MVA '92 IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

RECOGNIZING OBJECTS WITH VARIABLE APPEARANCES USING THE VAPOR MODEL

John Canning

NTT Basic Research Laboratories – Naito Group
3-9-11 Midori-cho, Musashino-shi, Tokyo 180 Japan
canning@apollo3.ntt.jp

ABSTRACT

Many real world objects have variable appearances because they are flexible and/or have a variable number of parts. These objects cannot be easily modeled using current object recognition techniques which require the models to have a certain number of recognizable features with fixed relationships. We propose the use of a knowledge representation called the *VAPOR* (Variable APpearance Object Representation) model to represent objects with variable appearances. The VAPOR model is an idealization of the object; all instances of the model in an image are variations from the ideal appearance. An energy function quantifies how much variation is necessary to change the ideal object prototype to match a given set of objects in the image. The energy function is defined as the *description length* of the data given the model, i.e. the number of information-theoretic bits needed to represent the model and the deviations of the data from the ideal appearance. The shortest length model is chosen as the best description. We demonstrate how the VAPOR model performs in a simple domain of circles and polygons and in the complex domain of finding cloverleaf interchanges in aerial images of roads.

1 INTRODUCTION

Most object recognition systems exploit the fact the fact that rigid objects have a fixed set of features (more or less invariant to viewpoint) and a fixed set of relations between the features. This prevents the modeling of non-rigid objects and objects with varying numbers of parts. For example, bushes have a variable number of branches and the relative branch positions are unknown, a priori. A candle has varying diameter, color, and shape when it is new, and the amount that has burned also changes its appearance. A lion may be seen walking, running, lying, or sitting; each condition implies different constraints on the relationships of its body parts.

The definition of an object as a variation from an ideal prototype is the basis for our object representation. Instead of modeling an object as a group of lower level features with fixed (or parameterized) spatial relationships, we will model an object as a set of parts with an associated set of constraints. All models can be "matched" to a given set of data, but as more constraints are violated the cost of the match will increase. Since we are only concerned with the appearance of these objects in images, we call our model the VAPOR model (Variable APpearance Object Representation) to signify that it does not represent other aspects of the object such as it's function or composition.

Even though the objects have variable appearance, we expect to be able to decompose the object into parts. The parts, in turn, will be represented by the same type of structure, forming a hierarchical model. In simple terms, an instance of a VAPOR model is a *set* of other VAPOR instances that is varied to minimize an energy function that measures the degree of fit of the constraints. We will search for low energy instances of a VAPOR model by adding and deleting parts to and from the set of current parts.

VAPOR models also explicitly represent the parameters associated with the appearance of the object. A parameter can be any quantity that can be estimated from the set of component VAPOR instances. Parameters can include such quantities as the six degrees of freedom for the position and orientation of a rigid part, or the number of parts in the object, or the partial ordering of the depths of the objects in a given scene. The parameters are functions of the current set and can influence the addition and deletion of other elements to and from the set.

In order to compare VAPOR instances using different models, we use a common energy function scale based on description length modeling theory [8]. Minimum description length theory claims that the best description of a set of observations using a given description language is the shortest description.

The VAPOR model was motivated by several other shape and object representation strategies. Terzopoulos, Witkin, and Kass use a finite element simulation of an elastic sheet undergoing deformation in response to "forces" [10] to create what have been termed "snakes" [7] and are more correctly called "active contour and surface models." Fua and Hanson use the smoothness constraint of Terzopoulos et al. (in the form of an energy function as opposed to simulated forces) plus new model constraints such as rectilinearity of edges in their implementation of active contours [4, 5]. They use description

This work was carried out while the author was at the Center for Automation Research at the University of Maryland, College Park. The support of the Air Force Office of Scientific Research under grant AFOSR-91-0239 is gratefully acknowledged.

length theory in the construction of their energy functions, but don't adhere strictly to Rissanen's definition.

