Rotation Invariant Object Recognition Using a Probabilistic Approach

Vadim von Brzeski
Department of Computer Science
University of California, Santa Cruz

June 9, 2005

Abstract

The goal of this project was to implement and test a system that can recognize instances of an object class from various angles of rotation. During the training phase, 2-D images of an object class are taken from three angles of rotation. A probabilistic model of the shape and appearance of an object is constructed as a collection of experts, with each expert corresponding to the probability distribution derived from one of the training angles.

1 Introduction

The motivation behind this project stems from the problem of viewpoint invariant object recognition in 2-D images. Having been shown a few examples of an object class from a few viewpoints (even if only in 2-D), humans can easily pick out instances of this class, even if the object is shown from practically an arbitrary viewpoint. The goal of this project is to implement and test a system that makes a step towards this ultimate goal by first dealing with the issue of recognition under rotation. This issue is important in image recognition and processing systems because real-world images contain objects in a variety of poses and orientations, rarely in the optimal orientation for recognition.

We follow the so-called probabilistic *constellation model*, which models objects as flexible collections of *parts*. Specifically, we setup a probabilistic model of an object as outlined by Burl et. al [2], Fergus et. al [1], and Li et. al [3]. We learn to recognize instances of an object class by learning a constellation model that defines the class, which amounts to learning a probability distribution (or more specifically its parameters) over the locations and appearance of the object's essential parts. In our case, we construct three independent constellation models. Each learned model corresponds to the object photographed at one of three angles of rotation. We then show how the system is able to recognize the presence of the object in images where the object is positioned at any angle of rotation, specifically at any angle in between the three angles used during training. Our training is conducted in a virtually unsupervised way : we only specify the presence or absence of the target object in an image; we do not specify the locations nor appearances of the parts.

2 Previous Work

This project follows on the work done initially by Burl et. al [2], and later expanded by Fergus et. al [1] and Li [3]. Fergus develops a system which makes a Bayesian decision as to the presence or absence of an object by calculating a ratio R of posterior probabilities :

$$R = \frac{p(Object \,|\, X, A)}{p(No\ Object \,|\, X, A)} \tag{1}$$

where X and A are the part location and part appearance attributes extracted from an image. However, Fergus does not follow the full Bayesian approach when evaluating the posterior probabilities, i.e. instead of integrating over the parameter space θ, Fergus evaluates the distributions at the maximum likelihood value of the parameter vector θ. Li [3] builds on Fergus's work, and does the full Bayesian calculations using Variational Bayes methods [4] by integrating over the parameter space θ. Li outlines the variational Bayes framework specifically in the context of a mixture model of constellation models; however, strangely enough the mixture model is not used to its full potential since "learning is performed using a single mixture component" [3]. Nevertheless, both produce good results - accuracy is above 90% for most object classes. The systems however fail in cases when the key part locations are not recognized. Both use the salient region detector developed by Kadir and Brady [5] to identify the parts. This detector is not affine invariant, and thus does not produce good results when applied to images under different viewpoints. Therefore, although both Fergus and Li perform learning in an unsupervised manner, both perform a significant image pre-processing step : all training and test images are first transformed such that the target object instances are all oriented in the same direction.

Our approach extends the work done by Li et. al by dealing with three different orientations of an object (instead of one). We also experiment with a different region detector, specifically one that happens to extract affine invariant regions, in order to measure its effectiveness at dealing with rotated objects.

3 Constellation Model

We begin the description of our methodology with a brief review of the constellation model as outlined in [1]. A constellation model describes an object class in terms of probability distributions over the *shape* (X) and *appearance* (A) of an object's essential *parts*. Although the number of parts is typically fixed, the locations and appearance of an object's essential parts (i.e. the parts that define objects in that class) are unknown. Thus a primary goal of the learning phase is to discover which *features* of an object best represent its essential parts. By features we mean the locations and appearance of all "interesting" regions extracted during image analysis, where interesting is defined by choice of region detector.

To facilitate the automatic discovery of the best representative parts, we introduce the concept of a *hypothesis*. A hypothesis $\mathbf{h}$ is a collection (vector) of P parts, where each part has associated location and appearance attributes (X, A). First, the coordinates of all key regions in a 2-D image are extracted (the X attribute), and then at each such location a patch is cropped from the full image, describing the appearance of that region (the A attribute). A hypothesis $\mathbf{h}$ is then any valid mapping from the set of (X, A) attribute pairs to the set of parts. The total number of possible hypotheses is thus $\binom{F}{P}$, where F is the number of key regions and P is the number of parts. Therefore, given a parameterized probability distribution over the space of possible hypotheses, the model attempts to find the most likely hypothesis that could have generated the image in question. The number of parts P is typically kept low due to performance reasons since F is typically on the order of 20 - 30. In our case, we set the number of parts at four, and the number of key regions at 16, yielding a total number of hypotheses $|H| = 1820$.

Given that the hypothesis $\mathbf{h}$ is a hidden (latent) variable in the model, we can write our constellation model likelihood as follows :

$$p(X, A \mid \boldsymbol{\theta}) = \sum_h p(X \mid \boldsymbol{\theta}, \mathbf{h})\, p(A \mid X, \boldsymbol{\theta}, \mathbf{h}) \tag{2}$$

The model makes the following assumptions about the shape and appearance distribution $p(X, A \mid \boldsymbol{\theta})$. First, the shape (X) and appearance (A) components are independent, meaning that the appearance of an image patch is independent of where that patch is located. This may be a naive assumption, but the model compensates for this simplification by restricting the order of parts in a given hypothesis. Specifically, when constructing a hypothesis vector $\mathbf{h}$ of (X,A) attribute pairs (the candidate parts), the model sorts the vector by the x-coordinate of each candidate part, increasing from left to right.

