The Image Understanding Environment: Image Features

Keith Price and IUE Committee*
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, California 90089-0273

Abstract

An IUE image feature is a spatial object that represents some information extracted from an image. These features include image points, curves, edges, lines, regions, and complex collections of these kinds of features, such as perceptual groups. Image features without reference to the underlying image correspond to basic objects in the spatial object mathematical hierarchy. The *image feature* class shares a common development with this mathematical structure. In practice, image features will be kept in collections (the features extracted from an image) and will be used as an element of that collection of through the use of a *spatial index*. This paper describes the philosophy behind the *image feature* with a few limited examples of how they are described and used.

1 Introduction

This paper assumes that the reader is familiar with the basic concepts used by the Image Understanding Environment (especially the basics of object oriented development and the basic classes used by the IUE) and should be read in conjunction with the other papers on the IUE included in these proceedings. The complete documentation on image features in the IUE requires many pages and this paper is intended to provide a general description of the image feature classes not to be the complete description.

Central to any Image Understanding research or application program is the extraction and use of image features. Users of the IUE include applications users who are primarily interested in the user interface, applications developers who will use and extend image features, and other developers who will implement the more basic programs. For each of these groups, the *image feature* class definitions will be important.

*The members of the IUE Committee are: Tom Binford-Stanford; Terry Boult-Columbia; Bob Haralick, V. Ramesh-U. Washington; Al Hanson, Chris Connolly-U. Mass.; Ross Beveridge-Colorado State; Charlie Kohl- AII; Daryl Lawton-Georgia Tech; Doug Morgan-ADS; Joe Mundy-GE; Keith Price-USC; Tom Strat-SRI.

Image features form a portion of the larger spatial object hierarchy and are directly related to the mathematical structure of the hierarchy. IUE users such as application developers will be interested in how to use and extend the image feature classes with little consideration of the larger spatial object hierarchy. The spatial object hierarchy and the relationship of image features to it developed through the series of meetings of the IUE Committee and forms a clean way to describe and implement the hierarchy. There are several reasons why developing image features within the spatial object hierarchy is crucial.

First, there is a natural correspondence between the sequence of topological constructs, e.g. *vertex*, *edge*, and *face* used for spatial objects, and the descriptions of image features for points and junctions, edges, and regions. Access to this topological representation is especially important for describing composite image features such as linked line segments, adjacencies between regions found in segmentations, and perceptual groups. Second, since image features generally correspond to the projection of three dimensional object models, it is useful to have the same underlying operations and representations used for both of them. Third, image features are characterized by having a wide range of possible attributes. We wanted to be sure that it would be possible for users to easily define and extend the attributes associated with image features. Many of the shape attributes associated with image features are described by fitting a curve or two dimensional shape corresponding to a spatial object.

2 Relationship to Spatial Objects

Image features are defined as spatial objects that represent some information extracted from an image. These features include simple features such as points, curves, and regions and complex collections of image features in perceptual groups. An image feature has very little meaning separate from its underlying image—without the relationship to the image, image features fit into the basic spatial object hierarchy. In practice, image features will be kept in collections (i.e. a collection containing all the edge features extracted from one image) and will be stored, referenced, and manipulated as elements of that collection. A common alternative for the collection would be the *spatial index* that allows image-like (efficient) access to the features based on position.

Since image features are usually kept in collections some of the slot values associated with image features are only stored once for the collection. For example, the *coordinate-system* associated with an individual *image feature* is that of the underlying image, and since most expensive processing with the image feature is fixed to the *coordinate-system* of the image, there is no cost associated with this information.

Methods for image features are inherited mostly from the *spatial-object* class. There will be varying needs for specializations for methods relating to extraction (from the image), property value computations (i.e. color), display on the image, spatial indexing operations, input, output, grouping, and the various iterators over sets of spatial objects (subsets, all in an area, etc.). Image features will also be used as the basis for the *region-of-interest* in the image processing operations.

Image features are developed following the mathematical structures used for *spatial-object* where the *vertex*, *0-chain*, *edge*, etc. of the *spatial-object* hierarchy have analogs in the image feature hierarchy. The most apparent difference between the mathematical structure of the *spatial-object* classes and image features is the inclusion of image related property values and the use of named object classes that correspond to the kinds of image features used in image understanding research and development programs. An important requirement of the IUE has always been to support rather than hinder research, so we do not define these objects in absolute final forms but indicate what property values (slots) are possible, the names that will be used, and the associated semantics. Between different programs, the slots may vary, but the creation, reading, and writing programs will allow for missing and extra information without harm. Our first description of image features will parallel the mathematical hierarchy of *spatial-objects*. Most image features fit clearly into this parallel hierarchy, but some may not be obvious at first glance.

