Rochester Institute of Technology
RIT Digital Institutional Repository

Theses

2005

Guiding object recognition: a shape model with co-activation networks

Timothy Lebo

Follow this and additional works at: https://repository.rit.edu/theses

Recommended Citation
Lebo, Timothy, "Guiding object recognition: a shape model with co-activation networks" (2005). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact repository@rit.edu.

Guiding Object Recognition: A Shape Model with Co-activation Networks

Thesis report submitted in partial fulfillment of the requirements for the degree

Master of Science in Computer Science
at the
Rochester Institute of Technology

July 2005

Timothy M. Lebo

Advisor: Dr. Roger S. Gaborski _____

Reader: Dr. Andrew M. Herbert _____

Observer: Dr. Peter G. Anderson _____

Contents

List of Figures

List of Algorithms

1 Introduction

The goal of image understanding research is to develop techniques to automatically extract meaningful information from a population of images. This abstract goal manifests itself in a variety of application domains. Video understanding is a natural extension of image understanding. Many video understanding algorithms apply static-image algorithms to successive frames to identify patterns of consistency. This consumes a significant amount of irrelevant computation and may have erroneous results because static algorithms are not designed to indicate corresponding pixel locations between frames. Video is more than a collection of images, it is an ordered collection of images that exhibits temporal coherence, which is an additional feature like edges, colors, and textures. Motion information provides another level of visual information that can not be obtained from an isolated image. Leveraging motion cues prevents an algorithm from "starting fresh" at each frame by focusing the region of attention. This approach is analogous to the attentional system of the human visual system. Relying on motion information alone is insufficient due to the aperture problem, where local motion information is ambiguous in at least one direction. Consequently, motion cues only provide leading and trailing motion edges and bottom-up approaches using gradient or region properties to complete moving regions are limited.

Object recognition facilitates higher-level processing and is an integral component of image understanding. We present a components-based object detection and localization algorithm for static images. We show how this same system provides top-down segmentation for the detected object. We present a detailed analysis of the model dynamics during the localization process. This analysis shows consistent behavior in response to a variety of input, permitting model reduction and a substantial speed increase with little or no performance degradation. We present four specific enhancements to reduce false positives when instances of the target category are not present. First, a one-shot rule is used to discount coincident secondary hypotheses. Next, we demonstrate that the use of an entire shape model is inappropriate to localize any single instance and introduce the use of co-activation networks to represent the appropriate component relations for a particular recognition context. Next, we describe how the co-activation network can be combined with motion cues to overcome the aperture problem by providing context-specific, top-down shape information to achieve detection and segmentation in video. Finally, we present discriminating features arising from these enhancements and apply supervised learning techniques to embody the informational contribution of each approach to associate a confidence measure with each detection.

2 Image Understanding Recipe

Humans have little difficulty understanding images when presented in the suitable format. We have an innate ability to process and understand visual information. Most of the human brain is devoted to visual processing, so it is no surprise information presented visually is more easily understood. This understanding is so effortless it may be difficult to understand why computers are not capable of interpreting images as well as humans. This difficulty becomes apparent when forced to interact with images in the only format available to a computer: a sequence of numbers. At this level, images become an indiscernible sea of confusion.

An image understanding algorithm manipulates this sequence of numbers to postulate higher-level meaning. The construction of most image understanding tools is relatively similar. Each approach follows the same general steps with particular choices for each component. The components of an image understanding tool include representation via decomposition and classification. Successful systems are those that comprise a good combination of these components for a particular problem domain. This section will present the "Image Understanding Recipe", an outline of the options available when creating an image understanding tool.

2.1 Pattern Classification

Statistical analysis [20, 26] is a well established field that successfully addresses problems characterized by numbers. This methodology views signals from the world as random variables. Signals are usually multidimensional, indicating responses from a battery of sensors. This section describes how an image understanding problem is framed within the statistical analysis methodology. Our specific interest, object detection, is posed as a signal detection problem by identifying when the object signal is present in the noise signal.

2.1.1 Choose Pattern Representation

Choices made for the representative signal will affect the final accuracy and performance of the image understanding system. The signal representation should be considered carefully because it ultimately determines separability. There are many potential choices for this decision.

All of the various choices for image representation are available to create the signal representing the object. An image may be represented in a variety of signals. At the lowest level, and image may be represented with a sensor for each pixel. This would create an $m \times n \times b$-dimensional signal representing the color at each location on the imaging surface. An alternative representation for each pixel is the HSV color space. This represents hue, saturation, and value for each pixel. Image information may also be transformed to the

Fourier domain via a linear transformation. The Wavelet Transform may also be used. There is a key property of these representations that is not often explicitly addressed. They are *equivalent representations of the same information.* RGB is converted to HSV via a linear transformation. RGB is converted to the Fourier domain via a linear transformation. The RGB representation may be converted to and recovered from the Wavelet transformation without a loss of information. Thus, the image data may assume a variety of forms without changing what it is: image data. RGB and HSV are commonly used since their spatial representation is more intuitive. However, each representation has an advantage in particular contexts by describing the information in a more natural way. Since each representation is equivalent, it is appropriate to use any for the signal representation.

Because the size of the signal is large when all image information is represented, it is desirable to reduce the dimensionality for efficacy and tractability purposes. Much of the information in the raw image data is redundant. This is expressed by the assumption of continuity [38]. Identifying decomposition techniques to extract distinguishing features can suppress irrelevant information, facilitate separability, and reduce the size of the representation. Reducing the representation size reduces computation time and storage space. Decomposing a signal vector with a decomposition technique produces a feature vector that becomes a new representation of the object. A feature is a random variable describing a particular characteristic of an observation. This process of resolving the constituent parts of a compound into its elementary parts may significantly aid the classification process. Successful results rely on careful consideration of which constituent parts to emphasize. All of the various choices for image decomposition are available to create the feature vector representing the object. Some of these are described in Section 2.2.

2.1.2 Partition Data

The process of pattern classification creates a working knowledge of the sampled world and uses this to distinguish between unknown observations. Both the working knowledge and novel information are represented solely by the chosen feature vector representation.

When creating a classification problem, the designer must learn as much about the data as possible. This understanding will suggest which representations and learning techniques should be used. An understanding of how the data were collected provides a foundation for the classification process. The designer should understand the phenomenology and be able to describe what characteristics will be expected. A description of the noise characteristics should also be addressed. The first and second order statistics aid in understanding the distributions of the data and help determine if a simple gaussian assumption is appropriate. The chi squared test is also useful in this evaluation. If the data are multi-variate, an investigation of the marginal distributions can aid in understanding the distribution. Knowing whether the data represents a uni-modal or multi-modal distribution and identifying existing outliers are also useful pieces of information.

Supervised learning is a common form of pattern classification. With this form, samples are provided along with the class from which it was obtained. This allows samples to be partitioned into each class for further investigation. If sufficient data samples are available, separation into training, validation, and testing sets is appropriate. The classifier creates the discriminant functions based on the training data while the classification efficacy is evaluated with the validation set. The classifier is not created from the validation set, so its performance with the samples in the validation set are a prediction of the performance on unknown data. Partitions of the data are also useful for boosting techniques, which are discussed in Section 2.1.6.

2.1.3 Choose Classifier

Once random variables have been defined, we can use a variety of learning techniques to distinguish between two or more populations [16]. Making choices based on Bayesian Decision Theory will minimize classification error. The least error is obtained by choosing the class with the largest posterior probability, given by Bayes Formula:

$$p(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)} \quad \text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}} \tag{1}$$

Unfortunately, this requires full knowledge of the conditional probability distributions and the prior probabilities of each class, which is not usually available or accurately estimated. Specific parametric distributions may be estimated using the provided samples by finding the parameters that maximizes the probability of the sample data. This maximization may be determined by Maximum Likelihood or Bayesian Learning. If the data do not fit the chosen parametric model, this approach is inappropriate and will provide poor results.

Non-parametric techniques may also be used to estimate the conditional probability distributions to facilitate classification. The value of the probability distribution at a particular location may be estimated by finding the number of samples surrounding the location and the volume the samples occupy. The Parzen window technique fixes the volume size and finds the number of samples within the volume. The K_n Nearest Neighbor technique fixes the number of neighbors and expands the volume until subsuming the fixed number of samples. In both techniques, the probability is found by

$$p(\omega_i|x) \simeq \frac{k/n}{V} \tag{2}$$

where k is the number of samples in the volume V and n is the total number of samples in the class ω_i. A technique similar to the K_n Nearest Neighbor distribution estimation is the k-Nearest Neighbor classification. The k closest neighbors of the test observation are found, and the test point is assigned the identity of the majority of its neighborhood. The

value of k is usually odd to avoid ties and the case of $k = 1$ is known as the *nearest neighbor* technique.

Generalized Linear Discriminants (GLDs) are an alternative technique that specifies the form of the discriminant function instead of fitting the sample data to a specific or arbitrary probability distribution. GLDs use geometric properties of linear algebra to describe the distribution and perform classification. The discriminant function takes the form of a hyper-plane that partitions the universe into two regions, each of which represent the residence of a single class. The hyper-plane is represented by a normal vector. The sign of the inner product between the hyper-plane normal and the test sample indicates the class assignment, since the sign indicates which side of the plane the sample resides. The separating hyper-plane can be found by gradient descent of a chosen criterion function and can act on the original feature representation or a larger-degree transformation of the feature representation. These larger-degree transformations, known as Φ functions, allow nonlinear boundaries of the original feature representation by finding a linear separation in the higher dimension. The Support Vector Machine (SVM), a relatively new and popular classification technique, finds the separating hyper-plane that maximizes the distance between the separating hyper-plane and the data samples. A parametric Φ function for the SVM is chosen by hand, which may not be ideal for any particular problem space. Neural networks are a natural extension to GLDs and determine an appropriate Φ function to transform the representation that achieves separability.

Evolutionary techniques such as Boltzman Learning, Simulated Annealing, Genetic Algorithms, and Genetic Programming may also be used to identify approximate solutions to the separation task. These techniques are modeled after various physical and biological processes observed in nature.

2.1.4 Obtain Baseline Error Rates

It is beneficial to have a general idea how much error to expect from the chosen representation and classifier. If certain distributions are assumed, the theoretical Chernoff or Bhattacharyya Bounds are idealistic estimates for expected error. Additional assumptions, such as a diagonal covariance, may be used to simplify the expected error estimation.

2.1.5 Refine Feature Selection

After initial attempts at classification, feature selection refinement is appropriate. Using subsets of the original features may allow more accurate estimates of parametric distributions. This also allows faster learning and classification, since the runtime is a function of the dimensionality of the feature vector. Principle Component Analysis or Multiple Discriminant Analysis may be an appropriate technique to identify dominant feature components that most concisely represent the data. Choosing an entirely new representation is also an

option. These representations are discussed in Section 2.1.1.

2.1.6 Consider Other or Combination of Classifiers

Each classifier has specific strengths and weaknesses with particular data population characteristics. When facing poor results, considering other classifiers is appropriate. Combining additional types of classifiers may also be an alternative. Allowing each classifier to classify the test sample and combining the results of all classifiers can increase accuracy. Boosting techniques [16] may also be used to classify samples that do not fit within general trends. This process involves creating subsequent classifiers to address the misclassified samples of previous classifiers. This allows the ability to classify nonlinear population behavior, although over-learning becomes a concern. Although the feature choices and learning techniques may be addressed independently, it is important to emphasize that the interaction between the choices for each influence the resulting performance.

2.1.7 Evaluate Performance

Once the choices have been made for the representation and learning technique, the system is tested within the intended problem domain. Error estimates and modifications are made empirically and parameters are customized to suit the specific domain.

2.2 Digital Image Processing Techniques

2.2.1 Normalized Grayscale Correlation

Normalized Grayscale Correlation (NGC) is a method to measure similarity of two image patches. For two ordered data sets p and q of length n, the NGC is given by

$$\mathrm{NGC}(p,q) = \frac{\sum_{i=1}^{n}(p_i - \bar{p}_i)(q_i - \bar{q}_i)}{\sqrt{\sum_{i=1}^{n}(p - \bar{p}_i)^2 \sum_{i=1}^{n}(q_i - \bar{q}_i)^2}} \tag{3}$$

The two image patches may be vectorized and concatenated to obtain a $n^2 \times 2$ observation matrix. The correlation may be illustrated with a scatter plot showing the values of the corresponding pixels. A high correspondence will be expressed by points along the identity function. Figure 1 illustrates this conceptualization. The NGC provides an intuitive evaluation of the similarity, as it varies in the range $[-1, 1]$ with 1 indicating identical and -1 indicating exactly opposite. Figure 2 shows two patches producing a high NGC value. This

metric incorporates spatial information only in the sense that the aligned pixels are compared. The pairings may be rearranged in an identical manner and retain the same NGC value.

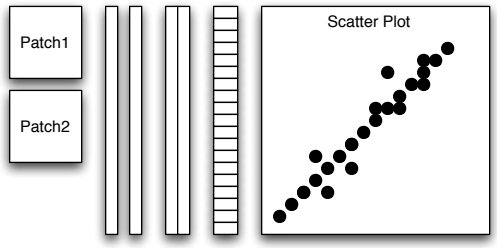
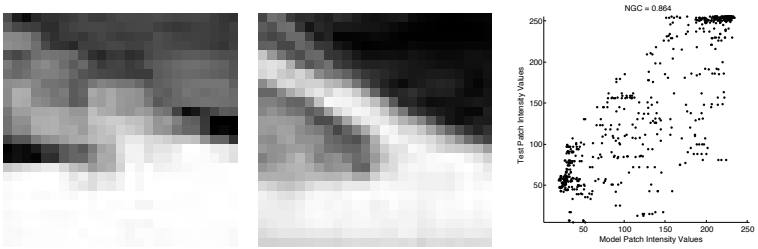


Figure 1. Reforming image patches into a two-dimensional observation matrix



(a) Image Patch $\vec{\mathbf{m}}_1$ (b) Image Patch $\vec{\mathbf{m}}_2$ (c) Scatterplot of gray values in $\vec{\mathbf{m}}_1$ and $\vec{\mathbf{m}}_2$

Figure 2. NGC of two similar patches

2.2.2 Convolution and Filtering

The NGC provides a measure of similarity between two image patches of the same size. When one image is significantly larger than another, the smaller patch may be compared to all unique sub-images of the larger image. This will provide a third image that indicates the NGC of the small patch with each sub-image of the larger image. This operation is formalized by the convolution operator [50]. The small patch is referred to as the kernel. The response of a kernel to an image neighborhood is proportional to how similar it is to the neighborhood. This behavior can be leveraged to search the image for particular local structures by designing a kernel resembling the structure of interest. Kernels formed from the Difference of Gaussians (DoG) are used to find points of high contrast. Kernels formed from sine-modulated DoGs, called Gabor filters, are used to determine points on edges of a particular orientation.

2.2.3 Gradient Estimation

The gradient of an image is motivated by the gradient operator of a bivariate function, which finds the partial derivative with respect to each variable and scales the associated unit vector for the corresponding dimension:

$$\nabla\phi(x,y) = \frac{\partial\phi}{\partial x}\hat{x} + \frac{\partial\phi}{\partial y}\hat{y} \tag{4}$$

The result is a vector within the plane that indicates the direction and magnitude of change. Viewing the bivariate function ϕ as a sampled image I and constructing the convolution kernel

$$ker = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

to reflect the computations desired to approximate the derivative, the components of $\nabla I(x,y)$ may be approximated by

$$\frac{\partial I}{\partial x}\hat{x} \simeq I(x,y) * ker(x,y) = I_x \qquad \frac{\partial I}{\partial y}\hat{y} \simeq I(x,y) * ker(x,y)^T = I_y$$

Where $I_x(x,y)$ and $I_y(x,y)$ at the locations (x,y) represent the magnitude of $\nabla I(x,y)$ in the x and y directions, respectively. Since $\hat{x}$ and $\hat{y}$ are unit vectors in two dimensions, $\nabla I(x,y)$ is also a vector. The angular direction of a vector may be determined by the inverse tangent of the ratio of the comprising components. The magnitude of a vector is determined by the square root of the individual d components squared. Thus, in our bivariate case applied to image data:

$$\text{Magnitude} : m(x,y) = \sqrt{I_x(x,y)^2 + I_y(x,y)^2} \tag{5}$$

$$\text{Orientation} : \theta(x,y) = \tan^{-1}\left(\frac{I_y(x,y)^2}{I_x(x,y)^2}\right) \tag{6}$$

2.2.4 Corner Detection

Points with high intensity change, indicated by the magnitude of the gradient, may be considered interesting because it exhibits a change in continuity. Corners are a subset of these points that exhibit large change in both directions. Harris [22] observed that the size of the magnitude alone was insufficient to indicate adequate corners. The Hessian matrix can be formed from the second partial derivatives at a pixel location:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 I}{\partial^2 x} & \frac{\partial \frac{\partial I}{\partial x}}{\partial y} \\ \frac{\partial \frac{\partial I}{\partial y}}{\partial x} & \frac{\partial^2 I}{\partial^2 y} \end{bmatrix} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \tag{7}$$

Since the eigenvalues of a linear system describe the principle curvature of the system, Harris observed that the relative sizes of the eigenvectors of the Hessian could be used to identify corner locations. Harris also showed how to avoid explicitly computing the eigenvectors. When the matrix H is viewed as a function mapping a domain to a range, the eigenvectors are the subset of the domain that are mapped to scaled versions of themselves. The eigenvectors can be found by the solution to the equation

$$\mathbf{Hx} = \lambda\mathbf{x} \tag{8}$$

$$|\Lambda I - \mathbf{H}| = 0 \tag{9}$$

The scalar amount by which each eigenvector is scaled is the corresponding eigenvalue. For a matrix $\mathbf{H}$ in $\mathbb{R}^{2\times 2}$, we can denote the eigenvalues of $\mathbf{H}$ as $\sigma(\mathbf{H}) = \{\alpha, \beta\}$ and assume $\mathbf{H}$ is positive definite. From linear algebra, we know the sum of the eigenvalues of $\mathbf{H}$ is equal to the trace of $\mathbf{H}$, $Tr(\mathbf{H}) = \alpha + \beta$, and the determinant of $\mathbf{H}$ is the product of the eigenvalues $|\mathbf{H}| = \alpha \times \beta$. Formulating this knowledge with the image representation in (7), we obtain

$$Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta \quad \text{and} \quad Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \tag{10}$$

The ratio of the eigenvalues may be represented as $\frac{\alpha}{\beta} = r$ and the following may be derived:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H}} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{11}$$

which is at a minimum when the two eigenvalues are equal and increases with r. The value r may be used as a threshold value to indicate how much curvature at the point must be present.

2.2.5 Multi-Scale Corner Detection

Lowe [37] points out that the Harris corner detector is very sensitive to changes in image scale. Although the Harris algorithm identifies localizable locations with high gradient in more than one direction, the scale at which this is performed is a parameter to the algorithm and directly affects which scales are searched. Lindeberg [35] presents work to identify appropriate and consistent scales for feature detection and describes it as a problem of scale selection. Witkin [57] describes the entity of the convolution of an image with all filter sizes as a scale space:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{12}$$

where $G(x, y, \sigma)$ is the bivariate spatial Gaussian with a standard deviation of σ. Lowe identifies local maxima and minima of the difference of the scale space evaluated at different σ values separated by a multiplicative factor. Lowe constructs this as a close approximation to the scale-normalized Laplacian of Gaussian studied by Lindeberg. Lindeberg showed

that $\sigma^2\nabla^2 G$ produces more stable image features compared to image functions such as the gradient, Hessian, or Harris corner detector. The maxima and minima are retained if the location is larger than its neighbors in the lower, current, and larger scale DoGs. Lowe mentions that maxima may be arbitrarily close together but are unstable to perturbations in the image if they are too close. Since the DoG also identifies locations on edges, which are poorly localized and unstable, it is also desirable to remove these. Lowe finds the principle curvature [37] at the point by computing the 2×2 Hessian matrix in the scale of the keypoint, similar to Harris.

3 Prior Work

Detection, recognition, and segmentation algorithms rely on some description of the target category appearance. This is embodied in an object model. Creating a reliable model is often difficult because instances from the same target class have a wide range of variations in appearance. These variations cause algorithms to misinterpret new images that are similar to modeled samples. **Noise** applied to a learned sample may affect behavior since the values are modified in an unpredictable way. Noise distribution parameters may be assumed to account for these variations. With a Gaussian assumption, if the testing value is within a certain variance threshold of the learned value, the test value is assumed to match. **Translations** of an object are usually not a difficulty since shape and size are preserved and their quantifying metrics are consistent. If the target is smaller than the input image, each sub-image may be compared to the learned samples. **Affine transformations** of a learned image, which modify the scale and rotation of an object, are more difficult to handle. **Clutter** in an image is problematic since it provides additional features that may lead to false matches. Clutter may be viewed as organized noise because it exhibits structure that is not part of the target. **Occlusion** of an object by other foreground objects prevents an algorithm from obtaining a uniform sampling of the object of interest. This may cause variants that are not accounted for, failing to obtain a holistic sense of the objects presence. **Deformable**, non-rigid, objects also cause problems since they change shape but maintain the same identity. **Illumination** and shadows cause an inconsistency across the object that can be misinterpreted as an object boundary. **Viewpoint** also changes the appearance of the object.

The goal is to represent the target class in a manner that is invariant to these intraclass variations but at the same time distinguishes instances of the object class from other images [1]. There are a variety of representational choices for object detection [53, 56]. In the simplest form, recognition may be performed at the pixel level. This level of recognition is justified for applications such as remote sensing, where parts of the earth are classified by terrain type. Each pixel can be independently classified according to the techniques described in Section 2.1 because non-local structures are mostly irrelevant. Ignoring local structure can be viewed as incorporating information from a neighborhood of size zero. Detection in most other applications, including our own, involves more information than the value at a particular pixel. The spatial relation with surrounding pixels becomes very important.

The model for an object may assume a two or three dimensional form. Object-centered models represent the spatial arrangements among parts in a three-dimensional coordinate system that is centered on the object itself. In this case, the detection becomes an alignment problem. Recognition by Components is a prominent object-centered model. View-based models form collections of view-specific features from previously seen objects. This two dimensional form is viewpoint-based and involves matching known views with the novel input [49].