Second, given a hypothesis $\mathbf{h}$, the appearance of one part is independent of the appearance of other parts in that hypothesis. This is a reasonable assumption since the appearance of one part should not be tied to the appearance of another part. However, the shape distribution is modeled as a full joint distribution over the locations of the candidate parts, i.e. the location of one part is dependent on the locations of other parts.

Third, the shape and appearance distributions are modeled as Gaussian distributions with mean and precision parameters $\boldsymbol{\theta}^X$=$\{\boldsymbol{\mu}^X, \boldsymbol{\Gamma}^X\}$ and $\boldsymbol{\theta}^A = \{\boldsymbol{\mu}^A, \boldsymbol{\Gamma}^A\}$, and thus our final forms for the terms in equation (2) are as follows :

$$p(X \mid \boldsymbol{\theta}, \mathbf{h}) = G(X(\mathbf{h}) \mid \boldsymbol{\mu}^X, \boldsymbol{\Gamma}^X) \tag{3}$$

$$p(A \mid \boldsymbol{\theta}, \mathbf{h}) = \prod_{p=1}^{P} G(A(\mathbf{h}) \mid \boldsymbol{\mu}^A, \boldsymbol{\Gamma}^A) \tag{4}$$

4 Bayesian Inference

The goal of training is to derive the probability distributions shown in equation (1]). Equation (1) can be written as follows, where we make the distinction between *test* image features X and A, and *training* image features X$_t$,A$_t$.

$$R = \frac{p(Object \mid X, A, X_t, A_t)}{p(No\ Object \mid X, A, X_t, A_t)} = \frac{p(X, A \mid X_t, A_t, Object)\, p(Object)}{p(X, A \mid X_t, A_t, No\ Object)\, p(No\ Object)} \tag{5}$$

We make the assumption that $\frac{p(Object)}{p(No\ Object)} = 1$, and expand the distributions by writing them as marginalized versions over the unknown parameters $\mathbf{h}$ and $\boldsymbol{\theta}$:

$$R \approx \frac{\sum_h \int p(X, A \mid \boldsymbol{\theta}, \mathbf{h})\, p(\boldsymbol{\theta}, \mathbf{h} \mid X_t, A_t)\, d\boldsymbol{\theta}}{\sum_h \int p(X, A \mid \boldsymbol{\theta}_{bg}, \mathbf{h})\, p(\boldsymbol{\theta}_{bg}, \mathbf{h} \mid X_t, A_t)\, d\boldsymbol{\theta}_{bg}} \tag{6}$$

where $\boldsymbol{\theta}$ corresponds the parameters of the distribution representing the target object, and $\boldsymbol{\theta}_{bg}$ corresponds to the parameters of the distribution representing the background (i.e. the absence of the object).

To compute R, we need the likelihood $p(X, A \mid \boldsymbol{\theta}, \mathbf{h})$, already specified for a given $\boldsymbol{\theta}$ by equations (3, 4), and we also need the posterior distribution for the parameters $\mathbf{h}$ and $\boldsymbol{\theta}$ =$\{\boldsymbol{\mu}, \boldsymbol{\Gamma}\}$, namely $p(\boldsymbol{\mu}, \boldsymbol{\Gamma}\, , \mathbf{h} \mid X_t, A_t)$. Note that since the shape (X) and appearance (A) components are independent, we can perform each of these derivations independently, and since the likelihoods in equations (3, 4) have the same form, the derivations will be identical for the X and A components. Thus we drop the X and A notation and deal with $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Gamma}\}$.

First, we assume the distribution for $\mathbf{h}$ is uniform and independent of $\{\boldsymbol{\mu}, \boldsymbol{\Gamma}\}$, i.e. $p(\mathbf{h}) = 1/|H|$, where $|H|$ is the total number of hypotheses, and thus we focus on deriving $p(\boldsymbol{\mu}, \boldsymbol{\Gamma} \,|\, X_t, A_t)$.

To derive the posterior in a Bayesian framework, we need to specify a prior distribution on $\{\boldsymbol{\mu}, \boldsymbol{\Gamma}\}$. In order to simplify the derivation of the posterior, we assume the prior distribution on $\{\boldsymbol{\mu}, \boldsymbol{\Gamma}\}$ has a form that is *conjugate* to the likelihood. The advantage of conjugate analysis is that it allows us to immediately arrive at a closed form for the posterior distribution without performing the complex integration involved in Bayes rule. Specifically, if we assume a Gaussian-Wishart prior distribution with (hyper)parameters $\mathbf{m}_0, \beta_0, a_0, \mathbf{B}_0$:

$$p(\boldsymbol{\mu}, \boldsymbol{\Gamma}) = p(\boldsymbol{\mu} \,|\, \boldsymbol{\Gamma})\; p(\boldsymbol{\Gamma}) = G(\boldsymbol{\mu} \,|\, \mathbf{m}_0, \beta_0 \boldsymbol{\Gamma})\; W(\boldsymbol{\Gamma} \,|\, a_0, \mathbf{B}_0). \tag{7}$$

