A review of distance-based and feature-based shape-matching techniques

Vittal PREMACHANDRAN¹, Ramakrishna KAKARALA¹, and Jian XU²

¹School of Computer Engineering, Nanyang Technological University, Singapore. email: {vitt0002, ramakrishna}@ntu.edu.sg
²Precision Measurements Group, SIMTech, Singapore. email: jxu@simtech.a-star.edu.sg

Abstract

Detecting anomalies is crucial in industries such as aerospace engineering. Advancements in image processing and computer vision technologies enable the use of imaging technologies in performing non-destructive testing of structural elements. In this paper, we review the state-of-the-art in contour-based image retrieval and point out how these techniques can be used in anomaly detection. We will also review some of our previous works in shape-matching and shape-based image retrieval in this paper.

Keywords: shape-matching, image retrieval.

1 Introduction

This is a review paper about shape matching techniques. Similarities/dissimilarities between shapes can be identified by measuring the cost of matching the two shapes. The more similar they are, the lesser is the cost of matching, and the more dissimilar they are, the more is the cost of matching them. We review two different approaches for matching shapes.

1. Distances between shapes can be obtained by using some distance metric such as the Hausdorff distance or the Chamfer distance.

2. Distances can also be measured as the cost of matching the respective features or the deformation of the respective locations on shapes. We review some many such shape matching algorithms that are present in the literature, in the upcoming sections.

2 Distance-Based Shape Matching

Given two shapes, their similarity could be computed using certain distance-based techniques such as the Hausdorff distance [1, 2] and the Chamfer distance [3]. These distance-based techniques have been used for template matching. These techniques primarily compute the
“distance” between the two given shapes and see how similar they are to each other. The smaller the distance value, the more similar the shapes are to each other. Equation 1 gives the general equation of Hausdorff distance.

\[ h(A, B) = \max_{a \in A} \{\min_{b \in B} \{d(a, b)\}\} \]  \hspace{1cm} (1)

Where \( h(A, B) \) is the Hausdorff distance between two sets of points \( A \) and \( B \), \( a \) is a point in the set \( A \), \( b \) is a point in the set \( B \), and \( d(a, b) \) is the distance between the point \( a \) and point \( b \). The distance metric can be Euclidean distance or any other distance metric, based on the application. For shape matching, the most common distance metric that is used is the Euclidean distance. The distance obtained using Equation 1 is usually not used as the final metric for shape matching. This is because, \( h(A, B) \) is not always symmetric (\( h(A, B) \neq h(B, A) \)). Also, \( h(A, B) = 0 \) does not imply that \( A = B \). It only implies that \( A \subseteq B \). If the Hausdorff distance is modified as given is Equation 2, it can then be used as a metric for shape matching.

\[ H(A, B) = \max\{h(A, B), h(B, A)\} \]  \hspace{1cm} (2)

Early implementations of these techniques for shape matching, were not scale and rotation invariant [4]. Also, clearly, Equation 2 does not account for translation. Recent techniques have been developed to make them invariant to the common transformations [5]. However, [5] does not account for rotation. The main idea that the authors of [5] have used for achieving translation and scale invariance is to make use of a sliding window and a multi-scale window, respectively. Such methods are considered as brute force solutions and it usually takes a lot of time to scan the complete image.

Chamfer distance is another distance metric, which was was first introduced by [6]. It is defined as the average of the distances from a nearest point in a set \( B \) to all the points in a template set \( A \). Equation 3 defines the chamfer distance mathematically.

\[ d_{cham}^{A,B} = \frac{1}{|A|} \sum_{x_a \in A} \arg \min_{x_b \in B} \|x_a - x_b\|_2 \]  \hspace{1cm} (3)

Where \( A \) and \( B \) are two sets that are being matched, \(|A|\) is the number of points in the set \( A \), \( x_a \) is a point in set \( A \) and \( x_b \) is a point in set \( B \). The above equation is one of the most basic versions of the Chamfer distance. As in the case for Hausdorff distance, it can be seen from Equation 3 that the distance metric does not account for similarity transformations. Some modifications have been made to the above equation to incorporate invariance to translation. However, scale still remains a problem. Some authors have tried to introduce invariance to transformations using other techniques that are not related to distance matching. One example is the use of stereo to calculate the scale and then use the Chamfer distance to compute the shape similarity [7]. Many of these invariance techniques are either application specific or require the use of external equipment such as stereo vision cameras for scale analysis, which prevent them from being ported onto other systems. Hence, such distance-based techniques cannot be relied upon for robust shape matching.
3 Feature-Based Shape Matching

The distance-based matching that we just saw in the previous section is not invariant to basic transformations and is susceptible to output false distances in the presence of noise. Therefore, recent research has moved towards feature-based matching of shapes. The question that was being asked about a decade ago is what kind of features to use. Primitive-based features were the most favorable direction at that point in time and was actively researched upon since primitives provided a somewhat semantic description of object parts.

