ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD), Philadelphia, August 2006

# New EM Derived from Kullback-Leibler Divergence

Longin Jan Latecki
CIS Dept.
Temple University
Philadelphia, PA 19122
latecki@temple.edu

Marc Sobel
Statistics Dept.
Temple University
Philadelphia, PA 19122
Marc.Sobel@temple.edu

Rolf Lakaemper
CIS Dept.
Temple University
Philadelphia, PA 19122
lakamper@temple.edu

## ABSTRACT

We introduce a new EM framework in which it is possible not only to optimize the model parameters but also the number of model components. A key feature of our approach is that we use nonparametric density estimation to improve parametric density estimation in the EM framework. While the classical EM algorithm estimates model parameters empirically using the data points themselves, we estimate them using nonparametric density estimates.

There exist many possible applications that require optimal adjustment of model components. We present experimental results in two domains. One is polygonal approximation of laser range data, which is an active research topic in robot navigation. The other is grouping of edge pixels to contour boundaries, which still belongs to unsolved problems in computer vision.

## Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: General

## General Terms

Algorithms, Performance, Experimentation

## Keywords

EM, Expectation Maximization, Kullback-Leibler divergence

## 1. INTRODUCTION

Our goal is to approximate the ground-truth density $q(x)$ with a member $p_\Theta(x)$ of a parametric family $\{p_\Theta(x) : \Theta \in \mathcal{S}\}$ of densities. We use Kullback-Leibler divergence (KLD) to measure dissimilarity between the ground-truth and parametric family of densities. By definition, the KLD between the ground truth $q(x)$ and the density, $p_\Theta(x)$ is:

$$D(q(x)||p_\Theta(x)) = \int \log \frac{q(x)}{p_\Theta(x)} q(x)dx$$
$$= \int \log q(x)q(x)dx - \int \log p_\Theta(x)q(x)dx \qquad (1)$$

The data itself, being noisy, do not directly correspond to the ground truth density. We demonstrate below that the ground-truth density $q(x)$ can be estimated from the data. The use of Kullback-Leibler divergence (KLD) enables us to fit an optimal model to the ground truth rather than the noisy data.

Observe that KLD is able to approximate the optimal number of model components of $p_\Theta$. This is due to the fact that KLD $D(q||p_\Theta)$, viewed as a functional on the space $\{p_\Theta\}$ of Gaussian mixtures, is convex and hence has a unique minimum. However, this minimum does not have to be a finite mixture of Gaussians, since the space of finite Gaussian mixtures is not closed. On the other hand, the set of finite Gaussian mixtures is dense in the space of continuous functions. Therefore, we can approximate the minimum with any required precision when we minimize KLD in the space of finite Gaussian mixtures. In particular, this means that we can estimate the number of mixture components, but it is impossible to determine the optimal number of components, since this number may be large or infinite (e.g., some ground truth model components could be very small). Therefore, using KLD we are able to correctly estimate the number of 'large' (or significant) model components.

It is known that the Expectation Maximization (EM) algorithm can be derived from KLD (Section 2). However, in the EM framework the number of model components must be known and fixed. This is due to the fact that the log likelihood function that is optimized in the EM framework increases when the number of model components is increased. Thus, when optimizing the log likelihood, we cannot estimate the number of model components.

An important question which arises is: why is the ability to estimate the optimal number of model components lost in the derivation of EM from KLD? In this paper we provide an answer to this question and derive a new EM target function from KLD that allows us to optimize not only the model parameters but also to estimate the number of the model components. Moreover, in the proposed framework, EM converges to an optimal solution even if the initial values of model parameters are not close to the global optimum.

There exist many possible applications that require optimal adjustment of model components. We illustrate our approach on polygonal approximation of laser range data and object contours in digital images. Polygonal maps obtained by polygonal approximation of laser range data are very attractive means to represent range scan data due to their very compact size and simplicity. Hence they lead to huge data compression and make it easier to access higher level features. Therefore, several approaches have been proposed to obtain such maps, the most recent ones being [17, 11, 8]. An excellent overview can be found in [17]. Although approximation with higher order curves is possible, approximation with lines is more stable in the presence of noise (e.g., see Ch. 5 in [14]), which is the case for laser range scans. Therefore, we focus on polygonal approximation in this paper. However, the proposed EM framework has a broader scope of possible applications. Polygonal approximation of edge pixels in digital images can be interpreted as grouping of edge pixels to parts of object contours, which belongs to unsolved problems of computer vision. The approaches to grouping of object contours date back to the the first results of Gestalt psychology in the beginning of 20th century [18], and they remain an active research topic in computer vision. An overview of techniques for polygonal approximations of curves, which require that the order of data points is known, can be found in [10].

The main difficulty of fitting polylines in the above applications is that the segmentation (or correspondence) of data points to line segments as well as the order of data points are unknown. The Expectation Maximization (EM) algorithm [2] provides a particularly useful framework to solve this correspondence problem. Actually EM applied to line fitting is known as the Healy-Westmacott procedure in statistics, and predates EM by many years [6]. However, polygonal approximation of point data requires that not only the model parameters but also the number of model components (line segments) are estimated, but as observed above in the EM framework the number of model components must be known and fixed. Moreover, EM produces an optimal solution only if the number of model components is well estimated and the initial values of model parameters are close to the global optimum.

We give a simple example that illustrates the fact that EM yields a locally optimal solution if the initial values of model parameters are not close to globally optimal values. In Fig. 1 we see data points that follow the horizontal and vertical lines in a cross like pattern. Fig. 1(a) shows two diagonal lines that form the initial configuration of the standard EM algorithm. The number of model components (two lines) is correctly initialized, but their initial position is not close to the global optimum. Fig. 1(b) shows the final, locally optimal, result obtained by the classical EM algorithm. Finally, Fig. 1(c) shows the globally optimal approximation obtained by the proposed method on the same input.

Due to the problem of getting stuck in local optima, a correct estimation of the number of components and the parameter values of a statistical model is crucial in all EM applications, and therefore, belongs to one of the most challenging problems in statistical reasoning. Before we describe the proposed approach, we review existing solutions.

The existing solutions can be divided into two categories. The first category is based on using penalty functions like the Bayesian Information Criterion (BIC), or alternatively, the Minimum Description Length (MDL), and Akaike Information Criterion (AIC), to determine the optimal number of model components. The approaches in this category require that EM is run until it converges, whatever the initial number of components assumed, with the goal of selecting the components exemplified by the ground truth. As we show below, approaches of this sort cannot be guaranteed to correctly estimate the optimal number of model components because EM may get stuck in local optima.

