to midpoint and midpoint to endpoint) sufficiently modelled the change in radius. Thus for a dog, its torso size decreases in the middle of its body which is revealed in the plot (see figure 6)

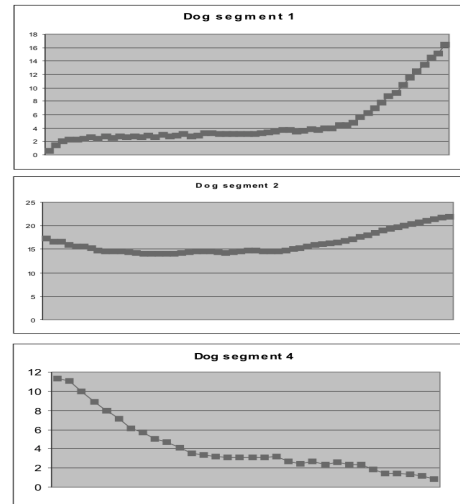


Figure 6 The radius of flux points along segments 1, 2 and 4 of the dog.

Segment Significance

An issue that has been raised several times already is whether or not a particular skeletal segment is significant in describing the overall shape of the object. As in the former example of the panther, complications can arise since such a long segment reconstructs too

small of a unique part of the overall shape. Therefore it's overall importance is somewhat diminished especially when once considers that pose changes could and often do contribute to the creation of extra segments such as that one. Therefore a ranking or weight of how much a certain segment contributes to the overall reconstruction of the shape is a worthwhile attribute.

4. IMPLEMENTATION

The implementation of the Hamilton-Jacobi skeleton from an input shape was completed by Dimitrov for [14] and [9]. A simple two bit (monochrome) image of a closed contour is input and the generated skeleton is output. Modifications were made to the program such that the program could output the relevant information (such as nodes, edge relationships and flux point values).

Additionally, as stated in the previous section, an attribute of interest is the overall relevance that the any particular segment has towards the re-construction of the original shape. To determine the significance that each edge segment contributes to the shape, a brute force method of reconstruction was employed. Since each flux point contains the required information to reconstruct its part of the original shape, a series of disks were constructed at each flux point with the appropriate radius. However, in order to quantify the uniqueness that each segment added to the shape, each segment was reconstructed separately. Then, the shape was reconstructed by adding together all segments except for the one under consideration. By examining only the unique pixels that the given segment contributes to the overall shape, two important ratios are calculated to describe it's significance and importance:

$$R_I = \frac{\text{unique pixels}}{\text{total pixels}} \quad R_S = \frac{\text{unique pixels}}{\text{original pixels}} \quad (8)$$

where R_I is the *Importance Ratio* which describes the amount of unique pixels each segment contributes to the overall shape while R_S is the *Significance Ratio* which measures the amount of unique pixels that the segment adds to the shape with respect to itself.

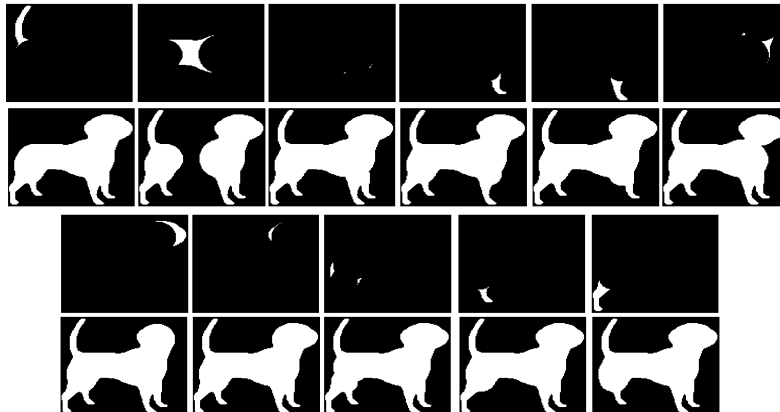


Figure 7 Above is each segment's unique contribution to the shape and below is the reconstruction without it. (top left is 1, clockwise)