Brooks' ACRONYM system has a similar recursive part-whole hierarchy of object models and allows objects to have a variable number of parts [1]. In contrast, the VAPOR system is not limited to a fixed geometric representation of parts such as generalized cylinders and the constraints between parts can be more than simple inequalities of scalar parameters. The description length energy function provides a measure of how well a model fits the data as opposed to ACRONYM's binary decision of whether or not all the inequalities can be satisfied simultaneously.

Segen models nonrigid objects with probabilistic hypergraphs [9]. The nodes of the hypergraph represent shape primitives and their relations. The system builds a hypergraph of relations between each shape primitive in the image and n of its "nearest neighbors." The resulting hypergraph can be matched to a probabilistic hypergraph representing an object model. Segen's system learns a class model from a group of training images, and can effectively adjust the probabilities to allow object recognition that is invariant to some non-rigid shape variations. The method, however, is sensitive to the stability of the extracted shape primitives, and some changes in appearance – such as changing the number of parts – will require multiple models.

2 THE VAPOR MODEL

VAPOR instances perturb a set of other VAPOR instances, representing parts of an object, in order to find a minimum of an energy function. The VAPOR instance perturbation proceeds by adding elements to or deleting elements from the set.

The VAPOR model itself represents an object and the set of other VAPOR models represents the component parts of the object. For example, a polygon VAPOR model could consist of a set of straight edges that make up its sides. Each straight edge, in turn, can be represented as a VAPOR model containing edge pixels that are grouped together to form a line segment.

The flexibility of active contour and surface models make them good for representing constrained variations in shape. The VAPOR model is good for describing the component parts of an object as well its variable shapes. Many degrees of freedom are represented in such models, including some that are unrelated to object shape. To represent these other degrees of freedom, we allow our models to have any number of *parameters* in addition to the set of component VAPOR models. The parameters are estimated from the set of components and the input data. A polygon VAPOR model might have a parameter that quantifies the color of the interior of the polygon.

2.1 VAPOR instance optimization

Since a VAPOR model does not always correspond to a physical boundary in the way active surfaces do, we cannot use the analogy of a sheet deforming in a viscous fluid to motivate the optimization. Instead, we adopt the use of an energy potential function for our VAPOR models and minimize that potential to find the "best instance" of the object. The energy function uses the set of component VAPOR instances and the estimated parameters to determine how well the model fits the data. In Section 2.2, we describe why description length is a good choice for the energy function.

Since the energy function will depend on the contents of the set of parts, we will not know the "derivatives" of the energy function – there is no derivative with respect to a set. Several methods exist for optimizing multivariate functions without knowing the derivatives with respect to each variable. We use a non-deterministic method that can avoid most of the local minima in the energy function [3]. The method is an iterative one that adds or removes an element to or from the set at each iteration. After each addition or deletion, the parameters of the model instance are estimated and recorded. The new parameter values are used to determine the instance's energy and are often used by a likelihood function which determines the probabilities of adding candidate elements to the set. For example, a straight edge VAPOR instance which collects edge points from the image (pixels of high gradient magnitude) might estimate the parameters of a line passing through the set of points. The likelihood function for an edge point, p, could be the reciprocal of the distance of p from the estimated line, so edge points lying nearest to the line are the most likely to be added to the set. The other edge points may still be chosen by the non-deterministic selection procedure, but they have lower likelihood.

2.2 Minimum description length

In order for competing VAPOR models to be compared, we need some kind of metric to measure which model is more appropriate for any given data. Since our model, by definition, has a related energy function, it would be advantageous to use that energy function as the metric for comparing instances of different model types. The energy represents the amount of deviation from the ideal appearance of the object. The problem then is to find a *common scale* for comparing the deviations of all the VAPOR models. This is not an easy problem. Consider comparing a lion without a tail against a table missing one leg (even though it is very unlikely that any given scene could lead to both of these descriptions being compared). Are these equivalent variations (deviations)?

Since we must choose the most appropriate model using the energy function, the energy function must measure the complexity of the model as well as how well the data fits the model. If model complexity were not a factor, then models with more degrees of freedom would always be chosen over simpler models, because the fit to the data would be better. The theory of minimum description length [8] incorporates both factors by using the length of the description of the data in terms of the model.