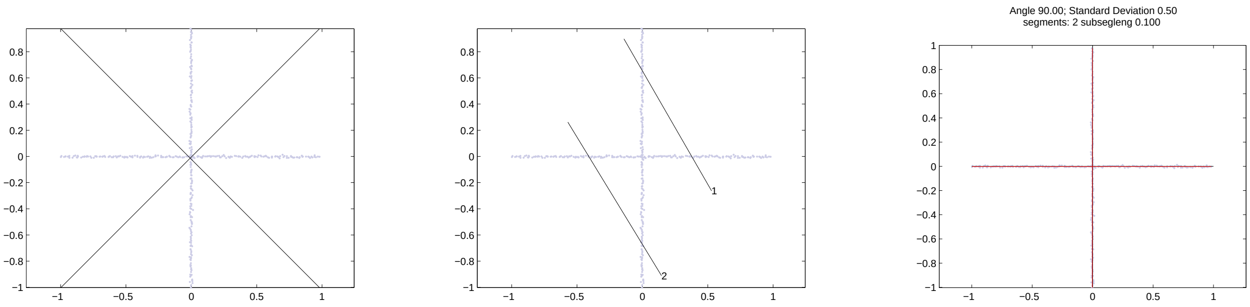
In [1] the use of BIC and AIC to estimate the number of model components is discussed. We focus here on BIC but our arguments also apply to AIC and MDL. For a fixed number of data points, which is the case in our application at each given time $t$, the use of BIC represents a trade-off between emphasizing the importance of model complexity and the likelihood of the data. Typically a model that has the greatest BIC values is selected by repeatedly comparing these values for all possible numbers of model components. The problem with this approach is that its success depends on the convergence of the EM algorithm to the global optimum whatever the initial number of model components assumed. If, for some given initial starting configuration, EM gets stuck in a local optimum, the BIC estimate will incorrectly estimate the ground truth number of model components. For example, the correct number of model components could not be determined (using BIC methodology) for the situation in Fig. 1(a,b). Since EM got stuck in a local optimum in (b), the likelihood of the model with two components is very low, and consequently the ground-truth model with two components is not selected. To the best of our knowledge this problem is not addressed by any existing approach designed to estimate the number of model components.

Moreover, in practice there is a hidden parameter that is manually adjusted to obtain the desired number of model components in BIC. This parameter is the standard deviation of the measurement process. In BIC this standard deviation acts as a tradeoff weighting factor between the likelihood of the data points and the model complexity. As determined experimentally on ground-truth data in [1], BIC tends to over weight the penalty on model complexity, which leads to a too small number of model components.

The second category of approaches to estimate the optimal number of components is based on steps involving splitting and merging of EM model components after each algorithm iteration. Our approach belongs to this category. It is important to mention that the approaches in the second category yield a quicker convergence since they adapt the number of model components and model parameters to the given environment after every algorithm iteration while BIC requires convergence for each given initial number of model components.

We will first show that the existing split and merge approaches cannot be guaranteed to correctly estimate the optimal number of model components due to the fact that they cannot recognize locally optimal solutions that are not globally optimal.

In 1995 Green [3] proposed a solution based on iterative merging and splitting components of a mixture model with the goal of obtaining a better mixture model in the case of univariate normal mixtures. Green's solution is based on a fully Bayesian mixture analysis that makes use of jump Markov chain Monte Carlo (MCMC) methods. The jumps are realized by split and merge moves that are reversible. Since Green's merge move is evaluated using the data points,



(a) (b) (c)

**Figure 1: (a) shows the data points and the initial position of model lines. (b) shows the optimal approximation of the data points obtained by EM. (c) shows the optimal approximation result obtained by the proposed method.**

it requires an additional penalty for the number of model components. The number of model components depends largely on this penalty, which is not directly related to the model quality assessment, as is the case in our approach. Green's approach is used to fit polygons to contours in digital images in [9]; in this setting split moves correspond to inserting a new vertex into the polygon and merge moves correspond to removing a vertex. Greens algorithm requires a huge number of iterations (Green reported the need for 20,000 iterations). This is due to a random selection of vertices, which is counterintuitive from the point of view of human visual perception. Humans are able to identify good and bad fitting parts of a given polygon by visual inspection. In consequence of this, it makes more sense to base algorithm moves on local visual inspection rather than on random selection.

In 2000 Ueda et al. [16] proposed a split and merge extension of the EM framework for mixture models. Their split and merge rules do not require any penalty as is the case for Greens approach. However, as we will now show, their approach is not able to recognize some locally optimal solutions that are not globally optimal. Their merge criterion is based on posterior probabilities associated with the model components. Two model components $\omega_i$ and $\omega_j$ are merged if they have almost equal posterior probabilities over the data points; this means that the probability of being generated by either component is approximately equal for all data points (formula (15) in [16]). Defining model components as line segments, this means that data points are approximately the same distance to either one of components that are under consideration to be merged. Observe that the two model components (diagonal line segments) in Fig. 1(a) are merged by their rule. This, however, incorrectly results in a single line segment that cannot provide good support for the cross-shaped data points.

A single model component is split if the local data density is significantly different from the global density; both densities are estimated using the actual component parameters of this component (formula (16) in [16]). This split criterion fails in our application, where the model components are line segments. The single line segment in Fig. 2 is not split by this criterion, since both densities are identical (i.e., match perfectly). However, clearly two line segments are needed to obtain an optimal fit to the data points. This critique also applies to the approach in [19] that uses the same split criterion.

The above problems also explain why the algorithm by Ueda et al. [16] needs a relatively large number of iterations to converge. [16] reports that about 350 iterations are needed to fit lines to data points. The proposed algorithm usually converges in less than 20 iterations.

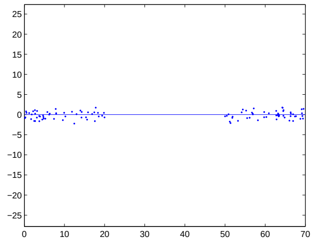


**Figure 2: Clearly two line segments are needed to obtain an optimal fit do the depicted data points.**

We observe that all the split and merge steps presented in the literature optimize different target function than the function optimized by the classical M step of the EM algorithm. Here we propose split and merge steps that optimize the same target function (Sections 3 and 4).

## 2. OPTIMIZING KLD

It can be easily derived that the parameters $\widehat{\Theta}$ minimizing (1) are given by

$$\widehat{\Theta} = \text{argmax}_{\Theta}\{ \int [\log p_{\Theta}(x)] q(x) dx \} \tag{2}$$

We obtain the classical maximum likelihood estimator by applying the MC (Monte Carlo) integral estimator to (2) under the assumption that the observations $x_1, ..., x_n$ are i.i.d. (independently and identically distributed) sample points selected from the ground truth distribution $q(x)$.

$$\widehat{\Theta} = \text{argmax}_{\Theta} \sum_i \log p_{\Theta}(x_i) \tag{3}$$

However, as we derive below (equation (9)), if some proportion of the observations $x_1, ..., x_n$ is noisy, a more accurate estimator of $\Theta$ in (2) is given by:

$$\widehat{\Theta} = \text{argmax}_\theta \sum_i \log p_\theta(x_i) sdd(x_i), \qquad (4)$$

where $sdd$ is called the **smoothed data density** and is defined in Section 5 by the means of nonparametric density estimation.

