An aspect graph plays an important role in three-dimensional object recognition. It represents the three-dimensional shape of an object by its two-dimensional qualitative views as seen from various viewpoints. To create the aspect graph of an object, the viewpoint space is partitioned into regions, each of which corresponds to qualitatively similar projections of the object. Algorithms for creating aspect graphs of polyhedral objects have been developed. We developed and algorithm to compute the aspect graph of a curved object. Our approach partitions the viewpoint space by computing boundary viewpoints from the shape descriptions of the object given in a CAD database. These computations are formulated from the understanding of visual events and the locations of corresponding viewpoints. We also studied new visual events for piecewise smooth objects.
An aspect graph plays an important role in three-dimensional object recognition. An aspect graph represents the three-dimensional shape of an object by its two-dimensional qualitative views as seen from various viewpoints. To create the aspect graph of an object, the viewpoint space is partitioned into regions, each of which corresponds to qualitatively similar projections of the object. Algorithms for creating aspect graphs of polyhedral objects have been developed.

We developed an algorithm to compute the aspect graph of a curved object. Our approach partitions the viewpoint space by computing boundary viewpoints from the shape descriptions of the object given in a CAD database. These computations are formulated from the understanding of visual events and the locations of corresponding viewpoints. We also studied new visual events for piecewise smooth objects.
0.1 Introduction

Three-dimensional object recognition has been very active research topics in the computer vision community [8, 12, 22]. An intelligent vision system should be capable of recognizing arbitrary three-dimensional objects from their two-dimensional projections as seen from arbitrary viewpoints. It should also determine the positions and orientations of the recognized objects in the scene, so that an automated system can effectively manipulate the objects for a specific task. Model-based vision systems actively utilize geometric object models, which contain three-dimensional descriptions of objects, to perform object recognition. The vision systems analyze input sensory data, construct scene descriptions at appropriate levels of abstraction, and compare the scene descriptions with object models to obtain correct scene interpretations.

Most object recognition systems use object models described by view-independent object-centered representations. There are three general classes of object representations used in computer vision: volumetric representations, boundary representations, and generalized cones. Volumetric representations describe the shape of an object by the space occupied by the object. For example, constructive solid geometry representation is specified in terms of simple solid primitives, such as spheres, cylinders, blocks, and a set of Boolean operators to combine these primitives. In boundary representations, an object is represented by the surfaces that bound the volume of the object. Generalized cone or sweep representation describes the shape of an object by a space curve that acts as an axis, a 2-D cross section, and a sweeping rule specifying how the cross section is to be swept and smoothly transformed along the axis curve. Among these object-centered object representations, boundary representations seem to be more suitable for computer vision since what we perceive directly are surfaces of objects.

Though these object representations precisely describe the shape of an object, they do not provide any explicit information of its appearance as seen from various viewpoints. The object may look completely different from one viewpoint when compared with its appearance from a second viewpoint. And yet an object recognition system will be expected to determine that it is the same object in both cases. The lack of knowledge about object appearance makes object recognition become a difficult problem. During recognition process, we must establish correspondence matches between extracted image features and entities on object models. This direct 2-D to 3-D matching is very complicated and time
consuming since extracted features and object models are described in different coordinate systems. 3-D to 2-D transformations must be performed before the observed features can be compared with the object models.

Therefore model-based vision systems should make use of a prior knowledge of appearance of object models. One approach is to use multiple-view representations which consist of projections of an object from a discrete set of uniformly distributed viewpoints [32, 50]. Using multiple-view representations, recognition problem is reduced to 2-D to 2-D matching problem. Recognition can be achieved easily by comparing an image with the computed projections. However, this approach is not desirable since it requires a large amount of storage space and computation time. Computation time is wasted since projections of an object from neighboring viewpoints are usually similar. The recognition process will be very slow, especially when the geometric database contains many object models.

It is very desirable to have complete information about what kinds of features and their spatial relationships that we can expect in projections of an object from various viewpoints. This feature information is very useful for generating efficient recognition strategies. Deriving recognition strategies can be done during one time off-line phase, and the efficient real-time recognition can be achieved during the run-time phase. For example, feature indexing schemes (e.g. [18, 46]) can be developed to generate hypotheses that certain objects are present in particular orientations, based on the extracted features in an input image. These hypotheses can be verified by projecting the hypothesized object models back to the image and determining the "goodness" of matches. From feature information of different objects, we can also determine what are "salient" or "discriminant" features that are unique for a given object. Recently several researchers have proposed object recognition systems that utilize a prior feature information [10, 18, 36, 38, 40, 41, 68]. Different systems differ in the uses of different types of features, organizations of feature information, and recognition strategies.

One important issue is how to derive feature information from object models. This can be achieved by computing the aspect graph of an object. The aspect graph was introduced by Koenderink and van Doorn for representing object shape [47, 48]. An aspect is defined as a qualitatively distinct view of an object as seen from an open set of viewpoints. Every viewpoint in each set gives qualitatively similar projection of the object (i.e. having the same number and types of features). As an observer moves from one set to another adjacent set, the view of the object suddenly changes at the boundary, and a visual event is said to
occur. A new visible surface of the object may emerge or disappear. Two aspects are said to be connected by a visual event if their corresponding sets of viewpoints are adjacent. In an aspect graph, nodes represent aspects and arcs denote visual events. Each node is associated with a representative view of the object, from which we can determine the feature information.

