2010 International Conference on Pattern Recognition

Detection of Shapes in 2D Point Clouds Generated from Images

Jingyong Su
Department of Statistics
Florida State University
Tallahassee, FL 32306 USA
jingyong@stat.fsu.edu

Zhiqiang Zhu
Department of Computer Science
Colorado State University
Fort Collins, Colorado 80523 USA
napozhu@hotmail.com

Anuj Srivastava, Fred Huffer
Department of Statistics
Florida State University
Tallahassee, FL 32306 USA
anuj@stat.fsu.edu, huffer@stat.fsu.edu

***Abstract*—We present a novel statistical framework for detecting pre-determined shape classes in 2D cluttered point clouds, which are in turn extracted from images. In this model-based approach, we use a 1D Poisson process for sampling points on shapes, a 2D Poisson process for points from background clutter, and an additive Gaussian model for noise. Combining these with a past stochastic model on shapes of continuous 2D contours, and optimization over unknown pose and scale, we develop a generalized likelihood ratio test for shape detection. We demonstrate the efficiency of this method and its robustness to clutter using both simulated and real data.**

***Keywords*-shape detection, poisson process, generalized likelihood ratio test.**

I. Introduction

The central problem in the area of image analysis and computer vision is to characterize objects present in the given images. A vast majority of image understanding problems - indexing, classification, detection, recognition, and understanding - rely on analysis, statistical or otherwise, of certain features of objects present in images. An important feature of many objects is their **shape**. One way to use shape analysis is to estimate the boundaries of the objects (in images) and to analyze the shapes of those boundaries. For 2D images, the object boundaries usually are simple closed curves and there is now a large literature on shape analysis of continuous, closed, planar curves. While such continuous formulations are important, the practical situations mostly involve heavily under-sampled, noisy, and cluttered discrete data, often because the process of estimating boundaries uses low-level techniques that extract a set of primitives (points, edges, arcs, etc.) in the image plane [1]. Figure 1 shows some examples of generating point clouds from images. Therefore, an important problem in object recognition is to relate a given cluttered cloud to the pre-determined (continuous) shape classes and to classify this cloud using a *fully statistical framework*.

While the problem of fitting 3D shapes (surfaces) to point clouds in $\mathbb{R}^3$ has been studied frequently, see for e.g. [2], the literature is very limited for the 2D version. The biggest challenge in such problems is to select and organize a large subset of the given primitives into configurations that

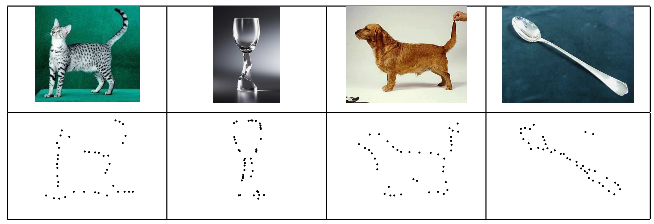

Figure 1. Examples of pre-processing an image into a 2D point cloud.

resemble the shapes of interest. The number of permutations for organizing primitives into shapes is huge and to evaluate all these shape permutations is impossible. One needs a principled framework: (i) to decide which points are from a shape and which are background clutter, (ii) to order the selected primitives into polygons resembling shapes, and (iii) to associate the resulting (under-sampled) polygons with given shape classes. Note that depending upon where the primitives are placed on the curve, the resulting polygons can have very different shapes. To reach a statistical framework for this classification, one has to develop models for the variabilities associated with shapes, the generation of primitives (i.e. sampling of contours into primitives), and the observation noise and clutter. In this paper we restrict to point primitives only.