Representations with large neighborhoods capture a holistic appearance while represen-

tations with smaller neighborhoods capture the appearance of the parts. It is often difficult to obtain a complete match for each object using a holistic model due to occlusion and illumination variations, so some level of component representation is favorable. Lowe [37] uses a local keypoint descriptor that provides illumination, scale, and rotation invariant descriptions of regions surrounding points identified by the approach described in Section 2.2.5. These Scale-Invariant-Feature-Transform (SIFT) keypoints have been shown to be successful on a wide range of image domains, including indoor scenes, outdoor scenes, human faces, aerial photographs, and industrial images. An orientation histogram, representing the dominant direction of the local gradients, is formed from the gradient orientations surrounding the keypoint location at the appropriate scale of the keypoint. Each value is weighted by its gradient magnitude and by a Gaussian-weighed circular window. Lowe demonstrates the scalability to image databases of up to 100,000 keypoint entries and suggests a linear growth with the size of the database. Sivic et al. [19] quantize SIFT-like region descriptors to formulate the recognition problem into a text retrieval problem. The quantized descriptors become the visual vocabulary describing the database.

Object representations vary in their focus on features or geometrical relationships. There is a trade-off between allocating effort to the description of either of these [49]. Carneiro and Jepson [10] indirectly address the use of pairwise geometric constraints for their SIFT-like region descriptors that does not retreat to a Maximum Likelihood approach. Neural networks are one approach in this trade-off space that allows all sub-images to neighbor all other sub-images [4] or the entire image itself [55]. Bileschi [5] uses a components-based approach and pairwise position statistics between component locations to locate parts of the face. Joint Probability Distributions (PDF) of the relations between components are often estimated with Maximum Likelihood (ML) such as [18, 56]. Kersten and Yuille [27] survey the use of Bayesian models for object perception. This approach originates from Hemholtz's notion of unconscious inference and uses Bayesian probability theory in which prior knowledge about visual scenes is combined with image features to infer the most probable interpretation of the image. Kersten and Yuille suggest that human visual perception can be close to ideal for visual tasks of high utility and under visual conditions that approximate those typically encountered. The Bayesian inference of object properties relies on probabilistic descriptions of image features as a function of their causes in the world and their 'prior' descriptions of these causes independent of the images. It is largely an open question of how the human visual system learns the appropriate statistical priors, but some priors and strategies for learning priors may be rooted in our genes [27]. Kersten and Yuille demonstrate that the distribution of the difference in intensity values between pairs of pixels is highly non-Gaussian. Some perceptions may be driven more by prior knowledge and some more by data. The less reliable the image features, the more the perception is influenced by the prior knowledge. This trade-off is seen in visual phenomena. Influence graphs may be constructed to describe the relations between the components of the natural world $S_1, S_2, S_3, ...$ and the features of an image $I_1, I_2, I_3, ...$ The distribution can be expressed as $p(S, I) = p(S_1, S_2, ..., I_1, I_2, ...)$. Such

decompositions may not be straightforward and may be hard to learn. However, the graphical structure of these models often makes it straightforward to map them onto networks for suggested neural implementations. The prior knowledge and likelihood functions are implemented by synaptic weights. Neural mechanisms for representing uncertainty can be realized by population encoding. The Bayesian models are also suggestive of the feedforward and feedback connections known to exist in the visual areas of primates.

An alternative approach creates component classifiers for each part of the object and an additional supervisory classifier to determine presence. Dorko et al. [15] create classifiers for each component and choose the best classifier based on the likelihood ratio or mutual information criteria. This retains classifiers that provide distinctive category information and eliminates redundant classifiers. This approach has been used to detect different target categories. Everingham et al. [17] and Mohan et al. [41] apply it to human detection, Heissele et al. [23] apply it to face detection, and Lueng [33] applies it to car detection. Fergus et al. [18] model rigid and non-rigid objects as flexible constellations of parts and use a probabilistic representation for the shape, appearance, occlusion, and relative scale between the parts. The parameters are learned using maximum-likelihood with a Gaussian assumption. Classification is achieved by Bayesian methods using the learned model and the same Gaussian assumption.

Other object-recognition algorithms work with edges. Mikolajczykto et al. [40] use scale invariant edge detection and apply progressively tighter geometric restrictions. Borenstein et al. [7] partition the original image into regions and combine them to create a unified and consistent whole. Schaffalitzky and Zisserman [48] find geometric groupings of repeated elements for region segmentation. Iqbal and Aggarwal [24, 25] present a technique to group line segments in an image. This technique is applied to identifying images of buildings in a content based image retrieval system based on the "principle of non-accidentalness" of the presence of parallel lines, "L" junctions, and "U" junctions. Burn's straight line detector [9] was used to identify straight line segments. Segments are combined into longer lines by searching near each line segment and combining them if 1) they are pointed in the same direction or 2) the endpoints of the fragments are close. A single representative line replaces the original fragments. "L" junctions are found by identifying fragments that terminate in a small window and have directions that are close to $\frac{\pi}{2}$. "U" junctions are found by aligning two "L" junctions with an additional joining line. Parallel lines are found by grouping all remaining lines that have a similar orientation. The algorithm creates a three-dimensional feature vector representation of these visual aspects. The first dimension represents the proportion of "L" junctions, the second dimension represents the proportion of "U" junctions, and the third dimension represents the proportion of parallel lines. They make a multivariate Gaussian assumption for the Bayesian posterior probabilities. The model parameters μ_i and Σ_i are estimated with maximum likelihood estimation. Discriminant functions are created with these parameters and the largest posterior probability is chosen.

Swain and Ballard [51] demonstrate the use of color histograms to identify and localize

objects of interest. They use a 66 image database to identify the same objects in new images. They present the "what" versus "where" dichotomy of the primate cortex, where the parietal cortex addresses the localization and the temporal cortex addresses the identification. They suggest that performing both concurrently is difficult, and addressing each individually is sufficient. Their Histogram Intersection algorithm compares a histogram of a region of the image with the learned histogram from the database. The Intersection is defined as the sum of the minimum value in each bin. This value is normalized by the volume of the known histogram. This metric tells how many of the pixels in the model histogram are found in the image. They note that most of the information is carried by the largest bins of the histograms and describe a technique to index via this characteristic. They note that the histogram space is substantially large, which allows a capacity to uniquely store many different objects. This technique is largely independent of view and resolution and can be performed without figure-ground segmentation. It is invariant to translation and rotation about an axis perpendicular to the image. Histograms change slowly with rotation about other axes, occlusion, and change of distance. The histogram is used as an equivalence function on the set of possible colors. Because histograms are binary natured, other techniques using Gaussian bins could be used but Swain and Ballard demonstrate simple histograms perform well enough. They use color opponent axes as inspired by the human visual system. However, they also use traditional color axes with equivalent performance.

Dickinson et al. [14] present an algorithm for active object recognition. This domain includes the ability to intelligently change the intrinsic and extrinsic sensor parameters to more effectively solve the vision task. They argue an attentional mechanism aids the search for objects of interest and is good for avoiding a "sweeping window" approach. They use several graph structures to represent the object models in two and three dimensions. They discuss the desire to use distinguishing, or low entropic, features to use in object recognition. They begin with 10 volume structures similar to Biederman's geons. They rotate each of these by 10 degrees and orthographically projected to an image plane. They take advantage of symmetry to reduce the number of different views to 688. Conditional probabilities are then estimated by observing how many times these views appear in the objects. The objects they train on are the same geons that are used to describe the objects. They derive a metric for average inferencing uncertainty to quantify the ability of a view to identify a volume identity. They then use simple segmentation procedures to obtain image regions, which are described by a Minimum-Description-Length using the geon projections. An interpretation tree is used to determine the region classification. They formulate a probabilistic formulation of the object prediction using the object, volume, and aspects of the hypothesis and the target. They present an additional graph representation that predicts view events as a function of camera movement. This allows them to expect what the shape will look like and to make decisions about where to move the camera if the current viewpoint is ambiguous. This graph may be used for object tracking.

4 Physiological Analogue

Frequently, algorithms addressing image data often quickly disregard the fact that the input is visual. The various decomposition techniques provide data points that are classified without a clear understanding of the association between the responses and the results. An understanding of the decomposition and statistical techniques is necessary but not sufficient. Understanding of the human visual system and human perception is instructive in developing computer vision algorithms. We look to the human visual system to understand how humans are successful at interpreting diverse and under-constrained input while providing the tools to test and constrain theories of human object perception [27]. Three supporting operations are discussed in the following sections. First, low-level feature detection is presented. These primitives provide the fundamental building blocks required for subsequent processing. Next, object recognition is discussed in terms of these building blocks. Finally, attention is discussed as a coordinating and guiding process to restrict operations to obtain task-dependent knowledge.

4.1 Early Visual System

The early regions of the visual pathway are responsible for decomposing the retinal input into constituent features. This is achieved by a vast network of independent but cooperating processing units called neurons. Neurons have a branching input system that accept signals from other neurons and output a single signal along their axon. Neurons are found in a layered arrangement. Neurons in any particular layer receive signals from previous layers, interact within the current layer, and transmit signals to the subsequent layer. The neurons in the first layer are connected to the photoreceptors in the retina. These neurons respond to points with high contrast at a range of spatial resolutions and their responses can be modeled by a convolution with a Difference of Gaussian kernel. Neurons that are found later in the visual pathway respond to progressively complex spatial patterns, including points on edges with high contrast and points on oriented edges with high contrast. The neurons that respond to these features are known as simple and complex cells and are found in the primary visual cortex. A computational model reflecting the operations of the early visual system are presented by Koch and Ullman [29].

4.2 Object Recognition

Object recognition is performed by the ventral pathway in the human visual system with a significant contribution from the inferotemporal cortex (IT). The neurons further down the pathway show increasing receptive field sizes and tend to prefer even more complex stimuli. There is considerable evidence that object recognition in primates is based on the detection of local image features of intermediate complexity that are largely invariant to

imaging transformations [36]. Tanaka [28, 52] showed that object recognition makes use of neurons in the IT that respond to features of intermediate complexity. These features are typically invariant to a wide range of changes in location, scale, and illumination, while being very sensitive to particular combinations of local shape, rotation, color, and texture properties. Although some neurons in anterior IT cortex responded to very simple lines or bar features, in most cases the optimal response was obtained by features of intermediate complexity, such as a dark five-sided star shape, a circle with a thin protruding element at a particular orientation, or a green horizontal textured region within a triangle boundary. Some neurons responded only to more complex shapes, such as moderately detailed face or hand images. These intermediate-complexity neurons were often highly sensitive to small variations in shape, such as the degree of rounding of corners or relative lengths of elements. On the other hand, the neurons exhibited a wide range of invariance to other parameters, such as retinal location, size, and contrast. Neurons that were close together in cortex often responded to small variations of the same feature. Tanaka uses the size of the columns and size of the region to estimate 1300 unique feature columns.

Feature responses have been shown to depend on previous visual learning from exposure to specific objects containing the features. Booth and Rolls [6] reported that after 10 plastic objects were placed in a monkey's cage for a period of weeks or months without training, many neurons responded only to particular views of these shapes while exhibiting the usual invariance to large ranges of scale and location. In addition to the usual view-dependent neurons, they found a small population of neurons that responded to any view of a particular object. Experiments with paperclips and monkeys show that the neurons in the IT were tightly shape-tuned to the training objects and responded only to a specific view. In contrast, face cells in IT argue for a distributed representation of the object class with the identify of a face being jointly encoded by the activation pattern over a group of neurons [47].

Riesenhuber and Poggio [46] claim that object recognition performance crucially depends on previous visual experience. They report psychophysical experiments comparing the discrimination performance between subjects who received viewpoint-specific training and those who did not. They found that the visual system is very well able to perceive novel objects even without training; there is a baseline discrimination performance for any novel class. However, extensive experience with an object class builds a representation of that object class that generalizes to unseen class members and facilitates their recognition. Training builds a viewpoint- and class-specific representation that supplements a pre-existing representation. They also note that the advantages of the training did not transfer to angles of rotation beyond a 45° view. They suggest a representative scheme where a group of units, broadly tuned to representatives of the object class, code for the identity of a particular object by their combined activation pattern.

Riesenhuber and Poggio [47] discuss computational and neurophysiological models of object recognition. They emphasize the differences between neuroscience and computer vision when addressing the tasks of identification and categorization. Computer vision is very good

at identification but meets categorization with much more difficulty. Biological investigations show that categorization is suggested to be simpler. Computer vision approaches recognition as a supervised learning problem that trains a classifier with positive and negative examples. Positional invariance can be achieved by affine transformation normalization before the recognition process. Although the debate between the appropriate types of models continues, Riesenhuber and Poggio favor the view-based systems. They focus on view-based models of object recognition and show how they provide a common framework for identification and categorization. The scanning approach in computer vision techniques is unlikely in biological models. A feedback model is favored, where the difference between the guess and the input is continually refined. Unfortunately, the small latency in recognition tasks suggests that this cannot be performed for very long. Riesenhuber and Poggio discuss the hierarchical models and their ability to avoid combinatorial explosion of the number of units in the system. This suggests an over-complete dictionary similar to the computer vision approaches.

4.3 Attention

The early feature decomposition discussed in Section 4.1 is performed in parallel. This characterizes the input in all representations enumerated by the Cartesian product of scale, orientation, and location. The primitives provided in these representations may be used by subsequent visual processing, including the formation of intermediately complex receptive fields found in the IT discussed in Section 4.2. It is known that object recognition in human vision uses a serial process to bind features to object interpretations, determine pose, and segment an object from a cluttered background [54, 58]. This appears to involve the determination of object pose and other parameters, as well as selection and integration of features consistent with these parameters [36]. Since enumerating the Cartesian product of all receptive fields in regions after the visual striate appears to be computationally intractable, higher level processes must compete for a limited computational resource. The brain manages this computational limitation by selectively determining which processing to perform depending on task-related goals. This management manifests itself as a top-down attentional system.

Humans have a strong impression of seeing all surrounding objects simultaneously and in great detail. The representations are coherent and complete. The change blindness phenomenon, the inability to detect changes made during a visual disturbance, argues against the idea that our brains contain a detailed visual buffer representing the entire scene. Evidence shows that only specific, goal-oriented object properties are extracted and remembered during tasks, and revisiting the object is common for identifying new characteristics. The expert-level of the observer also affects their ability to detect change. This was shown with groups that learned mug versus groups that learned Toms mug. The specific was more likely to identified the change. Similar results were found for scenes of football with experts and non-experts.

Rensink's [45] coherence theory of attention describes a dynamic, just-in-time represen-

tation where low-level proto-objects are rapidly, inattentively, and continually formed across the visual field. These proto-objects are volatile and are replaced when any new stimulus appears at their location. Focused attention selects a small number of proto-objects from this constantly-regenerating flux and stabilizes them. Feedback from a higher-level nexus form links to create a coherence field. This field enables a high degree of coherence over space and time. When focused attention is released, the object loses its coherence and dissolves back into its constituent proto-objects. Lifetimes of the coherence fields are quite brief. A structure is endowed with coherence for only as long as attention is directed to it. Rensink argues that attention allocation is coordinated to create a stable object representation whenever needed, achieving a *virtual representation* that allows higher levels to operate as if all objects in the scene are simultaneously represented in detail. This retains all of the power of a full internal buffer, while using much less processing and memory resources. This creates a sort of "time-sharing" of information with access on request. Rather than being the main gateway of all visual perception, Rensink views attention is just one of several concurrent streams, namely the stream concerned with the conscious perception of coherent objects [45].

Focused attention is needed to see change. Under normal circumstances, a change is accompanied by a motion signal, causing attention to be attracted to its location. When attention is directed to the location, the structure is granted coherence. Any new stimulus at a location within the coherence field is treated as a change of an existing object rather than the appearance of a new one. When the local signal is swamped, the guidance for attention allocation is lost and change blindness is induced.

Rensink's coherence theory leads to an intricate interplay between the internal information based on knowledge and the external information about visual detail based on the image. Maintaining a large internal buffer doesn't make sense when one considers the environment as an external buffer, since the world doesn't change much and the information can always be available from the world itself. High level areas may explain away the image and cause the early areas to be completely suppressed. Alternatively, high-level areas might sharpen the responses of the early areas by reducing activity that is inconsistent with the high level interpretation [27].

5 Implicit Shape Model

5.1 Approach

Torralba et al. [53] describe the commonly-held distinction between three related image understanding tasks. Object detection traditionally assumes the object is buried in a cluttered scene and the detector is tasked with identifying if and where the object is present. Object segmentation traditionally assumes the object has been detected and identifies all pixels that the object occupies. Object recognition traditionally assumes the object has been segmented from the background and asserts its membership. Object recognition can range from categorization at the class-level (e.g. all cars) to identification at the instance-level (e.g. my red Toyota). Riesenhuber and Poggio describe the difference between categorization and identification as a tradeoff between invariance and specificity [47].

Although the detection, segmentation, and recognition tasks can follow a logical sequence, they are intrinsically interrelated and potentially interdependent. Each task is able to leverage information provided by any of the other tasks. This dependence implies that there is no natural ordering of the tasks and suggests each should operate concurrently and perhaps even interactively. Psychophysical experiments support this alternative perspective. Humans demonstrate strong effects of prior shape-specific familiarity during image segmentation. A number of behavioral studies have shown that human subjects are more likely to regard a familiar region, and not a less familiar region, as figure [42, 43]. This indicates that object recognition facilitates segmentation. Complexity analysis also supports this alternative perspective. Torralba et al. discuss the infeasibility of traditional approaches to object recognition. It is impractical to form a specific classifier for each class of interest because it does not scale with respect to computation time or number of training examples [53]. This is even worse when attempting to explicitly create a classifier for each high-level component of the target class and entrusting a final classifier for recognition, as in [17, 33]. Even if the one-classifier-per-class approach was tractable, segmentation would still not be achieved and additional processing would be required.

Torralba et al. observe that common features can be shared across object classes and claim that this redundancy can be leveraged to achieve logarithmic complexity with the number of classes. They train classifiers for multiple classes using shared features. The representation forms a binary feature vector indicating the presence or absence of the feature. A joint-boosting algorithm is employed to converge to a fault-tolerant classifier. Agarwal and Roth [2] also create a binary feature vector indicating the presence of each particular part, but additionally encode its relative location to other parts. Each part, represented by a small image patch, is obtained during training and is combined with similar parts to create a compact vocabulary. The enumeration of relative positions creates a very large and sparse representation, which Agarwal and Roth claim is well suited for the Sparse Network of Winnows (SNoW) classifier. The classifier is only capable of working on fixed-window sizes,

so the typical window-sliding approach must be used. Their approach is also only capable of identifying the bounding box in which the object resides and does not provide refined segmentation information.

Leibe et al. [32] use a visual vocabulary similar to that of Agarwal and Roth, but avoid the window-sliding restriction and achieve segmentation. The approach combines the segmentation and recognition processes, allowing interaction to guide mutual processing. The integration of the learned knowledge about the category and the supporting information in the image parallels the current understanding of human object perception [45]. Segmentation is achieved by gathering an additional segmentation patch during training. This requires a segmentation mask to accompany the training image to indicate the region occupied by the training instance. Since human object recognition performance crucially depends on previous visual experience [46], it is reasonable to rely on sample target objects during a training phase. Vote displacements accompany the model patches to indicate where the object center was observed when the patch was present. Matching patches independently vote in a generalized Hough Transform and an agreement among interpretations gives rise to object presence hypotheses. This framework can interpolate between local parts seen on different training objects, allowing a relatively small number of training examples to recognize and segment the category instance. This allows a robust recognition system under occlusion, since a match may be found even if the test image does not exhibit all features.

The meaning of the shared features presented by Torralba et al. can be enhanced by incorporating these additional discussions. Agarwal et al. refer to the parts as a visual vocabulary, while Leibe et al. refer to local "part" structures as *interpretations*. The inferotemporal cortex (IT), discussed in Section 4.2, comprises cells with receptive fields that respond to intermediately complex patterns that may be described visually. Since an image patch is capable of representing arbitrarily complex structure, the algorithmic analogue to the IT receptive fields can be realized by image patches. The preference [47] for view-based models for human perception, as opposed to object-based models, supports the use of the image patch visual vocabulary used by [2, 32]. The use of arbitrarily complex spatial patterns in the image patch is different from the hand-crafted Haar-like orthogonal basis patches that are used elsewhere [3, 8, 15, 34, 41].

As discussed in Section 3, approaches that represent sub-parts of an object often include the geometric relationships between the parts. Fergus et al. [18] describe the geometrical relationships as a "constellation of parts" and argue shape is represented by the mutual position of the parts. Their parts are represented by a point in an arbitrary appearance space, while a representation closer to human perception is preferred. Human models of representation consist of a group of units, broadly tuned to representatives of the object class, that code for the identity of a particular object by their *combined activation pattern* [45]. Wolf describes psychophysical experiments that have shown preattentive object descriptions consist of only a *collection of isolated features* [58] . Serial attention is necessary to represent shape relationships and integrate these features into a *common object description*. This

agrees with the whole-object detection described by Rensink [45], where the nexus embodies the whole object perceived at any given moment via *connections to its parts*. The coherence field represents a local hierarchy with object- and part-level descriptions that is an extremely useful device and a natural way to represent objects [45].

The work presented in this thesis extends the work of Leibe et al. by following four principles formulated by an integration of the insights from the preceding observations:

1. The traditional detection-segmentation-recognition trichotomy should be abandoned and new algorithms should embody all of these tasks.

2. Algorithms should leverage the commonality of shared features across target categories.

3. Model representations and algorithms reflecting the current physiological or psychophysical understanding of human perception should be preferred.

4. Retreating to a traditional classification algorithm should be used as late in the algorithm as possible.

5.2 Shape Model Creation

This section describes how the shape model is created. Leibe represents an Implicit Shape Model for a given class ω as $ISM(\omega) = (I_\omega, P_{I,\omega})$, where the codebook I_ω contains prototypical local appearances and the spatial probability distribution $P_{I,\omega}$ specifies where each codebook entry may be found on the object. The framework Leibe presents facilitates a probabilistic treatment of localization and segmentation dynamics. However, the spatial probability distribution that Leibe uses is not an explicit or static function. It is embodied by the aggregation of the displacement information associated with each codebook entry. We reformulate[1] Leibe's Implicit Shape Model in a set-theoretic framework to more accurately reflect the computational process and facilitate discussions regarding system implementation. This new and equivalent framework allows a natural and expressive language to discuss and manipulate specific shape model elements based on a variety of implicit and behavioral characteristics and provides the basis for the enhancements discussed in Section 7. Each representation of the shape model may be used to concisely describe different aspects of the localization, recognition, and segmentation processes.