2.1 Image Points as Special Cases

Image points are simple features for representing image positions. As was used for point in the spatial object hierarchy, the simple *image-point* does not inherit from the *image-feature* class, but is a primitive element that only contains the image positions. Sequences of points are stored in specialized versions of the *ordered-pointset*.

2.2 Vertex Objects and Image Features

The mathematical object *vertex* is a 0-D object. A 0-D image feature is a point in the image, possibly with some associated neighborhood. This point can be something extracted directly (e.g. points from an interest operator, a corner detector, etc.) or the point could be the result of an intersection of two lines (or *line-segments*). The primitive slots for an image feature vertex are the image positions, but, depending on the feature being modeled, many other property values are possible.

A common image feature corresponding to *vertex* is the *edgel*—extracted in a neighborhood and representing the step between two intensity surfaces. A simple edgel that encodes the yes/no result of an edge operator at every pixel in the image is treated as a single point with no associated properties and is thus a *vertex*. The main aspect which distinguishes the class *edgel* from the basic *vertex* is that a local model for the geometry of the image intensity surface is assumed. The local neighborhood is defined by a disk about each potential edgel location.

In the IUE core, we allow a number of simple models for the intensity surface in the local neighborhood around each edgel location. The neighborhoods are characterized by the local structure of image intensity surfaces. The neighborhoods are defined as follows in order of frequency of occurrence in an image:

A The interior of a single surface.

B Two surfaces intersect at the edgel.

C Three or more surfaces. Often corresponds to a corner.

D The image intensity surface is too complex to be described by a simple model.

The intent of the cover with small disks is to divide and conquer, to make small neighborhoods that are sufficiently simple that it is feasible to describe the surface adequately over the neighborhood. This is a fine to coarse to fine approach. In this approach, the first level of disks is the smallest meaningful. It is simple to describe patches of class A that include a single continuous surface. Continuous patches are described in differential geometry by a tuple: (point, tangent plane, and curvature tensor).

It is reasonably simple to describe the compound surface over a disk of type B that includes two surfaces. This model is the most common representation used to define an edgel. Two surfaces are bounded either by a curve at which two surfaces intersect or by a limb, an apparent boundary. On a small disk, the boundary curve is locally straight. The model of type C is typically called a corner and is best represented as an attributed *vertex*.

In addition to the position slot that is inherited from *vertex*, the basic set of attributes that is associated with the edgel model are as follows:

Line Segment The parameters of the edgel line segment.

Tangent Vector For efficiency the local line segment may be represented as the tangent vector.

Tangent angle Another alternative is a quantized tangent angle.

Strength The local slope of the discontinuous transition between the two surfaces in the edgel neighborhood.

Left Surface Normal The surface normal of the intensity surface on the left of the edge boundary.

Right Surface Normal The right surface normal.

Covariance Matrix The variances of the parameters of the edgel model computed from the actual intensity distribution in the neighborhood. Gives a likelihood that the model holds for the local disk.

These point operation results can be grouped in a number of ways depending on the user's concept of the

underlying geometry of the boundary or region. The simplest group is the *0-chain* which is a set of vertices with an implicit linear ordering. Usually a full topological description is not applied until a local neighborhood analysis is done to "link" edgels into connected sets called *edgelchain*.

2.3 0-Chains of Image Features

The mathematical *0-chain* describes ordered collections of objects of class *vertex*, such as that formed by linking edgels into a discrete curve. For some purposes the *ordered-pointset* class may be used to group edgel sequences and for others the more complete *image-0-chain* is proper. The main difference is that the *image-0-Chain* is a topological concept which supports algebraic operations on the points while the *ordered-pointset* is a set of feature points with no topological interpretations.

A sequence of corner objects also forms a *image-0-chain* but here there is often no local neighborhood relation assumed between the corners. However, the corners usually correspond to junctions of two or more edges and it is reasonable to use the topological structure of the *vertex* for the basis of the individual junctions.

2.4 Edge, One-Dimensional Image Features

The name edge has a clear meaning in describing graphs and in topology, and has a very different meaning in most image understanding work. In the IUE descriptions, we use edge for both of these meanings, but usually the context will indicate which one is meant. The mathematical *edge* corresponds to many image features, especially bounded line segments and curves. The *image-line-segment* and *image-curve-segment* are the most obvious objects in this class. These features may be computed directly from the image or derived from other image features.