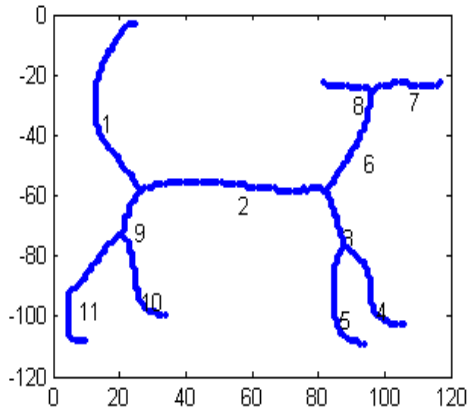


Figure 8 The Skeleton of the Dog with each edge labelled

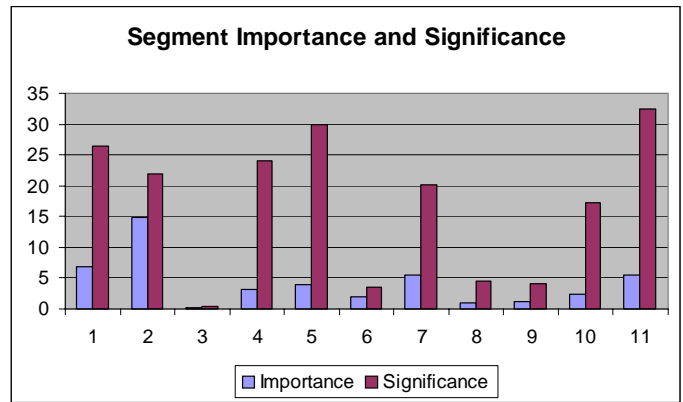


Figure 9 Percent of unique pixels contributed by each reconstructed edge of the dog. Importance is the overall ratio of contributing towards the global shape while significance is with respect to itself.

Finally, the initial implementation of the graph matching algorithm was completed by Brooks in [6]. The algorithm is employed through MatLab using the Bayes Net Toolbox (BNT) for MatLab [13] and the GraphViz software. As mentioned in [6], optimization was not a goal in implementation. Hence due to the inference procedure selected through the BNT, the algorithm breaks down when comparing graphs of order 6 or greater (six nodes). Therefore that greatly reduced the number of examples possible to create through the skeletal generating program. Further, the specific inference procedure did allow for different weights of importance to be utilized but only comparison of attributes through a sum of square differences calculation.

5. RESULTS AND DISCUSSION

Due to the implementation issues that arose with the graph matching algorithm, and its lack of ability to match higher order graphs, a simple paw shape was created and skeletonized. The model and sample are exact duplicates except that a small protrusion was added to the sample (figure 10) and their resulting skeletons (figure 11). The resulting Bayesnet for the match is displayed in figure 12.



Figure 10 Two paws that are to be skeletonized and matched: (i) model and (ii) is the sample.

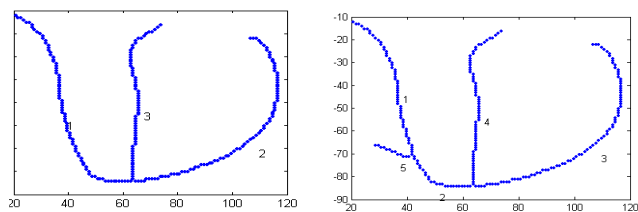


Figure 11 The output skeletal graph with labelled edges (i) model and (ii) sample.

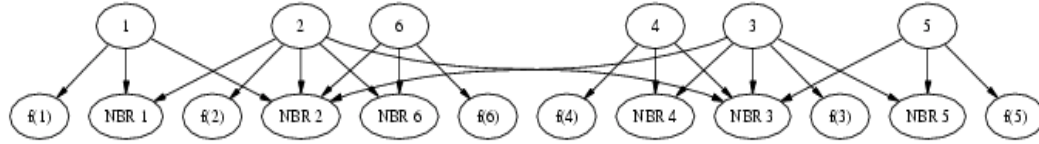


Figure 12 The Bayesnet that was created by the matching algorithm for the two paws.

5.1 Matching Results

Although many trials were conducted with the two shapes (and others that did not fair so well due to the algorithm's current limitations), below are some rudimentary results of different situations.

Table 1 Matching results with different conditions

Condition	Number of Results	Best Match
Spatial Coordinates only	1	
Here the nodes that did not change due to the addition of the protrusion matched perfectly. However the three nodes that are affected are matched to the null node. NB (In this case, the dual of the graph was not used, hence the nodes being matched are endpoints and junctions of the segments)		
No Attributes (no XOR)	48	<p>no XOR node</p>
XOR Added	12	
With the XOR node not enabled and no attributes to restrict the potential, there are no restrictions upon the matching algorithm with respect to matching the same node more than once. Upon a restriction with the XOR node, 12 matches occur.		
Arc Length, Derivative of Radius	1	
With the weight (w_{att}) of the arc length set at 0.2 and that of the radius derivative at 1.0, the following match occurred. Notice that once again, those nodes segments that were affected by the protrusion did not map correctly but to the <i>null node</i> .		

Additionally, if the attributes are too restrictive, no match would occur as all nodes would match to the *null node*. This included having too high a weight factor for arc length, or using contour slope.