and if we assume the likelihood is $p(X \mid \boldsymbol{\mu}, \boldsymbol{\Gamma}) = G(X \mid \boldsymbol{\mu}, \boldsymbol{\Gamma})$, then the form of posterior distribution given the training data X_t will also be a Gaussian-Wishart, namely :

$$p(\boldsymbol{\mu}, \boldsymbol{\Gamma} \,|\, X_t) = G(\boldsymbol{\mu} \,|\, \mathbf{m}, \beta \boldsymbol{\Gamma})\; W(\boldsymbol{\Gamma} \,|\, a, \mathbf{B}) \tag{8}$$

where
$\mathbf{m} = \dfrac{\beta_0}{\beta_0 + n} \mathbf{m}_0 + \dfrac{n}{\beta_0 + n} \bar{X}$
$\beta = \beta_0 + n$
$a = a_0 + n$
$\mathbf{B} = \mathbf{B}_0 + \sum_{i=1}^{n} (X - \bar{X})(X - \bar{X})^T + \dfrac{\beta_0 n}{\beta_0 + n} (\bar{X} - X)(\bar{X} - X)^T$

Now that we have the posterior distribution $p(\boldsymbol{\mu}, \boldsymbol{\Gamma} \,|\, X_t)$, we need to evaluate
$p(X \mid X_t) = \int p(X \,|\, \boldsymbol{\mu}, \boldsymbol{\Gamma})\; p(\boldsymbol{\mu}, \boldsymbol{\Gamma} \mid X_t)\; \mathbf{d}\boldsymbol{\mu}\, \mathbf{d}\boldsymbol{\Gamma}$
in order to compute the ratio R in equation (6). Conjugacy helps here as well since the posterior (8) is conjugate to the likelihood (3) , and the result of this integral is a t-distribution with the following parameters [7] :

$$p(X \mid X_t) = t_{a-d+1}(\mathbf{m}, \frac{\beta + 1}{\beta(a - d + 1)} \mathbf{B}) \tag{9}$$

where d is the dimension of the data. As mentioned above, the derivation for the appearance (A) component is identical. Thus the final form for the predictive distribution, including both X and A components, is :

$$p(X, A \,|\, X_t, A_t) = \sum_{h=1}^{|H|} p(X \mid X_t)\; p(A \mid A_t) \tag{10}$$

Therefore the **goal of learning is to learn the parameters $\mathbf{m}, \beta, a, \mathbf{B}$** from the training data for the shape (X) and appearance (A) distributions, once for the target object and once again for the background.

4.1 Variational Bayes EM

We use the Variational Bayes Expectation Maximization (VBEM) framework outlined in detail by Li [3] to estimate the parameters. We utilize the VBEM algorithm because we are optimizing two unknowns simultaneously : the parameters $\boldsymbol{\theta}$ and the hypothesis $\mathbf{h}$, a hidden variable. If we knew $\boldsymbol{\theta}$, we could find the $\mathbf{h}$ that best describes a scene in an image by computing $p(X, A, h \mid \boldsymbol{\theta})$ for each $\mathbf{h}$; conversely, if we knew the best $\mathbf{h}$ (i.e. the best mapping from features to parts for a given scene) we could solve for $\boldsymbol{\theta}$ by maximum likelihood methods.

Variational Bayes EM differs from the standard EM algorithm in that standard EM searches for the mode (the most likely estimate, MLE) of a distribution and then approximates the entire distribution with a delta function located at the mode. This makes the integrals in equation (6) easy to evaluate - they collapse to $p(X, A \mid \boldsymbol{\theta}_{MLE}, \mathbf{h})$ - but may be an oversimplification of the entire distribution. Variational Bayes on the other hand searches for a distribution that is *closest* (Kullback-Leibler distance) to the entire target distribution [9], in our case the posteriors $p(\boldsymbol{\theta} \mid X_t)$, and $p(\boldsymbol{\theta} \mid A_t)$. In the interests of space, we do not go into the details of Variational Bayes here, but rather refer the reader to an excellent explanation by Beal [9]. For an exact description of how Variational Bayes is applied to this problem, including the exact EM update equations we implemented in Matlab, please see Li [3].

5 Methodology

5.1 Data Collection

We work with only a single object class : handguns, which includes pistols and revolvers. Our training and test data consists of digital images of 138 various handguns, where we collect three images per gun, one image from one of three angles of rotation :+90, 0, and -90 degrees, where 0 degrees corresponds to looking at the gun in profile, facing left - see figure 2. We obtained these images from gun manufacturers' websites; all of the original images were at the "0 degree" orientation, and we generated the -90 and +90 copies by rotating the original images. All images were converted to gray scale.

5.2 Region Detection and Extraction

We chose an intensity based affine region detector (IBR) developed by Tuytelaars and van Gool [8]. We use the compiled Linux executable of the region detector, which is available for download from Oxford Vision Research Group's website [6].