A recent paper [3] developed a Bayesian approach by estimating the posterior probability of a shape class in a point cloud using the Monte Carlo estimation. In this top-down approach, the authors estimate the probability that a given shape is present in a cloud by integrating over unknown variables such as pose, scale, and point labels (clutter or not clutter). Even the shape variability within a shape class is accounted for using stochastic shape models. In the current paper we take a similar top-down approach except for the choices of data models and the techniques for inference are different. We will setup the detection problem as a generalized likelihood ratio test rather than a Bayesian inference, and will utilize fully Poisson models for points coming from shapes and clutter. This framework is more efficient as it does not involve Monte Carlo estimation, but uses maximum-likelihood estimation instead. In Section 2 we describe the chosen models for 2D point clouds

1051-4651 2010

DOI 10.1109/ICPR.2010.647

IEEE computer society

generated by sampling shapes in cluttered environments and in Section 3 we demonstrate some detection results.

II. Model-Based Shape Detection

We take a model-based approach where we evaluate the likelihood of a point cloud observation to be associated with a shape class, and compare that likelihood with the likelihood of having pure clutter. This estimated likelihood helps solve both the detection and the classification problems in a principled manner.

Assume that we are given a cloud $\mathbf{y} = \{\mathbf{y}_i \in \mathbb{R}^2 | \ i = 1, 2, \ldots, m\}$ in a square domain $D = [0, 1]^2$ and we want to develop a statistical framework for deciding if there is a pre-determined shape class contained in this set. We will treat this problem as that of binary hypothesis testing – the null hypothesis is that $\mathbf{y}$ is simply clutter, and the alternate hypothesis is that $\mathbf{y}$ is generated from a curve from that shape class:

$$\begin{aligned} H_0 &: \text{Shape is absent, Likelihood} \quad P(\mathbf{y}|C) \\ H_1 &: \text{Shape is present, Likelihood} \quad P(\mathbf{y}|S) \end{aligned}$$

Here, S denotes a shape class and C denotes the clutter. Now we describe the probability models associated with the observed points. The points present in a given point cloud can be one of two types: (i) points belonging to the shapes and (ii) the points associated with the background clutter. We will propose an observation model for each of them separately.

A. Detection of a Fixed Curve

In this section we focus on detection of a shape formed by a curve β, where $\beta : [0, 1] \to \mathbb{R}^2$ is a closed, continuous contour, with shape belonging to S. It is fixed momentarily and is parameterized by the arc-length, i.e. $\|\dot{\beta}(t)\| = 1$ for all $t \in [0, 1]$. This fixes the length of the curve β to be one.

1) **Points belonging to** β: We assume that these points are a realization of a **one-dimensional** Poisson process on the parameterized contour β. Let $\gamma : [0, 1] \to \mathbb{R}_+$ be the intensity function of the Poisson process along β; the number of points generated from any part of the curve is a Poisson random variable with mean being the integral of γ on that part. Let the points sampled from β be denoted by $\mathbf{x}$. The actual observations are assumed to be noisy versions of the elements of $\mathbf{x}$; the noise is i.i.d. Gaussian with mean zero and variance $\sigma^2 I$.
2) **Points associated with clutter**: This subset of observations, independent of the first subset, comes from the clutter and we model them as realizations of a **two-dimensional** Poisson process with the intensity $\lambda : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$. Let $\Lambda = \int_{\mathbb{R}^2} \lambda(y) dy$.

We simplify by assuming that both the Poisson intensities are constant, i.e. $\Lambda = \lambda$ and $\Gamma = \gamma$.

The full point cloud $\mathbf{y}$ can now be modeled as a Poisson process with the intensity function: $\xi(y) = \int_0^1 f(y|\beta(t))\gamma(t) \ dt + \lambda(y)$, where $f \equiv N(0, \sigma^2 I)$. The probability density function of $\mathbf{y}$, given β, γ, λ, and m, is given by $P_m(\mathbf{y}|\beta, \gamma, \lambda) = (\prod_{i=1}^m \xi(\mathbf{y}_i))e^{-\lambda-\gamma}$. The null hypothesis is that all the points belong to the Poisson clutter. In that case, the likelihood functions is given by: $Q_m(\mathbf{y}|\lambda) = e^{-\lambda} \prod_{i=1}^m \lambda(\mathbf{y}_i)$. The likelihoods for both the cases, with and without shape, involve certain parameters that are generally not known beforehand. Thus, taking a simple likelihood ratio is not possible and we resort to the *generalized likelihood ratio test* (GLRT). This uses the maximum likelihood estimates of the parameters, under the respective hypotheses, and uses them for evaluating the likelihood ratio:

$$\begin{aligned} \frac{Q_m(\mathbf{y}|C)}{P_m(\mathbf{y}|S)} &= \frac{\max_\lambda (e^{-\lambda} \prod_{i=1}^m \lambda)}{\max_{\lambda,\gamma,\sigma}(e^{-\gamma-\lambda}(\prod_{i=1}^m (\lambda + \gamma\alpha_i(\sigma)))} \ . \\ &= \frac{e^{-m} m^m}{\max_{\lambda,\gamma,\sigma}(e^{-\gamma-\lambda}(\prod_{i=1}^m (\lambda + \gamma\alpha_i(\sigma)))} \ . \end{aligned}$$

where $\alpha_i(\sigma)$ is a scalar quantity given by $\frac{1}{2\pi\sigma^2} \int_0^1 e^{-\frac{1}{2\sigma^2}\|\mathbf{y}_i - \beta(t)\|^2} dt$. Notice that $\alpha_i(\sigma)$ is high if the point $\mathbf{y}_i$ is close to the curve β, with the closeness being measured relative to the scale σ. Let $\theta \in \mathbb{R}^3$ represent the vector of unknown parameters associated with the hypothesis S (presence of the shape in the point cloud), namely $\theta = [\gamma, \lambda, \sigma]$. Then, define the log-likelihood function $L : \mathbb{R}^3 \to \mathbb{R}_+$ given by

$$L(\theta) = -\gamma - \lambda + \sum_{i=1}^m \log(\lambda + \gamma\alpha_i(\sigma)) \qquad (1)$$

and let $\hat{\theta} = \operatorname{argmax}_\theta L(\theta)$ be the MLE of θ. With the estimated parameters, the log-likelihood ratio becomes $R(\mathbf{y}) \equiv -m + m\log(m) - L(\hat{\theta})$. The generalized likelihood ratio test is then given by: $R(\mathbf{y}) \gtrless \mu$, where μ is a chosen threshold.

B. Detection of a Variable Curve

So far we have assumed a fixed curve β, but of course a shape can be present in an image at an arbitrary pose and scale. In fact, within the same shape class one can have curves with different shapes allowed under that class. Therefore, β can have variable shape, position, rotation, and scale. Let $O \in SO(2)$ denote the orientation, $x \in \mathbb{R}^2$ denote its translation, and $\rho \in \mathbb{R}_+$ denote its scale. First let β_0 be a standardized curve, with a fixed shape, and define $\beta = \rho O\beta_0 + x$ be a transformed version of that curve. When we allow unknown transformations, the cost function L keeps the same form, except the function α_i now depends on these other variables also, i.e. $\alpha_i(\sigma, x, O, \rho) = \frac{1}{2\pi\sigma^2} \int_0^1 e^{-\frac{1}{2\sigma^2}\|\mathbf{y}_i - x - \rho O\beta_0(t)\|^2} dt$. These additional variables are also estimated, along with θ, using maximum-likelihood estimation. Shown in Figure 2 is an illustration of this gradient process: for a fixed point cloud, we estimate the rotation O (left), translation x (middle), and scale ρ (right) using the gradient of L.

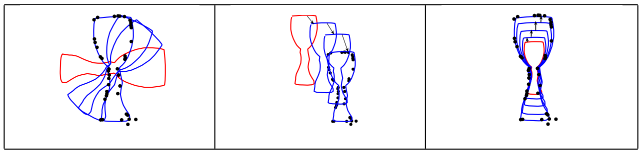

Figure 2. Estimation of pose and scale by maximizing log-likelihood function L.