[1]Several notational modifications have also been made to Leibe's original discussion in the pursuit of clarity. Leibe [32] uses the notation C and o_n to define a class and object category, respectively, while our discussions confound these meanings and use the notation ω, adopting conventions presented by Duda et al. [16]. The variable x was replaced with $\vec{\mathbf{x}}$ to express the bivariate nature of a position in an image. The variable $\mathbf{e}$ was replaced with $\vec{\mathbf{e}}$ to express the multidimensional nature of an image patch. The use of I_i was changed to $I_i^{\vec{\mathbf{e}}}$ to express the dependence on $\vec{\mathbf{e}}$.

In the object detection domain, we are concerned with identifying specific locations and category membership of instances in an unknown image. These two pieces of information form an object hypothesis and can be expressed by elements from the sets

$$\Omega = \{\omega : \omega \text{ is the target category}\} \quad (13)$$
$$\Lambda = \{(r, c) : r \in \mathbb{N}, c \in \mathbb{N}\} \quad (14)$$
$$H = \{(\omega, \vec{\lambda}) : \omega \in \Omega, \vec{\lambda} \in \Lambda, \text{object } \omega \text{ may exist at location } \vec{\lambda}\} \quad (15)$$
$$A = \{a \in H : a \text{ is an asserted hypothesis}\} \quad (16)$$

where Ω is the set of objects the algorithm attempts to identify and $\vec{\lambda} \in \Lambda$ is the specific location in the image relative to the origin.

(a) Training image (b) Training segmentation

Figure 3. Sample of input required for shape model creation

The input[2] required for shape model creation is illustrated in Figure 3 and comprises a set of (image, segmentation) pairs where the image contains an instance of the target object in natural surroundings and the segmentation image identifies the region the instance occupies. If r represents the patch radius, a patch from these images can be described[3] as a $(2r + 1)^2$-dimensional vector. Let P denote the set of all patches.

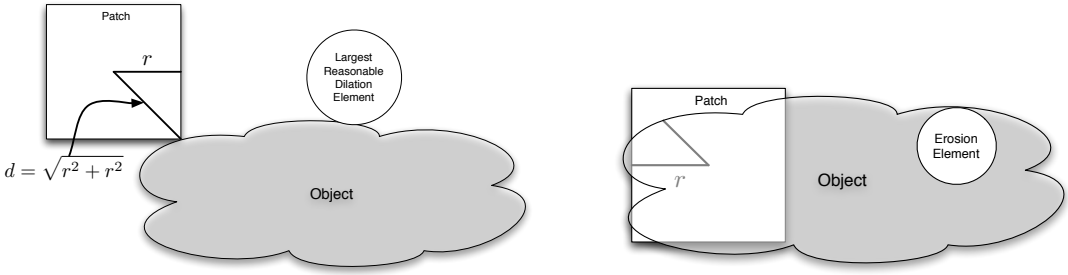
$$P = \{\vec{\mathbf{p}} : \vec{\mathbf{p}} \in \mathbb{R}^{w \times w}, w \in \mathbb{N}\} \quad (17)$$

Harris corner detection is applied to the training images to identify a set of points from which to extract training and segmentation patches. If the distance from the Harris corner to the segmentation mask is greater than the radius of the image patch, none of the patch will contain object information. This is illustrated in Figure 4(a). The locations from which patches are extracted must be within a certain distance from the provided segmentation mask. This is to ensure that the patch contains at least some object information. The distance restriction can vary and defines a region surrounding the input segmentation.

[2]Training and testing images presented in this work were obtained from Leibe [31] and UIUC [1] while the processing presented is the product of our own implementations.

[3]The use of the variables ω and w should be distinguished. The ω is lower case Ω and describes a particular object class. The w represents patch width. The distinction should be clear by context.

This permitted region can be determined by the appropriate morphological process of dilation or erosion. The patch size should be considered when determining the appropriate distance restriction. These trade-offs are illustrated in Figure 4, which shows an approximation of the patch area covering the object as a function of the patch's distance from the provided segmentation. The permitted region size directly affects the size of the model and consequently affects the time to create the model and to use the model for localization. Permitting background information in the shape model also raises concerns for efficacy during the localization process described in Section 5.3. Section 7.2.2 discusses a neutralization technique that can be used during the model creation process to reduce the adverse affect of background information in model patches.



(a) Center farthest away from border (b) Center must be within border

Figure 4. Range of distances from patch center to object boundary

(a) Learning interpretation $I_{\vec{\mathbf{t}}_1} = \{(\vec{\mathbf{t}}_1, \vec{\mathbf{d}}_1)\}$ (b) Learning interpretation $I_{\vec{\mathbf{t}}_2} = \{(\vec{\mathbf{t}}_2, \vec{\mathbf{d}}_2)\}$ (c) Collection of patches and displacements $(\vec{\mathbf{t}}, d)$ from training image

Figure 5. Observed patches noting relative displacement to object center

The observed patches from all training images create sets T and S of training and segmentation patches, respectively. Let Δ be the set of possible displacements relative to a particular point in an image; in our case the reference is the center of an extracted patch. When a training patch is extracted, the displacement from its location to the known object center[4] and the corresponding segmentation patch are stored, creating an interpretation

[4]The center of the object is estimated as the centroid of the segmentation mask.

$I_{\vec{\mathbf{t}}} = \left(\vec{\mathbf{t}}, \omega, \vec{\mathbf{d}}, \vec{\mathbf{s}}\right)$. These associations are illustrated in Figure 5. An interpretation indicates that if a patch $\vec{\mathbf{t}}$ is similar to an unknown image patch at position $\vec{\lambda} \in \Lambda$, an object of type ω is at location $\vec{\lambda} + \vec{\mathbf{d}}$.

$$T \subset P = \{\vec{\mathbf{t}} : \vec{\mathbf{t}} \in \mathbb{R}^{w \times w}, w \in \mathbb{N}, \vec{\mathbf{t}} \text{ observed during training}\} \quad (18)$$

$$S \subset P = \{\vec{\mathbf{s}} : \vec{\mathbf{s}} \in \mathbb{B}^{w \times w}, w \in \mathbb{N}, \vec{\mathbf{s}} \text{ observed during training}\} \quad (19)$$

$$\Delta = \{(\theta, r) : 0 \leq \theta < 2\pi, 0 < r\} = \{(\partial r, \partial c) : \partial r \in \mathbb{Z}, \partial c \in \mathbb{Z}\} \quad (20)$$

$$I_{\vec{\mathbf{t}}} = \{(\vec{\mathbf{t}}, \omega, \vec{\mathbf{d}}, \vec{\mathbf{s}}) : \vec{\mathbf{t}} \in T, \omega \in \Omega, \vec{\mathbf{d}} \in \Delta, \vec{\mathbf{s}} \in S\} \quad (21)$$

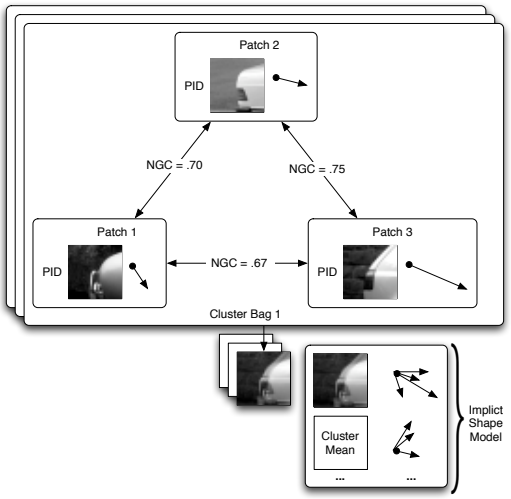


Figure 6. Interpretations $\{I_{\vec{\mathbf{t}}_1}, I_{\vec{\mathbf{t}}_2}, I_{\vec{\mathbf{t}}_3}\}$ consolidated by visual similarity to form model entry $I_{\vec{\mathbf{m}}_1}$

Many of the training patches in T are visually similar. The Normalized Grayscale Correlation (NGC), as described in Section 2.2.1, can be expressed in vector notation and the neighborhood $N_{\vec{\mathbf{p}}}$ of a given patch $\vec{\mathbf{p}}$ is defined as the set of all patches that are within a certain NGC distance of $\vec{\mathbf{p}}$. Patches in the same neighborhood are "visually similar" because the values at their corresponding pixels are correlated.

$$NGC(\vec{\mathbf{p}}, \vec{\mathbf{q}}) = \frac{(\vec{\mathbf{p}} - \vec{\mu}_p)^T(\vec{\mathbf{q}} - \vec{\mu}_q)}{\sqrt{(\vec{\mathbf{p}} - \vec{\mu}_p)^T(\vec{\mathbf{p}} - \vec{\mu}_p) + (\vec{\mathbf{q}} - \vec{\mu}_q)^T(\vec{\mathbf{q}} - \vec{\mu}_q)}} \quad (22)$$

$$N_{\vec{\mathbf{p}}} \subset P = \{\vec{\mathbf{q}} : \vec{\mathbf{p}} \in P, \vec{\mathbf{q}} \in P, NGC(\vec{\mathbf{p}}, \vec{\mathbf{q}}) > .7\} \quad (23)$$

The visual similarity of the training patches suggests that the interpretations associated with the similar patches may be consolidated. Consider interpretations $I_{\vec{\mathbf{t}}_1} = \left(\vec{\mathbf{t}}_1, \omega_1, \vec{\mathbf{d}}_1, \vec{\mathbf{s}}_1\right)$

and $I_{\vec{\mathbf{t}}_2} = \left(\vec{\mathbf{t}}_2, \omega_2, \vec{\mathbf{d}}_2, \vec{\mathbf{s}}_2\right)$ where $\vec{\mathbf{t}}_1 \in N_{\vec{\mathbf{t}}_2}$. Both of these interpretations may be replaced by the consolidated interpretation $I_{\vec{\mathbf{t}}_1,\vec{\mathbf{t}}_2} = \begin{pmatrix} \vec{\mathbf{t}}_1 & , & \omega_1 & , & \vec{\mathbf{d}}_1 & , & \vec{\mathbf{s}}_1 \\ \vec{\mathbf{t}}_2 & , & \omega_2 & , & \vec{\mathbf{d}}_2 & , & \vec{\mathbf{s}}_2 \end{pmatrix}$. The interpretations observed from the training set may be agglomeratively combined in this manner. Interpretations are combined if the average NGC among all training patches in the interpretation exceed a threshold.[5] The interpretation similarity is defined as

$$\text{similarity}(I_1, I_2) = \frac{\sum_{\vec{\mathbf{t}}_1 \in I_1} \sum_{\vec{\mathbf{t}}_2 \in I_2} \text{NGC}(\vec{\mathbf{t}}_1, \vec{\mathbf{t}}_2)}{\|I_1\| \times \|I_2\|} > t \tag{24}$$

where $\|.\|$ is the number of training patches in the consolidated interpretation and t is the NGC threshold.

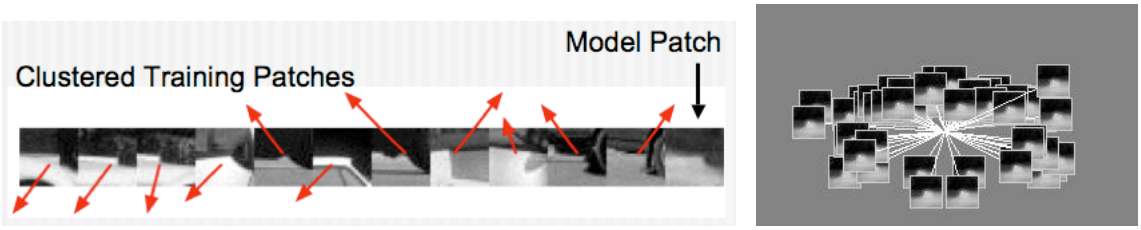
When interpretations can no longer be combined without violating this condition, a single representative patch $\vec{\mathbf{m}}$ is obtained for each consolidated interpretation by finding the vector mean of the constituent patches. In our previous example, this allows $I_{\vec{\mathbf{t}}_1,\vec{\mathbf{t}}_2} = \begin{pmatrix} \vec{\mathbf{t}}_1 & , & \omega_1 & , & \vec{\mathbf{d}}_1 & , & \vec{\mathbf{s}}_1 \\ \vec{\mathbf{t}}_2 & , & \omega_2 & , & \vec{\mathbf{d}}_2 & , & \vec{\mathbf{s}}_2 \end{pmatrix}$ to become model entry $I_{\vec{\mathbf{m}}} = \begin{pmatrix} \vec{\mathbf{m}}, & \omega_1 & , & \vec{\mathbf{d}}_1 & , & \vec{\mathbf{s}}_1 \\ & \omega_2 & , & \vec{\mathbf{d}}_2 & , & \vec{\mathbf{s}}_2 \end{pmatrix}$. This model entry embodies multiple interpretations with a single model patch $\vec{\mathbf{m}}$. The agglomeration and model entry processes are illustrated in Figure 6. The set of model patches is denoted by M and all interpretations that involve a specific class ω can be aggregated to obtain the set I_ω. This set embodies the shape model for the target category. Figure 7 shows eleven training patches within a consolidated interpretation, the model patch derived from them, and the relative locations of each interpretation from a common object center. This illustration shows that the single patch can be interpreted as the top of a light colored car or the bottom of a dark colored car.

$$M \subset P = \left\{\vec{\mathbf{m}} : \vec{\mathbf{m}} = \frac{1}{n}(\vec{\mathbf{t}}_1 + \vec{\mathbf{t}}_2 + \cdots + \vec{\mathbf{t}}_n), \vec{\mathbf{t}}_i \in T \text{ agglomerated}\right\} \tag{25}$$

$$I_\omega = \left\{\bigcup (I_{\vec{\mathbf{m}}}) \text{ such that } \omega \in I_{\vec{\mathbf{m}}}\right\} \tag{26}$$

All training patches $\vec{\mathbf{t}}$ in a consolidated interpretation are similar to the model patch $\vec{\mathbf{m}}$, but the training patches in the consolidated interpretation are not the only patches in the training database T that are similar to the model patch. A consolidated model interpretation $I_{\vec{\mathbf{m}}} = \begin{pmatrix} \vec{\mathbf{m}}, & \omega_1 & , & \vec{\mathbf{d}}_1 & , & \vec{\mathbf{s}}_1 \\ & \omega_2 & , & \vec{\mathbf{d}}_2 & , & \vec{\mathbf{s}}_2 \end{pmatrix}$ can accumulate additional object vote information if the model

[5] As discussed in Section 2.2.1, the range of the NGC provides an intuitive and accurate metric for similarity, so determining this threshold is straightforward. The NGC threshold throughout this thesis is .7, while another common value in the literature is .8.



(a) Training patches and model patch

(b) Relative locations to object center

Figure 7. Consolidated Interpretation

patch $\vec{\mathbf{m}}$ is visually similar to a training patch $\vec{\mathbf{t}}_k$ not originally in $I_{\vec{\mathbf{m}}}$. This results in the addition of the $(\omega_k, \vec{\mathbf{d}}_k, \vec{\mathbf{s}}_k)$ triplet observed during extraction of $\vec{\mathbf{t}}_k$. $I_{\vec{\mathbf{m}}}$ then becomes $I_{\vec{\mathbf{m}}} = \begin{pmatrix} & \omega_1 & \vec{\mathbf{d}}_1 & \vec{\mathbf{s}}_1 \\ \vec{\mathbf{m}}, & \omega_2 & , \vec{\mathbf{d}}_2 & , \vec{\mathbf{s}}_2 \\ & \omega_k & \vec{\mathbf{d}}_k & \vec{\mathbf{s}}_k \end{pmatrix}$ if $\vec{\mathbf{t}}_k \in N_{\vec{\mathbf{m}}}$. This additional processing may be omitted to obtain a linear-time heuristic. Figure 8 compares the number of votes obtained for the shape model when performing the algorithm and heuristic.

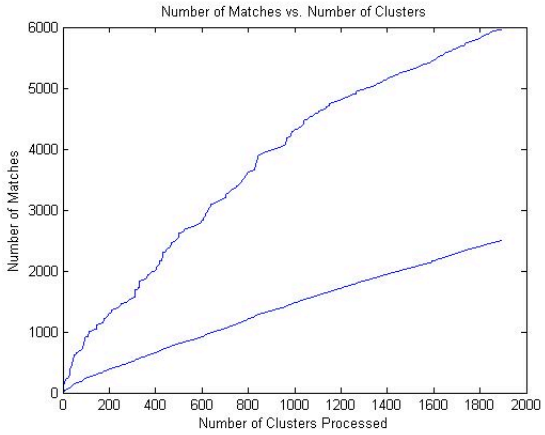


Figure 8. Number of vote displacements in shape model: algorithm vs. heuristic.

5.2.1 Complexity

This section describes the runtime complexity of the shape model creation. We are interested in the computation cost as a function of the number of accumulated patches and the number of consolidated interpretations. Operations such as reading an image, corner detection, and morphological processing are considered to be constant since they are independent of the shape model growth.

Two supporting data structures are maintained to avoid repeating identical computations. Since the similarity between two consolidated interpretations is the average NGC between each pairing of constituent training patches $\vec{\mathbf{t}}$, the NGC value is consistently required. A matrix containing the NGC between the corresponding patches is maintained to avoid recomputing this static measure. The Similarity Matrix (SIM) is maintained to record the similarity between each pair of consolidated interpretations. When a training patch $\vec{\mathbf{t}}$ is added to the database T, the NGC values between it and all other patches are added to the NGC matrix. This results in the addition of a new row and column. When two consolidated interpretations are merged, the similarities between the modified interpretation and all other interpretations are also updated. This results in a row and column deletion of SIM since two consolidated interpretations become a single consolidated interpretation.

Since the size of the patch is constant, the correlation coefficient computation may be considered constant. Although the cost is constant, it is the single largest computation required in the algorithm and is performed an amount proportional to the size of the shape model. This leads us to be concerned with minimizing the number of operations used. Let k patches be in the existing training patch database. The cost of creating NGC and updating it p times is:

$$NGC(p) = O\left(\frac{1}{2}k^2 + \sum_{i=1}^{p} p\right) = O\left(\frac{1}{2}k^2 + \frac{p(p+1)}{2}\right) = O(p^2)$$

The size of p is bounded by the size of the training patch set T.

The computation of SIM involves the averaging of a subset of the values in the NGC matrix. This involves little more than memory access. Regardless, the computation grows with the number of consolidated interpretations:

$$SIM(c) = O\left(\frac{1}{2}k^2 + \sum_{i=1}^{c} c\right) = O\left(\frac{1}{2}k^2 + \frac{c(c+1)}{2}\right) = O(c^2)$$

where c is the number of consolidated interpretations. Since the number of consolidated interpretations is smaller than the number of training patches, the cost of NGC(p) is greater than SIM(c). The algorithm to consolidate the shape model is $O(\|P\| \times \|C\|)$, where P is the set of patches and C is the set of consolidated interpretations. Since $\|C\| < \|P\|$, the total operation is $O(\|P\|^2)$.

Algorithms 1 through 3 outline the operations used to create the shape model. Algorithm 2 extracts patches from the training set and consolidates them into an existing consolidated interpretations structure. The first call to Algorithm 2 obtains all patches in the image, agglomerates them, and returns the consolidated interpretations. The subsequent calls to Algorithm 2 include the accumulated interpretations. Algorithm 3 reduces the training patches in the consolidated interpretations to the single model patch $\vec{\mathbf{m}}$ and accumulates

additional object vote information. Figure 9(a) shows the run time as a function of training patches in T and Figure 9(b) shows the number of consolidated interpretations and number of extracted training patches.