Considering importance of aspect graphs, many algorithms have been proposed to construct the aspect graph of the object [20, 24, 34, 35, 40, 50, 58, 70, 71]. Most previous research focused on polyhedral objects, or used exhaustive search in the viewpoint space to locate aspects of the object. In the literature of singularity theory, many researchers have investigated visual events and their corresponding viewpoints for smooth objects and piecewise smooth objects [2, 3, 33, 45, 63]. However, the catalog of studied visual events is not yet complete for arbitrary objects. Recently, Eggert and Bowyer [27] and Kriegman and Ponce [51] have presented algorithms for computing aspect graphs of solid of revolution under orthographic projection.

0.2 Problem descriptions

Motivated by the importance of aspect graphs for three-dimensional object recognition, we propose to develop an efficient algorithm for constructing the aspect graph of an arbitrary curved opaque object, assuming orthographic projection model. Our algorithm is designed so that extensions to the case of perspective projection can be done easily. Our algorithm is also applicable for arbitrary objects that may contain both curved and planar surfaces. Input of the algorithm will be boundary representations of object models in the geometric database. Each geometric object model contains descriptions of surfaces and boundary curves in parametric forms. Each surface is assumed to consist of $C^3$ patches joining with $C^3$ continuity. Our object models are constructed by using Alpha 1 geometric modeling system [1], where the surfaces are B-spline surfaces. The outputs of the algorithm will be the aspect graph of the object, and partition of the viewpoint space into regions, each corresponds to an aspect.

Understanding visual events is basis for constructing aspect graphs. In this proposal, we studied of new visual events for curved objects and a mathematical framework for computing boundary of the partition of the viewpoint space.

Our algorithm for aspect graph generation can be outlined as the following steps:
1. Compute all potential bifurcation surfaces by locating candidate event participation points, critical rulings and planar surfaces on the object for all visual event types. Details of computations are given in the later sections as we study visual events.

2. Prune away portions of potential bifurcation surfaces using the interaction between each ruling and local geometry of event participation points. This step basically determines the potential visibilities of event participation points from directions along the rulings.

3. Calculate and record loci of accidental viewing directions from the remaining parts of potential bifurcation surfaces. The loci of accidental viewing directions intersect each other into arcs on the viewing sphere. Each arc is associated with a set of visual events and sets of connected rulings on potential bifurcation surfaces. For each visual event on the arc, record the loci of potential event participation points, the singular ruling or the planar surface that define the arc.

4. Determine the validity of each arc on the viewing sphere. Select a representative direction at the middle of the arc. For every associated visual event, check the visibilities of the corresponding event participation points, singular ruling or planar surface from the representative direction by using ray-tracing techniques. If some event participation entities are totally occluded, the visual event is deleted from the arc. If all visual events are removed, delete the arc from the viewing sphere.

5. At this step, the viewing sphere is correctly partitioned. Compute the representative view and the aspect descriptions for each region on the viewing sphere.

6. Generate the aspect graph of the object by examining the adjacency relationships between regions on the viewing sphere. Assign a node for each region and connect two nodes by an arc if their corresponding regions are adjacent. For each node, store the representative view and the feature configuration. Each arc in the aspect graph is associated with the description of visual events, and loci of accidental viewing directions.

Our proposed algorithm has several advantages over an exhaustive approach (e.g. [50]), which groups equivalent stable viewing directions by sequential search over the viewing sphere. Our approach fully utilizes the shape information in an object model, not just
for obtaining projections of the object. The required computations in our approach are proportional to the shape complexity of the object; while the exhaustive approach must examine all possible viewing directions regardless of the object shape. Simple objects usually take less time because of fewer visual events. Our approach is also independent of the resolution of the viewing sphere tessellation which effects correctness of the exhaustive approach. Moreover, our approach also computes the bifurcation surfaces for perspective projection.

0.3 Summary

The aspect graph of an object is a very useful representation for object recognition. Aspect graphs provide the knowledge of what are possible qualitatively different feature configurations that objects can assume from various viewing directions. This information is very useful for generating an effective strategy for object recognition.

In this proposal, we developed an efficient algorithm for constructing the aspect graph of an arbitrary piecewise smooth opaque object from its boundary representation. Our strategy is to compute all the accidental viewing directions that partition the viewing sphere into set of stable viewing directions. These computations are formulated from the understanding of all possible visual events, the loci of their accidental viewing directions, and bifurcation surfaces. We present our study of new visual events for piecewise smooth objects, and develop a general mathematical framework to compute accidental viewpoints. We are currently implementing the proposed algorithm, using Alpha system as our geometric modeling system. We believe that our research will make significant contributions to the field of object recognition.
Bibliography