Rissanen defines the description length of a model and the data it describes as "the total number of bi-

nary digits required to rewrite the observed data, when each observation is given with some precision." Rissanen mathematically defines the length as a sum of two terms for fixed families of models:

$$L(\mathbf{x}, \theta) = -\log_2 P(\mathbf{x}|\theta) + L(\theta)$$

where $\mathbf{x}$ is the observed data, θ is the model in the form of a vector of k parameters, and $L(\theta)$ is a term expressing the complexity of the model θ as the number of bits necessary to represent the k parameters of the model. The first term on the right side of the equation expresses how well the data fit the model; when the probability of the data given the model is low, the first term grows large. The negative binary logarithm of the probability yields the smallest number of bits required to encode the data "relative to the assumed statistical model of the data" [8]. The second term quantifies the number of bits needed to represent the model alone. The model is usually a collection of parameters that are represented by integers within a specific range or reals with a certain precision. Rissanen developed a universal prior for quantifying the number of bits needed for both types of parameter.

VAPOR models can use description length as their energy function, even though their shapes may have numbers of degrees of freedom that approach infinity. This is possible for two reasons. Most variable objects can be modeled with a finite number of parameters and a set of variations that allows an infinite number of shapes. Secondly, the finite resolution of the data means that only a finite subset of the infinite number of appearances are perceivable.

2.3 Implementation notes

To put minimum description length theory into practice, the probability densities, $P(\mathbf{x}|\theta)$, and the length of the model, $L(\theta)$, must be defined for each model. To specify the description length of a variable object, a minimal set of parameters that fully characterize the object must be defined. The conditional probability is what determines the "cost" of the variations from the ideal model.

One important aspect of the conditional probabilities used for the description length computation is that they do not need to be normalized so that the sum of $P(x_i|\theta)$ over all possible values of x_i is 1. This is because there may be multiple possible values for x_i that all have a probability of 1 (e.g. the positions of the vertices of an n-gon). In effect, relaxing this normalization constraint permits the object to have multiple appearances that are all equivalent to the ideal appearance of the object.

In our implementation, the description length is the number of bits required to represent the essential parameters of the model. A VAPOR model can have other parameters which are used only for the convenience of computing the conditional probabilities.

In order to use minimum description length, the data must be partitioned into groups such that all the data in a group belong to the same model. The description length of each group and its model are summed to compute the description length of the entire data set. In order to find the minimal length description, *all* possible partitionings and labelings must be explored. In general, our computational resources are insufficient for an exhaustive search of such a huge search space, so other methods must be used.

Using a "divide and conquer" strategy causes problems because description length should be computed globally while local interpretations of regions of data will not always form a complete, mutually exclusive partitioning of all the data. We can partially solve this problem by assuming that each VAPOR instance covers a part of the total data set and that all the rest of the data is modeled by some simple default model. The default model is the model of the "background" on which the objects will be seen. The background, of course, is not always known a priori, but all that is necessary for our purposes is a simple model of "uninteresting" data such as uniform color or gray level. Using the background model, VAPOR instances increase in size until the energy caused by modeling more image data with the VAPOR model is higher than that of modeling image data with the background model. We refer to these two components as the internal and the background energies, respectively. The sum of the two energies allows different vapor instances to be compared on the same data set (i.e. the entire image). To compute these quantities for every VAPOR instance, we require each instance to maintain a parameter named `area` that represents the region in the image that is considered internal to the instance.

3 SYNTHETIC DATA EXAMPLE

To illustrate VAPOR models we will describe how the system works in a simple synthetic domain. The domain consists of images containing combinations of uniform gray level polygons and circles. We use VAPOR models for polygons, circles, their combinations, and their constituent pieces. The MOSS search procedure described in [3] will be used to find instances of these models.

3.1 Part/whole hierarchy

The VAPOR models are organized by a part/whole hierarchy that makes explicit the relationships between the different types of models. Each model's `component parts` and `parameters` are marked by special fonts when used in this text. At the highest level we have a VAPOR model for scenes of polygons and circles. The scene model's `component parts` are, of course, polygons and circles. The polygons and circles are composed of straight edges and circular edges, respectively. Note that a circle normally only has a single circular edge, but we assume that the data are noisy and several concentric circular edge fragments could be grouped together to form a circle. The models for straight and circular edges are composed of edge points which are pixels in the image that are local maxima of gradient magnitude.