Algorithm 1 Find Consolidated Interpretations in Training Set

Require: set of training images
Ensure: consolidated interpretations consInterp
  consInterp, NGC, SIM = GetPatchClusters(I_1)
  for all images I_i in training set **do**
    consInterp, NGC, SIM = GetPatchClusters(I_i, consInterp, NGC, SIM)
  end for
  return consInterp

Algorithm 2 GetPatchClusters (Find Consolidated Interpretations in Image)

Require: image
Ensure: clusters with patches from image added
  find corners in image
  if clusters provided **then**
    for all patches $\vec{\mathbf{t}}$ centered around corner in image **do**
      clusters = AddPatchToClusters(p, clusters)
    end for
  else
    create new extractedPatches data structure
    for all patches $\vec{\mathbf{t}}$ centered around corner in image **do**
      add $(\vec{\mathbf{t}}, \omega, \vec{\mathbf{d}}, \vec{\mathbf{s}})$ to extractedPatches
    end for
    clusters = AgglomeratePatches(extractedPatches)
  end if
  return clusters

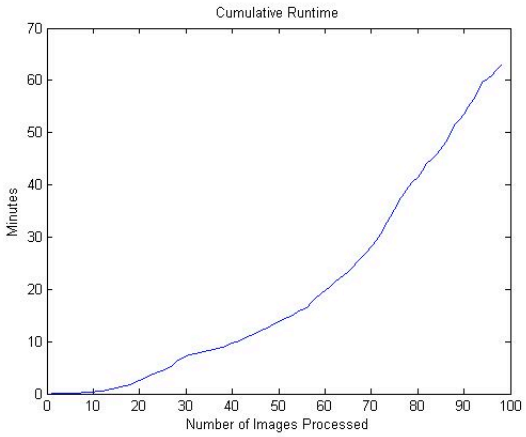
5.3 Object Localization

5.3.1 Voting Space

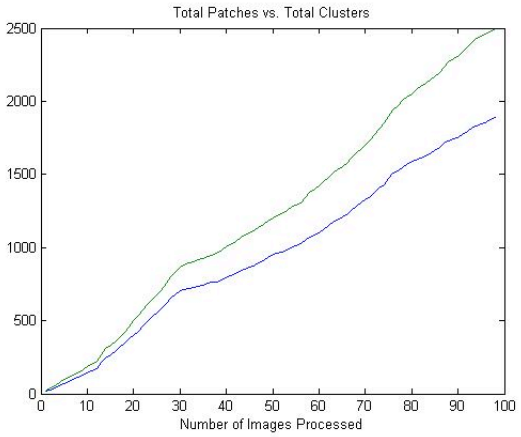
The shape model may be used to identify the location and category membership of objects in an unknown image. First, the Harris corner detector is used to identify locations from which to extract evidence patches. Let E describe the set of evidence patches from the image

Algorithm 3 Create Shape Model

Require: consolidated interpretations consInterp
Ensure: implicit shape model
  find representative model patch for each consolidated interpretation
  for all model patches $\vec{\mathbf{m}}_i$ in consInterp **do**
    for all patches $\vec{\mathbf{t}}_j \in T$ **do**
      if NGC($\vec{\mathbf{m}}_i$,$\vec{\mathbf{t}}_j$) > SIM_THRESH **then**
        add $(\omega_j, \vec{\mathbf{d}}_j, \vec{\mathbf{s}}_j)$ to consolidated interpretation i
      end if
    end for
  end for
  return consInterp



(a) Runtime Plot



(b) Number of patches and clusters

and the location at which they were extracted.

$$E = \{(\vec{\mathbf{e}}, \vec{\lambda}) : \vec{\mathbf{e}} \in P, \vec{\lambda} \in \Lambda, \vec{\mathbf{e}} \text{ extracted from novel image at position } \vec{\lambda}\} \quad (27)$$

For any patch $\vec{\mathbf{e}}$ extracted from an image centered on location $\vec{\lambda}$, the set $I^{\vec{\mathbf{e}}} \subset I_\omega$ represents the interpretations in the shape model that are visually similar to the patch $\vec{\mathbf{e}}$.

$$I^{\vec{\mathbf{e}}} \subset I_\omega = \left\{ \bigcup I_{\vec{\mathbf{m}}}, NGC(\vec{\mathbf{m}}, \vec{\mathbf{e}}) > .7 \right\}$$

When $\vec{\mathbf{e}}$ matches a model patch $\vec{\mathbf{m}}$, we can move $I_{\vec{\mathbf{m}}}$ on top of $\vec{\mathbf{e}}$ at position $\vec{\lambda}$ and cast votes to the locations $\vec{\lambda} + \vec{\mathbf{d}}$ for all $\vec{\mathbf{d}} \in I_{\vec{\mathbf{m}}}$. Each vote has a weight of $\frac{NGC(\vec{\mathbf{e}}, \vec{\mathbf{m}})}{|I_{\vec{\mathbf{m}}}|}$. Figure 9(c) shows five interpretations of a model patch casting five votes for the object center. Vote casting for the single i^{th} interpretation in $I^{\vec{\mathbf{e}}}$ is represented[6] by the probability density function

$$p(\omega, \vec{\mathbf{x}} | I_i^{\vec{\mathbf{e}}}, \vec{\lambda}) \quad (28)$$

This action can be performed for all interpretations associated with the matching model patch, creating the marginalization

$$p(\omega, \vec{\mathbf{x}} | \vec{\mathbf{e}}, \vec{\lambda}) = \sum_i p(\omega, \vec{\mathbf{x}} | I_i^{\vec{\mathbf{e}}}, \vec{\lambda}) p(I_i^{\vec{\mathbf{e}}} | \vec{\mathbf{e}}, \vec{\lambda}) \quad (29)$$

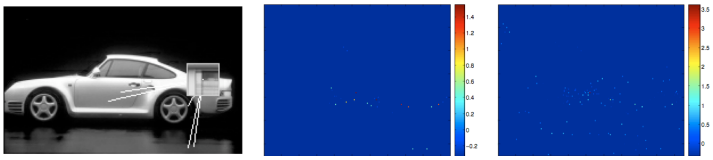
and can be illustrated, as in Figure 9(d), with an image containing mostly zeros and a few isolated positive pixels surrounding location $\vec{\lambda}$. Since $\vec{\mathbf{e}}$ and $\vec{\mathbf{m}}$ are matched independently of the location $\vec{\lambda}$ and all interpretations are assumed to be equally likely, $p(I_i^{\vec{\mathbf{e}}} | \vec{\mathbf{e}}, \vec{\lambda}) \rightarrow p(I_i^{\vec{\mathbf{e}}} | \vec{\mathbf{e}}) = \frac{1}{\|I^{\vec{\mathbf{e}}}\|}$.

Accumulating $p(\omega, \vec{\mathbf{x}} | \vec{\mathbf{e}}, \vec{\lambda})$ for all $\vec{\mathbf{e}} \in E$ creates the multivariate probability distribution function

$$\text{Object Presence: } p(\omega, \vec{\mathbf{x}}) = \sum_{k=1}^{|E|} p(\omega, \vec{\mathbf{x}} | \vec{\mathbf{e}}_k, \vec{\lambda}_k) \quad (30)$$

and provides the probability of a particular object at a given location in the image. Figure 10 illustrates the accumulation of interpretation votes. Figure 11 illustrates the process on an unknown image.

[6]Both $\vec{\lambda}$ and $\vec{\mathbf{x}}$ are elements of Λ. The variable $\vec{\lambda}$ is used to indicate a location of interest such as a harris corner, the center of an extracted patch, or an object hypotheses. The variable $\vec{\mathbf{x}}$ is used to parameterize the probability function describing all locations in the image. The bounds of $\vec{\lambda}$ and $\vec{\mathbf{x}}$ may extend beyond the bounds of the image if votes are not clipped to the image edge.



(c) An evidence patch $\vec{\mathbf{e}}$ matches a model patch $\vec{\mathbf{m}}$ and casts all interpretations associated with $\vec{\mathbf{m}}$

(d) $p(\omega, \vec{\mathbf{x}}|\vec{\mathbf{e}}, \vec{\lambda})$

(e) $p(\omega, \vec{\mathbf{x}})$

Figure 9. Casting interpretations of the input image

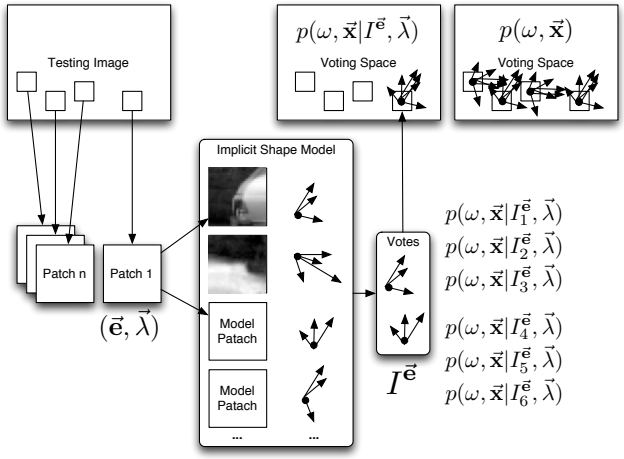
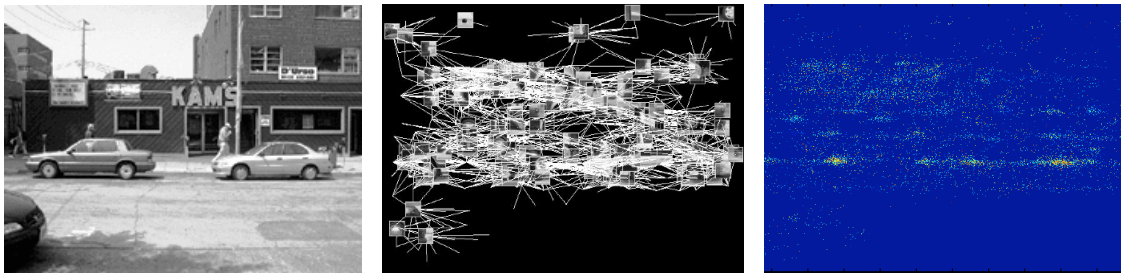


Figure 10. Using shape model to create object-hypothesis vote space



(a) Original image

(b) Matching interpretations

(c) Vote space $p(\omega, \vec{\mathbf{x}})$

Figure 11. Localization process on unknown image

5.3.2 Mean-Shift Clustering

Matching patches cast votes for the object identity and location in a Generalized Hough Transform voting space. The object hypotheses $(\omega, \vec{\lambda})$ may be identified by finding the maximum and dense regions in Equation 30. Mean shift clustering [11, 13], illustrated in Figure 12, is used to create these object hypotheses. Let the universe of data samples exist within R^2, the 2-dimensional Euclidean space. The algorithm begins by assigning each point in the vote map to a single cluster, creating the original set $S \subset R^2$. Each cluster is iteratively repositioned according to the sum of weighted displacements caused by nearby clusters. There are two weights for each cluster. The first weight is the vote mass from Equation 30 and is represented by $w(\vec{s})$. This is the result of the shape model vote casting. The second weight is from a kernel K, which is a function accepting a vector and returning a weight proportional to its magnitude. A simple kernel is a window function of a fixed radius. Gaussian kernels are also appropriate and reduce the weighting of more distant neighbors. The kernel size should be chosen appropriately because it is related to the expected cluster size and the distance between the final clusters. Let T_i be the current cluster centers and $T_1 = S$. Then $T_{i+1} \leftarrow m(T_i)$ such that

$$
\begin{aligned}
m(\vec{x}) &= \frac{\sum_{\vec{s}\in S} K(\vec{s}-\vec{x})w(\vec{s})\vec{s}}{\sum_{\vec{s}\in S} K(\vec{s}-\vec{x})w(\vec{s})} && (31)\\
&= \left(\frac{K(\vec{s}_1-\vec{x})w(\vec{s}_1)}{\sum_{\vec{s}\in S} K(\vec{s}-\vec{x})w(\vec{s})}\right)\vec{s_1} + \cdots + \left(\frac{K(\vec{s}_n-\vec{x})w(\vec{s}_n)}{\sum_{\vec{s}\in S} K(\vec{s}-\vec{x})w(\vec{s})}\right)\vec{s_n} && (32)\\
&= \left(\frac{1}{\sum_{\vec{s}\in S} K(\vec{s}-\vec{x})w(\vec{s})}\right)(K(\vec{s}_1-\vec{x})w(\vec{s}_1)\vec{s}_1 + \cdots + K(\vec{s}_n-\vec{x})w(\vec{s}_n)\vec{s}_n) && (33)\\
&= \left(\frac{1}{\sum \hat{w}_i}\right)(\hat{w}_1\vec{s}_1 + \cdots + \hat{w}_n\vec{s}_n) && (34)
\end{aligned}
$$

for each $\vec{x} \in T_i$ and where $\hat{w}_i = K(\vec{s}_i - \vec{x})w(\vec{s}_i)$. We find it instructive to represent the linear combination derived in Equation 31 with the matrix operation

$$
\frac{1}{\sum_i \hat{w}(\vec{s}_i)}
\begin{bmatrix}
\hat{w}(\vec{s}_1) & \hat{w}(\vec{s}_2) & \hat{w}(\vec{s}_3) & \ldots & \hat{w}(\vec{s}_n)\\
\hat{w}(\vec{s}_1) & \hat{w}(\vec{s}_2) & \hat{w}(\vec{s}_3) & \ldots & \hat{w}(\vec{s}_n)
\end{bmatrix}
\begin{bmatrix}
R_{\vec{s}_1} & C_{\vec{s}_1}\\
R_{\vec{s}_2} & C_{\vec{s}_2}\\
R_{\vec{s}_3} & C_{\vec{s}_3}\\
\ldots & \ldots\\
R_{\vec{s}_n} & C_{\vec{s}_n}
\end{bmatrix}
\qquad (35)
$$

where $R_{\vec{s}_i}$ is the row component of the vector $\vec{s}_i$ and $C_{\vec{s}_i}$ is the column component of the vector $\vec{s}_i$. This avoids an iteration for the scaling of each individual neighbor and achieves the result in a single efficient matrix operation. When multiple clusters are moved to the same location, they are confounded and their weights are summed.

The entity T_{i+1} may be the previous iteration state T_i or it may be the original set S. Using the previous iteration is termed *blurring* and reusing the original set S is termed *non-blurring*. The distinction between these two techniques is important with respect to computational complexity and the resulting cluster positions. The blurring technique may be performed faster since the number of clusters to compare each element in $\|T_i\|$ attenuates over the iterations. Blurring also allows an aggregation of votes to obtain stronger hypotheses. However, the clusters positions are able to travel further from their original locations. Since the final position of the clusters becomes the object hypotheses and we are concerned with localization accuracy, this is an undesirable behavior. Using the non-blurring technique prevents clusters from deviating from their current location, since they are continually influenced by the original cluster locations. However, a larger number of object hypotheses result. Section 5.3.4 discusses a compromise between these two alternatives.

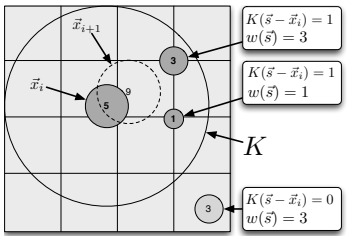
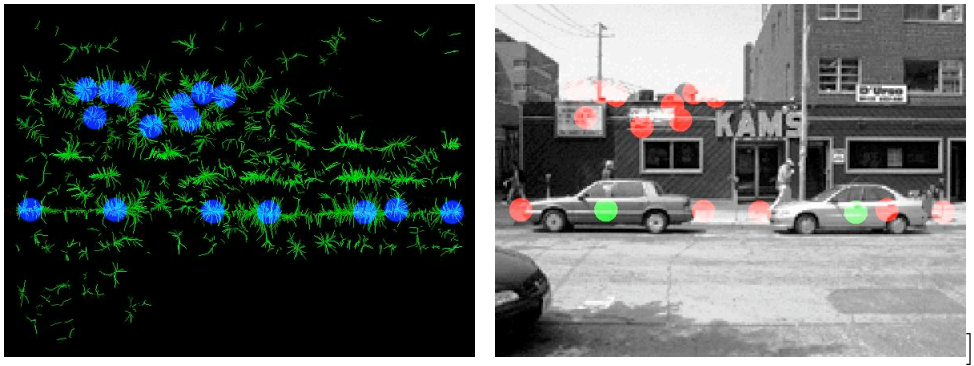


Figure 12. Cluster relocation caused by weighted influences of neighbors



(a) Mean-shift clustered votes from image (b) Test image with highlighted final object hypotheses

in (b) with votes consuming 60% critical mass highlighted

Figure 13. Patch voting leading to object localization

Significant reduction in computation may be achieved when the fact that only a restricted subset of the factors in Equation 31 are nonzero. Since we are working in R^2, the clusters in an image are indexed by position when stored in a matrix. This allows an efficient retrieval

Algorithm 4 Mean-shift Algorithm

Require: Initial vote locations with weights $\{(\vec{\lambda}, w)\}$
Ensure: Dense vote locations with large weights $\{(\vec{\lambda}, w)\}$
  Create candidate clusters
  while Candidate clusters are different from previous iteration **do**
    for all Candidate clusters $(\vec{\lambda}_c, w) \in \{(\vec{\lambda}, w)\}$ **do**
      Get locations and weights $\{(\vec{\lambda}_n, w_n)\}$ in neighborhood of $\vec{\lambda}_c$
      Weight weights by Gaussian: $\hat{w} = K(\vec{\lambda}_c - \vec{\lambda}_n, w(n))$
      Obtain weight and location vectors of neighbors as in Equation 35
      Find inner product of weight vector with row locations of neighbors
      Find inner product of weight vector with column locations of neighbors
      Form a vector with the two inner products and scale by the sum of the weights
    end for
    Combine weights of clusters that were relocated to same position
  end while
  return converged locations and associated weights

of only the clusters that are within the desired distance of the cluster $\vec{x}$. This implicitly performs as a flat kernel and avoids unnecessary computation. A Gaussian Kernel weighting is trivially obtained by scaling the retrieved subimage via element multiplication of a pre-computed kernel with unit volume.

5.3.3 Critical Mass

The critical mass technique is a technique to determine how many of the weighted object hypotheses obtained after mean-shift clustering should become candidate object hypotheses for further investigation. The runtime of the remaining localization algorithm depends on the number of candidate locations obtained, so restricting the size of this set is desirable. The hypotheses are sorted by their final vote mass, and the largest k hypotheses are selected such that

$$\sum_{i=1}^{k} w(\vec{s}_i) \leq \alpha \sum_{i} w(\vec{s}_i) < \sum_{i=1}^{k+1} w(\vec{s}_i) \tag{36}$$

where $0 < \alpha \leq 1$ and $w(s_1) \geq w(s_2) \geq ... \geq w(s_k)$. We have found that $\alpha = .6$ works well. An example of this critical mass restriction is illustrated in Figure 13(a), where the locations permitted to become candidate hypotheses after the mean-shift clustering are highlighted with blue circles.

5.3.4 Postblur Mean-Shift Clustering

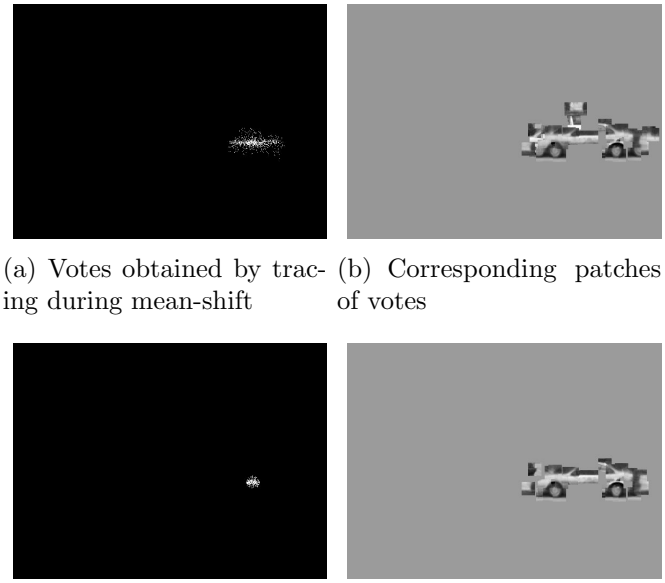
Section 5.3.2 describes the tradeoff between blurring and non-blurring during the mean-shift clustering. We introduce a recursive implementation of mean-shift clustering. The algorithm accepts a binary encoding of which method to use at each level of execution. The recursive call assigns the set S to the converged positions of the previous execution. This allows a compromise between the blur and nonblur methods. The method ordering"(Nonblur, blur)" allows the vote masses to combine within the restrictions of the original vote locations, truncates with a critical mass restriction, and finally blurs cluster locations. This final blurring is more appropriate after the insignificant vote mass clusters are removed.

5.3.5 Resolving Vote Equivalencies

It is useful to know which interpretation votes contributed to each candidate location. Given a vote location $\vec{\lambda}_v$ from an interpretation and a candidate hypothesis location $\vec{\lambda}_h$, there are two ways to consider *contribution*. The natural definition would declare contribution if the vote mass cast by the interpretation was relocated to the candidate hypothesis during the mean-shift clustering. Mean-shift clustering allows votes from a large region to combine. If the vote location is relocated a significant distance, it may be undesirable to allow contribution to the candidate hypothesis. Figure 14(a) shows original vote-mass locations that were relocated to the candidate hypothesis. A distance restriction can be used to filter the initial votes to only those that are close to the resulting hypothesis. This is shown in Figure 14(c). The extra patches further away from the radius are more likely to match erroneous background patches and do not provide good segmentation contributions. By comparing Figures 14(b) and 14(d) it is evident that the radius restriction does not reduce the appropriate matches while omitting undesirable patches. An optional step may be included to reposition the original vote vectors by the interpretations so that its hypothesis aligns with the candidate hypothesis instead of where it actually voted.

5.3.6 Complexity

Algorithm 5 finds the probability density function in Equation 30 for each object of interest by accumulating the probability distribution functions provided by Algorithm 6, which represents the patch vote information in Equation 28. Let the number of extracted patches from the unknown image be $e = \|E\|$, the number of model patches in the shape model be $m = \|I_\omega\|$, where E and I_ω are the sets defined in Section 5.2 and d is the dimension of the model patches $\vec{\mathbf{m}}$. Operations such as image reading, corner detection, and patch extraction are assumed to be constant since their costs do not increase with e or m. The operation that takes longest during interpretation vote casting is the NGC calculation between two patches. The NGC calculation comprises two mean of vector elements, two subtraction of



(a) Votes obtained by tracing during mean-shift

(b) Corresponding patches of votes

(c) Votes obtained by distance from hypothesis

(d) Corresponding patches of votes

Figure 14. Results of Radius vs Tracking Equivalencies

vectors, three inner products, one addition of scalars, and a square root calculation. A straight forward implementation is outlined in Algorithm 7.

Algorithm 5 ObjectPresence($image, \Omega$) (Implementation of Equation 30)

Require: n × m *image*
Require: Ω, set of objects to search for
Ensure: voteMap of size ($|\Omega|\times$ n × m) containing $p(\omega, \vec{\mathbf{x}})$
  voteMap ← empty matrix of size ($|\Omega| \times n \times m$)
  for all patches $\vec{\mathbf{e}}$ at corner locations $\vec{\lambda}$ in unknown image **do**
    for $\omega = 1{:}|\Omega|$ **do**
      voteMap(ω) ← voteMap(ω) + PatchVote($\vec{\mathbf{e}}, \vec{\lambda}, \omega$)
    end for
  end for

Each patch $\vec{\mathbf{e}} \in E$ must be compared to each model patch $\vec{\mathbf{m}} \in I_\omega$. Since e can be much less or much greater than m, we will not combine or bound these terms. Thus, the vote casting algorithm is $O(e \times m)$ or $O((7d + 3)^2) = O(d^2)$ where d represents an addition, subtraction, multiplication, or division of scalars. If e is between 200 and 600 and m is approximately 1000, 60,000 NGC calculations are performed on a single test image. Redundant calculations can be avoided by memoizing repeating calculations and accessing

Algorithm 6 PatchVote($\vec{\mathbf{e}}, \vec{\lambda}, \omega_s$) (Implementation of Equation 28)

Require: Patch $\vec{\mathbf{e}}$ from unknown $n \times m$ image
Require: Location $\vec{\lambda} = [r_e, c_e]$ of patch $\vec{\mathbf{e}}$
Require: Object class ω_s to search for
Ensure: voteMap containing $p(\omega, \vec{\mathbf{x}}|\vec{\mathbf{e}})$ for all $\vec{\mathbf{x}}$
  voteMap $\leftarrow n \times m$ matrix of zeros
  for all interpretations $I_i = (\vec{\mathbf{m}}, \omega, dr, dc)$ in shape model I_ω **do**
    if NGC($\vec{\mathbf{e}}$, $\vec{\mathbf{m}}$) > SIM_THRESH **then**
      {Since $p(I_i^{\vec{\mathbf{e}}}|\vec{\mathbf{e}}) > 0$, add $p(\omega, \vec{\mathbf{x}}_j|\vec{\lambda}_i)$}
      voteMap$(r_e + dr, c_e + dc)$ = NGC($\vec{\mathbf{e}}$, $clusterMean$) / $|I_{\vec{\mathbf{m}}}|$
    end if
  end for

the result when needed. These calculations may be avoided entirely by normalizing the data in the patches so that the mean and standard deviation are zero and one, respectively. The shape model patches may be stored in this normalized form and the evidence patches $\vec{\mathbf{e}}$ may be normalized during extraction. Despite these minimization techniques, the algorithm remains $O(d^2)$. The mean-shift algorithm is also $O(d^2)$ due to the distance comparison between the current vote mass and the vote masses in T_i. This can be reduced by a constant factor when using the indexing procedure described in Section 5.3.2 and heuristics may be used to reduce the initial set of cluster centers.