The IBR detector picks out patches from the image that are invariant to affine transformations. It does this by first identifying locations of local intensity extrema in the image. From each local extremum, rays are traced outwardly in a circular fashion. The following function is computed along each ray :

$$f(t) = \frac{abs(I(t) - I_0)}{\max\left(\frac{\int_0^t abs(I(t) - I_0)dt}{t}, d\right)} \tag{11}$$

where t is the Euclidean arclength along the ray, $I(t)$ the intensity at position t, and I_0 the intensity at the local extremum; d is a small constant included to prevent division by zero. To understand what $f(t)$ is doing it is helpful to look at figure 1. The maxima of $f(t)$ occur at points on the rays at which the

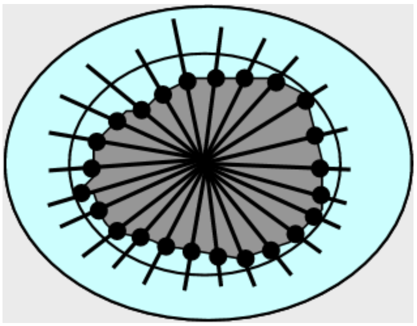

Figure 1: Rays are traced from a point of local extremum of intensity. The points on the rays at which the intensity changes most dramatically define the border of the affine invariant region. The final ellipse defining the region is double the size of the border found.

intensity changes most dramatically compared to the intensity changes seen so far, and thus these points define the border of an affine invariant region in the image.

Given an image, the region detector produces a long list of locations (anywhere from 50-200). These locations are clustered using k-means to reduce their number to a reasonable number of key locations, in our case 16. These 16 key locations form the X attribute (feature) of each training image. To extract the appearance (A) attribute, for each (x,y) location in X, we crop a patch from the image, with the size of the patch equal to the size of the ellipse produced by the region detector. This patch is then resized to a smaller 11x11 patch of intensity values. This still leaves a 121-dimensional vector, and so this is further reduced to a 10-dimensional vector using principal component analysis (PCA). The basis for the PCA mapping is pre-computed from all such extracted patches from all training images. We ignore object scale for this project - all images are at approximately at the same scale. The k-means clustering step for X attributes, as well as the pixel path extraction and PCA calculations for the A attributes are implemented using Matlab.

5.3 Training

During the training phase, we independently learn five sets of distributions for the following five sets of objects :

1. 138 uncluttered images of guns oriented at -90 degrees (down)
2. 138 uncluttered images of guns oriented at 0 degrees (left)
3. 138 uncluttered images of guns oriented at +90 degrees (up)
4. 100 images of background clutter
5. 100 uncluttered images of motorcycles facing right, from Caltech's Vision Group dataset (see testing section below)

For each training run, the initial values for the $\mathbf{m}_0$ and $\mathbf{B}_0$ parameters were set proportional to the sample mean and sample variance of a subset of the training data, and then the EM algorithm was run on the remaining subset. For example, for the sets of gun images, the first 38 images were used to arrive at reasonable estimates for the initial values, and then the EM algorithm was run on the remaining 100 images. The number of EM iterations required for convergence was between 40 and 50, depending on the distribution being learned. The total time required to learn one distribution was approximately 1

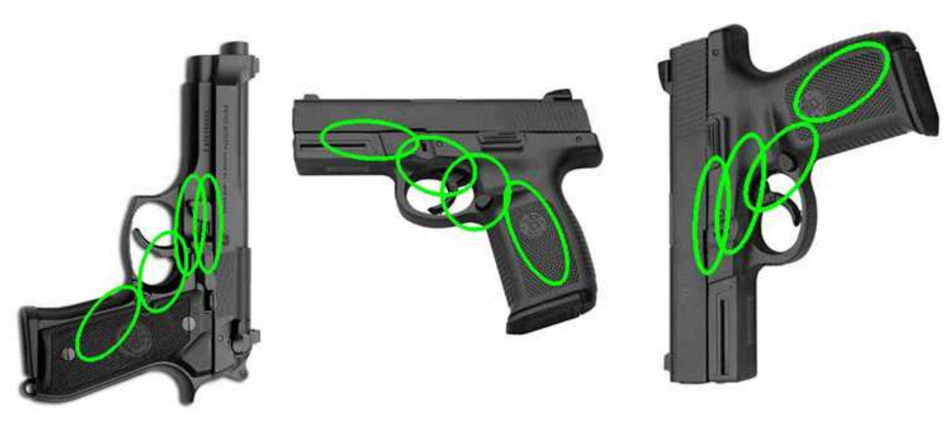

Figure 2: Shape distributions for the three different orientations : +90, 0, -90 degrees. The ellipses indicate the 95% predictive regions for the locations (X) of the four parts in each model.

hour on a Pentium 2 GHz machine.

The training was "virtually" unsupervised, meaning that the algorithm was "told" if the training image contains an instance of the object class or not. No other information about location, pose, orientation, etc., is provided. The EM update equations are implemented in Matlab.

Some of the results of training are shown below in figures 2 and 3. Figure 2 shows the shape distributions against the backdrop of sample training images, one from each orientation, and one can observe how the model has identified the most probable locations of the four parts.

Visualizing the appearance distribution is not very easy since each part's appearance is a 10-dimensional vector, so thus a four part model has a 40-dimensional appearance distribution. However, we can get a sense of it by looking at its mean and by finding the patches in our training images that most closely match the mean. This is show in figure 3. Some of the parts' appearances are clearly identifiable, e.g. the trigger region for part 3; however, the rest are harder to interpret, indicating one of the drawbacks of this approach - the appearance density is not very tight (i.e. it has large variance), potentially leading the system to associate an image patch from one object with a similar image patch from a completely different object.

5.4 Testing

The testing phase is composed of two separate test runs. First, we measure the system's ability to recognize images of guns versus images of background. This may not be a difficult task since the background images are cluttered and very different from the uncluttered gun images. Therefore, we perform a second test where we measure the system's ability to discriminate between uncluttered images of guns and uncluttered images of another object, in this case : motorcycles. We test the system on the following sets of the images (all previously unseen) : 101 images of guns at *different orientations*, 95 images of background scenes, and 100 images of motorcycles. Figures 4, 5, and 6 show a sample of the images used during the testing phase.