Equation (4) is the basis of the proposed approach. To demonstrate the significance of (4), we consider the problem of estimating the optimal number of model components by minimizing the KLD $D(q(x)||p_\Theta(x))$ in $\Theta$. It is well known that (3) cannot be used to estimate the correct number of model components, since (3) increases when the number of model components increases. In contrast, we are able to determine the correct number of model components when using (4) to estimate the KLD, $D(q(x)||p_\Theta(x))$. Thus, the modified EM algorithm that maximizes (4) is not only able to estimate model parameters but also the right number of model components.

One of the key steps in the derivation of (4) is the Monte Carlo (MC) estimate of the integral given by the right hand side of equation (1). Let $x_1, \ldots, x_n$ be i.i.d. sample points drown from the probability density function (pdf) $q(x)$. Then we can approximate the integral of a continuous function $f$ by its MC estimate:

$$\int f(x)q(x)dx \approx \frac{1}{n}\sum_i f(x_i) \qquad (5)$$

In the usual approach to inference, it is a commonly accepted assumption that sample data points $x_1, \ldots, x_n$ are distributed according to the (estimated) density $q(x)$. This assumption is the key to insuring that maximum likelihood estimators are appropriate for purposes of estimating parameters of interest. However, in all real applications, the sample data points are corrupted by a certain amount of noise. Usually the proportion of noisy points does not decrease when the number of sample points is increased. We quantify this corruption by assuming that the data follow a distribution consisting of a mixture of an unknown ground-truth distribution $q(x)$ and an unknown noise distribution $\eta(x)$. Let $u(x) = \alpha q(x) + (1-\alpha)\eta(x)$ denote this mixture distribution. The quantity, $\alpha$ is the probability that an observation comes from the ground-truth distribution $q(x)$ and $(1-\alpha)$ is the probability that it comes from the noise distribution. Since the observed sample data points do not follow the ground truth distribution $q(x)$ but the mixture of noise and true distribution $u(x)$, we obtain a more accurate MC estimate of the integral in (5) $\int f(x)q(x)dx =$

$$\frac{\int f(x)q(x)dx}{\int q(x)dx} = \frac{\int f(x)\frac{q(x)}{u(x)}u(x)dx}{\int \frac{q(x)}{u(x)}u(x)dx} \approx \frac{\sum_i f(x_i)\frac{q(x_i)}{u(x_i)}}{\sum_i \frac{q(x_i)}{u(x_i)}} \qquad (6)$$

In Section A we show that equation (6) leads to a substantially smaller mean squared error in the estimation of the integral than equation (5). The ratio

$$\frac{\alpha q(x)}{u(x)} = \frac{\alpha q(x)}{\alpha q(x) + (1-\alpha)\eta(x)} \qquad (7)$$

is equivalent to the conditional probability, $P(ground\, truth|x)$, that an observed data point $x$ is selected from the ground truth density $q(x)$. We note that large values of $P(ground\, truth|x)$ indicate that the data point $x$ is of significant interest for inference purposes; small values indicate the reverse.

In Section 5 we show that it is possible to estimate a ratio proportional to (7) with the smoothed data density $sdd(x)$. Consequently,

$$\int f(x)q(x)dx \approx \frac{\sum_i f(x_i)sdd(x_i)}{\sum_i sdd(x_i)} \qquad (8)$$

By identifying $sdd(x_i)$ with its normalized value $\frac{sdd(x_i)}{\sum_j sdd(x_j)}$ for $i = 1, ..., n$, we can rewrite equation (8) in the form

$$\int f(x)q(x)dx \approx \sum_i f(x_i)sdd(x_i) \qquad (9)$$

Finally equation (4) clearly follows from (9) and (2).

## 3. E AND M STEPS

We introduce latent variables $z_1, ..., z_n$ which serve to properly label the respective data points $x_1, ..., x_n$. It is assumed that the pairs $(x_i, z_i)$ for $i = 1, \ldots, n$ are i.i.d. with common (unknown) joint (ground truth) density, $q(x,z) = q(x)q(z|x)$; $q(x)$ is the marginal x-density and $q(z|x)$ is the conditional density of the label $z$ given $x$. In this new framework, the KLD between the joint density $q(x,z)$ and a parametric counterpart density $p_\Theta(x,z)$ is

$$D(q(x,z)||p_\Theta(x,z)) = D(q(x)q(z|x)||p_\Theta(x)p_\Theta(z|x))$$
$$= \int_x \int_z \left\{ \log\left[\frac{q(x)}{p_\Theta(x)}\right] + \log\left[\frac{q(z|x)}{p_\Theta(z|x)}\right] \right\} q(x)q(z|x)dzdx$$
$$= \int_x \log\left[\frac{q(x)}{p_\Theta(x)}\right] q(x)dx + \int_x q(x) \int_z \log\left[\frac{q(z|x)}{p_\Theta(z|x)}\right] q(z|x)dz \qquad (10)$$

We are now ready to introduce the expectation (E) and maximization (M) steps. Both steps aim at minimizing the same target function (10) in our framework. The expectation step yields the standard EM formula; considerations discussed above lead to a different solution for the maximization step.

**Expectation Step:** For a fixed set of parameters $\Theta$, we want to find a conditional density $q(z|x)$ that minimizes $D(q(x,z)||p_\Theta(x,z))$. Since KLD is always nonnegative, and the second summand in (10) is minimized for $q(z|x) = p_\Theta(z|x)$ (in which case it is equal to zero), we obtain from (10) that

$$q(z|x) = p_\Theta(z|x) \quad \text{minimizes} \quad D(q(x,z)||p_\Theta(x,z)).$$

In particular, for given sample points $x_1, \ldots, x_n$, we obtain

$$q(z_i = l|x_i) = p_\Theta(z_i = l|x_i) = p(z_i = l|x_i, \Theta) \qquad (11)$$
$$= \frac{p(x_i|z_i = l, \Theta)p(z_i = l|\Theta)}{p(x_i|\Theta)} = \frac{p(x_i|z_i = l, \Theta)\pi_l}{\sum_{j=1}^k p(x_i|z_i = j, \Theta)\pi_j} \qquad (12)$$

where $\pi_l = p(z_i = l|\Theta)$ and $\pi_j = p(z_i = j|\Theta)$ are the prior probabilities of component labels $l$ and $j$ correspondingly.