Algorithm 7 Straight-forward $O(e(m(7d+3))) = O((7d+3)^2) = O(49d^2)$ Patch Matching

Require: Shape model I_ω and extracted patches from unknown image $E = \{(\vec{\mathbf{e}}, \vec{\lambda})\}$
Ensure: Collection of cast interpretations $I^{\vec{\mathbf{e}}}$
  for all extracted patches from unknown image **do**
    for all patches in shape model **do**
      Calculate $\vec{\mu}_p$
      Calculate $(\vec{\mathbf{p}} - \vec{\mu}_p)$
      Calculate $(\vec{\mathbf{p}} - \vec{\mu}_p)^T(\vec{\mathbf{p}} - \vec{\mu}_p)$
      Calculate $\vec{\mu}_q$
      Calculate $(\vec{\mathbf{q}} - \vec{\mu}_q)$
      Calculate $(\vec{\mathbf{q}} - \vec{\mu}_q)^T(\vec{\mathbf{q}} - \vec{\mu}_q)$
      Calculate $(\vec{\mathbf{p}} - \vec{\mu}_p)^T(\vec{\mathbf{q}} - \vec{\mu}_q)$
      Calculate addition, square root, and division of scalars
    end for
  end for

5.4 Segmentation

A localized object is segmented from the remaining image on a per-pixel basis using the object hypothesis $(\omega, \vec{\lambda})$ obtained from the discussions in Section 5.3. This produces a grayscale mask $p(\mathbf{p} = figure|\omega, \vec{\lambda})$ representing the probability each pixel $\mathbf{p}$ in the image contains the object.

Only a subset of all interpretations cast during the initial voting phase contribute to the object hypothesis. This can be represented by $I^E|_{(\omega,\vec{\lambda})}$. As discussed in Section 5.2, each interpretation has an associated segmentation patch $\vec{\mathbf{s}}$ indicating the foreground pixels in the patch. A weighted average of this pixel-level segmentation information is obtained for each pixel occupied by the internal representation. The influence weight that a given interpretation $I_{\vec{\mathbf{m}}}$ has is proportional to the amount it contributed to the object hypothesis and can be expressed as:

$$p(\vec{\mathbf{e}}|\omega, \vec{\mathbf{x}}) = \frac{p(\omega, \vec{\mathbf{x}}|\vec{\mathbf{e}}, \vec{\lambda})p(\vec{\mathbf{e}})}{p(\omega, \vec{\mathbf{x}})} \tag{37}$$

The factor $p(\vec{\mathbf{e}})$ is assumed to be constant, leaving the patch influence to be the ratio of the votes from one patch to the sum of all winning votes. The probability that the pixel is figure is obtained by summing over all interpretations in $I^E|_{(\omega,\vec{\lambda})}$ that contain the pixel:

$$p(\mathbf{p} = figure|\omega, \vec{\lambda}) = \sum_{\mathbf{p}\in\vec{\mathbf{e}}} p(\mathbf{p} = figure|\vec{\mathbf{e}}, \omega, \vec{\lambda})p(\vec{\mathbf{e}}|\omega, \vec{\mathbf{x}}) \tag{38}$$

where $p(\mathbf{p} = figure|\vec{\mathbf{e}}, \omega, \vec{\lambda})$ denotes patch-specific segmentation information from $\vec{\mathbf{s}}$. Figure 15 illustrates the weighting calculation when $I^E|_{(\omega,\vec{\lambda})}$ contains four patches voting for an object hypothesis. The total vote weight is represented in the horizontal bar and is partitioned according to the contribution from each patch. The segmentation masks are shown for interpretations A and B. The probability of foreground for the pixel at the intersection of A and B is found by weighting the segmentation asserted by each interpretation by the proportion that it contributed to the final hypothesis. Patch A suggests that the pixel should be background, while patch B suggests that it should be foreground. Since patch B contributed more to the final hypothesis, its interpretation is weighted more and the final probability of foreground is $\frac{9}{13}$. A similar calculation is performed for all pixels occupied by a voting patch, while $p(\mathbf{p} = figure|\omega, \vec{\lambda})$ for all other pixels is zero. Figure 16 shows the sample results of this segmentation process. Part (a) shows the unknown image, referred to as the "external representation," and the highlighted object hypotheses. Part (b) shows the model patches from the shape model that matched the image during the initial voting phase and is referred to as the "internal representation." Part (c) is the representation of $p(\mathbf{p} = figure|\omega, \vec{\lambda})$ for all $\mathbf{p}$ in the image and is referred to as the "internal segmentation." Part (d) is the original image masked by the grayscale mask in (c) and is referred to as the "external segmentation." The terminology for these representations are inspired by Rensink's coherence theory [45].

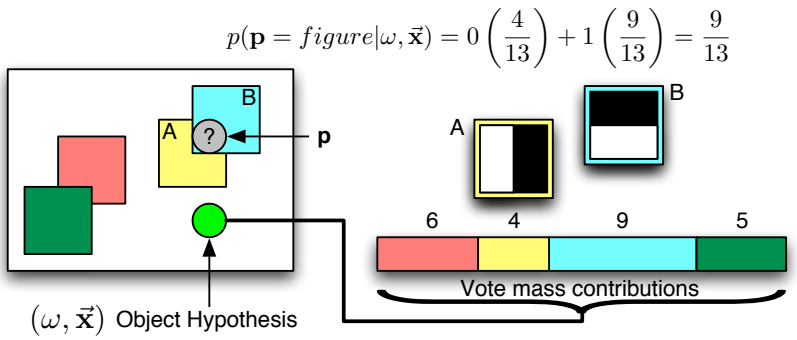
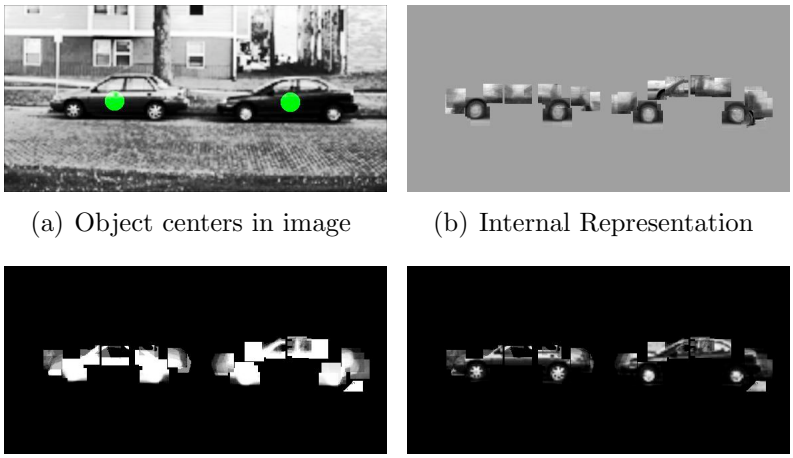


Figure 15. Weighted contribution for segmentation calculation



(a) Object centers in image

(b) Internal Representation

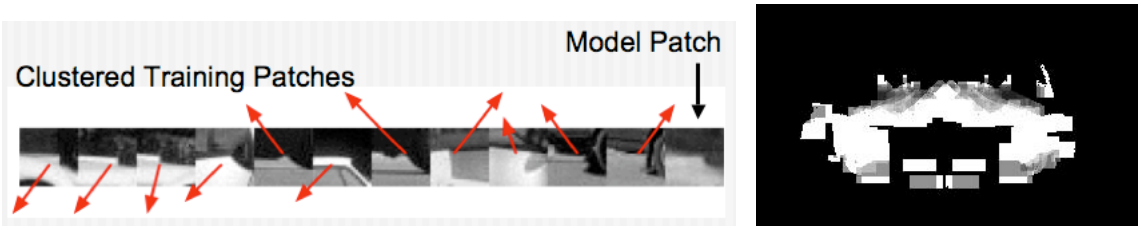
(c) Internal Segmentation

(d) External Representation

Figure 16. Segmentation results using object hypotheses $(\omega, \vec{\lambda})$

The discontinuous segmentation provided by the initial filtered interpretation set $I^E|_{(\omega,\vec{\lambda})}$ suggests that additional interpretations could be added before finding $p(\mathbf{p} = figure|\omega, \vec{\lambda})$. Additional interpretations from the shape model may be found by comparing all model patches with the patches in the unknown image at locations $\vec{\lambda}_h - \vec{\mathbf{d}}$. If the newly extracted patch is similar to the model patch, the associated interpretations are added[7] to the set $I^E|_{(\omega,\vec{\lambda})}$. Leibe ambiguously reports a "uniform sampling" of points surrounding the object hypothesis and refers to this operation as a "refined segmentation." A significant shortcoming of this approach is a high computational cost. Less intensive techniques providing more accurate results are introduced in Section 7.3.2.

There are two goals during segmentation refinement. First, the empty regions within the object region need to be filled in. Second, the edge transitions between object and background need to be confirmed. Only the pixels surrounding the object are used for refinement. These pixels are found by thresholding the initial segmentation obtained from $I^E|_{(\omega,\vec{\lambda})}$ such that $p(\mathbf{p} = figure|\omega, \vec{\lambda}) > .5$. Each object hypothesis is processed individually. The convex polygon of all remaining pixels defines the region of uniform sampling. These additional votes fill in unattended regions within the object regions and refine segmentation decisions regarding the object border. Because edge segmentation is more ambiguous and the patch-matching operation is time-intensive, attention should be focused on the border regions. We introduce an alpha- and beta-sampling technique to achieve this non-uniform sampling. The α-sampling region occupies the border and is indicated as the regions lost after morphological erosion. The β-sampling region occupies the interior regions and is indicated by the regions remaining after morphological erosion. Figure 18 presents a diagram of the differing sampling regions with respect to the initial internal segmentation patches and Figure 19 shows a sample from image data. The α and β values typically used in this thesis are .15 and .1, respectively.



(a) Training patches and their representative model patch $\vec{\mathbf{m}}$

(b) Individual segmentations $\vec{\mathbf{s}}$ of each interpretation associated with $\vec{\mathbf{m}}$

Figure 17. Segmentation information of model patch

[7]This action inspires the model support and activation network techniques introduced in Section 7.3.

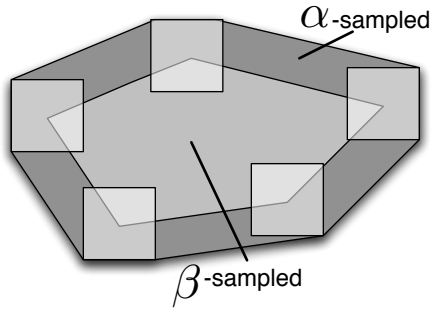
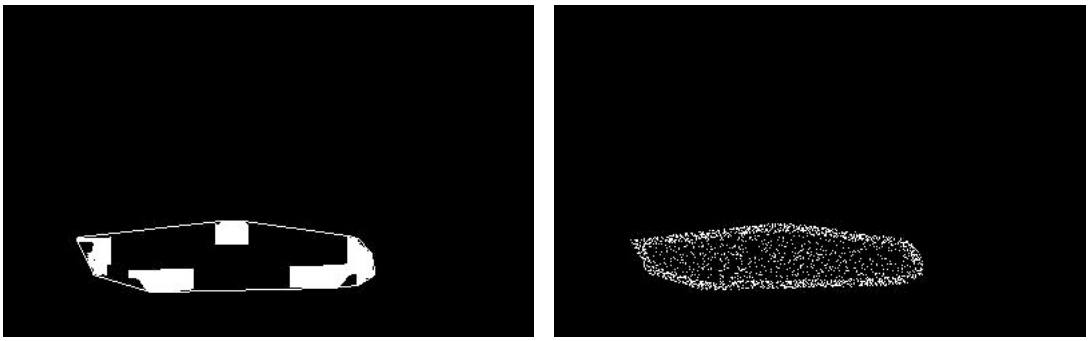


Figure 18. Non-uniform sampling of initial segmentation to augment $I^E|_{(\omega,\vec{\lambda})}$



(a) Segmentation of initial votes (b) Sampling for potential additions to $I^E|_{(\omega,\vec{\lambda})}$

Figure 19. Determining sampling region for refined segmentation

5.4.1 Complexity

The segmentation $p(\mathbf{p} = figure|\omega, \vec{\lambda})$ can be found using $I^E|_{(\omega,\vec{\lambda})}$ with a time and memory efficient algorithm by maintaining accumulators for each of the internal and external representation and segmentations. When using $I^E|_{(\omega,\vec{\lambda})}$, the initial interpretation votes are already known and NGC calculations are not performed. The algorithm runs in $O(i)$ where $i = |I^E|_{(\omega,\vec{\lambda})}|$ represents the number of initial interpretation votes cast. Algorithm 8 formalizes the procedure. The accumulators are initialized to zero. For each interpretation vote, the location from which it voted[8], $\vec{\lambda}_s$, is obtained. The model patch of the interpretation is added to the patch in the internal representation accumulator at the location $\vec{\lambda}_s$. The segmentation patch is added to the patch in the segmentation accumulator at the location $\vec{\lambda}_s$. An additional counter accumulator is maintained and the patch surrounding the location $\vec{\lambda}_s$ is incremented for each interpretation encountered. This achieves a per-pixel count of interpretations occupying the image. The refined segmentation takes substantially longer because it relies on NGC calculations to find matching patches. This operation assumes the same $O(p^2)$ complexity as the initial voting operation described in Section 5.3.6, but is parameterized by the number of pixels to sample and the size of the shape model instead of the number of extracted pixels and the size of the shape model. The number of sampled pixels is significantly larger than the number of extracted patches.

Algorithm 8 Segmentation

Require: $I^E|_{(\omega,\vec{\lambda})}$
Ensure: internalRep, internalSeg, externalRep
  Weight each mask with vote(e) / vote(alle)
  {All accumulators are size of image}
  Initialize countAccumulator, internalRepAccumulator, and internalSegAccumulator
  for all winning interpretations $(\vec{\mathbf{m}}, \omega, \vec{\mathbf{d}}, \vec{\mathbf{s}}) \in I^E|_{(\omega,\vec{\lambda})}$ **do**
    Increment countAccumulator around $\vec{\lambda} + \vec{\mathbf{d}}$
    Add $\vec{\mathbf{m}}$ to internalRepAccumulator around $\vec{\lambda} + \vec{\mathbf{d}}$
    Add $\vec{\mathbf{s}}$ to internalSegAccumulator around $\vec{\lambda} + \vec{\mathbf{d}}$
  end for
  internalRep = internalRepAccumulator ./ countAccumulator
  internalSeg = internalSegAccumulator ./ countAccumulator
  externalRep = image .* internalSegmentation

[8]The subscript s indicates interpretation voting *source*.

6 Algorithm Evaluation

This section describes the procedures used to evaluate the performance of the recognition system. The recognition community agrees that common image sets should be used to compare techniques. For this reason, we use publicly available and frequently reported image sets [1, 31]. Unfortunately, the evaluation of the different algorithms processing the identical image sets often differ, preventing direct comparison. We review the common evaluation methods and advocate strict standards that the recognition community does not consistently embrace.

6.1 Detection Tradeoff

In the object detection task, the Cartesian product of the hypothesis and truth spaces may be partitioned into one of four sets. Each set is defined by its relation to two fundamental sets as described formally below.[9]

$$\begin{aligned} H(\text{image}, \theta) &= \{(r_h, c_h) : \text{algorithm predicts an object in image at location } (r_h, c_h)\} \\ T(\text{image}) &= \{(r^*, c^*) : \text{an object exists in image at position } (r^*, c^*)\} \\ C(\text{image}) &= H \cap T \\ N(\text{image}) &= (H \cup T)' \end{aligned}$$

The set H contains the locations that the detector identifies as an object presence. The set T contains the true locations of object presence. The intersection of these two sets, $H \cap T = C$, contains the correct hypotheses of the algorithm. The set N represents the universe that is not represented in any of these three sets, namely, all possible locations of objects that are not true and have not been hypothesized. Figure 20 illustrates these sets in Venn diagram and confusion matrix forms.

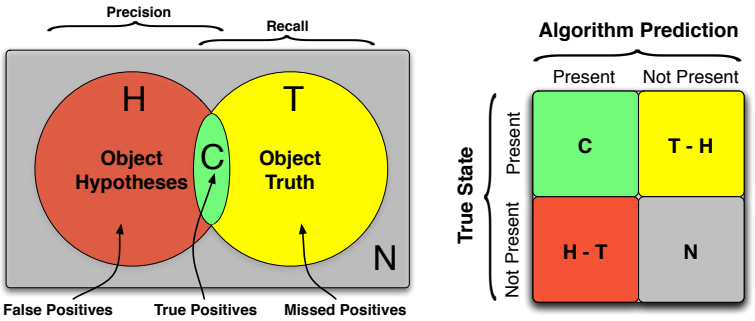


Figure 20. Possible outcomes in detection tradeoff

[9]The symbol * is used to indicate optimality, as used by [16].

6.2 Receiver Operating Characteristic Curve

The set T, containing ground truth, does not change. However, the partitioning of the universe into the remaining three sets is determined by the sensitivity of the detector. This sensitivity is denoted by the parameter θ in the definition for H. All targets may be detected by maintaining a sufficiently high sensitivity. An extreme case would be to identify *everything* as a target by maintaining the highest sensitivity. This causes H to occupy the universe and results in a large number of mistakes. An opposite extreme is to identify nothing as a target, allowing $H = \emptyset$. Certainly, this avoids making a mistake, but the detector fails to be a detector. This tradeoff for a detection system can be described by the Receiver Operating Characteristic (ROC), which is a parametric plot of the correct detections and false detections the system makes when the parameter θ is varied. For each value of θ, the set $H(\text{image}, \theta)$ determines the following quantities that can be plotted as in Figure 21.

$$\text{Correct Detection Rate} = \frac{\|\text{H} \cap \text{T}\|}{\|\text{T}\|} = \frac{\text{Number of Correct Hypotheses}}{\text{Number of Targets in data set}}$$

$$\text{False Detection Rate} = \frac{\|\text{H} - \text{T}\|}{\|N\|} = \frac{\text{Number of Incorrect Hypotheses}}{\text{Number of Non-Targets in data set}}$$

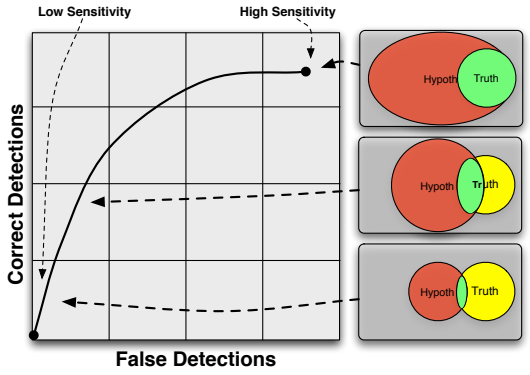


Figure 21. Trade-off values with varying sensitivity

6.3 Ill-Defined Quantity

Although the number of non-targets , $|N|$, is naturally defined for the image classification task [2], it is not well-defined for the object recognition task. In recognition tasks, the number of non-targets, $|N|$, is much larger, causing the false detection rate to appear artificially small.

Many object detection algorithms train a classifier that operates on an image of fixed but arbitrary size. Detection is then performed on an unknown image by extracting all sub-images of the fixed size and feeding it to the classifier, which provides a binary or continuous response. The number of sub-images obtained is often used for the number of non-targets, but this is clearly inappropriate since the number of non-targets would change if the size of the sub-image changed. This metric is characteristic of the internal implementation of the classifier, but $|N|$ should be strictly characteristic of the problem domain itself, i.e. the input. This causes the ROC to be biased in ways that depend on the system implementation [2].

The algorithm evaluation should be independent of the algorithm. Two alternatives are presented in the following sections. The first, proposed by Agarwal and Roth [2], sidesteps the ill-defined value by replacing the false detection rate with the precision metric. The second, developed in this paper, introduces a method to obtain a scale-invariant and algorithm-independent quantification of $|N|$.

6.4 Modified Recall-Precision Curve

Agarwal and Roth [2] propose the use of the recall and precision metrics to avoid the use of the ill-defined quantify $|N|$. Although Section 6.5 presents a sound approach to quantify this value, we present the modified Recall-Precision Curve (RPC) for completeness. The RPC is found in the same fashion as the ROC, except for plotting

$$\text{Recall} = \frac{\|\mathrm{H} \cap \mathrm{T}\|}{\|\mathrm{T}\|} = \frac{\text{Number of Correct Hypotheses}}{\text{Number of Targets in data set}}$$

$$\text{Precision} = \frac{\|\mathrm{H} \cap \mathrm{T}\|}{\|\mathrm{H}\|} = \frac{\text{Number of Correct Hypotheses}}{\text{Number of Hypotheses}}$$

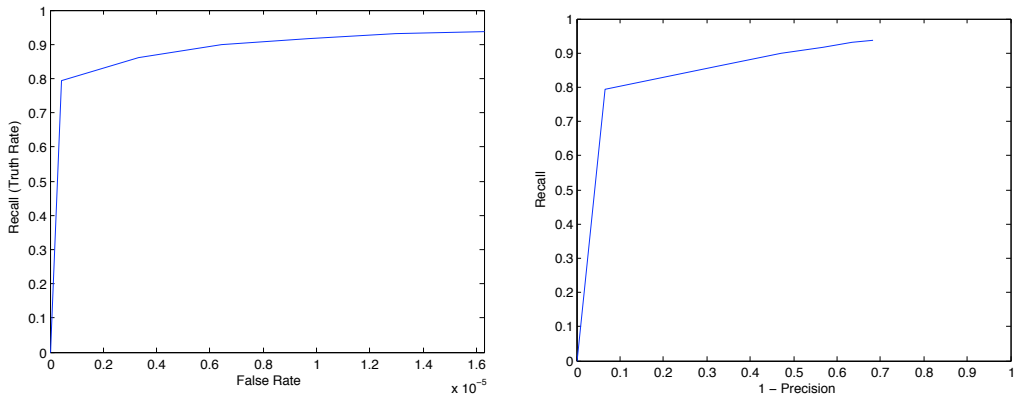
instead of the Correct Detection Rate and False Detection Rate.[10]. Since a high precision is good and a low false detection rate is good, a modified RPC can assume a shape similar to the ROC curve by using plotting the complement

$$1 - \mathit{Precision} = \frac{\|\mathrm{H} - \mathrm{T}\|}{\|\mathrm{H}\|} \tag{39}$$

for the false detection axis.