The *image-line-segment* potentially has a large number of attributes, but the basic set of attributes include:

V0 and **V1** The primary information of the line segment is the beginning and ending points. These are are vectors of image locations.

0-chain A 0-chain (ordered list) of the pixel locations corresponding to the line segment.

Strength The difference across the line segment, a float.

Segment-length The length across the line segment, a float.

tangent-angle The direction (in radians, a float) of the line segment.

2.5 1-Chain, Connected 1-D Features

The mathematical construction continues with the description of 1-Chains or ordered sets of 1-D objects such as sequences of line segments. 1-Chain structures can also intersect at junctions represented by objects of the class *vertex* and be extended to define closed region boundaries. In practice, it is often difficult to form complete topologies of these types in a bottom-up fashion. An *image-0-chain* may be analyzed to produce a *image-1-chain* composed of *image-line-segments* that approximate the original curve and the individual segments maintain the order given by the original edge elements.

2.6 Face, 2-D Image Features

The *spatial-object face* corresponds to 2-D image features most clearly represented as a *2d-image-region* in an image. The low level representation of a region can vary according to the ultimate use due to time and space efficiency considerations, and can include point sets, binary masks, interval codes, boundaries, etc. Each of these can be derived from the others. A nice aspect of treating regions as faces is that all of the *edge* mathematical structures can be used directly to determine adjacent regions and represent the geometry of the image structures.

There are several typical attributes associated with *2d-image-regions*. Some of these are simple scalar and matrix attributes for describing shape such as Area (integer number of pixels), Euler number, Centroid (the 2-D point of the centroid), Scatter-Matrix-of-Pixel-Positions, and Compactness. Some shape attributes correspond directly to instances of 1D and 2D spatial objects which describe the shape of a region and are instantiated by applying their corresponding fitting methods to the edgel chain of a region. Other attributes such as average intensity, variance, and so forth, are computed using a spatial index to register a region with an image and to access the corresponding image values. These attributes are stored as a *gaussian-distribution* with two slots, mean and variance.

2.7 2-Chains, Linked 2-D Features

A *2-chain* is a sequentially linked *face* feature. It is likely that the more conventional region adjacency graph is the more effective data structure to group faces as image regions, although it is essential that the underlying mathematical descriptions of edges and faces be used. Composite regions produced by an image segmentation procedure are represented as multiply connected faces corresponding to a set of 1-Chains enclosing pixel areas in the image plane. Region merging operations are supported by the topological operations for removing common edges between faces. Merging operations require methods to access properties of adjacent regions sharing a common boundary. These descriptions of region adjacencies are similar to the attributes used with edgel models.

2.8 Blocks, 3-D Image Features

Images are typically 2-D objects, but with range sensors and time sequences, we can expect to deal with extending the *block* to image features.

2.9 Other Collections of Image Features

Not all collections of image features fit the definitions of the X-chains. For example, *perceptual-groups* formed by clustering some number of image features into perceptually meaningful structures may result in a *image-0-chain* or *image-1-chain* where points or lines are grouped into

single curves, but frequently they are grouped into either simple shapes with few features (e.g. two parallel lines, rectangles, etc.) or clustered into an area feature where order and linking are unimportant. The IUE will provide direct support for basic groups of image features through the *2d-segment-pair*, *2d-segment-triple*, and *2d-segment-quad* classes. The specific shapes will be handled by classes such as *2d-segment-parallel*, *2d-image-corner*, *2d-segment-junction*, *2d-u-shape*, etc. which will inherit from the appropriate *spatial-object* class and contain the links to the image features that contributed to the object.

Perceptual grouping produces hypotheses that include cluster of interior cells of a region and n-tuple of edgels along a smooth curve boundary between surfaces.

Basic images features may be grouped by a variety of properties such as proximity, alignment, curvature, etc. Many different data structures can be important to use in this grouping process, such as K-D Trees and quadtrees Additionally, the Hough transform performs directional grouping. Proximity and directional grouping also often use different forms of spatial indexing. One example of the grouping process involves the generation of two sets of hypotheses from a tuple of edgels. The first set is that there is not a smooth curve between two surfaces. The second set is that there is a smooth curve between two surfaces. Hypotheses for individual edgels are that they are on the curve, that they are random spatially-invariant as a result of camera noise, and that they are "clutter," non-random and not on the smooth curve, e.g. another edge.