Recall that the outputs of the training phase are the following five distributions ($\boldsymbol{\theta}$ in this case refers to the parameters in the t-distribution in equation (9)) :

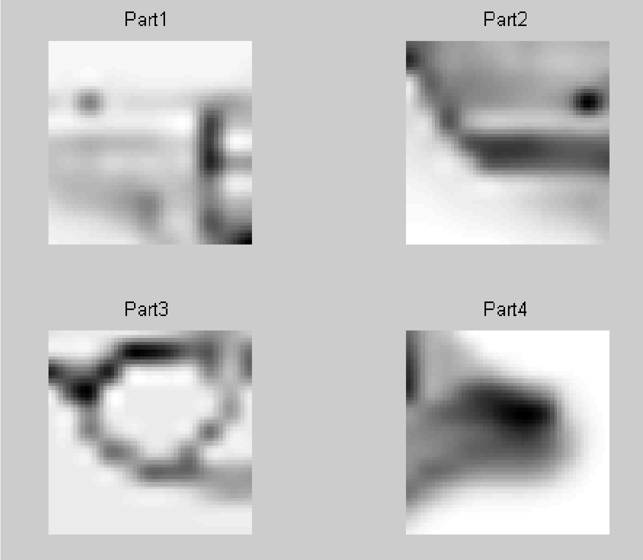

Figure 3: The four patches in the training image set (left orientation) that are closest (in Euclidean distance) from the mean of the appearance distribution.

1. $p(Gun_{down} \mid \boldsymbol{\theta}_{gun-down})$
2. $p(Gun_{left} \mid \boldsymbol{\theta}_{gun-left})$
3. $p(Gun_{up} \mid \boldsymbol{\theta}_{gun-up})$
4. $p(Background \mid \boldsymbol{\theta}_{bg})$
5. $p(Motorcycle \mid \boldsymbol{\theta}_{motorcycle})$

Therefore during each test run, we compute the following quantity for each test image X, and we decide that the image contains a gun if $R > 1$.

$$R_{gun-vs-bg} = \max\left(\frac{p(X \mid \boldsymbol{\theta}_{gun-down})}{p(X \mid \boldsymbol{\theta}_{bg})}, \frac{p(X \mid \boldsymbol{\theta}_{gun-left})}{p(X \mid \boldsymbol{\theta}_{bg})}, \frac{p(X \mid \boldsymbol{\theta}_{gun-up})}{p(X \mid \boldsymbol{\theta}_{bg})}\right) \tag{12}$$

$$R_{gun-vs-cycle} = \max\left(\frac{p(X \mid \boldsymbol{\theta}_{gun-down})}{p(X \mid \boldsymbol{\theta}_{cycle})}, \frac{p(X \mid \boldsymbol{\theta}_{gun-left})}{p(X \mid \boldsymbol{\theta}_{cycle})}, \frac{p(X \mid \boldsymbol{\theta}_{gun-up})}{p(X \mid \boldsymbol{\theta}_{cycle})}\right) \tag{13}$$

The time to test a single image is approximately 5-6 seconds.

6 Results and Analysis

The results of the analysis are shown below in table 1. Table 1 shows that the system performs much better on the easier task of distinguishing between an image of a gun and an image of background clutter, as compared to discriminating between two uncluttered images of different objects. We attribute this to the fact that although the shape distribution of the parts is representative of the object (see figure 2), the appearance distribution of the parts does not represent all essential features of the target object. The results for "gun vs. background" compare are in-line with the results achieved by Li and Fergus, where they reported an error rate of 5.6 - 10% [1] [3].

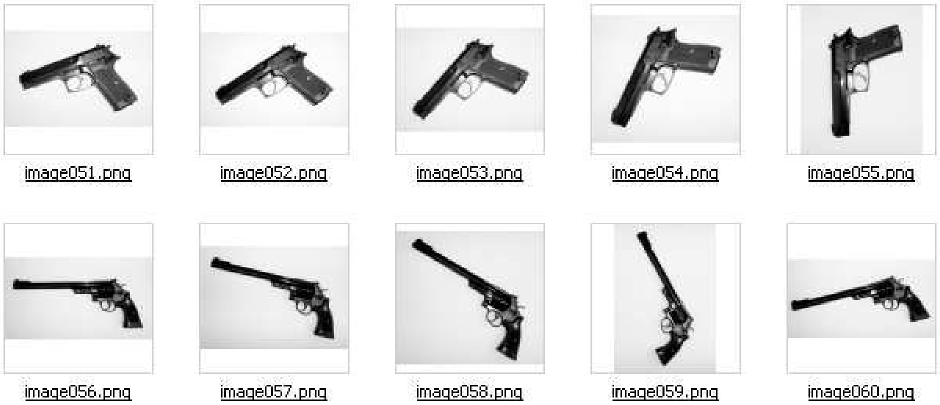

Figure 4: Sample gun images used in the testing phase.