**Maximization Step:** For the fixed marginal distribution $q(z|x) = p_\Theta(z|x)$, we want to find a set of parameters $\Theta$ that maximizes (10). Substituting $q(z|x) = p_\Theta(z|x)$ in (10), we obtain

$$D(q(x,z)||p_\Theta(x,z)) = \int \log(\frac{q(x)}{p_\Theta(x)})q(x)dx = D(q(x)||p_\Theta(x)) \qquad (13)$$

Thus, minimizing $D(q(x,z)||p_\Theta(x,z))$ in $\Theta$ is equivalent to minimizing $D(q(x)||p_\Theta(x))$ in $\Theta$. Using the estimate derived in equation (4), minimizing (13) in $\Theta$ is equivalent (in the MC setting discussed above) to maximizing the weighted marginal density

$$WM(\Theta) = \sum sdd(x_i) \log p_\Theta(x_i) = \sum sdd(x_i) \log p(x_i|\Theta)$$
$$= \sum_{i=1}^{n} sdd(x_i) \log[\sum_{l=1}^{k} p(x_i|z_i = l, \Theta) p(z_i = l|\Theta)]$$
$$= \sum_{i=1}^{n} sdd(x_i) \log[\sum_{l=1}^{k} p(x_i|z_i = l, \Theta)\pi_l] \quad (14)$$

where $\pi_l = p(z_i = l|\Theta)$ are the prior probabilities of component labels $l = 1, \ldots, k$.

Now we explicitly use the incremental update steps of the EM framework. Using the prior probabilities of component labels $\pi_l^{(t)} = p(z_i = l|\Theta^{(t)})$ obtained at stage $t$ for $l = 1, ..., k$, we obtain from (14) that an update of $WM(\Theta)$ is estimated by maximizing

$$WM(\Theta; \Theta^{(t)}) = \sum_{i=1}^{n} sdd(x_i) \log[\sum_{l=1}^{k} p(x_i|z_i = l, \Theta)\pi_l^{(t)}] \quad (15)$$

in $\Theta$ with $\Theta^{(t)}$ denoting the value of $\Theta$ computed at stage $t$ of the algorithm. The crucial difference between this and the standard EM update is that our target function is weighted with terms $sdd(x_i)$. We note that the known convergence proofs for the EM algorithm apply in our framework, since adding the weights $sdd(x_i)$ in (15) does not influence the convergence.

## 4. SPLIT AND MERGE

The proposed split and merge steps adjust the number of model components by performing component split and merge steps only if they increase the value of our target function (15). Since the proposed split and merge steps are computed in the sparse EM framework, the convergence of our algorithm follows from [7].

Our framework is very general in that it allows many possible selections of the candidate components for the split and merge steps. We present specific selection methods of the candidate components in Section 8. They are based on a Maximum A Posteriori principle. In the following formulas, we assume that the candidate components are given.

**Split:** Assume that we are given two candidate model components $l_1, l_2$; we consider replacing the model component $l$ with components $l_1, l_2$. Since our goal is maximizing $QM(\Theta; \Theta^{(t)})$ in formula (15), we simply need to check whether replacing $l$ with $l_1, l_2$ increases $WM$, where $j \in \{1, \ldots, k\}$:

$$WM(\Theta; \Theta^{(t)}) = \sum_{i=1}^{n} sdd(x_i) \log[\sum_{j} p(x_i|z_i = j, \Theta)\pi_j^{(t)}]$$
$$< \sum_{i=1}^{n} sdd(x_i) \log[\sum_{j \neq l} p(x_i|z_i = l, \Theta)\pi_l^{(t)}$$
$$+ \quad p(x_i|z_i = l_1, \Theta)\pi_{l_1}^{(t)} + p(x_i|z_i = l_2, \Theta)\pi_{l_2}^{(t)}] \quad (16)$$

We only need to perform 'local' computation to perform this test, i.e., we only need to compute the corresponding probabilities for the candidate components $l_1, l_2$, subject to the condition that $\pi_l^{(t)} = \pi_{l_1}^{(t)} + \pi_{l_2}^{(t)}$. The parameters are estimated following the sparse EM step in Neal and Hinton [7], (see equation (15)). In accordance with the results of [7] this local computation guarantees that the target function increases after each iteration (if (16) holds). Convergence is also guaranteed in this way.

**Merge:** Given a candidate component $l$, we merge two existing model components $l_1, l_2$ to $l$ if for $j \in \{1, \ldots, k\}$

$$WM(\Theta; \Theta^{(t)}) = \sum_{i=1}^{n} sdd(x_i) \log[\sum_{j} p(x_i|z_i = j, \Theta)\pi_j^{(t)}]$$
$$> \sum_{i=1}^{n} sdd(x_i) \log[\sum_{j \neq l} p(x_i|z_i = l, \Theta)\pi_l^{(t)}$$
$$+ \quad p(x_i|z_i = l_1, \Theta)\pi_{l_1}^{(t)} + p(x_i|z_i = l_2, \Theta)\pi_{l_2}^{(t)}] \quad (17)$$

Again we only need to perform 'local' computations to perform this test. For merge, we only need to compute the corresponding probabilities for the candidate component $l$, subject to the same constraint $\pi_l^{(t)} = \pi_{l_1}^{(t)} + \pi_{l_2}^{(t)}$. If (17) holds and we replace $l_1, l_2$ with $l$, the convergence of our algorithm follows from the results of [7].

We note that the proposed split and merge steps do not work in the classical EM framework. To see this, consider $sdd(x_i) = 1$ for all the data points ($i = 1, \ldots, n$). The merge inequality (17) is not satisfied even if the ground truth model is assumed to be a single component, since multiple components can better fit the data, and consequently have a larger log likelihood value. Analogously, if the split inequality (16) holds for a reasonable selection of candidate component models, the classical EM framework incorrectly splits ground truth components. Thus, a mixture model of larger number of components is always prefered in the classical EM framework. In the proposed framework, $sdd$ represents an estimated density of the data points (estimated in a nonparametric way as described in Section 5). Consequently, in the proposed split and merge steps, the divergence of parametric components $l, l_1, l_2$ from the ground truth is evaluated with respect to this nonparametric density.

## 5. ESTIMATING THE DATA DENSITY

In this section, we construct the function $sdd(x)$ that estimates the ratio (7). Following the assumption made in calculating bootstrap samples, we can estimate the density, $u(x)$ on the observed i.i.d. sample points $x_1, \ldots, x_n$ drawn from $u(x)$ by $\widehat{u}(x_1) = \cdots = \widehat{u}(x_n) = \frac{1}{n}$.

We use a kernel estimate, which is the most widely-used nonparametric density estimation method, to estimate the ground truth density $q(x)$. Thus, under the assumption that $x_1, \ldots, x_n$ are i.i.d. sample points, we estimate the ratio (7) with a smoothed data density obtained by

$$sdd(x_j) \propto \frac{q(x_j)}{u(x_j)} \approx nq(x_j) = n \sum_{i=1}^{n} K(\frac{d(x_j, x_i)}{h})$$
$$= \frac{n}{nh} \sum_{i=1}^{n} G(d(x_j, x_i), 0, h), \quad (18)$$

where proportionality refers to the fact that $\sum sdd(x_i) = 1$, $d(x, y)$ is the Euclidean distance, and $G(d(x, y), 0, h)$ is a Gaussian with mean zero and the standard deviation (std) $h$. An intuitive motivation for (18) is as follows:

If a given data point $x_j$ was sampled from the true distribution $q(x)$, then the ratio $\frac{q(x_j)}{u(x_j)}$ would be large. Since the ratio is proportional (see equation (7)) to the probability, $P(groundtruth|x_j)$, this too would be large. As a consequence of this latter fact, $x_j$ would be likely to lie in a dense region of the observed sample points and consequently $sdd(x_j)$ would be large.