Next, we want to account for the shape variability of the curve β, while remaining in the same shape class S. For the class S, we estimate a stochastic shape model of the type described in [4] and generate many random shapes from the model. For details, we refer the reader to that paper. The Figure 3 (left) shows an example of random shapes generated from a shape model for the shape class "runner". Given these samples, we maximize the log-likelihood function L over each of the random samples generated for a shape class. More specifically, these curves substitute for β_0 in the earlier optimization over L. The maximum value is then used in performing the likelihood ratio test.

The likelihood ratio $R(\mathbf{y})$ is to be compared with a threshold μ to decide if a shape is present in the data or not. Ideally, this threshold is dictated by the probability distributions of $R(\mathbf{y})$. In practical problems, where it is difficult to ascertain these distributions, one uses either the asymptotic theory or an empirical approach. Taking an empirical approach, we estimate the probability density function of the log-likelihood ratio $R(\mathbf{y})$ under the null hypothesis. We generate hundreds of realizations of $\mathbf{y}$, each using $\gamma = 0$ and a fixed m, say 40, and compute a histogram of $R(\mathbf{y})$ values. The right panel in Figure 3 shows an example. Once we have this histogram, we can find a value that allow a certain fixed value, say 0.05, of probability of Type I error, and use that value for μ.

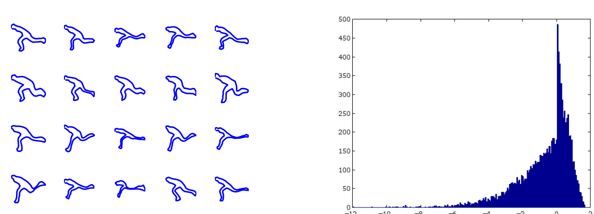

Figure 3. Left: Random samples from a shape model for the "runner" class. Right: A histogram of the likelihood ratio $R(\mathbf{y})$ under H_0, used for estimating the threshold μ.

III. Experimental Results

In this section, we describe some experimental results for detecting shapes of interest in given images.

Simulated Point Clouds: In this case we select a random curve from a shape class S, transform it with random rotation, translation and scaling, select random points on it using a homogeneous Poisson process, and add Gaussian noise to the selected points. Then we generate clutter points from a 2D Poisson process and mix the two set of points. For this cloud $\mathbf{y}$, we find the generalized likelihood ratio under S and perform the hypothesis selection by comparing it with μ. Figure 4 shows two examples of this setup. In both cases, the data was simulated for the runner class, and a large negative value of $R(\mathbf{y})$ supports rejecting H_0.

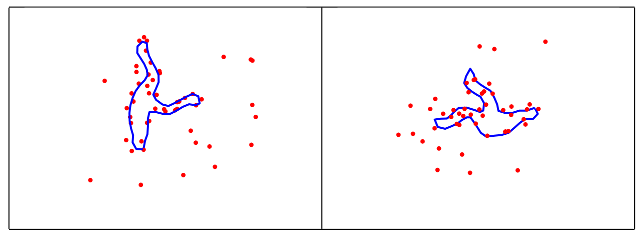

Figure 4. $R(\mathbf{y}) = -90.4$ (left) and -96 (right) strongly support H_1, $\mu = -3.8$.

As the number of sampled points on the curve decreases, or as the noise or clutter increases, the detection performance suffers. We have performed Monte Carlo averaging to study the performance of shape detection as a function of r = (# of points on the curve)/(# of clutter points) and noise std dev. σ, and have plotted the Type II error versus these variables for the shape class runner in Figure 5.

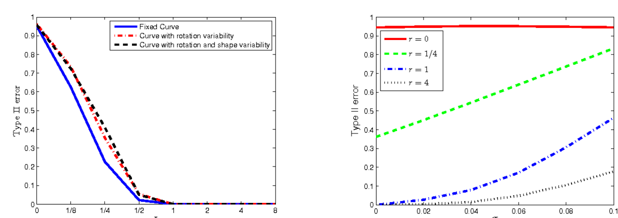

Figure 5. Detection error versus r and σ.