Corner detectors gathered some momentum as they were easy to compute and since corners were considered to be the “important” locations on the object’s boundary. Some notable corner detectors include the Harris corner detector [8], SUSAN corner detector [9] and the FAST corner detector [10]. The appeal of higher-order primitives have also resulted in other higher-order primitive detectors such as the Line Segment Detector [11] and rectangle detectors [12]. However, such detectors always make us ask whether the specific primitive under consideration is the best primitive to use in order to describe object shapes.

The fact that the above question has, to date, not be answered, and the fact that we find it difficult to describe a shape in words, shows the abstract nature of shapes. Bridging the gap between higher-level object semantics and lower-level image pixels has been an extremely difficult task and is a largely unsolved problem. The difficulty in developing semantically meaningful features is more-or-less accepted in the community and there is now a widespread interest in the use of statistical techniques. The following subsections give a sense of how the field has progressed in the past decade. The evolution of shape features and shape matching techniques are laid out in the rest of the section.

3.1 Shape Descriptors

Early work used shape silhouettes as the input data and tried to extract descriptors out of it. Fourier descriptors [13, 14] are an example of such shape descriptors. Fourier descriptors consider the shape boundary as a periodic function and represent it using the frequency coefficients. A lack of invariance of the Fourier descriptors to shape deformations has somewhat diminished the interest in this area. Following the success story of the SIFT feature descriptor [15, 16], lots of attention has been diverted towards the development of histogram-based shape features.

**Shape Context:** A major milestone in the advancement of shape descriptors came from the work of Belongie et al. [17]. They proposed a histogram-based shape descriptor namely, Shape Context (SC), which effectively captured the shape properties of an object. The SC descriptor is computed at a finite set of points, which are uniformly distributed across the shape’s boundary. The first major contribution came from the claim (verified experimentally) that these set of finitely sampled points need not be corner points or “special” in any way. The experimental results validated this claim and showed that there was nothing “special” about an object’s corner points. Two notable findings have paved way for the development of statistical shape descriptors. Firstly, the subjectivity in the definition of corner points made it hard to develop corner detectors that produced semantically meaningful points on the contours, thus diminishing interest in this area. Secondly, the claim that the corner points did not contain any “special” value also diverted the attention from developing more sophisticated corner detectors onto the development of more robust histogram-based shape
Figure 1: [Best viewed in color] The 2-D shape context histogram. The value in each bin is the number of points that fall in that bin. Histograms at similar locations look similar.

descriptors.

The Shape Context is basically a 2-D histogram of distances and angles. The distances are obtained by calculating the Euclidean distance between the point under consideration and every other sampled point. Similarly, the angular information between any two points is obtained by finding the angle between the tangent drawn at the point under consideration and the displacement vector between the two points. Armed with the distance and the angular information, a histogram is then created at each sampled point, which is the shape context histogram of that point. Similar such shape contexts are generated for every other sampled point on the object’s boundary. Furthermore, the binning is done in such a way that points that are close to each other are given more importance than the points that are far away from each other. To specify such importance, instead of uniform binning, the authors of [17] adopt a log-polar style of binning.

Figure 1 gives an example of the shape context at a particular sampled point on the star. The red circular disc is a log-polar histogram with angular and distance bins. The value in bin \((i, j)\) is the number of points that fall in the intersection of the \(i\)-th distance bin and the \(j\)-th angular bin.

More formally, we could define the shape context as follows. Suppose we are given a shape, \(S\), we first sample the shape into a set of \(n\) equally spaced points, \(S = \{p_1, p_2, ..., p_n\}\). Each point \(p_t\) is a two dimensional vector, with the \(x\)-coordinate and the \(y\)-coordinate as its dimensions. The histogram at point \(p_t\) is defined as

\[
\mathcal{H}_t^S(k) = \#\{p_s : p_s \in \text{bin}(k)\},
\]

where the value in the \(k\)-th bin is the number of points, \(p_s\), falling into \(\text{bin}(k)\), \(\forall s \in \{1, ..., n\}\). The shape context and the matching stages have since been improved by the original authors leading to a series of publications [19, 20, 21, 22]. The Shape Context does have a lot of advantages. Since the histogram is composed of relative distances, it is inherently invariant to translations. The angles are computed relative to the tangent at the point under consider-
In order to make the descriptor invariant to scale, the distances are all normalized by the mean of the distances, before binning.