The various points in the Recall vs. 1 - Precision curve are obtained by varying the activation threshold, just as with finding the ROC. The RPC provides a complete view of the detection trade-off, with both axes spanning the range [0,1], as opposed to the ROC which has a false detection range from 0 to a very small fraction of one. The ROC and Modified RPC are shown in Figures 22(a) and 22(b), respectively.

[10]The Recall metric is the same as the Correct Detection Rate, but have different names in different problem domains.



(a) ROC Curve (b) RPC

Figure 22. Detection Performance on UIUC Set

6.5 Regions of Tolerance

The number of pixels in the input image could be used to estimate $|N|$ instead of the number of extracted sub-images. However, the false detection rate would become artificially small if the image is scaled to a larger image, since the number of targets remains the same while the number of non-targets increases. This can be corrected with the notion of Regions of Tolerance (ROT).

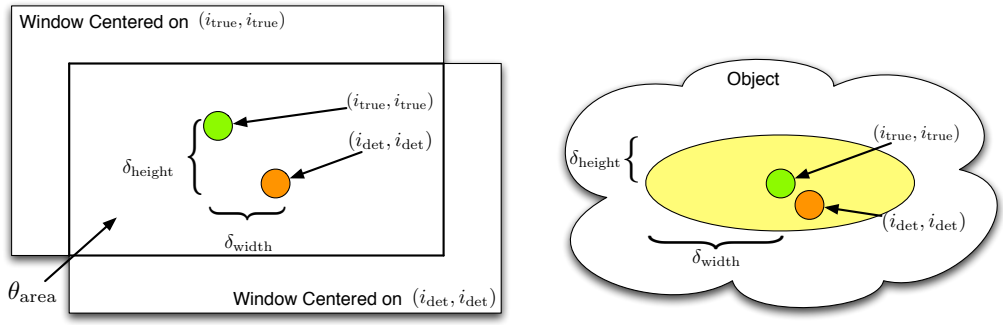
A Region of Tolerance is an implicit assumption when evaluating hypothesis locations. If the detector provides a hypothesis that is 2 Manhattan-units away, should it be considered a hit or a miss? Typically, this small distance is arguably insignificant and the hypothesis should be considered a hit. But at what distance from the truth does the hypothesis become a miss? This is clearly application dependent. Two factors contribute to this decision. First, the size of the input images affects the tolerated distance. Second, the shape of the target category is also important. If the ROT of the target category is circular, Euclidean distance is an appropriate measure. If the ROT of the target category is not radially symmetric, more detailed representations of what locations surrounding the true location should be acceptable. A reasonable extension, presented by Agarwal and Roth [2], is to use an ellipsoidal volume surrounding the true location. An image has a set of true locations $\vec{\lambda}^* = (i^*, j^*)$ and the algorithm provides a set of object hypothesis locations $\vec{\lambda}_h = (i_h, j_h)$. An object hypothesis is deemed correct if

$$\frac{|r_h - r^*|^2}{\alpha_1^2} + \frac{|c_h - c^*|^2}{\alpha_2^2} \leq 1 \tag{40}$$

where α_1 and α_2 define the major and minor axes of the ellipse.

We propose to formalize the implicit assumption used by virtually all recognition practitioners into an explicit and fundamental component of algorithm development and evaluation. Using a more complex model for the ROT requires additional information during the

ground-truth creation process. In the model presented above, truth is uniquely defined with four values: 1) the row and column of the ROT center and 2) the sizes of the major and minor axes forming the ROT. The value $|N|$ can now be the number of non-target pixels in the image outside of the ROT. This metric enables a scale-invariant and algorithm-independent quantification of the false detection rate. Figure 23 shows the evolution of the parametric ROT in recent recognition literature.



(a) 2002 ROT by Agarwal and Roth [2] (b) 2004 ROT by Agarwal and Roth [1]

Figure 23. Evolution of a parametric model for Regions of Tolerance

More complex models for ROT would require more parameters. The most complex ROT would be a binary mask indicating which pixels the object occupy. This would require a single parameter for every pixel in the image. Considering the evaluation of the detection algorithm within this ROT framework shows how the segmentation procedure is reduced to a complex detection procedure by *detecting* all pixels the object occupies. Further, if a ROT is represented by a binary mask, why would a single location in the ROT be acceptable? When viewing a detector as a segmenter, why is the single-pixel performance acceptable when there are considerably more pixels the object occupies?

6.6 Object Hypothesis Equivalence

When given the set H of hypothesis locations and the set ROT of regions of tolerance for each object in the image, different methods for determining the size of the set C and the number of false alarms $|H - T|$ are reported in the literature. Often, if two object hypotheses land on the same object, both are considered to be correct detections. This is expressed formally in Equation 41. Using this method, an algorithm is unjustly rated better. Considering both as correct detections unjustly bolsters the accuracy metric. Although this bolsters the performance measure, it is fundamentally incorrect since it is unable to recognize the equivalence of two distinct hypotheses. This behavior should reflect a poorer rating. A more appropriate measure is to consider only a single hypothesis as a correct detection and force the remaining hypotheses in the same region to be false detections. This is formalized in

Equation 42. A more appropriate measure is to only permit a single hypothesis be correct and force remaining hypotheses in the same region to be false detections. Although this hinders the reported performance of the algorithm, pretending it does better than it really does will not aid in the advancement of recognition research.

$$\hat{C} = H \cap T = \{\lambda_2 : \lambda_2 \in H \wedge \lambda_2 \in \bigcup_{i=1}^{|T|} ROT_i\} \quad (41)$$

$$C^* = H \oslash T = \{\lambda_2 : \lambda_2 \in H \wedge \lambda_2 \in ROT_i \wedge \nexists(\lambda_1 \in H)[\lambda_1 \in ROT_i]\} \quad (42)$$

6.7 Data Sets

The Leibe image set[32] contains 100 high-resolution images of 50 cars. Each car image is reflected horizontally. Each image contains one car, which assumes most of the image. Each image is accompanied by a segmentation mask indicating the object region. This image set is used to create the shape model and activation network. Figure 24(a) shows a sample from this data set. The UIUC image set[1] contains 170 images of 200 cars. Each image contains one to three cars. Figure 24(b) shows a sample from this data set. The McEuen image set[39] contains 143 images. Each image contains one to three vehicles, where the vehicle may be a car, truck, van, or station wagon. Figure 24(c) and 24(d) show samples from this data set. The drive-by with pan video contains 326 frames. A frame contains zero to two vehicles. The vehicle was either a car or a SUV. The camera pans to the right using a tripod. Figure 24(e) and 24(f) show samples from this data set. The drive-by stop sign video contains 45 frames. A frame contains zero to one car. The camera is stationary and the car occupies a large portion of the frame. Figure 24(g) shows a sample from this data set.

(a) Leibe image set

(b) UIUC image set

(c) McEuen image set

(d) McEuen image set

(e) Panning video

(f) Panning video

(g) Stop sign video

7 Shape Model Enhancements

There are two motivating concerns when pursuing improvements for the approach described in Section 5. The first is performance quality and the second is computational cost. As discussed in Sections 5.3.6 and 5.4.1, there is an inherent computational cost attributed to the patch-matching framework. Although quality and cost often exist within a trade-off, there exist improvements to both that do not compromise either. This section describes how computational effort may be focused to more quickly achieve improved discrimination and segmentation. These improvements are especially useful when motivated to apply the algorithm to video sequences, where interpretation involves more than independently processing each frame.

7.1 Characterization of Model Dynamics

The shape model is a collection of interpretations. Each interpretation embodies an implication indicating object presence. Many interpretations within the model are grouped by similar receptive fields, which are represented by model patches. This grouping may be illustrated in matrix form, as shown in Figure 24. When a model patch matches an input patch, the interpretations associated with it are cast. Some of these interpretations may correctly coincide with the true state of nature and some may be incorrect. Interpretations may contribute during other processes by contributing segmentation refinement (Section 5.4) or coactivation support (Section 7.3). As a result, any particular model patch may assume one or more roles during the localization, recognition, and segmentation processes. We introduce a characterization framework that provides insight of the model dynamics during these processes.

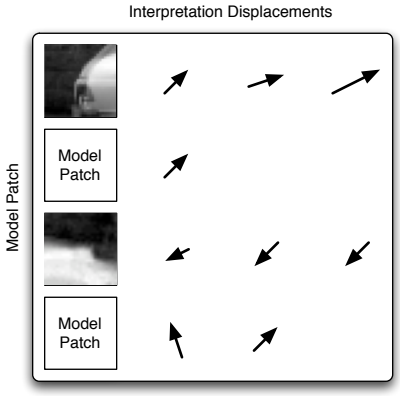


Figure 24. Illustration of shape model. All interpretations are grouped by their common model patches.

7.1.1 Interpretation Tally

An interpretation tally is created for each interpretation the model may provide. Let $m = |M|$ be the number of model patches in the shape model and $i = \arg\max_x(|I_{\vec{\mathbf{m}}_x}|)$ be the most number of interpretations any single model patch has. Then the interpretation tally is a $m \times i$ matrix where each element represents a different interpretation. An interpretation tally for the sample shape model in Figure 24 would result in a 4×3 matrix. A separate interpretation tally is created to record interpretation participation in one of the possible roles described above. This includes CASTING, CORRECT, and SUPPORTING. When an interpretation is cast during object localization, the tally for that interpretation is incremented in the CASTING tally. When an interpretation was correctly cast, the corresponding CORRECT tally is incremented. The CORRECT tally can be obtained during the same secondary training process used to create the activation networks, which is described in Section 7.3. When an interpretation provides coactivation or segmentation support, the corresponding SUPPORTING tally is incremented. The interpretation tallies provide a quantitative description of the activity exhibited by any particular interpretation or group of interpretations.

7.1.2 Model Patch Role Classification

Interpretations are represented in a three-dimensional space and grouped into eight general classes depending on their relative contribution for each of the three roles in the algorithm. These classes are illustrated in Figure 25. Four desirable role classifications exist. The ideal interpretation, known as the *Super Star*, frequently votes, is frequently correct, and frequently supports other interpretations. Two other reasonable roles are the *Hot Shot* and *Wingman* classes. The Hot Shot frequently votes and is frequently correct, but does not support other interpretations. The Wingman is complementary to the Hot Shot because it rarely votes by itself but frequently and correctly supports other votes. The final desirable role classification is the *Wise Man* class. This class rarely votes, but when it does it is correct. It also correctly supports other votes. The four remaining role classifications are less desirable because of their low correctness frequency. The *Social Fool* is a Super Star patch that is never correct. This means that it votes and supports a lot but is rarely correct. The *Supportive Fool* is a Social Fool that does not vote frequently by itself. The *Reclusive Fool* is a Social Fool that rarely supports other votes. The *Dormant* interpretations are those that rarely vote, are rarely correct, and rarely support others.

7.1.3 Profile Axes

Activity alone may not be a strong indicator of the quality or representative capability a particular patch has with respect to the target category. Although interpretations may be partitioned by their activity and correctness, they may also be characterized in a variety of

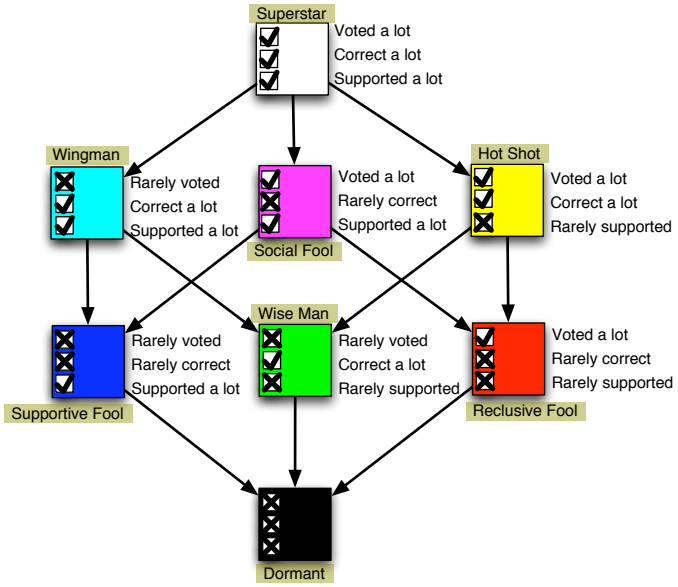
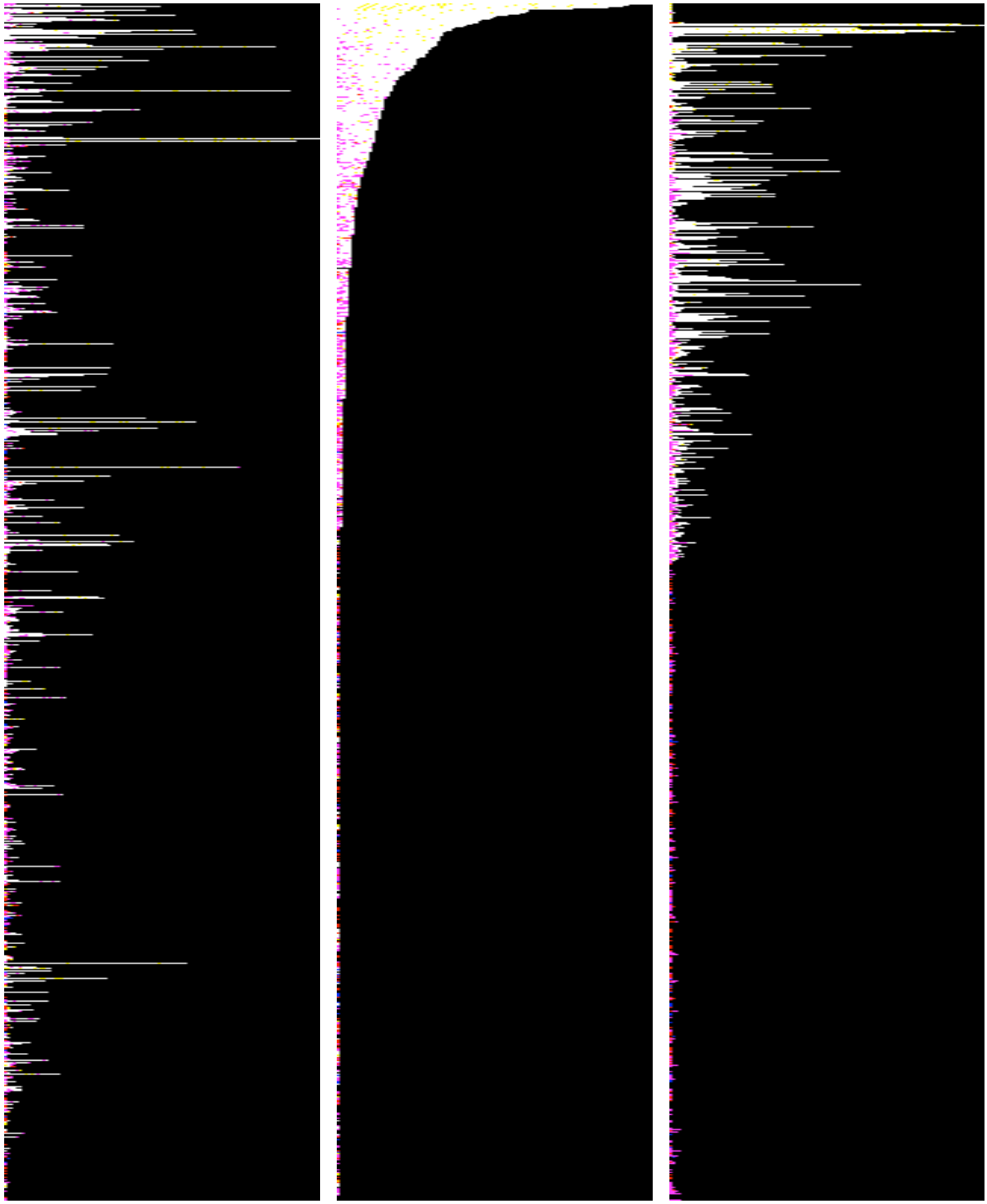


Figure 25. Interpretation classifications

other ways. There are a variety of implicit orderings for the shape model components and each may be used to create an explicit ordering to rearrange the shape model with no change in performance. The initial ordering of the shape model reflects the exposure order during creation. This is shown in Figure 26(a).

Defining an explicit ordering may be considered a different axis around which the shape model may be rotated. An axis may reorder the rows or columns of the shape model. Reordering the rows addresses the interpretations grouped by their common model patch. Figure 26(b) shows this along the *number of interpretations* axis. Reordering the columns addresses the individual interpretations within each grouping. The *probability of correctness* axis may be used to relocate the reliable interpretations to the beginning of the shape model. Figure 26(c) shows the model along this axis. The probability of correctness may be obtained by dividing the CORRECT tally by the CASTING tally and represents how likely the interpretation will be correct when cast. Computational costs may be reduced with minimal localization efficacy loss by truncating the end of the shape model. This can be viewed as model pruning. Any other axis may be defined to suit individual investigations. These may include the number of interpretations associated with each patch, the dominant vote angle direction, the entropy of vote angle directions, the displacement magnitude, the percentage of foreground information, or the class representativeness measure presented by Borenstein and Ullman in [7].



(a) Unsorted shape model (b) Sorted by $|I_{\vec{\mathbf{m}}}|$ (c) Sorted by $p(i \in I_\omega$ is correct)

Figure 26.

7.2 Imposed Localization Constraints

7.2.1 Vote Weighting

The probabilistic framework presented in Section 5.3 assumes a uniform distribution for the factor $p(I_i^{\vec{e}}|\vec{e})$ to simplify several derivations. The characterization framework introduced in this section allows us to easily demonstrate the empirical nonconformance of this assumption. The empirical distribution of $p(I_i^{\vec{e}}|\vec{e})$ shown in Figure 27 is clearly nonuniform. Instead of forcing the collected data to fit a specific statistical model, the interpretation tally becomes a non-parametric model that accurately and efficiently embodies the distribution during localization. This probability distribution replaces the $\frac{1}{|I^{\vec{e}}|}$ assumption used in Section 5.

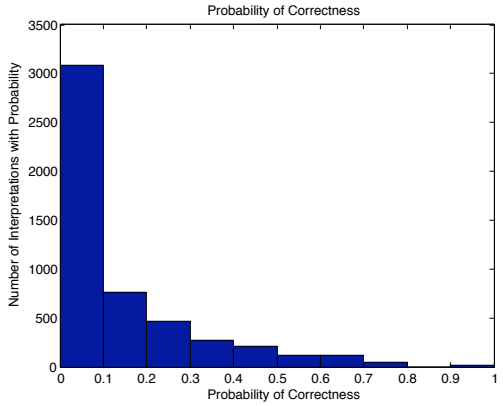


Figure 27. Empirical $p(I_i^{\vec{e}}|\vec{e})$

7.2.2 Background Information in Extracted Patches

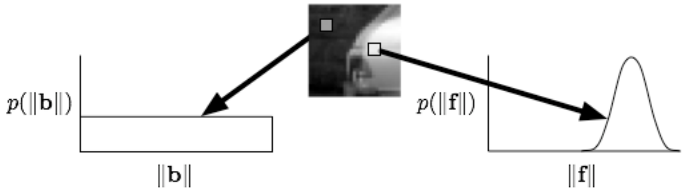


Figure 28. Underlying distributions of background and foreground

Section 5.3 discusses a restriction on the region from which the training patches may be extracted. This was done to ensure that part of the model patch would contain object information. The inclusion of background information in model patches is necessary, since contour information of the target category shape occurs at the boundary edges. However,

background information raises concern during the patch-matching localization process. Consider the model patch in Figure 28 with two pixels **f** and **b** existing at positions on and off the object, respectively. The background pixel will vary more and approximates a uniform underlying distribution. The foreground approximates a Gaussian underlying distribution that is likely centered at a mean different than the uniform mean. The expected values of these two distributions are different. If we look at the converging value of each pixel as more patches are added to the cluster, we see an emerging distinction:

$$\begin{aligned} \lim_{\|C_i\|\to\infty} \|\mathbf{b}\| &\neq \lim_{\|C_i\|\to\infty} \|\mathbf{f}\| \\ \mathrm{E}[\mathrm{uniform}(\|\mathbf{b}\|)] &\neq \mathrm{E}[\mathrm{normal}(\|\mathbf{f}\|)] \\ 128 &\neq \mu \end{aligned}$$

The uncorrelated background information in the non-object region of the model patch reduces the ability to agglomeratively cluster during shape model creation. It also inhibits the ability to match a model patch to a novel patch during localization. We can leverage the knowledge of what happens to the background pixel as the size of the cluster increases by weighting them towards this expected value. This neutralization can be expressed as

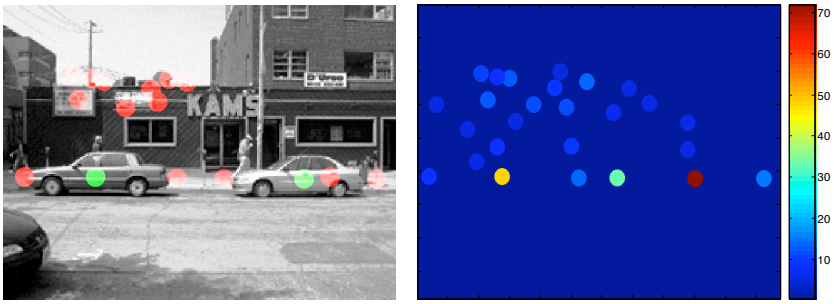
$$\text{patch} = p(\text{foreground}) * \text{patch} + p(\text{background}) * 128 \tag{43}$$

Patches with neutralized values are more likely to merge during model creation and are more likely to match novel input. The $p(\text{foreground})$ and $p(\text{background})$ factors in Equation 43 can be found via the segmentation occurrence $\vec{\mathbf{s}}$ obtained during training.

An understanding of what occurs as the size of the cluster tends to infinity does not aid an agglomerative process that reflects local hill climbing. The initial neutralization process imposes too much unnatural modifications by forcing the background information to assume a predefined value. The neutralization method should set all background patches to a value such as the mean value of the present background pixels. This neutralization also incorrectly assumes that all interpretations for the model patch correspond. Figure 17 demonstrates that this is clearly not the case. Accounting for this observation would require different background pixels to be neutralized for each interpretation and would require $\arg\max_x(|I_{\vec{\mathbf{m}}_x}|)$ times more patch-matching operations during localization because the model patch is no longer able to represent all interpretations it is associated with.