Point Clouds from Real Images: Starting with an image, we use a median filter to remove the noise and calculate image gradients with an auto-adaptive threshold. We then merge rectangular regions that contain objects and find the largest one, and then obtain a binary image for the rectangular region. Next, we obtain a thinning image for the rectangular region and find the contour of the object. Finally, we sample random points on the contour. An example of this whole process is shown in Figure 6. Additional examples of the resulting $\mathbf{y}$ are shown earlier in Figure 1.

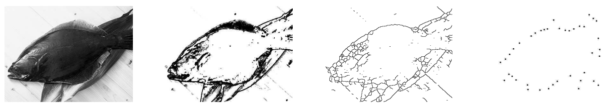

Figure 6. Processing images into clouds.

To perform shape detection in real images, we consider the four point clouds shown in Figure 1, and estimate the likelihood of the 16 shape classes shown in Figure 7. Also, we manually extract the ground truth from the images in Figure 1, and compute their likelihoods against the corresponding cloud data. The true curves help us evaluate the results that are presented in Table I. For the point cloud $\mathbf{y}_1$, the likelihood ratio is minimum (-64) for class 6 (cat) and is -78 for the ground truth. Thus, the model strongly supports the presence of a cat shape in the given data and illustrates success of this method.

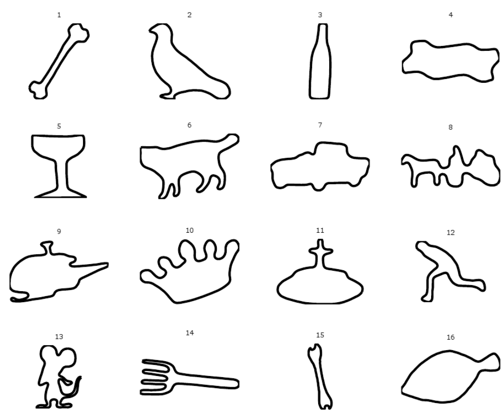

Figure 7. Training shape classes.

Table I
LIKELIHOODS OF REAL DATA.

$R(\mathbf{y})$	$\mathbf{y}_1$	$\mathbf{y}_2$	$\mathbf{y}_3$	$\mathbf{y}_4$
Truth	-78	-89	-65	-63
1	-31	-48	-25	-44
2	-43	-32	-45	-21
3	-34	-42	-27	-30
4	-35	-36	-33	-22
5	-52	-26	-50	-22
6	-64	-25	-56	-18
7	-34	-33	-34	-20
8	-51	-34	-39	-25
9	-49	-29	-41	-19
10	-56	-26	-40	-21
11	-43	-15	-48	-12
12	-57	-26	-48	-21
13	-60	-18	-38	-15
14	-53	-25	-39	-26
15	-29	-47	-22	-45
16	-22	-26	-25	-20

IV. CONCLUSION

We have presented a statistical framework for detecting shapes in cluttered point clouds and have demonstrated it using real and simulated data. This model is based on a composite Poisson process: one for primitives generated from the shape and another for primitives belonging to the background clutter. This model allows computation of a log-likelihood ratio for each class against clutter and this ratio provides sufficient statistic for detection and classification of shape classes.

ACKNOWLEDGMENT

This research was supported in part by AFOSR FA9550-06- 1-0324, ONR N00014-09-1-0664 and NSF DMS-0915003.

REFERENCES

[1] J. Thuresson and S. Carlsson, "Finding object categories in cluttered images using minimal shape prototypes," in *SCIA, Springer LNCS 2749*, 2003, pp. 1122–1129.

[2] T. K. Dey, "Computing shapes from point cloud data," in *Algorithms ESA 2002, vol. 2461*, 2002, pp. 395–399.

[3] A. Srivastava and I. H. Jermyn, "Looking for shapes in cluttered, two-dimensional point clouds," *IEEE Patt. Analysis and Machine Intell.*, vol. 31, no. 9, pp. 1616–1629, September, 2009.

[4] A. Srivastava, S. Joshi, W. Mio, and X. Liu, "Statistical shape analysis: Clustering, learning and testing," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 590–602, 2005.