With all its advantages, the Euclidean distance that the SC uses does not allow it to be invariant to articulations in shapes. This problem led to the next major milestone in the form of Inner Distance Shape Context (IDSC).

**Inner Distance Shape Context:** Ling and Jacobs [23, 24] identified the lack of invariance of the Shape Context to articulations and proposed a variant of the SC in the form of Inner Distance Shape Context. The IDSC at a point is also made of the same two ingredients; distances and angles. However, it differs in the manner in which the distances and angles are calculated. For any two given points, $p$ and $q$, the inner distance between the two points is calculated as the shortest distance along the path joining the two points, such that the path lies completely within the boundary of the object. The angle between the points is the angle between the tangent drawn at point $p$, and the first part of the path leading to point $q$. The authors name the angle as the inner angle. Ling and Jacobs found that their descriptor worked well with 8 distance bins, as opposed to the 5 distance bins that was used in the original SC. Therefore, the IDSC is a 96-D descriptor at each sampled point on the object’s boundary. Figure 2 gives an illustration of how the inner distance and the inner angle is calculated.

In addition to the new descriptor, Ling and Jacobs also followed the idea suggested by Thayanathan et al. [25] to make use of the continuity constraints of the sampled points. The continuity constraints basically states that, on an object contour, a point $p_i$ neighbors just two other points $p_{i-1}$ and $p_{i+1}$ and that they follow a sequential ordering. This observations allows for the use of efficient matching techniques such as dynamic programming, thus allowing for a quick matching of shapes.

**Non-Planar Shapes:** A final variant of the SC that is worth mentioning is that of Gopalan et al. [26]. IDSC assumes planarity of shapes and thus, it does not perform well when the silhouettes under consideration are non-planar projections of 3-D objects. This issue was identified by Gopalan et al. and they propose to first affine-normalize the object parts before calculating the IDSC. They assume that the parts are near-convex regions of the shape boundary and propose an algorithm to extract all near-convex regions. They then affine normalize all the parts and then use IDSC to compute the shape features.

**3-D Shape Descriptors:** While this paper concentrates predominantly on 2-D shape descriptors, for completeness, we would like to mention some works in 3-D shape descriptors as well. Kokkinos et al. [27] have extended the 2-D shape descriptor, Shape Context, to help describe 3-D shapes. Their descriptor is still made distances and angles. However, their distances are intrinsic to the shape. Therefore, they name their 3-D shape descriptor as

![Figure 2: Illustration of how to compute the inner distance and the inner angle.](image)
Intrinsic Shape Descriptor. Kalogerakis et al. [28] try to learn an automatic segmentation algorithm on 3-D shapes and use a list of features (curvature, principal components, shape diameter, average geodesic distance, orientation features, etc.) to train a discriminative labeling algorithm using Conditional Random Fields (CRF). The area of 3-D shape descriptors and retrieval is quite new and has lots of potential for future work.

3.2 Shape Matching

Once the shape features have been computed, matching respective shapes boils down to matching their respective features. The cost of matching a point $p_i$ in shape, $S_1$, to a point, $q_j$, in shape, $S_2$, is the cost of matching the shape contexts at the two points. Shape contexts, being histogram-based, allows for the use of standard histogram matching functions such as the $\chi^2$ test statistic,

$$C_{ij} \equiv C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{[H_{S_1}^i(k) - H_{S_2}^j(k)]^2}{H_{S_1}^i(k) + H_{S_2}^j(k)},$$

where $C_{ij}$ is the cost of matching point $p_i$ to point $q_j$. Given all pairwise matching costs, the objective is to minimize the total cost of matching,

$$\Psi(\pi) = \sum_{i} C(p_i, q_{\pi(i)}),$$

where $\pi(.)$ is a one-to-one mapping function that maps points in the first shape to the points in the second shape. Standard bipartite graph matching algorithms such as the Hungarian algorithm [29] were previously used to find the optimal matches. However, as stated in the previous subsection, adding figural connectivity constraints allows us to make use of more efficient algorithms such as dynamic programming.