If a given data point $x_j$ were sampled from the noise distribution $\eta(x)$, then the ratio, $\frac{q(x_j)}{u(x_j)}$ would be small. For analogous reasons, this implies that $x_j$ would be likely to lie in a sparse region of the sample space, and consequently $sdd(x_j)$ would be small.

To estimate the bandwidth parameter $h$, we can draw from a large literature on nonparametric density estimation [12, 13]. As we show in the presented experimental results, an accurate bandwidth estimation in not crucial in our approach. It is also possible to use variable bandwidth [15].

## 6. SPECIFIC DETAILS OF THE M STEP

In equation (15) of Section 2 it was shown that minimizing the Kullback Leibler Divergence in the parameters $\Theta$ amounts to maximizing the weighted marginal density $WM(\Theta)$. We use this fact throughout the discussion below.

The goal of this section is to show that formulas for maximizing (15) are analogous, except for multiplication by $sdd$, to log likelihood maximization in the standard EM algorithm. To illustrate this we compute a partial derivative of (15) over one of the model parameters $\theta_j$ from the parameter vector $\Theta$ that is a parameter of j'th model component.

$$\frac{\partial}{\partial\theta_j} WM(\Theta;\Theta^{(t)}) \quad (19)$$

$$= \sum_{i=1}^{n} sdd(x_i)\frac{1}{p(x_i|\Theta)}\frac{\partial}{\partial\theta_j}\sum_{l=1}^{k} p(x_i|z_i=l,\Theta)\pi_l^{(t)} \quad (20)$$

$$= \sum_{i=1}^{n} sdd(x_i)\frac{\pi_j}{p(x_i|\Theta)}\frac{\partial}{\partial\theta_j} p(x_i|z_i=j,\Theta) \quad (21)$$

$$= \sum_{i=1}^{n} sdd(x_i)\frac{\pi_j p(x_i|z_i=j,\Theta)}{p(x_i|\Theta)}\frac{\partial}{\partial\theta_j}\log p(x_i|z_i=j,\Theta) \quad (22)$$

$$= \sum_{i=1}^{n} sdd(x_i)p(z_i=j|x_i,\Theta)\frac{\partial}{\partial\theta_j}\log p(x_i|z_i=j,\Theta) \quad (23)$$

The transitions from (20) to (21) and from (21) to (22) are based on

$$\frac{\partial}{\partial x}\log f(x) = \frac{1}{f(x)}\frac{\partial}{\partial x}f(x).$$

The transition from (22) to (23) is based on the Bayes rule.

For example, in the 1D case when $\theta_j$ is the mean of one of the Gaussian mixture components, we can substitute $p(x_i|z_i=j,\Theta) = \exp(\frac{(x_i-\theta_j)^2}{-2\sigma^2})$ and set (23) equal to zero:

$$\sum_{i=1}^{n} sdd(x_i)p(z_i=j|x_i,\Theta)\frac{(x_i-\theta_j)}{\sigma} = 0 \quad (24)$$

Then we obtain in the 1D case

$$\theta_j = \frac{\sum_{i=1}^{n} sdd(x_i)p(z_i=j|x_i,\Theta)x_i}{\sum_{i=1}^{n} sdd(x_i)p(z_i=j|x_i,\Theta)} \quad (25)$$

## 7. ONE DIMENSIONAL EXAMPLE

Below, we use the notation, $G(x;\mu;\sigma)$ for the Gaussian density at $x$ with mean $\mu$ and standard deviation $\sigma$. We generated a 1 dimensional data set $x_1,...,x_n$ (with n=500) from the noisy density,

$$u(x) = \begin{cases} G(x;10;3) & wprob\ 30\% \\ G(x;20;3) & wprob\ 30\% \\ G(x;30;3) & wprob\ 35\% \\ G(x;30;30) & wprob\ 5\% \end{cases} \quad (26)$$

See Fig. 3(a) for a plot of the generated data with groundtruth groups marked with different symbols.

We employed a split and merge algorithm with 5 initial groups with group labels chosen randomly. For each component considered for possible splitting, our algorithm searched for a component point, whose density, as measured by $sdd$, is more than 1 standard deviation below the average component density. If no such point exists, the component is not split. Splits are accepted if they cause the objective function to increase its value. All pairs of components are considered for possible merging. Splits and merges are accepted if they cause the objective function to increase from its former value in accord with formulas 16 and 17, correspondingly.

The results obtained by the proposed algorithm are illustrated in Fig. 4. To illustrate the relationship between the smoothing bandwidth $h$ of $sdd$ and robustness properties of the parameter estimates, we repeat our algorithm for different values of $h$. Smaller values of (the bandwidth) $h$ result in less smoothing; larger values result in more smoothing. The bandwidth of $h = 1.2$, calculated using least squares cross-validation (see [5]), is optimal in this setting. This follows from a general theorem relating the optimal bandwidth to the standard deviation and sample size. The point labels obtained by our algorithm for $h = 1.2$ are shown in Fig. 3(b).

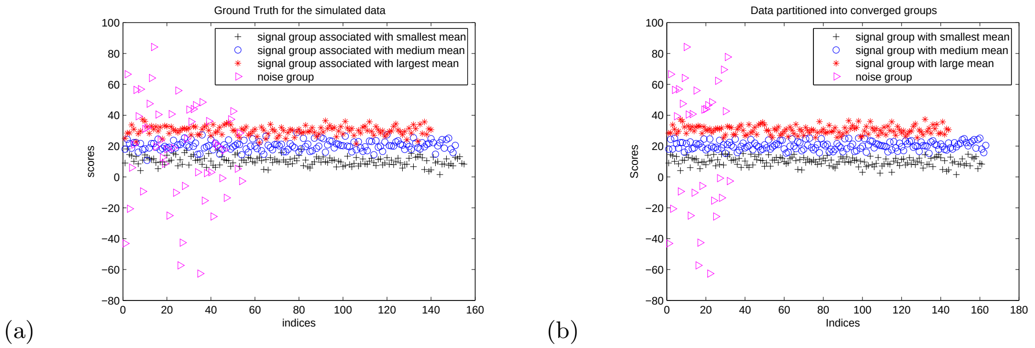
Observe a large stability of our algorithm with respect to the bandwidth $h$ illustrated by plots in Fig. 4. For each $h$ value, the algorithm was initialized with a randomly selected group labels consisting of 5 groups. Our algorithm always converged to the correct number of three signal model components. Small bandwidths did not adequately discriminate between noise and signal. Already moderately large bandwidths demonstrate adequate discrimination in that component means $\mu_j$ and weights $\pi_j$ $(j = 1, 2, 3)$ are accurately estimated.

## 8. LINE SEGMENTS AS COMPONENTS

We present specific details concerning our use of line segments as EM model components in the applications presented below. We stress that this section applies also to hyper planes in any dimensions, but the presentation is given in terms of line segments for purposes of simplification.