7.2.3 One Shot Rule

Section 5 discusses why dense regions in the vote space $p(\omega, \vec{\mathbf{x}})$ imply target category detections. As seen in Figure 29(a), more dense regions are identified even though many fewer instances are present. Because of this, locations provided by the mean-shift-clustered interpretation votes are considered *candidate locations*. These locations require additional



(a) Candidate locations highlighted on original image

(b) Candidate locations strengths

Figure 29. Initial strengths of candidate hypotheses

deliberation to determine which, if any, should be asserted as detections. Naturally, the strongest candidate hypotheses are likely, but how strong must a hypothesis be to assert a detection? Figure 29(b) illustrates the vote strengths for the example discussed in Section 5. It shows two strong candidates at locations corresponding to the two cars in the image. The third-largest candidate does not correspond to a category instance, but is appropriately strong. The shape model exhibited a false holistic interpretation between the two true cars because of the back wheel of the left car and the front wheel of the right car.

Clark [12] refers to these false strong hypotheses as *ghosts*, observed that any single patch within an image should contribute to only a single holistic interpretation, and addresses the issue with a vote-withdrawing technique that we term *starving*. Even for visual illusions such as the face-vase illusion, only a single recognition is perceived at any given moment. Because all interpretations associated with the matching model patch are cast, the remaining votes not corresponding to the dominant global interpretation should be disregarded. The diagram in Figure 30 shows how alternative interpretations become candidate hypotheses by aggregating moderate but insufficient weight. It also shows why the strongest hypothesis will not be the ghost, since additional patches independently contribute to the non-ghost holistic interpretation. Figure 31 illustrates the first two steps of the vote starving technique, while Figure 32 shows the hypothesis strengths before and after the starving process. Unfortunately, one of the correct hypotheses was starved to a weakened undetectable state and results in a missed positive. This demonstrates that the starving technique alone is insufficient for distinguishing candidate hypotheses.

7.3 Activation Networks

Although Clark [12] refers to false strong candidate hypotheses as *ghosts*, we prefer to consider them alternative holistic interpretations caused by an insufficient subset of local interpretations. The necessary subset of local interpretations minimally describes the appearance

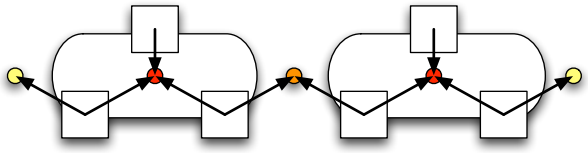
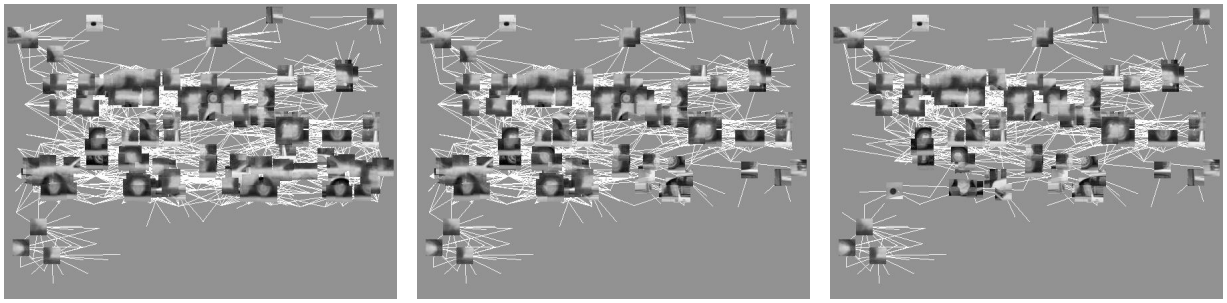
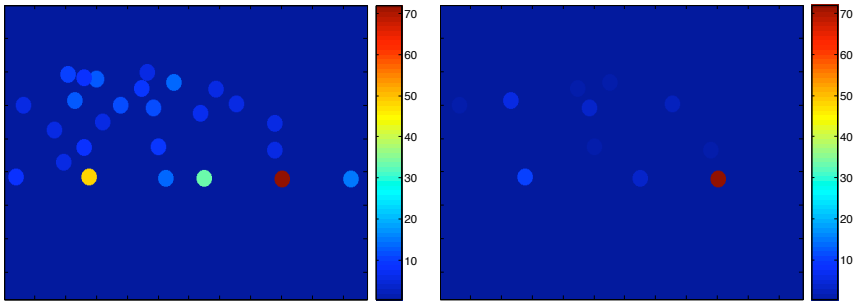


Figure 30. One Shot Rule



(a) Initial matching interpretations (b) Starved interpretations with strongest removed (c) Starved interpretations with second-strongest removed

Figure 31. Starving votes



(a) Initial vote weights (b) Starved vote weights

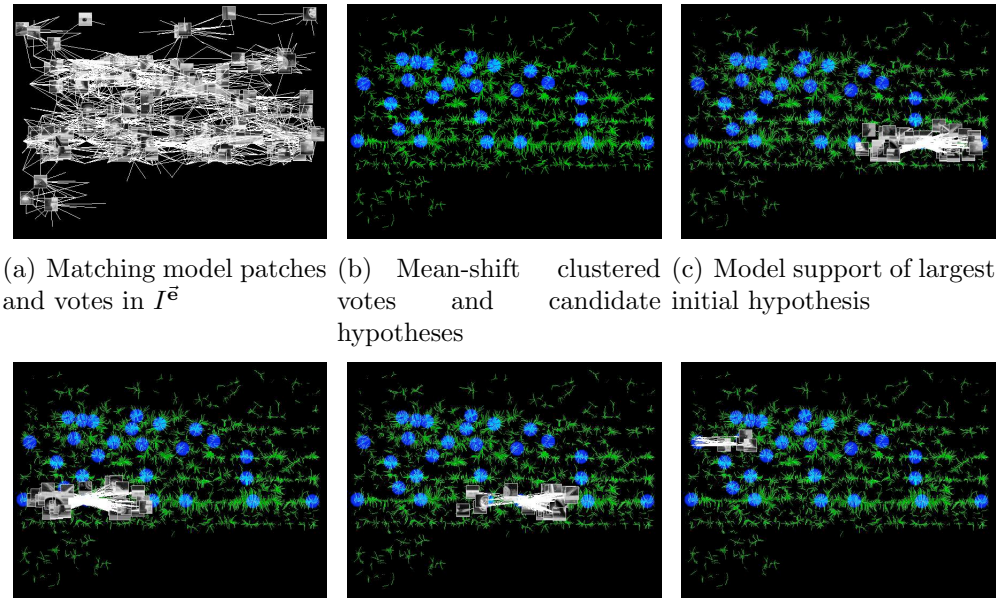
Figure 32. Starving results

of the target object, including the relative location of each fundamental component. For the example illustrated in Figure 29, the "necessary" components that were not present for the false hypothesis were the horizontal gradients indicating the roof and bottom of the car. The original method presented in Section 5.3 relies on a general agreement of independent actions by multiple interpretations and is incapable of expressing these higher-level interactions. Although all cast interpretations are appropriate at the local level, many do not continue to be appropriate at the global level because additional and necessary structures fail to support them. This section addresses the challenge of representing the necessary relations between components without manually asserting their existence for each target category. We present an approach that implicitly incorporates structural dependency information to improve the object hypothesis decision. This new process additionally reduces the reliance of unstable corner detection, extends the original probabilistic framework presented by Leibe, and can leverage the starving technique introduced by Clark.

Models representing sub-parts of an object often include the geometric relationships between the parts. Fergus et al. [18] describe the geometrical relationships as a *constellation of parts* and argue shape is represented by the mutual position of these parts. Their chosen parts are represented by a point in an arbitrary appearance space, while a representation closer to human perception is preferred. Human models of representation consist of a group of units, broadly tuned to representatives of the object class, that code for the identity of a particular object by their *combined activation pattern* [45]. Wolf describes psychophysical experiments showing that preattentive object descriptions consist of a *collection of isolated features* [58]. Serial attention is then necessary to represent shape relationships and integrate these features into a *common object description*. This agrees with the whole-object detection described by Rensink [45], where the nexus embodies the whole object perceived at any given moment via *connections to its parts*. The coherence field represents a local hierarchy with object- and part-level descriptions that is an extremely useful device and a natural way to represent objects [45].

Each interpretation in the shape model embodies an implication indicating the location of the object if the corresponding model patch matches the unknown input. If the evidence patch $\vec{\mathbf{e}}$ is similar to the model patch $\vec{\mathbf{m}}$ at the location $\vec{\lambda}$, then there is an object of type ω at position $\vec{\lambda} + \vec{\mathbf{d}}$. When given an object hypothesis $\vec{\lambda}_h$, the implication of the interpretation may be reversed to assert that the region surrounding $\vec{\lambda}_h - \vec{\mathbf{d}}$ should be similar to $\vec{\mathbf{m}}$. This alternative perspective allows all of the shape model to be relevant, instead of just the subset $I^{\vec{\mathbf{e}}}$. The shape model may now be used to *support* the candidate hypotheses $\{\vec{\lambda}_h\}$, as illustrated in Figure 33. Part *a* shows the internal representation of $I^{\vec{\mathbf{e}}}$, part *b* shows the candidate hypotheses after the interpretation votes were mean-shift clustered, parts *c*-*e* show the model support for the strongest three candidate hypotheses, and part *f* shows the model support for the weakest candidate hypothesis.

Two characterizations demonstrate the shortcomings of this model support technique. The scatter-plot in Figure 34 shows the high correlation between the initial and model



(a) Matching model patches and votes in $I^{\vec{e}}$

(b) Mean-shift clustered votes and candidate hypotheses

(c) Model support of largest initial hypothesis

(d) Model support of second-largest initial hypothesis

(e) Model support of ghost hypothesis

(f) Initial hypothesis that does not have large model support

Figure 33. Imposing the model support onto the initial hypothesis locations

support strengths. This should not be a surprise, since both metrics are provided by the same model. Many of the supporting interpretations are identical to the interpretations in the initial votes. The mean-shift clustering slightly relocates the object hypotheses, causing a portion of the initial interpretation votes to be lost and a portion of unique supporting votes to be added. This behavior is the source of high correlation. An illustration of the shape model, shown in Figure 35(a), should represent the ideal target object. The model does not appear as any single instance of the category it represents. This indicates that the use of the entire shape model for localization and support is inappropriate. Only specific subsets of the shape model should occur together, while other subsets should never occur together. One instance of an appropriate subset of interpretations is shown in Figure 35(b).

The appropriate subsets of the shape model are the interpretations that correctly occur together when observing a target instance. These relations are represented by an *activation network*. While the interpretation consolidation described in Section 5.2 combines model patches based on visual similarity, the creation of the activation network relates interpretations based on appropriate coactivation. This leverages mutual relationships between the interpretations instead of maintaining the original assumption of independence. Neurophysiological motivation for expressing these relationships is the phenomenon of long-range interactions between neural cells in the human visual system [21]. The activation network contains coactivation relations between individual interpretations reinforced during an aug-

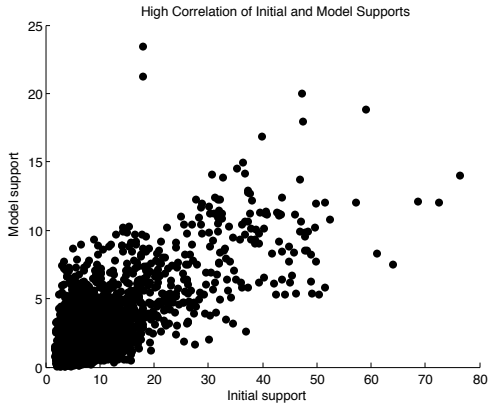
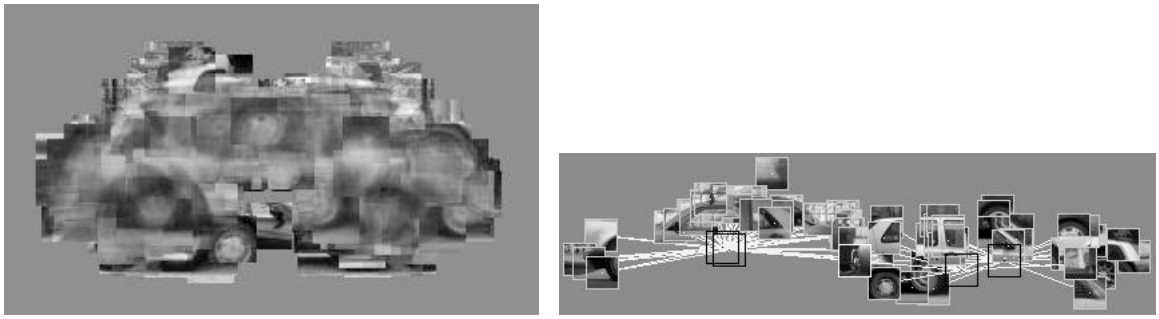


Figure 34. High correlation between initial and model support



(a) Entire shape model (b) Specific subset of shape model

Figure 35. Use of the entire shape model in 35(a) is inappropriate for matching any single category instance, while specific subsets of the shape model represent appropriate instance-level relations. 35(b) shows one such subset.

mented training phase. The training phase performs the localization, including the patch voting and mean-shift clustering. All interpretations voting for a candidate hypothesis that corresponds with a true object location are collected. The relative spatial arrangements of these correct interpretations are added to the activation network. The learned co-activation relations for a given target category[11] may be described by the set

$$\Psi_\omega = \{(i, \vec{\mathbf{m}}_j, \vec{\delta}) : i \in I_\omega, \vec{\mathbf{m}}_j \in M, \vec{\delta} \in \Delta\} \tag{44}$$

where I_ω, M, and Δ are defined in Section 5.2. When $\vec{\mathbf{m}}_i$ matches an input patch $\vec{\mathbf{e}}$ at location $\vec{\lambda}_i$, model patch $\vec{\mathbf{m}}_j$ should be matched against the patch existing in the test image at position $\vec{\lambda}_i + \vec{\delta}$. The coactivation relation is learned symmetrically and is translation-invariant, so the relation may be used independently of the learned location. Figure 36 shows how a coactivation relation is learned with a training image. The filled circles indicate corner locations. Model patches 2 and 47 match at these locations and cast interpretation votes 2α, 2β, 47α, and 47β for the object center. Since 2α and 47β correctly agree on the object center, the displacement between them is stored. Only the correct interpretations cast by the model patches form a coactivation relation.

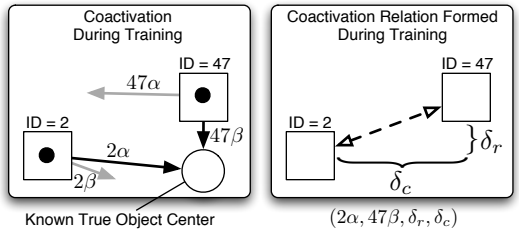
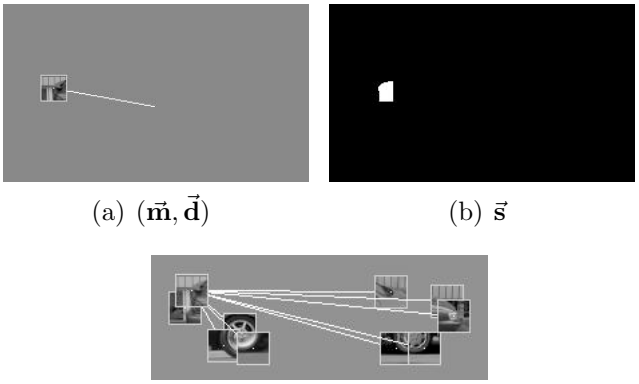


Figure 36. Learning activation network

Inspection of the activation networks reinforces the nomenclature used to describe the vote displacements associated with a model patch. Figure 37 shows a model patch with a single interpretation. Figure 38 shows a model patch representing two similar interpretations of a wheel. The model patch may be interpreted as a back wheel of a car facing to the left, or it can be interpreted as the front wheel of a car facing to the right. Figures 39 and 40 show a model patches with three distinct abstract groupings of interpretations. The interpretation in Figure 39 can be the bottom part of a bumper on the right side of the car, it can be the light part of the front of a car facing to the left, or it can be the roof of a gray car. The local interpretation is globally consistent with three separate subsets of interpretations and assumes different roles within each context.

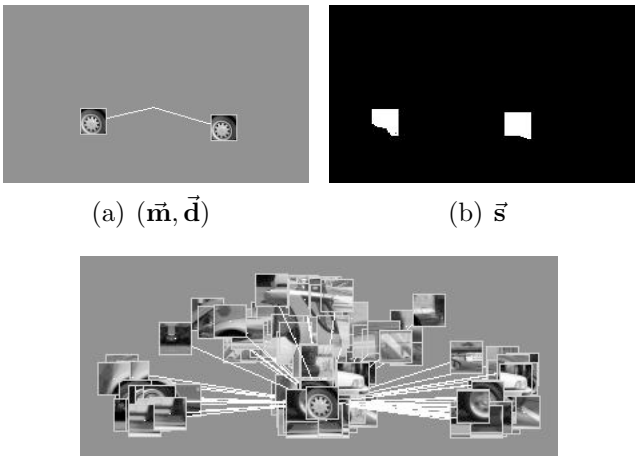
[11] The letter Ψ is used for its visual similarity with multiple interpretations voting towards a central location



(a) $(\vec{\mathbf{m}}, \vec{\mathbf{d}})$ (b) $\vec{\mathbf{s}}$

(c) Co-activations $\Psi_{\vec{\mathbf{m}}}$

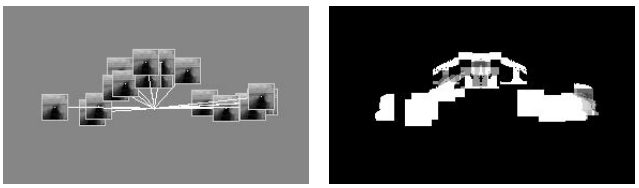
Figure 37. Model patch $\vec{\mathbf{m}}$ with a single interpretation $i \in I_\omega$



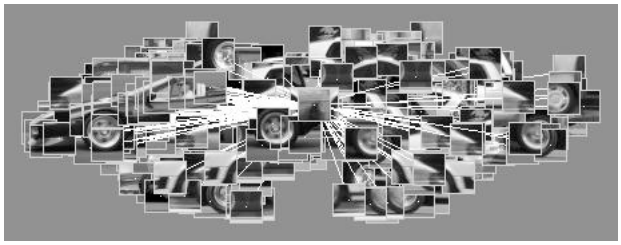
(a) $(\vec{\mathbf{m}}, \vec{\mathbf{d}})$ (b) $\vec{\mathbf{s}}$

(c) Co-activations with $\Psi_{\vec{\mathbf{m}}}$

Figure 38. Model patch $\vec{\mathbf{m}}$ with two Interpretations $I_{\vec{\mathbf{m}}} \in I_\omega$

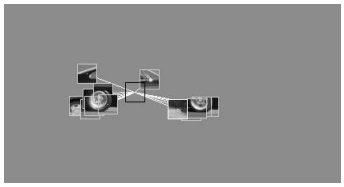


(a) $(\vec{\mathbf{m}}, \vec{\mathbf{d}})$ (b) $\vec{\mathbf{s}}$

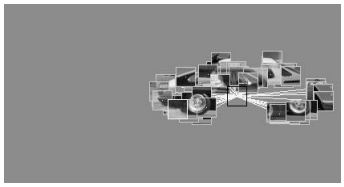


(c) Co-activations $\Psi_{\vec{\mathbf{m}}}$

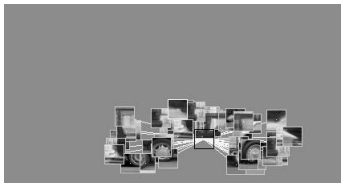
Figure 39. Interpretations associated with $\vec{\mathbf{m}} \in I_\omega$



(a)



(b)



(c)

Figure 40. A model patch with three interpretations. Each interpretation has a separate activation network. The model patch is centered on each image and the object center vote is highlighted with a black square. The activation network shows the relative locations of other patches that should agree with the object center vote.

7.3.1 Localization Supplementation

The activation network can provide top-down perceptual grouping to supplement the initial holistic interpretations associated with the candidate locations. Localization is aided by allowing a matching patch to trigger an investigation of patches in specific relative locations that were not identified by the corner detector. This leverages the knowledge of a previous co-occurrence to augment the current localization. Figure 41 shows a model patch matching the input patch at one of four detected corners. The matching model patch casts all interpretations associated with it. All coactivations associated with interpretation 47β are obtained from the activation network. In this case, the displacement (δ_r, δ_c) identifies the patch in the unknown image that model patch 2 should be compared to. If they do match, interpretation 2α supplements the original set $I^{\vec{\mathbf{e}}}$. If they do not match, the interpretation can assume an inhibitory role. The interpretation 47α does not trigger an investigation of model patch 2 because the activation network is stored at the interpretation level and not the model patch level. Figure 42 illustrates why the activation network needs this higher level of granularity. The candidate hypothesis A should be preferred over candidate hypotheses B or C due to the presence or absence of coactivation support. Figure 43 compares the internal representations of the initial interpretations $I^E|_{(\omega,\vec{\lambda})}$ and the coactivation interpretations. Figure 44 compares the coactivation vote strengths with the starved coactivation vote strengths.