3 An Example Image Feature

To illustrate the construction of a class in the image feature hierarchy we will use the *2d-image-line-segment*. This description shows the level of description available to the system developers and illustrates how slots and methods are inherited in the construction of objects. It must be pointed out that the format in the complete description documents is superior to what is given in this paper.

3.1 2d-image-line-segment

A major structure in image segmentation algorithms. Many of the slots are associated with the orientation of the line and there is a standard orientation ambiguity which arises in image coordinate frames. At times it is convenient to have the image coordinates with x along the increasing image column index and y downward along the increasing row index. This coordinate is left handed, and alters the meaning of the segment orientation. The sense of the coordinate frame is provided by its definition at the level of *spatial-object*. It is assumed that the inverted coordinate frame is normally used, i.e., 'y' downward. In order to put the features into a standard right-handed cartesian coordinate system for later processing, the 'y' orientations, θ_y, must be transformed to $\theta_y + 180$ which is a method on *coordinate-transform*.

- Superior Class *image-line-segment*
- Pseudo slots – the slots that are added with this definition.
  - *tangent-angle-x float* Another alternate orientation specification for the line segment. The angle in radians of the line segment. The sense is counter-clockwise with respect to the x-axis.
  - *tangent-angle-d-x float* Same as *tangent-angle-x* except the angle is in degrees.
  - *tangent-angle-y float* Another alternate orientation specification for the line segment. The angle in radians of the line segment. The sense is counter-clockwise with respect to the y-axis.
  - *tangent-angle-d-y float* Same as *tangent-angle-y* except the angle is in degrees.
  - *slope float* The direction of the line represented as the slope in the image coordinate space.
  - *x-intercept float* The position where the segment intersects the X axis, or the column value when the row is 0.
  - *y-intercept float* The position where the segment intersects the Y axis, of the row value when the column is 0.
  - *rho float* Hough Transform representation, distance.
  - *theta-x float* Hough Transform representation, angle of the *2d-line-segment* normal in radians with respect to the x-axis.
  - *theta-x-d float* Hough Transform representation, angle *theta-x* in degrees.

The inheritance structure shows how the slots for this object are derived. Most slots are derived from the objects earlier in the hierarchy with the direct 2-D image related slots added at this stage. Additionally, many of the slots are implemented as "pseudo" slots rather than as "hard" slots. That is, these behave like slots in terms of storing the values, but do not take any space if they are not needed. Additionally, some slots are defined early in the hierarchy and are refined as the hierarchy is developed (e.g. centroid changes from the general point to the more specialized nd-image-point). Slots such as coordsys (the coordinate system for the spatial object) are usually the same for all image features corresponding to one image (and are the same as the image).

Inheritance Structure:

Class where Defined	Slot Name	Type of Slot	How Implemented
spatialobject	coordsys	pointer(coordinate-system)	hard
spatialobject	bounding-box	aligned-box-neighborhood	hard
spatialobject	centroid	point	hard
image-feature	centroid	nd-image-point	hard
image-feature	image	pointer(image)	hard
parametric-curve	domain	pointer(spatial-object)	hard
parametric-curve	range	implicit-curve	hard
parametric-curve	parametric-mapping	pointer(function)	hard
parametric-line	range	implicit-line	hard
parametric-line	lmat	vector[n](vector[2](float))	pseudo
2d-parametric-line	range	2d-implicit-line	hard
2d-parametric-line	lmat	vector[2](vector[2](float))	pseudo
topology-node	superiors	list(pointer(topology-node))	pseudo
topology-node	inferiors	list(pointer(topology-node))	pseudo
edge	0-chn	0-chain	pseudo
edge	1-chn	list(pointer(1-chain))	pseudo
edge	v0	pointer(vertex)	hard
edge	v1	pointer(vertex)	hard
image-line-segment	tangent-vector	vector[n](float)	pseudo
image-line-segment	fitting-tolerance	float	pseudo
image-line-segment	edgel-fit	float	pseudo
2d-image-line-segment	tangent-angle-x	float	pseudo
2d-image-line-segment	tangent-angle-d-x	float	pseudo
2d-image-line-segment	tangent-angle-y	float	pseudo
2d-image-line-segment	tangent-angle-d-y	float	pseudo
2d-image-line-segment	slope	float	pseudo
2d-image-line-segment	x-intercept	float	pseudo
2d-image-line-segment	y-intercept	float	pseudo
2d-image-line-segment	rho	float	pseudo
2d-image-line-segment	theta	float	pseudo
2d-image-line-segment	theta-d	float	pseudo