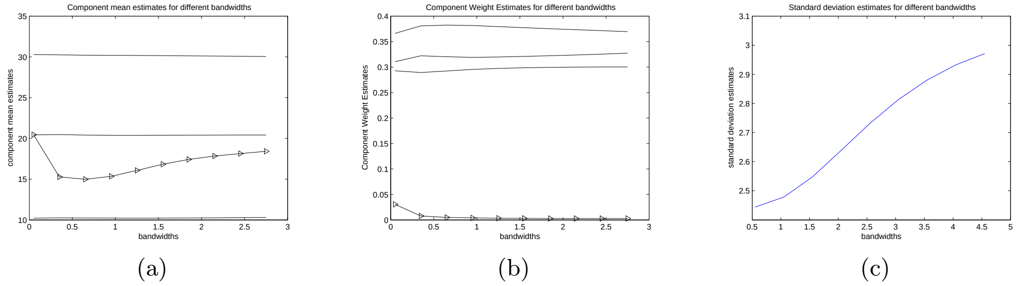
The proposed approach requires a minor extension of EM line fitting to work with line segments, which we will call Expectation Maximization Segment Fitting (EMSF). The difference between EMSF and EM line fitting is that our model components are line segments (rather than lines). The input, for our model, is a set of line segments and a set of data points. As with EM the proposed EMSF is composed of two steps:

(1) **E-step** The EM probabilities are computed based on the distances of points to line segments instead of the distances of points to lines.



(a) (b)

**Figure 3: (a) A plot of the simulated data with their ground-truth component labels. (b) A plot of data points with labels to which the EM algorithm converges.**



(a) (b) (c)

**Figure 4: Component means (a), weights (b), and sigma (c) as function of bandwidth $h$ used in $sdd$.**

(2) **M-step** Given the probabilities computed in the E-step, the new positions of the lines are computed by minimizing squared regression error weighted with these probabilities.

As in the case of EM line fitting, the output of the M-step is a new set of lines (not line segments). Since we need line segments as input to the E-step, we trim lines to line segment based on their support in the sample data. This is done by the split process described in Section 8.2.

Now we describe the specific details related to line segments for steps (1) and (2). In order to derive the solution of (23) for EM model components being line segments, we introduce so called EM weights. In the classical EM, the weight $w_{il}^{(t)} = p(z_i = l|x_i, \Theta^{(t)})$ represents the probability that point $x_i$ corresponds to segment $s_l$ for $l = 1, \ldots, k$. We use the notation $\theta_l$ for the parameters of the line segment $s_l$ itself. In our framework

$$w_{il}^{(t)} \propto sdd^{(t)}(x_i) \cdot p(z_i = l|x_i, \Theta^{(t)}), \qquad (27)$$

and the weights are normalized so that $\sum_{l=1}^{k} w_{il}^{(t)} = 1$ for each $i$. After the E-step associated with the $t$'th iteration is accomplished, we obtain a new matrix $(w_{il}^{(t)})$. Intuitively, each row $i = 1, ..., n$ of this matrix corresponds to weighted probabilities that the data point $x_i$ is associated with the corresponding line segments; each column $l = 1, ..., k$ can be viewed as weights representing the influence of each point on the computation of new line positions in the M-step. Below, we use the notation $x_i = (x_{ix}, x_{iy})$ with $(i = 1, ..., n)$ for the coordinates of the observed data points, and $(\bar{x}, \bar{y})$ for the coordinate averages. The line $L_l$, constructed below, is constructed to go through the point $(\bar{x}, \bar{y})$. To obtain the solution of (23), we perform an orthogonal regression weighted with the matrix $(w_{il})$. The solution is given as the normal vector to line $L_l$, which is the vector corresponding to the smallest eigenvalue of the matrix $M_l$ defined as

$$\begin{bmatrix} \sum_{i=1}^{n} w_{il}(x_{ix} - \bar{x})^2 & \sum_{i=1}^{n} w_{il}(x_{ix} - \bar{x})(x_{iy} - \bar{y}) \\ \sum_{i=1}^{n} w_{il}(x_{ix} - \bar{x})(x_{iy} - \bar{y}) & \sum_{i=1}^{n} w_{il}(x_{iy} - \bar{y})^2 \end{bmatrix} \qquad (28)$$

Finally the parameters $\theta_l^{(t+1)}$ are given as parameters of the line segment $s_l^{(t+1)}$ obtained by trimming the line $L_l$ to the data points.

We are now ready to introduce particular realization of split and merge for EM model components being line segments. The proposed split and merge EM segment fitting (SMEMSF) algorithm iterates the following four steps

(1) EMSF (2) Split (3) EMSF (4) Merge

Split step is presented in detail in Section 8.2 while Merge step is described in Section 8.1. Split evaluates the support in the data points of lines obtained by EMSF and removes the parts that are weakly supported. Since we have a finite set of data points, this has the effect of trimming the lines to line segments. Finally the merge step merges similar line segments. Thus, split and merge steps adjust the number of model components to better fit the data.

## 8.1 Merging

If inequality (17) holds, we merge two model components represented by parameters $l_1, l_2$ into one model componet given by parameter $l$. While components $l_1, l_2$ are present at step $t$ (they are line segments $s_{l_1}, s_{l_2}$), we did not yet specify how to compute the candidate component $l$. Now we describe a particular method to generate a candidate component $l$ in the particular case in which the model components are line segments. We stress that other methods are possible and that inequality (17) applies to them too.

A **support set** $S(s_j)$ for a given line segment $s_j$ (model component $l$) is defined as set of points whose probability of supporting segment $s_j$ is the largest, i.e.,

$$S(s_j) = \{x_i : w_{ij} = \max(w_{i1}, \ldots, w_{ik})\}.$$

This maps each data point to a unique segment using the *Maximum A Posteriori* principle. Given two line segments $s_{l_1}, s_{l_2}$, the merged segment $s_l$ is obtained by trimming the straight line obtained by regression on data points in $S(s_{l_1}) \cup S(s_{l_2})$. Trimming is performed by line split described in Section 8.2.

## 8.2 Line split (LS)

A classical case of EM local optimum problem is illustrated in Fig. 5(a), where the line segment is in a locally optimal position. Clearly, the problem here is that we have a model consisting of one line only, while two line segments are needed. Fig. 5(b) illustrates a split operation described in this section. It is based on removal of subsegments that do not have sufficient support in the data points. As the result we obtain two line segments. Finally, Fig. 5(c) shows the globally optimal approximation of the data points obtained by EM applied to the two segments.

The main idea is that higher point density along a segment indicates the presence of a linear structure in the data points around the segment. The amount of support that a line segment has is measured by the density of points around it. Each line or line segment is examined regarding whether it has sufficient support in the data. Only parts of segments that have this support are allowed to remain. This leads to a splitting of existing lines or segments.