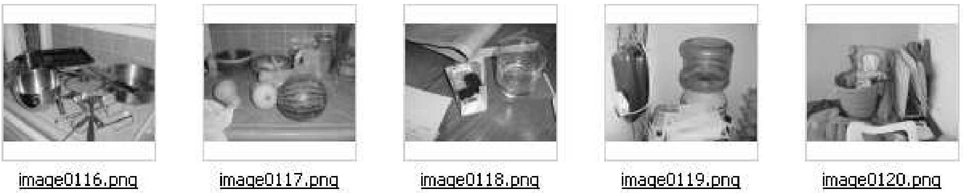

Figure 5: Sample background images used in the testing phase.

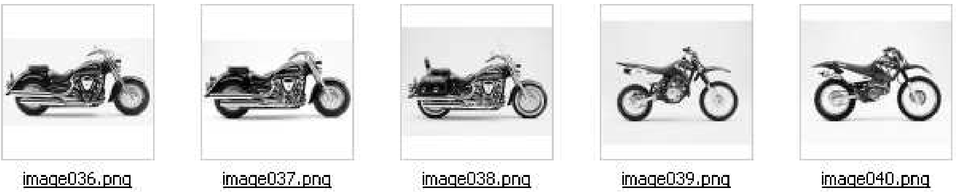

Figure 6: Sample motorcycle images used in the testing phase.

Test Set	**False Negative**	**False Positive**
Gun vs. Background	3 %	10.5 %
Gun vs. Motorcycle	17 %	13 %

Table 1: Test results.

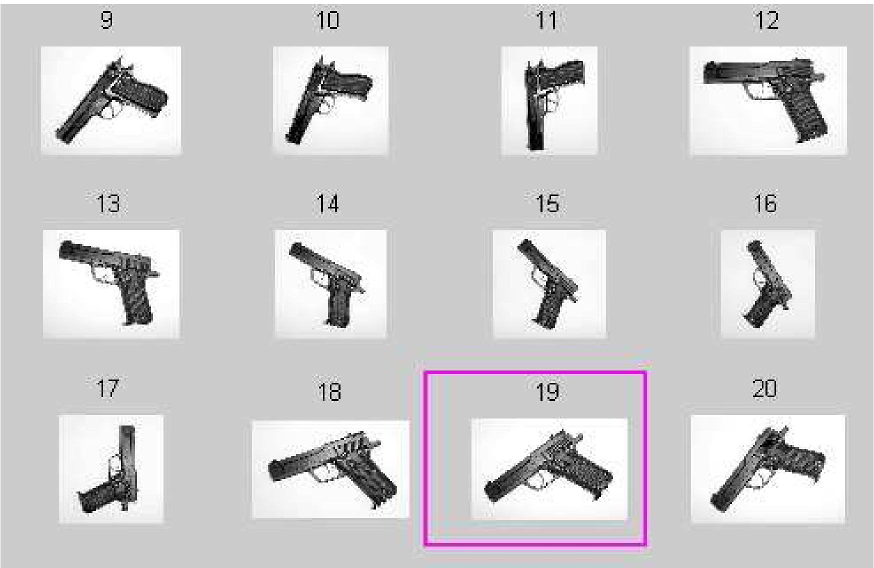

Figure 7: Image 19 shows a gun being incorrectly unrecognized.

To give the reader a sense of the behavior of the (sometimes unpredictable) nature of the probabilistic constellation model, some examples of correct and incorrect classifications are show in figures 7 - 14 below.

7 Conclusions and Future Work

In conclusion, the results obtained here are encouraging, but more work is needed to improve the accuracy, especially when it comes to discriminating between objects on a white background. The greatest strength of this method is that it is virtually unsupervised : the algorithm "learns" the locations, and to some extent the appearances, of the key parts of an object are without user intervention. However, this is not to say that the key parts it finds are the defining parts of an object, and this is the method's greatest weakness.

This method is highly sensitive to the region detector; if the region detector does not focus on the distinctive features of an object, i.e. on those features that make the object what it is, then accuracy suffers. Furthermore, k-means introduces an element of randomness. This is a problem, since this method relies on the region detector consistently producing the same set of key regions from one training image to another. If the k-means clustering is very inconsistent, then we get a high variance distribution over the locations and appearances of the parts.

In this project our approach focused on learning three distributions for three different orientations of an object. In retrospect, this is inefficient, since in theory all one needs to do is to learn a *single* distribution for a *single* orientation. Then for each test image, extract the features as usual, but before plugging the extracted (X, A) features directly into the predictive distribution (eq. 9), perform a sequence of affine transformations $T_1(X, A), T_2(X, A).....T_k(X, A)$ on the features first, and then compute the predictive distribution using the transformed features : $p(T_i(X, A) \mid X_t)$. Finally, compute the decision ratio R using these transformed features to decide if the image contains an instance, possibly transformed by an affine transformation $T_i(X, A)$, of the target object. However, this ideal approach relies even more