The edge points are given as input to the search; they are not created during the search process.

The straight edge VAPOR model collects edge points that are collinear. Similarly, the circular edge model collects edge points that are co-circular. Two parameters called `line` and `circle` hold the values defining the attributes for the straight and circular edge models, respectively. The parameters are estimated from the set of edge points after each addition or removal of an edge point from the set. All of the models have an `area` parameter and a *background is constant* energy component as described in Section 2.3. An `overlaps` parameter represents the partial ordering of the elements of the scene model. If one element of the scene occludes another element, the overlap relationship is recorded in this parameter. For more detail on the implmentation of these models, please see [3].

3.2 Example 1

Figure 1 shows a synthetic 100 × 100 image of three equal size circles with a wedge shaped section missing from each one. Kanizsa showed that a similar figure produces a *subjective contour* of a triangle that occludes parts of the circles (the wedges) [6]. Since the triangle has the same gray level as the background, no edges appear between the triangle and the background, only between the triangle and the circles.

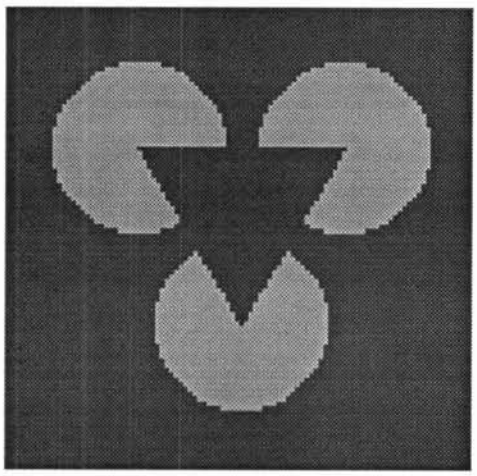


Figure 1: Test image 1: Three large, deformed circles

Our experiments show that the minimal length interpretation of Figure 1 is a (3 sided) polygon occluding three circles. After applying non-maximum suppression to the gradient values produced by the Sobel gradient operator, there were 426 edge pixels. These edge pixels were given to the MOSS procedure as edge point instances.

The procedure chooses an edge point instance having minimum energy and tries one of the models that uses edge points. The search procedure can take several actions, building a parent instance, building a child instance, building a sibling instance, splitting the current instance, or temporarily abandoning the instance to work or more promising instances or to let the instance become part of some existing, partial instance. In this way the procedure builds a part/whole instance graph.

In the case of Figure 1 a polygon instance collects the edges of the subjective triangle (3 subjective from 6 real straight edges – 3 pairs of collinear edges) and finds a model instance that has lower energy than the background model (even though its energy is non-zero due to the lack of complete edges along its perimeter). The scene instance collects the circle instances and the polygon instance into its `scene set` and determines that the polygon occludes the circles because that interpretation has lower energy. This is the final result of the search.

If we change the parameter controlling the description length of the background energy (namely increasing the standard deviation of the "uniform gray levels"), MOSS interprets the image as three deformed circles with no occluding polygon. Thus, the point at which the occluding triangle interpretation has shorter description length is controlled by the relative costs of the background and object models.

4 REAL DATA EXAMPLE

Cloverleaf interchanges in road networks are a good example of a structure with a variable appearance and a variable number of parts. At cloverleaf interchanges, n roads meet ($2 <= n <= 5$?) and there exist r ramps ($1 <= r <= 2(n-1)2n$) connecting them in a restricted topology. Figure 2 shows some of the many possible configurations of cloverleaf interchanges. The various different kinds of cloverleaf interchanges could be enumerated by the numbers and types of their components and then modeled with several different, deformable models (such as standard snakes), but the power of the VAPOR model allows us to use a single model definition.