We use the nonparametric density estimation $sdd$ to obtain the density along each segment. Although we defined $sdd$ only at the sample data points in (18), it is actually defined at every point $sdd(x) \propto \sum_{i=1}^{n} G(d(x, x_i), 0, h)$. Observe that $sdd_{|s_l}$ restricted to a segment $s_l$ is a one dimensional function. We obtain split point candidates (and consequently model segment candidates) as local minima of $sdd_{|s_l}$.

# 9. APPLICATIONS

An example application of our approach in robot mapping is outlined in Fig. 6. (a) shows an original data set of laser range scan points aligned with the algorithm presented in [4]. The original set is composed of 395 scans, each with 361 points. Thus, the original input map is composed of 142,595 points. We initialize our algorithm with only two segments, the two diagonals, as model components. (b) shows the output of the second iteration of our algorithm. The final polygonal map in (d), obtained after 12 iterations, is composed of 49 segments, i.e., of 98 points. Thus, the proposed approach yields the data compression ratio of 1455:1. The mean distance of scan points to the closest line segments is 5cm. We selected this map, since it contains surfaces of curved objects. The obtained polylines in (d) illustrate that the proposed approach is well suited to approximate linear as well as curved surfaces.

Now we apply the proposed approach to grouping edge pixels to polygonal curves representing object contours in digital images. Two example applications of this kind are outlined in Fig. 7. (a) shows an original input toy image. (b) shows the edges obtained by Canny edge detector with a substantial amount of incorrect edge pixels, and the initial model for our algorithm. It consists of only two line segments. (c) shows an intermediate step of our algorithm. The final polygonal approximation obtained after 27 iterations is shown in (d). (e) shows a simulated image obtained by sampling 3 ground truth segments (150 points) with a substantial amount of noise (2000 points). (f) shows the initial model segments for our algorithm. We present the results of our algorithm after 8 in (g) and 19 iterations in (h). We stress that we have only 150 signal points in comparison to 2000 background noise points.

# APPENDIX

# A. MONTE CARLO APPROXIMATIONS

We use the notation $q(x)$ for the ground truth density of the data. We assume that the data (including noise) is distributed as, $u(x)$ and let $\widehat{u}(x)$ be a standard density estimate of $u(x)$. We use the notation $sdd(x_i); i = 1, ..., n$ for the normalized estimates of the ratio $\frac{q(x)}{u(x)}$ at the given data points. As a result of Theorem 1 we obtain that mean squared error (MSE) for estimating the integral $\int f(x)q(x)dx$ using $H_{sdd} \sim \sum sdd(x_i)f(x_i)$ is significantly smaller than that using $H_m = \sum f(x_i)$ for any smooth function $f$.

THEOREM 1. *If $x_1, ..., x_n$ are data generated from the noisy density $u(x)$, then the approximate MSE for estimating the integral $\int f(x)q(x)dx$ using $H_{sdd} \sim \sum sdd(x_i)f(x_i)$ is, up to order $O(1/n)$,*

$$MSE(H_{sdd}) = O(1/n) \tag{29}$$

*The MSE for estimating the integral $\int f(x)q(x)dx$ using $H_m = \sum f(x_i)$ is up to order $O(1/n)$,*

$$MSE(H_m) = \left\{ \int f(x)\{u(x) - q(x)\}dx \right\}^2 \tag{30}$$

**Proof:** The variances of either Monte Carlo approximation are of order $O(1/n)$. Hence, the MSE's in either case correspond up to order $O(1/n)$ to the squares of the bias's. The asymptotic bias for the Monte Carlo approximation $H_{sdd}$ is, via the delta method, equivalent to:

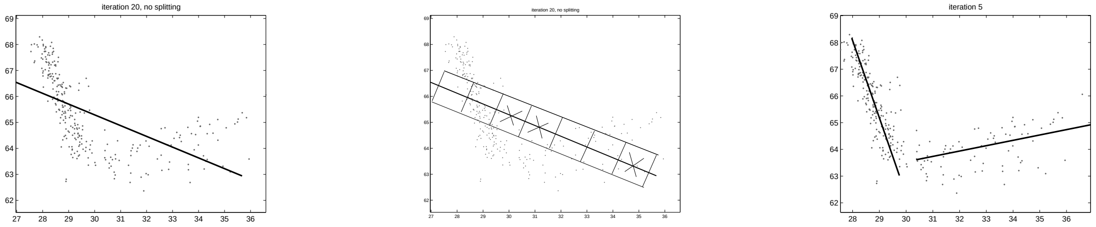
$$BIAS(H_{sdd}) \sim \left\{ \left[ (1/n) \sum_{i=1}^{n} sdd(x_i)f(x_i) - \int q(x)f(x)dx \right] + \left[ (1/n) \sum_{i=1}^{n} sdd(x_i) - 1 \right] \cdot \int q(x)f(x)dx \right\} \tag{31}$$

Due to the normalization, the rightmost term of equation (31) in square brackets is 0. Additionally, it follows from the Central Limit Theorem that

$$\left[ (1/n) \sum_{i=1}^{n} sdd(x_i)f(x_i) - \int q(x)f(x)dx \right] \leq O(1/\sqrt{n}) \tag{32}$$

It follows from equations (31) and (32) that

$$BIAS(H_{sdd})^2 \sim O(1/n) \tag{33}$$



(a) (b) (c)

**Figure 5: It is obvious to us that the approximation in (c) of the underlying data points is significantly better then the approximation in (a). (a) shows the best possible approximation of the data points obtained by EM. (b) The subsegments marked with crosses are removed, since their *sdd* values are too small, which results in splitting the segment to two parts. (c) shows the final approximation result obtained by EM after the split.**

Hence, by equation (33) and the remarks at the beginning of the proof,

$$MSE(H_{sdd}) = O(1/n) \tag{34}$$

The asymptotic bias for the monte carlo approximation $H_m$ is, equivalent to:

$$BIAS(H_m) = \left\{(1/n)\sum f(x_i) - \int f(x)q(x)dx\right\} \tag{35}$$

By the law of large numbers, up to order $O(1/\sqrt{n})$

$$(1/n)\sum f(x_i) \sim \int f(x)u(x)dx \tag{36}$$

Hence, by equations, (35) and (36), it follows that, up to order $O(1/\sqrt{n})$,

$$BIAS(H_m) \sim \int f(x)\{u(x) - q(x)\}dx \tag{37}$$

As a consequence, it follows from equation (37) that up to order $O(1/n)$,

$$MSE(H_m) \sim \left\{\int f(x)\{u(x) - q(x)\}dx\right\}^2 \tag{38}$$

The result follows.

## B. REFERENCES

[1] M. J. Beal and Z. Ghahramani. The variational bayesian em algorithm for incomplete data. In *BAYESIAN STATISTICS 7*. Oxford Univ. Press, 2003.

[2] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[3] P. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrica*, 82:711–732, 1995.

[4] G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *ICRA*, 2005.

[5] W. Hardle. *Smoothing Techniques with implementation in S*. Springer Verlag, 1991.