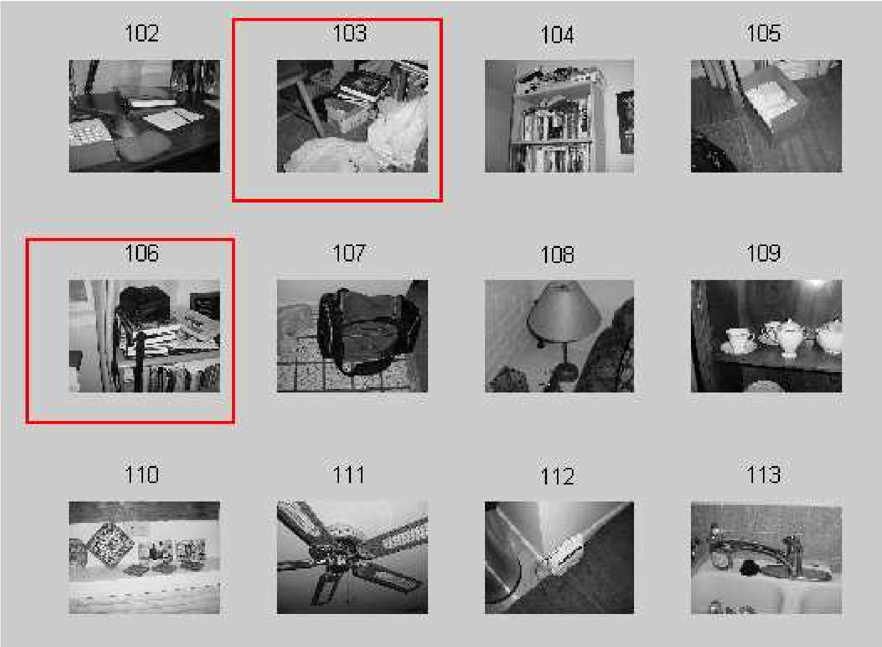

Figure 8: Images 103 and 106 are incorrectly classified as containing a gun.

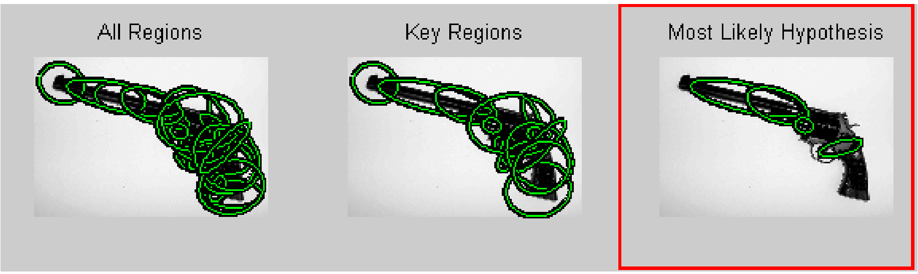

Figure 9: The images depict the flow of the recognition process. The image on the left shows the output from the IBR region detector. The middle image shows the 16 key regions after running k-means. The image on the right shows a gun being recognized, and the distribution of parts in the most likely hypothesis in the image.

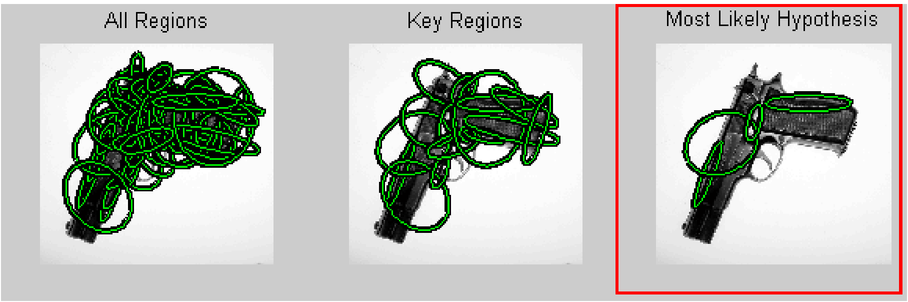

Figure 10: The image on the left shows the output from the IBR region detector. The middle image shows the 16 key regions after running k-means. The image on the right shows a gun being recognized, and the distribution of parts in the most likely hypothesis in the image.

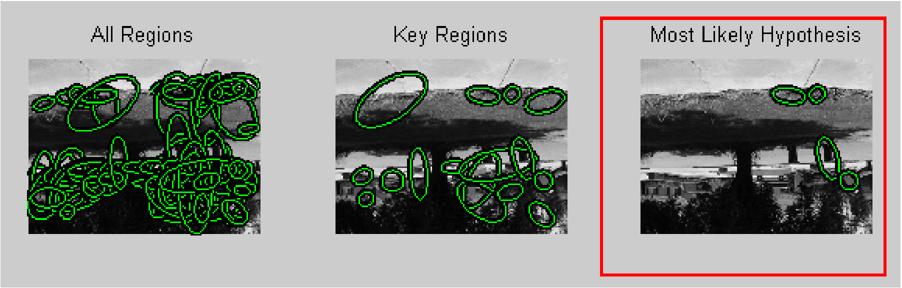

Figure 11: The image on the right shows a gun being recognized, which is incorrect. The distribution of parts in the most likely hypothesis in the image provides a clue to the misclassification - it resembles the distribution of parts seen in figure 2, the "0 degree" view.

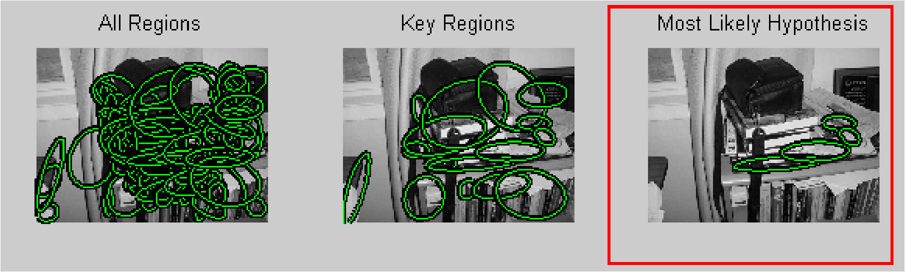

Figure 12: The image on the right shows a gun being recognized, which again is incorrect. The distribution of parts in the most likely hypothesis resembles the distribution of parts seen in figure 2, the +90 degree view.

