Polygonal Approximation of Point Sets

Longin Jan Latecki[1], Rolf Lakaemper[1], and Marc Sobel[2]

[1] CIS Dept., Temple University, Philadelphia, PA 19122, USA
{latecki, lakamper}@temple.edu
[2] Statistics Dept., Temple University, Philadelphia, PA 19122, USA
marc.sobel@temple.edu

Abstract. Our domain of interest is polygonal (and polyhedral) approximation of point sets. Neither the order of data points nor the number of needed line segments (surface patches) are known. In particular, point sets can be obtained by laser range scanner mounted on a moving robot or given as edge pixels/voxels in digital images. Polygonal approximation of edge pixels can also be interpreted as grouping of edge pixels to parts of object contours. The presented approach is described in the statistical framework of Expectation Maximization (EM) and in cognitively motivated geometric framework. We use local support estimation motivated by human visual perception to evaluate support in data points of EM components after each EM step. Consequently, we are able to recognize a locally optimal solution that is not globally optimal, and modify the number of model components and their parameters. We will show experimentally that the proposed approach has much stronger global convergence properties than the EM approach. In particular, the proposed approach is able to converge to a globally optimal solution independent of the initial number of model components and their initial parameters.

1 Introduction

Expectation Maximization (EM) is a very popular and powerful method that allows simultaneous estimation of model parameters and assignment of data points to components of the model. However, 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. If this is not the case EM is only guaranteed to produce a locally optimal solution. This is illustrated in Fig. 1, where (a) shows data points and the initial configuration of two straight line segments. The number of model components (2 line segments) is correctly initialized, but their position is not sufficiently close to the global optimum. Fig. 1(b) shows the final, locally optimal, result obtained by the classical EM algorithm. Fig. 1(c) shows the globally optimal approximation obtained by the proposed method on the same input.

Due to the local optimum problem, a correct estimation of the number of components and the initial parameters of a statistical model is crucial in all EM applications, and therefore, belongs to one of the most challenging problems in statistical reasoning. The proposed approach provides a solution to the problem

U. Eckardt et al. (Eds.): IWCIA 2006, LNCS 4040, pp. 159–173, 2006.

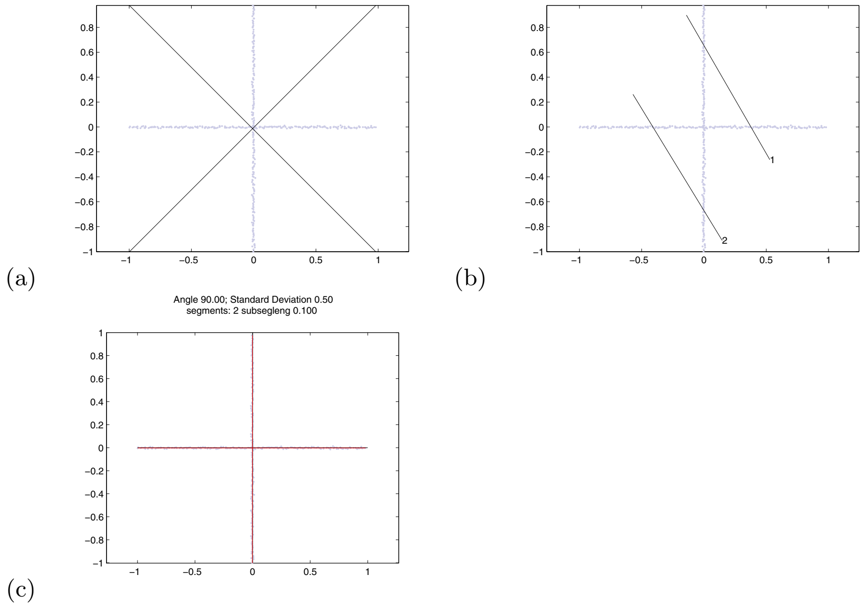


Fig. 1. (a) shows the data points and the initial position of model lines. (b) shows the optimal approximation of the data points obtained by EM. (c) shows the optimal approximation result obtained by the proposed method.

of local optimum in EM that is based on cognitively motivated local support evaluation of EM components. The example shown in Fig. 2 motivates the proposed approach. It is obvious to humans that the approximation in (c) of the underlying data points is significantly better then the approximation in (a). Observe the lack of local support in the data points of the middle part of the line in (a). This observation is the key argument for the proposed extension of EM. We will evaluate the support in data points of each EM component, and remove parts of components with insufficient support.