5.2 Discussion

The graph matching algorithm did not perform as well as expected, primarily due to the reason that it was not originally optimized to handle higher order graphs or attributes other than ones that would be calculated with a sum of square differences. Further, modifications to the algorithm were attempted but proved futile in the span of time allotted. However, the algorithm still shows promise since in sample trials with pre-configured graphs and added perturbations, the algorithm functioned well, therefore the potential exists for improvement.

5.2.1 Comments on Attributes

Upon running tests on the paws and on further observations regarding the nature and calculations of the attributes, the following observations are made:

- *Arc Length* posed problems when attempting to match segment 1 from the model paw with segments 1 and 2 in the sample (unless the weight w_{att} was kept low). It is clear to understand why this occurs since the sample's protrusion adds a node thus dividing the original segment in two.
- *Contour Slope* proved to be an unsatisfactory measure since it is extremely pose dependent. As mentioned earlier, it would also be of substantial use if in fact restrictions are placed upon the pose. Further, in for other shapes, many times the slope was undefined if there exists a vertical segment - proving more problematic.
- *Spatial Coordinates* could be of more use if they are calculated with respect to endpoints of segments, rather than just the (x,y) position of an arbitrary node (endpoint or junction) on the graph.
- The *Derivative of Radius* measurements along each segment worked well. However, since the trial involved only dividing the segment from origin to midpoint to endpoint, the issue that arose from arc length appears here as well since segment 5 bisects segment 1, thus creating two segments. If the slopes are calculated from minima to maxima (or vice versa), this issue may disappear.
- Importance and Significance Measures were unable to be incorporated into the inference process by the submission deadline since a fundamental change to the inference procedure is needed such that the algorithm may easily wish to disregard (or assign to *null*) those segments that do not contribute much towards the shape.

5.3 Further Considerations

Although the graph matching algorithm somewhat failed due to optimization problems, it nonetheless has the potential to contribute in the realm of the graph matching of shape since it takes an interesting notion of matching global structure by *inexact* means. By implementing some of the optimization recommendations in [6], it may be possible to effectively use this particular graph matching approach. Additionally, if the a way is found to best incorporate the importance and significance ratios into the inference procedure, a more robust inexact matching may in fact occur and allow the a more practical use of the Hamilton-Jacobi skeleton to and solve the lingering issues regarding less important or less significant skeletal segments from restricting potential matches.

6. REFERENCES

- [1] R. C. Wilson, E.R. Hancock, "Structural Matching by Discreet Relaxation." *IEEE Transactions on Pattern and Machine Intelligence*, Vol. 19, No. 6, June 1997.
- [2] S. J. Farmer, Using Probabilistic Networks for Graph Matching. PHD Thesis Proposal. <http://www-users.cs.york.ac.uk/~sara>, 2001.
- [3] H. Blum, R. N. Nagel, "Shape Description Using Weighted Symmetric Axis Features." *IEEE Conference on Pattern Recognition and Image Processing*, June 1997.
- [4] K. Siddiqi, S. Bouix, A. Tannenbaum, S. W. Zucker, "Hamilton-Jacobi Skeletons." *International Journal of Computer Vision*, vol 48, no. 3, pp. 215-231, 2002.
- [5] K. Siddiqi, A. Shokoufandeh, S.J. Dickinson, S. W. Zucker, "*Shock Graphs and Shape Matching.*" *International Journal of Computer Vision*, vol 35, no. 1, pp. 13-32, 1999
- [6] R. Brooks, "Exact Probabilistic Inference for Inexact Graph Matching", <http://www.cim.mcgill.ca/~rbrook>, 2003.
- [7] D. Rivière, J.-F. Mangin, D. Papadopoulos-Orfanos, J.-M. Martinez, V. Frouin, F. Régis, "Automatic recognition of cortical sulci of the human brain using a congregation of neural networks." *Medical Image Analysis*, vol. 6, no. 2, pp. 77-92, 2002.
- [8] D. Koller, N. Friedman, "Representing Complex Distributions." and "Inference with Bayesian Networks." Lecture notes, 2000.
- [9] P. Dimitrov, J. N. Damon, K. Siddiqi, "Flux Invariants for Shape." *International Conference On Computer Vision and Pattern Recognition*, 2003.
- [10] M.A. Turk, A. Pentland, "Eigenfaces for Recognition." *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991.
- [11] V. K. Balakrishnan, *Schaum's Outlines - Graph Theory*, McGraw-Hill, 1997.
- [12] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1998.
- [13] K. Murohy, "Bayes Net Tooldbox for MatLab." Software program available from <http://www.ai.mit.edu/~murohyk/Software/BNT/bnt.html>, 2003.
- [14] P. Dimitrov, "Flux Invariants for Shape." *Masters Thesis*, McGill University, 2003.