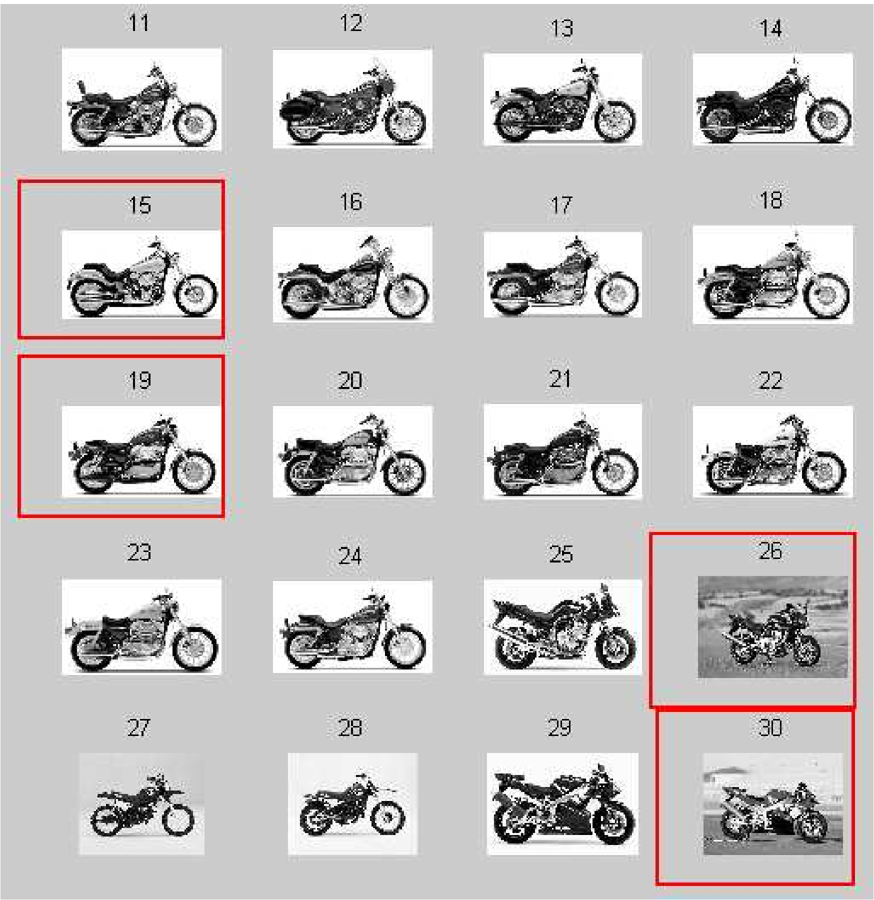

Figure 13: Images of motorcycles, with the boxed images being incorrectly classified as guns.

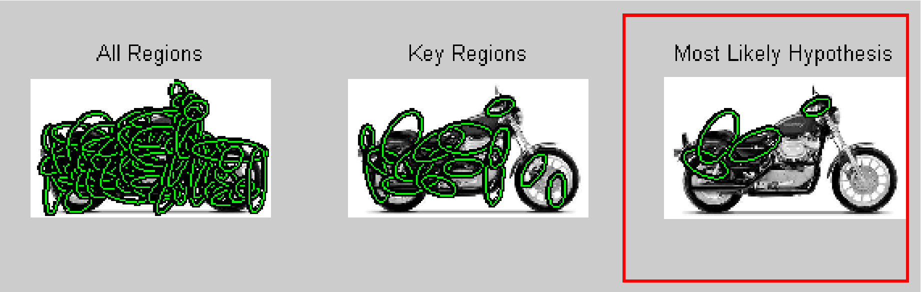

Figure 14: Detailed view of an incorrectly classified image. Image was classified as containing a gun.

heavily on the region detector being consistent and more "intelligent" in finding the essential features of an object.

References

[1] R. Fergus, P. Perona, A. Zisserman. *Object class recognition by unsupervised scale-invariant learning.* Proc. CVPR, vol. 2, pp. 264-271, 2003.

[2] M.C. Burl, M. Weber, P. Perona. *A probabilistic approach to object recognition using local photometry and global geometry.* Proc. ECCV, pp. 628-641, 1998

[3] L. Fei-Fei, R. Fergus, P. Perona. *A Bayesian approach to unsupervised learning of object categories.* Proc. ICCV, 2003.

[4] H. Attias. *Inferring parameters and structure of latent variable models by variational bayes.* 15th Conference on Uncertainty in Artificial Intelligence, pp. 21-30, 1999.

[5] T. Kadir and M. Brady. *Scale, saliency, and image description.* International Journal of Computer Vision, vol. 45, no. 2, pp.83-105, 2001.

[6] Affine Covariant Features. *Website : http://www.robots.ox.ac.uk/~vgg/research/affine/index.html*, containing all links to papers.

[7] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis*, 2nd Edition. Chapman & Hall/CRC, 2003.

[8] T. Tuytelaars and L. van Gool. *Matching Widely Separated Views Based on Affine Invariant Regions*, International Journal of Computer Vision 59(1), 2004.

[9] M. Beal. *Variational Algorithms for Approximate Bayesian Inference,* Ph.D. Thesis, 2003. Chapters 1-2.