3.3 Partial Shape Matching

The previous subsection described the well-known Shape Context shape descriptor and its variants. The SC, being a 2-D histogram of distances and angles, where the distances and angles are computed between every point and every other point on the shape, makes it a purely global shape descriptor. Such descriptors can be used when the object is well-segmented from the image. However, segmentation is still an unsolved problem. Also, most state-of-the-art edge detectors are nowhere close to producing smooth closed object boundaries. Therefore, ability to match shapes partially is extremely important while recognizing objects in real-world images.

Bronstein et al. [30, 31] tackle the problem of partial similarity and show how objects that have large similar parts (but not completely similar) can be matched. They define their own intrinsic shape features and find correspondences between shapes. They also present a novel approach, which shows how partiality can be quantified using the notion of Pareto optimality. Matching shapes at different levels of partiality gives rise to the Pareto boundary. Any point on the Pareto-optimal boundary has the same cost of matching. Different points on the boundary account for a trade-off between partiality and dissimilarities.
The interesting notion of Pareto optimality has since been applied by Donoser et al. [32] for matching partial shapes. They also develop a new angle-based shape descriptor that allows for efficient matching of partial shapes. They identified that combining the features into histograms make the descriptor purely global and, therefore, retained the true angular values (without binning) in the form of a matrix. The descriptor of a contour segment with \( n \) landmark points is a matrix of size \( n \times n \), with the \((u, v)\)-th entry given as,

\[
\alpha_{uv} = \angle(L_{u,v}, L_{v,v-\Delta}),
\]

where \( u \) and \( v \) are two points on the contour, \( L_{u,v} \) is the line segment joining the two points, and \( \Delta \) is an indicator of the number of points before point \( v \) with respect to which the angle \( \alpha_{uv} \) is calculated. The authors also propose an efficient way of partially matching shapes by making use of the integral images data structure, which was first introduced by Crow [33], and was later made popular by Viola and Jones [34] in their famous face detector.

Ma and Latecki [35] took motivations from the work by Donoser et al. [32] and utilized the partial shape matching technique for matching contour fragments extracted from real-world images. They not only used an angular matrix, similar to the one defined in [32], but also define a similar distance matrix between all pairs of points and use a combination of the distance and angle matrix to describe contour fragments.

We will use the notion of partial shape matching while performing object recognition from real world images. However, before moving on to recognizing objects in the wild, we will now explore the interesting area of going beyond pairwise matching of shapes to improve image retrieval scores.

4 Perceptually-Motivated Shape Context

While most of the shape information can usually be extracted from just the object’s contour, it is not true in cases where the objects have a strong base structure. In such cases, indentations in their boundaries have minimal effect on the human visual system. Figure 3 shows examples of objects that are visually similar to each other even though they have multiple indentations (and even breaks) within their contours. People tend to neglect these minor (or even major) indentations while perceiving the object’s shape. This is in accordance with Gestalt psychology, which maintains that the human eye sees objects in their entirety before perceiving their individual parts. The gestalt effect is the form-generating capability of our senses, particularly with respect to the visual recognition of figures, and whole forms, instead of just a collection of simple lines and curves. In this section, we summarize a recent work [36] of extracting the shape properties that capture the object’s shape in its entirety.
Motivated by the examples in Figure 3, it was identified that the object’s contour is not the only source of extracting the shape’s properties. The interiors of the shape also play an important role in discriminating between shapes. Therefore, a perceptually-motivated variant of the shape context shape descriptor was proposed in [36]. The shape descriptor was named as Solid Shape Context (SSC). The basic steps of extracting the SSC is as follows:

1. Extract the object’s boundary.
2. Sample the boundary into a set of uniformly spaced points.
3. Perform a Constrained Delaunay Triangulation (CDT) of the sampled points.
4. Sample a set of dense points from the triangles.
5. Extract the convex hull of the shape and sample from it, a set of uniformly spaced sparse points.
6. Compute the SSC shape descriptor at each sparse point, using the dense points for populating the bins.
7. Perform shape matching as before, using the new set of features.

The SSC is robust to minor, and major, indentations in the object’s contour. This allows us to match visually similar looking objects correctly. SSC can be used in applications that involve matching of defective objects to non-defective objects. Traditional techniques may classify defective objects as belonging to a different class because of the large cost of matching. However, if the goal is to identify objects belonging to the same class, even if they are defective, then the method of SSC is bound to provide more robust results.

5 Summary

In this paper, we have reviewed some of the standard shape-matching techniques that are present in the literature. We also listed out the standard pipeline for matching shapes. Finally, we introduced some of the recent advances in shape-matching that provides robust outputs, even in the case of indentations in the object’s contour.

References