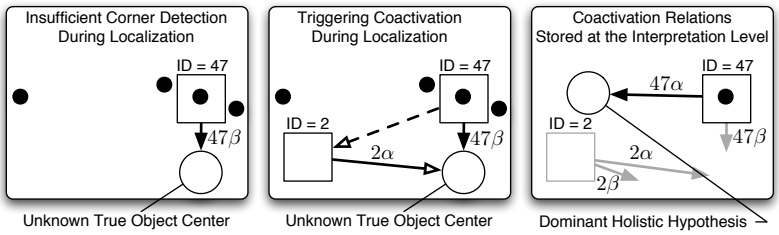


Figure 41. Using activation network

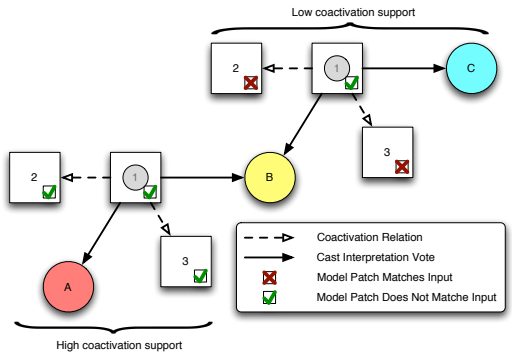
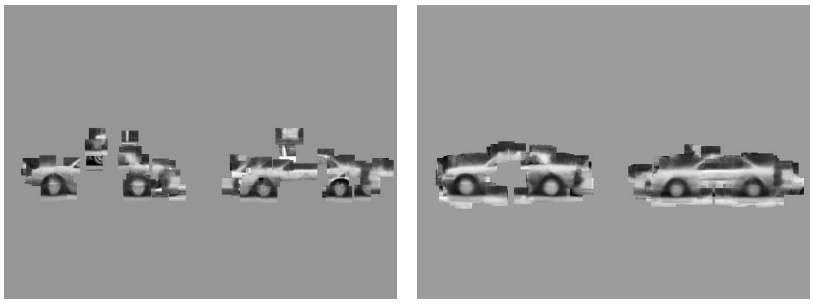


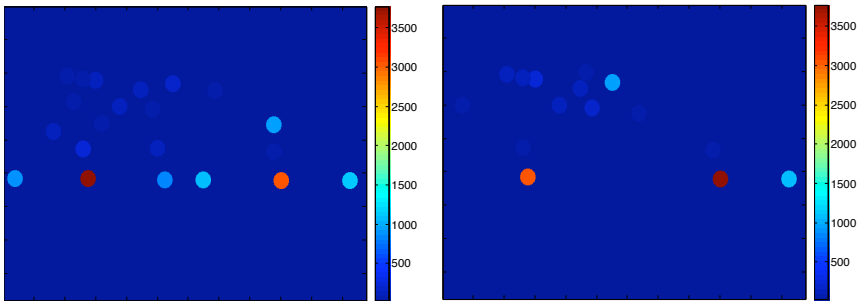
Figure 42. Coactivations are stored at the interpretation level and not the model patch level



(a) Internal representation of original interpretations

(b) Internal representation of coactivated interpretations

Figure 43. Supplementing localization



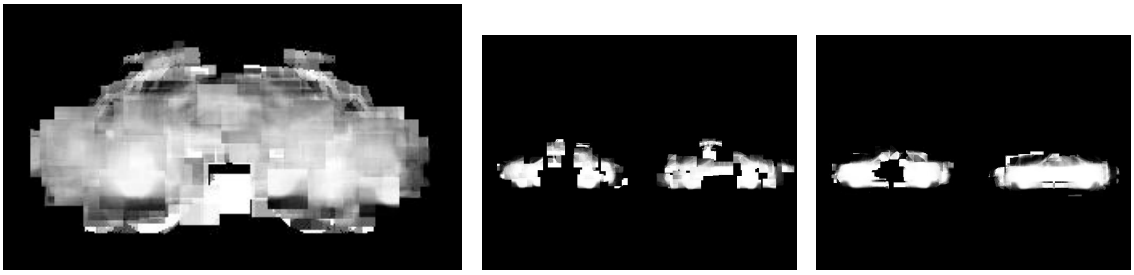
(a) Coactivation support

(b) Starved coactivation support

Figure 44. Performing the starving technique on the coactivated responses eliminates ghosts

7.3.2 Directed Segmentation

The activation network can provide top-down perceptual grouping to supplement the segmentation task. We argue and demonstrate that restricting the shape model to the appropriate interpretation subsets enhances the localization process. Figure 45(a) shows the segmentation provided by the entire unconstrained shape model. The reasons motivating use of the activation network for localization also motivate its use for the segmentation process, since the appropriate subsets of interpretations are learned and inappropriate support is avoided. Figure 45 compares Leibe's uniform sampling segmentation with the segmentation achieved by coactivation segmentation. The disjoint components of the segmentation in Figure 45(c) are caused by sparse interpretation cover along the border and can be resolved with methods such as Borenstein's Optimal Cover [7].



(a) Segmentation provided by entire shape model. Non-uniformity argues for use of activation network to restrict segmentation to appropriate subsets.

(b) Internal segmentation from uniform sampling

(c) Internal representation from coactivated responses

Figure 45. Using the activation network to efficiently guide segmentation

7.3.3 Supplementing Motion Cues

Motion information provides another level of visual information that can not be obtained from an individual image. The activation network can provide top-down perceptual grouping to supplement motion cues during tracking tasks. A shape model interpretation embodies an implication indicating object position. A coactivation relation $(i, \vec{\mathbf{m}}_j, \vec{\delta})$ embodies an implication indicating what another component of an instance should look like at a specific relative location. Point-level motion detection algorithms not relying on background models integrate information over a small local region, where significant intensity differences over time indicate a leading or trailing edge of a moving object. The coactivation relation overcomes the aperture problem by integrating over the appropriate neighboring and long-range apertures to enclose the tracked object. This enclosure is triggered by the matching patches centered on detected corners. The position invariance of the activation network maintains

accuracy through camera movements. Corner detection instability is also overcome, although more complex space-time interest points such as those by [30] may be explored.

Temporal coherence is an additional feature of the input just like edges, colors, and textures. Leveraging this feature prevents an algorithm from "starting fresh" at each frame. The subset of interpretations that are activated in previous frames can be used to create an Activated Shape Model (ASM) to represent the current expectations of the attended object.

7.4 False Alarm Reduction via Supervised Classification

The localization process provides more hypotheses than the number of present objects. This causes a large number of false alarms. The algorithm must determine which candidates, if any, should be asserted as final hypotheses. The original shape model approach provides only a single metric to discriminate between detection and non-detection. The low dimensionality of the hypothesis representation hinders separability. This section identifies additional metrics used to characterize an object hypothesis.

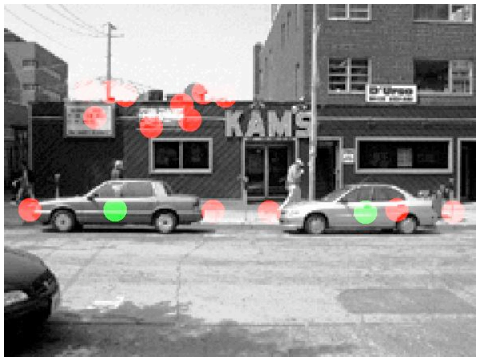
The candidate hypotheses generated during localization of the UIUC image database are partitioned by detection and non-detection to allow supervised learning and classification. A two-dimensional example is presented in Figure 46(a). The true positives are highlighted in green and the false positives are highlighted in red. Figure 46(b) shows these same hypotheses in feature space, where the Filled Green Circles correspond to the true locations and the Red Squares represent the false locations. This operation is performed for all UIUC images using all of the axes described in the following sections. Any number of classification algorithms existing in the machine learning community may be used to discriminate between these two classes. Some of these algorithms are described in Section 2.1.

7.4.1 Initial Support

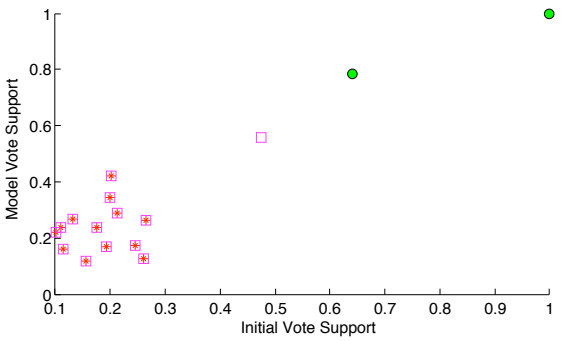
The magnitude of the vote mass accumulated during the patch vote and mean-shift clustering is referred to as the initial support. Candidate hypotheses are identified using this metric alone. The remaining metrics are obtained for each candidate hypothesis. The initial support partitioning is shown in Figure 47(a).

7.4.2 Starved Initial Support Derivative

The initial votes are starved using the procedure described in Section 7.2.3. The characteristic of interest is the drop in support before and after the starving procedure. Hypotheses with strong starved strengths tend to represent true detections. This metric is shown in Figure 47(c).



(a) Test image with potential object hypotheses highlighted.



(b) Bivariate feature space used to identify strongest object hypotheses. Filled Circles correspond to correct object hypotheses in (a). Filled Squares correspond to incorrect object hypotheses in (a). Empty Squares indicate object hypotheses that would be considered correct if another hypothesis had not already been accepted for that target.

Figure 46. Avoiding false alarms

7.4.3 Number of Starved Initial Support

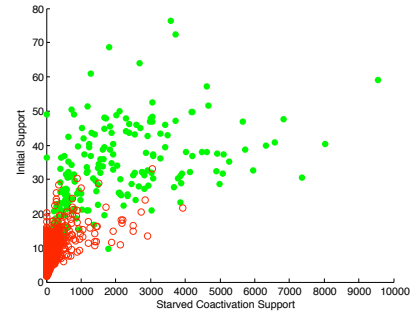
The initial votes are starved using the procedure described in Section 7.2.3. The number of unique interpretations remaining after the starving process then provides the support metric. The number of starved initial votes is shown in Figure 47(b).

7.4.4 Coactivation Support

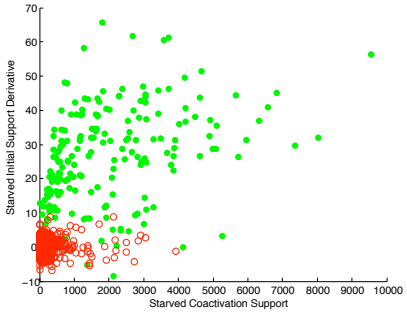
The votes augmenting the initial support via the activation network provide coactivation support. Section 7.3 describes the coactivation process. Coactivation support is a more concise metric, representing the known activations that should occur for specific instances instead of for the entire model representation that permits relations that do not occur together. The coactivation support partitioning is shown in Figure 47(d).

7.4.5 Subtending Votes

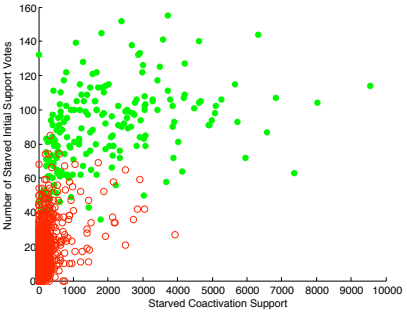
The region covered by the interpretations contributing to a particular object hypothesis is called the interpretation cover. If the object hypothesis is within the interpretation cover, the surrounding shape information represented in the model matched the input. If the object hypothesis is not within the interpretation cover, only a portion of the object in the input contributed to the hypothesis. This can be quantified by the *view angle* of the interpretation cover with respect to the hypothesis. Figure 47(e) shows how this view angle is calculated and Figure 47(f) shows the subtending angle partitioning.



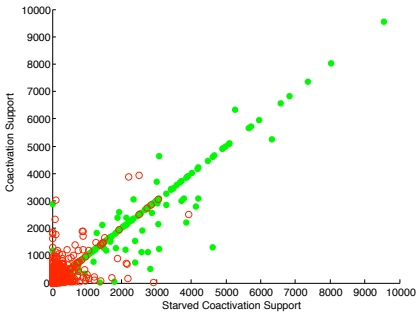
(a) Initial support



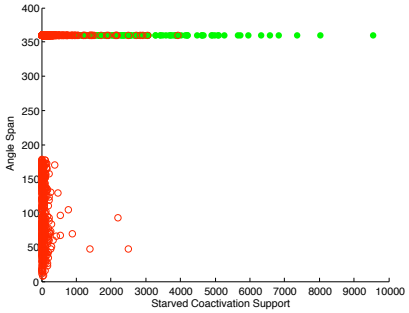
(b) Starved initial derivative



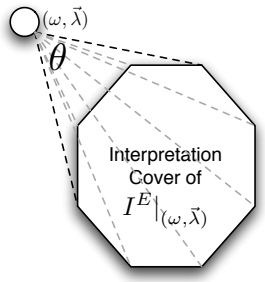
(c) Number of starved initial



(d) Coactivation support



(e) Angle span



(f) Determining degree span when object hypothesis is not surrounded by winning patches

Figure 47. Distinctive features for candidate hypotheses

8 Future Work

8.1 Improved Patch Similarity Metric

The Normalized Grayscale Correlation metric is a simple metric used to rate visual similarity between patches and has proved to be adequate for the presented work. The metric effectively handles textures and low frequency gradients, but does not handle other visual characteristics. Significant variance is an isolated region may change the local perceptual interpretation while maintaining a super-threshold NGC metric. The similarity metric is not as effective at comparing patches with contour or color information, while a variety of techniques in the Content-Based Image Retrieval domain address these concerns. It would be interesting to compare these techniques to determine agreement with the current understanding of human performance when determining isolated visual similarity.

8.2 Improved Patch Clustering

A concern regarding the model creation continues to be the number of model patches because it directly effects the computation time. Larger models become inconsistent and redundant. The activation networks reduce the inconsistency by eliminating the independence assumption. The initial patch clustering steps were performed to reduce the redundancy of the model. An improved patch similarity metric and alternative clustering techniques could determine similar groups in a better fashion. When a representative patch is created for a cluster, the arithmetic mean is found. Independence Component Analysis could replace this step by viewing the cluster as multiple signals representing an underlying structure. This underlying structure could replace the average patch to become the model patch.

8.3 Optimal Segmentation Cover

The optimal cover presented by Borenstein [7] identifies only consistent consolidations of patch-wise segmentation information. The established interaction between localization and segmentation can facilitate localization decisions. Interpretations removed because of inconsistent segmentation contributions would withdraw their initial and coactivation vote strengths.

8.4 Multiple Viewpoints

The current work addressed vehicles using only side views. Training on multiple views with the current algorithm confounds multiple interpretations with conflicting view-angle information. The view angle information would need to be included in the current interpretation implication $(\vec{\mathbf{m}}, \lambda \rightarrow \omega, \lambda + \vec{\mathbf{d}})$.

8.5 Activation Network for Discrimination

The excitatory and inhibitory nature of the activation network could be applied to discriminate between categories. If an activation network were created for two target categories, identifying cliques present in a single activation network that are not in present in another could be a distinguishing characteristic. Investigating computer vision applications with graph-theoretic techniques is a relatively rare approach, while the activation networks provide this opportunity.

8.6 Varying Specificity Shape Models

Although we described how to increase computation time by eliminating rarely-voting model patches, we found that these patches were tuned for specific instances and not general instances. The model patches in the shape model show a clear distinction between general-purpose interpretations and specific interpretations. It would be interesting to relocate the specific patches to an additional shape model and optionally evoke processing to discriminate between subcategories.

8.7 Temporal Activation Network

The current work addressed vehicles with only rigid bodies. Few modifications would be required to create shape models for non-rigid categories. However, the current shape model does not represent the temporal expectations of how attended objects will appear. This can be addressed with a temporal activation network. Work by Ramanan and Forsyth [44] use temporal coherency in place of a supervised indication of object presence to build appearance models of animals, although they do not build temporal models of the animals. Extending the shape model creation without supervised segmentation would be advantageous.

References

[1] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[2] Shivani Agarwal and Dan Roth. Learning a sparse representation for object detection. In *Proceedings of the 7th European Conference on Computer Vision*, 2002.

[3] J. Barreto, P. Menezes, and J. Dias. Human-robot interaction based on haar-like features and eigenfaces. *ICRA*, April 2004.

[4] I. Bax, G. Heidemann, and H. Ritter. A hierarchical feed-forward network for object detection tasks. In *Proc. SPIE Conf. on Independent Component Analyses, Wavelets, Unsupervised Smart Sensors, Neural Networks*, volume 5818, pages 144–152, Orlando, Florida, 2005.

[5] S. Bileschi and B. Heisele. Advances in component-based face detection, 2002.

[6] Michael C.A. Booth and Edmund T. Rolls. View-invariant representations of familiar objects by neurons in the inferior temporal cortex. *Cerebral Cortex*, 8:510–523, 1998.

[7] Eran Borenstein and Shimon Ullman. Class-specific, top-down segmentation. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part II*, pages 109–124, London, UK, 2002. Springer-Verlag.

[8] Tilo Burghardt, Janko Calic, and Barry Thomas. Tracking animals in wildlife videos using face detection. In *European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, October 2004.

[9] J. Brian Burns, Allen R. Hanson, and Edward M. Riseman. Extracting straight lines. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(4):425–455, 1986.

[10] Gustavo Carneiro and Allan D. Jepson. Flexible spatial models for grouping local image features. In *CVPR (2)*, pages 747–754, 2004.

[11] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):790–799, 1995.

[12] Dan Clark. Patch-level approach for removing ghosts. Personal Communication, 2005.

[13] Dorin Comaniciu and Peter Meer. Distribution free decomposition of multivariate data. In *SSPR '98/SPR '98: Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 602–610. Springer-Verlag, 1998.

[14] Sven J. Dickinson, Henrik I. Christensen, John K. Tsotsos, and Göran Olofsson. Active object recognition integrating attention and viewpoint control. In Jan-Olof Eklundh, editor, *ECCV (2)*, volume 2 of *Lecture Notes in Computer Science*, pages 3–14, Stockholm, Sweden, May 2-6 1994. Springer.

[15] G. Dorko and C. Schmid. Selection of scale invariant parts for object class recognition. In *ICCV*, 2003.

[16] Richard O Duda, P.E. Hart, and David G. Stork. *Pattern Classification*. Wiley, New York, 2nd edition edition, 2001.

[17] M.R. Everingham and A. Zisserman. Automated person identification in video. *In Proc. of the 3rd International Conference on Image and Video Retrieval (CIVR2004)*, 1:289–298, 2004.

[18] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 264, 2003.

[19] Josef Sivic Frederik. Efficient object retrieval from videos, 2004.

[20] Robert M. Gray and Lee D. Davidson. *An Introduction to Statistical Signal Processing*. Cambridge University Press, August 2004.

[21] Thorsten Hansen, Wolfgang Sepp, and Heiko Neumann. *Emergent Neural Computational Architectures Based on Neuroscience*, chapter Recurrent Long-Range Interactions in Early Vision, pages 139–153. Springer, 2001.

[22] C. Harris and M. Stephens. A combined corner and edge detector. *Proc. Alvey Vision Conf.*, pages 147–151, 1988.

[23] Bernd Heisele, Thomas Serre, Massimiliano Pontil, and Tomaso Poggio. Component-based face detection. *CVPR*, 2001.

[24] Qasim Iqbal and J. K. Aggarwal. Applying perceptual grouping to content–based image retrieval: Building images. In *Proceedings of the 1999 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)*, Fort Collins, Colorado, USA, 23–25 1999.

[25] Qasim Iqbal and J.K. Aggarwal. Perceptual grouping for image retrieval and classification. *Third IEEE Computer Society Workshop on Perceptual Organization in Computer Vision*, pages 19–1 – 19–4, July 2001.

[26] John P. Kerekes. Spatial pattern recognition. Personal Communication, February 2005.

[27] D Kersten and A Yuille. Bayesian models of object perception. *Current Opinion in Neurobiology*, 12(2), 2003.

[28] Kobatake, Eucaly, and Tanaka. Neuronal selectivities to complex object features in the ventral visual pathway of the macaque cerebral cortex. *Journal of Neurophysiology*, 71(3):856–867, 1994.

[29] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 1985.

[30] Ivan Laptev and Tony Lindeberg. Space-time interest points. *In Proc. ICCV 2003*, pages 432–439, 2003.

[31] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *British Machine Vision Conference (BMVC'03)*, pages 759–768, Norwich, UK, Sept. 2003.

[32] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. *ECCV'04 Workshop on Statistical Learning in Computer Vision*, May 2004.

[33] Brian Leung. Component-based car detection in street scene images. Master's thesis, Massachusetts Institute of Technology, May 2004.

[34] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *Proceedings of the International Conference on Image Processing*, pages 900–903, Rochester, USA, September 2002. IEEE.

[35] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *J. of Applied Statistics*, 21(2):224–270, 1994. (Supplement on Advances in Applied Statistics: Statistics and Images: 2).

[36] David G. Lowe. Towards a computational model for object recognition in IT cortex. In *Biologically Motivated Computer Vision*, pages 20–31, 2000.

[37] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[38] David Marr. *Vision.* W.H. Freeman and Company, San Fransisco, CA, USA, 1982.

[39] Matthew McEuen. Vehicle image data set. Personal Communication, April 2005.

[40] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. In *Proceedings of the British Machine Vision Conference*, 2003.

[41] A. Mohan, C. Papageorgiou, and Tomaso Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:349–361, April 2001.

[42] M. A. Peterson. Object recognition processes can and do operate before figure-ground organization. *Current Directions in Psychological Science*, 3:105–111, 1994.

[43] M.A. Peterson and B.S. Gibson. Shape recognition contributions to figure-ground organization in three-dimensional displays. *Cognitive Psychology*, 25:383–429, 1993.

[44] Deva Ramanan and David A. Forsyth. Using temporal coherence to build models of animals. In *ICCV*, pages 338–345, Nice, France, 14-17 October 2003. IEEE Computer Society.

[45] Ronald A. Rensink. Internal vs. external information in visual perception. In *SMARTGRAPH '02: Proceedings of the 2nd international symposium on Smart graphics*, pages 63–70. ACM Press, 2002.

[46] Maximilian Riesenhuber and Tomaso Poggio. The individual is nothing, the class everything: Psychophysics and modeling of recognition in object classes, 2000.

[47] Maximilian Riesenhuber and Tomaso Poggio. Models of object recognition. *Nature Neuroscience*, 3:1199–1204, Nove 2000.

[48] Frederik Schaffalitzky and Andrew Zisserman. Geometric grouping of repeated elements within images. In *Shape, Contour and Grouping in Computer Vision*, pages 165–181, 1999.

[49] H. Schneiderman. A statistical approach to 3d object detection applied to faces and cars, 2000.

[50] Linda G. Shapiro and George C. Stockman. *Computer Vision*. Prentice-Hall, Inc., 2001.

[51] Michael J. Swain and Dana H. Ballard. Color indexing. *Int. J. Comput. Vision*, 7(1):11–32, 1991.

[52] Keiji Tanaka. Neuronal mechanisms of object recognition. *Science*, 262:685–688, 1993.

[53] Antonio Torralba, Kevin Murphy, and William Freeman. Sharing visual features for multiclass and multiview object detection, April 2004.

[54] Anne M. Treisman and Nancy G. Kanwisher. Perceiving visually presented objects: recognition, awareness, and modularity. *Current Opinion in Neurobiology*, 8:218–226, 1998.

[55] R. Vaillant, C. Monrocq, and Y. LeCun. Original approach for the localization of objects in images. *IEEE Proc. Vision, Image, and Signal Processing*, 141(4):245–250, 1994.

[56] Markus Weber, Max Welling, and Pietro Perona. Unsupervised learning of models for recognition. In *ECCV (1)*, pages 18–32, 2000.

[57] Andrew P. Witkin. Scale-space filtering. In *International Joint Conference on Artificial Intelligence*, pages 1019–1022, 1983.

[58] Jeremy M. Wolfe and Sara C. Bennett. Preattentive object files: shapeless bundles of basic features. *Vision Research*, 37:25–43, 1997.