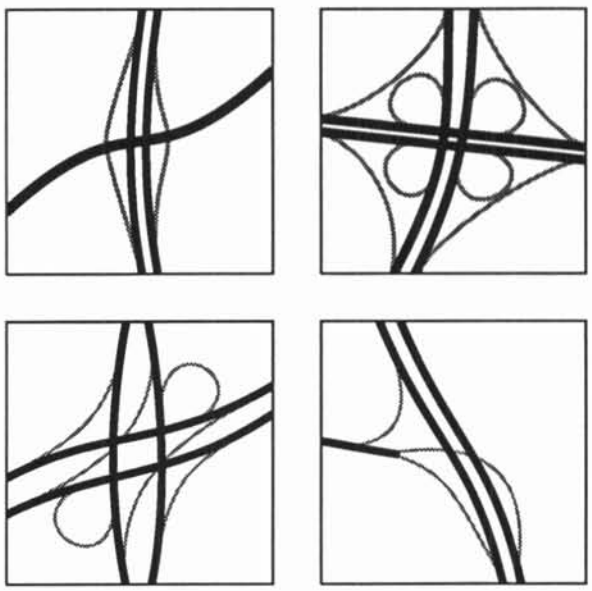


Figure 2: Some types of cloverleaf interchanges

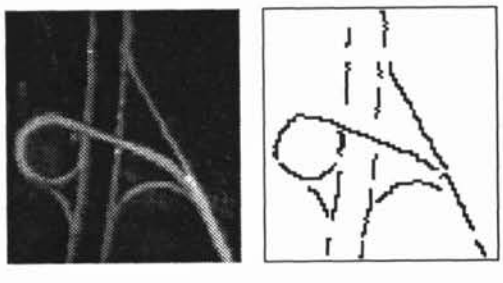
In our model, the long, intersecting roads will be called the axes of the interchange while the short roads that provide the connections between the axes will be called the ramps (shown as black and dark gray, respectively, in Figure 2). We define a part/whole hierarchy of VAPOR models for this domain like we did for the circles and polygons, but this time the inputs are curvilinear feature fragments [2]. The part/whole hierarchy has models for divided highways, roads (undivided highways), junctions, cloverleaf interchanges, and the input curvilinear features (CLF).

Figure 3a is an 80 × 88 window of an image generated by NASA's thematic mapper simulator. The original image is a seven band image, and the monochrome image used here is the result of a weighted sum of bands 1 and 5. Figure 3b shows the 28 CLFs extracted from Figure 3b (adjacent and overlapping segments appear as a single segment in the figure).

When we run MOSS to search for an instance of the cloverleaf model using at least 80% of these CLFs, the final result is the cloverleaf instance shown in Figure 4. The solid black lines indicate the highways labeled as the cloverleaf interchange axes, while the ramps are shown in dark gray with arrows indicating their traffic direction (assuming a right-hand driving rule). Some of the arrows are obscured by the black roads. All roads are drawn with a width of one pixel, since their true widths are not estimated. The circle indicates the center of the cloverleaf interchange, and the thin black line within the circle shows the estimated tangent for the divided highway at the cloverleaf center; a similar tangent line for the crossing road is obscured by that road. The light gray region indicates the pixels that are part of the cloverleaf model's `area` parameter.

The breakdown of the energy components shown in Figure 4 reveals that the the model instance fully satisfies the following constraints: *axes meet at junction*, *ramp directions are consistent*, *sensible continuations*, and *no redundant BCLFs*. There is, of course, some cost for the number of bits needed to represent the model (*cloverleaf base cost*), the costs of the component roads and junctions (*low road and junction energy*), and the cost of representing the rest of the image as a constant (*outside is constant*). The *axes are fully interconnected* constraint is not fully satisfied because the model shows that only four pairs of traffic directions are linked while up to eight links are possible (the more horizontal axis road is assumed to continue on to the left and thus is missing connections). The *high speed interchange* cost is non-zero because both connections of the leftmost ramp to the horizontal and vertical axis roads form angles greater than 45°. The five figure costs of these "insignificant" deviations may seem high on an absolute scale, but they are tiny compared to the background energy term. One of the ramps (the one in the lower right corner) is incorrectly labeled and should be part of the more horizontal axis road. This is due to the fact that the part of the axis road that crosses the divided highway is not the best extension of the mislabeled ramp; the ramp above it is almost collinear. The mislabeled ramp, however, does not greatly affect the cloverleaf interchange model.