[6] M. J. R. Healy and M. Wesmacott. Missing values in experiments analyzed on automatic computers. *Appl. Statist.*, 5:203–206, 1956.

[7] R. Neal and G. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, 1998.

[8] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick. Weighted line fitting algorithms for mobile robot map building and efficient data representation. In *ICRA*, 2003.

[9] A. Pievatolo and P. Green. Boundary detection through dynamic polygons. *Journal of the Royal Statistical Society*, B, 60:609–626, 1998.

[10] P. L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Trans. PAMI*, 19(3):659–666, 1997.

[11] D. Sack and W. Burgard. A comparison of methods for line extraction from range data. In *Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, 2004.

[12] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley and Sons, 1992.

[13] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[14] J. S. Simonoff. *Smoothing Methods in Statistics*. Springer-Verlag, 1996.

[15] G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, 20(3):1236–1265, 1992.

[16] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton. Smem algorithm for mixture models. *Neural Computation*, 12(9):2109–2128, 2000.

[17] M. Veeck and W. Burgard. Learning polyline maps from range scan data acquired with mobile robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[18] M. Wertheimer. Untersuchungen zur lehre von der gestalt ii. *Psycologische Forschung*, 4:301–350, 1923.

[19] Z. Zhang, C. Chen, J. Sun, and K. L. Chan. Em algorithms for gaussian mixtures with split-and-merge operation. *Pattern Recogntion*, 36:1973–1983, 2003.

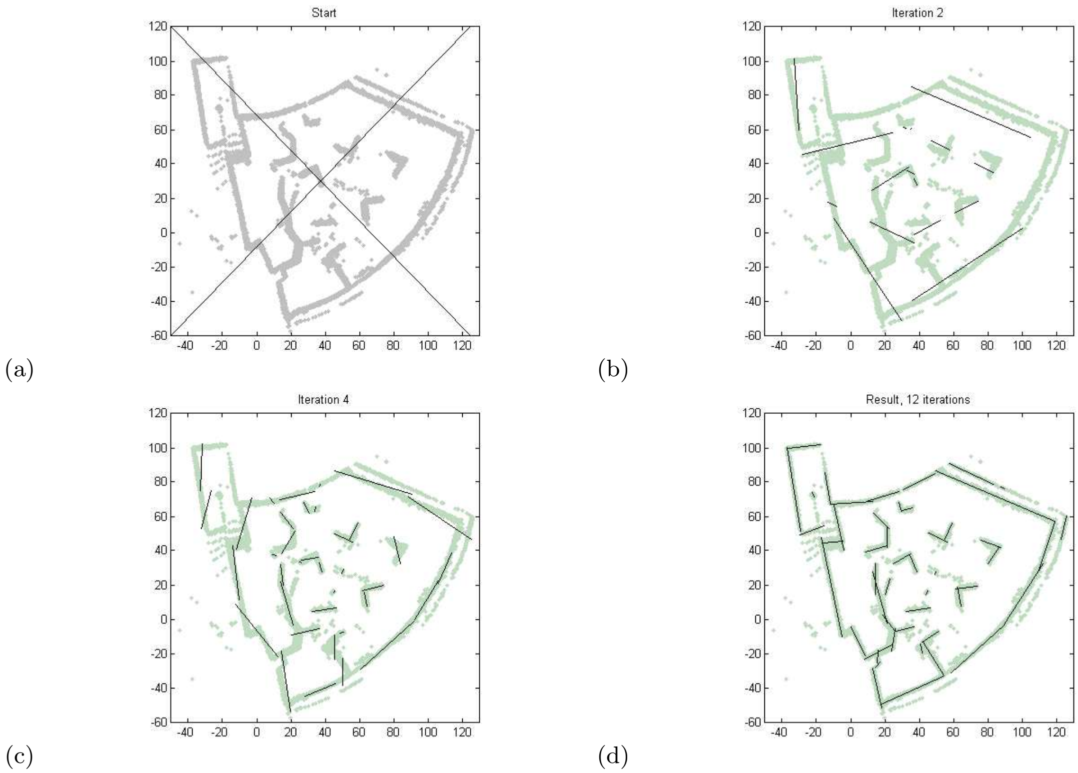


**Figure 6: (a) An original outdoor map is composed of 142,595 scan points obtained during the Rescue Robot Camp in Rome, 2004. We begin the approximation process with only two line segments that are the two diagonals. (b) shows the output of the second iteration of our algorithm. (d) The final polygonal map obtained after 12 iterations is composed of only 49 segments. The obtained compression rate is 1455:1, and the approximation accuracy is 5cm.**

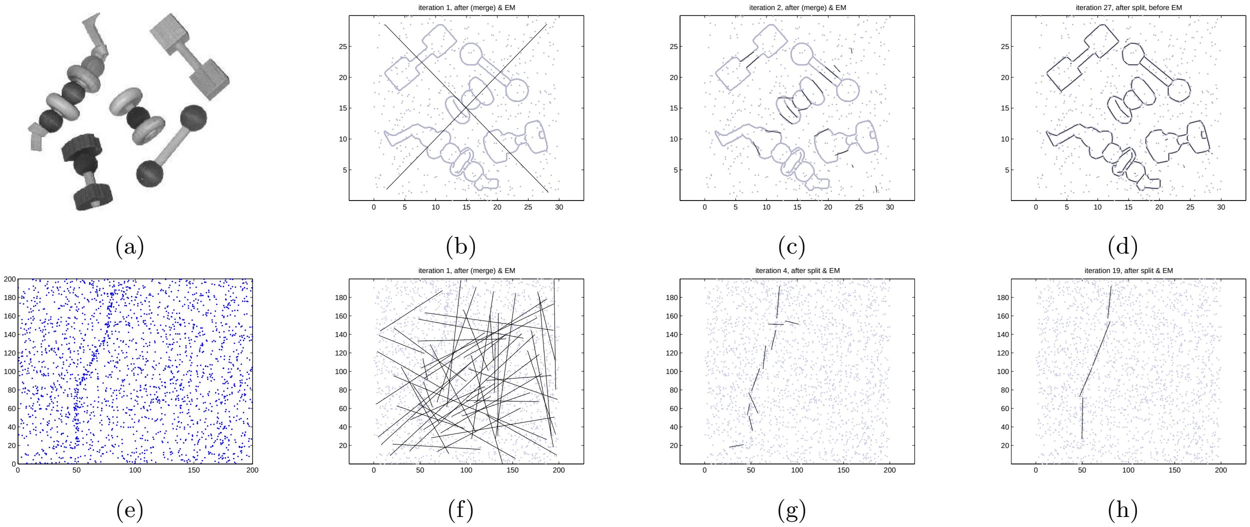


**Figure 7: (a) An original input image. (b) The edges obtained by Canny edge detector, and two initial line segments. (c) We see the polygonal approximation of the edge pixels obtained after (d) The final polygonal approximation obtained after 27 iterations is shown in (e)-(g) Illustrate our approach on simulated data generated by 3 ground truth segments with only 150 signal and 2000 noise points.**