The overall energy of the cloverleaf instance totals 1,126,815.0 which is less than the background reference energy, 1,292,064.9 (i.e. the cost of modeling the entire image as a constant gray level modified by additive white Gaussian noise with a standard deviation of 1 gray level). The difference may seem small, but that is due to the fact that the background model covers most of the image (about 75% for this instance of the model) and greatly outweighs the foreground components.



(a) (b)

Figure 3: Test image 2: Cloverleaf interchange (a) and its extracted CLFs (b)

CLOVERLEAF-43

CLOVERLEAF-BASE-COST	640.0
AXES-MEET-AT-JUNCTION	0.0
AXES-ARE-FULLY-INTERCONNECTED	41970.9
HIGH-SPEED-INTERCHANGE	51425.5
RAMP-DIRECTIONS-ARE-CONSISTENT	0.0
SENSIBLE-CONTINUATIONS	0.0
LOW-ROAD-AND-JUNCTION-ENERGY	58958.2
NO-REDUNDANT-BCLFS	0.0
OUTSIDE-IS-CONSTANT	973820.4
total =	1126815.0

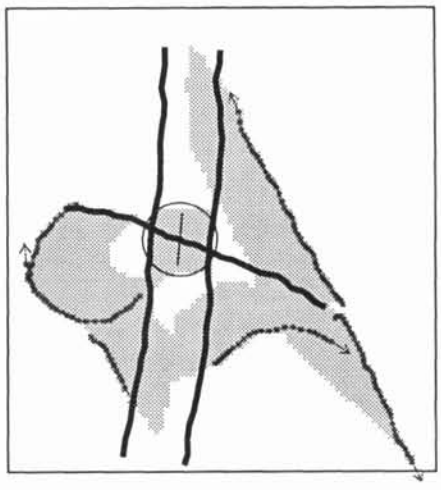


Figure 4: Interpretation for test image 2

5 CONCLUSIONS

The VAPOR model is a new representation that can be used to represent the variable appearances of a single object or object class. Objects with varying shape and number of parts can be successfully recognized using this model. The energy function is defined as the description length of the data in terms of the model. This allows the energy function for different kinds of objects to be measured on the same scale and, as a result, used to decide which model is better.

References

[1] R. A. Brooks, "Symbolic reasoning among 3-D models and 2-D images," *Artificial Intelligence*, vol. 17, pp. 285-348, 1981

[2] J. Canning, J. Kim, and A. Rosenfeld, "Symbolic pixel labeling for curvilinear feature detection," *Proc. DARPA Image Understanding Workshop*, pp. 242-256, 1987

[3] J. Canning, "Recognizing Objects with Variable APpearance - the VAPOR System," Tech. Rep. CAR-TR-563, CS-TR-2705, Center for Automation Research, Jul. 1991

[4] P. Fua and A. J. Hanson, "Objective functions for feature discrimination: applications to semiautomated and automated feature extraction," *Proc. DARPA Image Understanding Workshop*, pp. 676-694, 1989

[5] P. Fua and A. J. Hanson, "Objective functions for feature discrimination," *Proc. Int. Joint Conf. Artificial Intell.*, pp. 1596-1603, 1989

[6] G. Kanizsa, "Subjective contours", *Scientific American*, vol. 234, pp. 48-52, 1976

[7] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *Int. J. of Comput. Vision*, vol. 1, pp. 321-331, Jan. 1988

[8] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics*, vol. 11, pp. 416-431, 1983

[9] J. Segen, "Model learning and recognition of nonrigid objects," *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Patt. Recogn.*, pp. 597-602, 1989

[10] D. Terzopoulos, A. Witkin, and M. Kass, "Symmetry-seeking models and 3D object reconstruction," *Int. J. of Comput. Vision*, vol. 1, pp. 211-221, Oct. 1987