The existing approaches to determine the optimal number of EM components, of which AIC and BIC (Bayesian Information Criterion), which is equivalent to MDL (Minimum Description Length) [1], are most known, do not base their decision on the local support in data points of the model components. They assume only a fix cost per each model parameter. In particular, this means that a model component with high data support (i.e., positioned in a data region with high point density) costs the same as a component with low data support (i.e., positioned in a regions with low density of data points) although it is intuitively clear that a component with low data support is far less relevant than a component with large data support.

AIC, BIC, and MDL require separate EM runs until convergence for all possible number of model components, each run composed of several EM iterations, which may even be in the order of several thousands. For AIC, BIC, and MDL

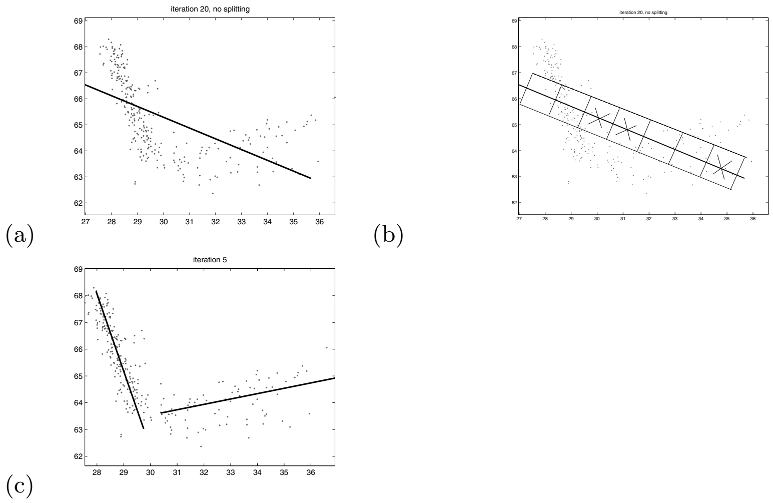


Fig. 2. It is obvious to us that the approximation in (c) of the underlying data points is significantly better then the approximation in (a). (a) shows the best possible approximation of the data points obtained by EM. (b) illustrates the line split (LS) based on subsegment removal. The removed subsegments are marked with crosses. (c) shows the final approximation result obtained by EM after the split.

to be successful, it is implicitly assumed that EM converges to global optimum in each run. However, as illustrated in Fig. 1(b) this is not always the case, since even with a correct number of components EM may get stuck in local minimum. Therefore, the correct number of two model components would not be selected by AIC, BIC, or MDL in our example. By locally evaluating the support in the data points of the two lines in Fig. 1(b), we can clearly determine that they form a bad approximation of the data points. By removing most of their parts, and retaining only small parts around the data points, we create a better input for the EM algorithm. This finally leads to a globally optimal approximation in Fig. 1(c).

The proposed approach provides a solution to the problem of local optimum in EM by adding two new steps that are well integrated with the standard E and M steps of EM. The two new steps are geometrically motivated and can be interpreted as split and merge steps in the context of line fitting. However, the proposed extension of EM is not restricted to any particular shape of model components. In the first new step, the split step, the model components obtained by a previous EM iteration are examined for support of the data points. The main idea (illustrated by the above example) is that higher point density around a model component (line segment in our application) indicates a presence of a linear structure in the data points around the segment. Parts of the segment that do not have sufficient support are removed. This may lead to segment removal but generally leads to a split of the segment into several subsegments. The second new step is merging similar model components. It prevents generating statistical models that overfit the data, i.e., fit noise in the data. This step requires a similarity measure of statistical model components. The merging step can be interpreted as perceptual grouping that dates back to the first results of Gestalt psychology in the beginning of 20th century [2].

We will show that integrating the split and merge operations in the EM framework leads to a globally optimal solution. In our experiments, we were able to obtain a globally optimal solution after just a few iterations (between 5 and 30). Two example applications of our approach are outlined in Fig. 3. (a) shows an original input toy image. (b) shows the edges obtained by Canny edge detector with a substantial amount of added noise, 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 an 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).

An overview of techniques for polygonal approximations of curves (when the order of data points is known), which have been studied at least since early seventies in computer vision, can be found in [3]. To some popular greed polygonal approximation methods in digital images belong [4] and [5, 6]. An overview of approaches to obtain polygonal maps from laser range data can be found in [7, 8].

We do not make any assumptions about the order of data points and extent of noise. The proposed method avoids the problem of a locally optimal solution and produces stable approximations not only to straight but also to curved lines. Moreover, the final number of fitted line segments depends on extent of noise. This means that the number of model components is adjusted to achieve the best possible approximation accuracy as the function of noise extent.

In order to show that the geometric and cognitively motivated split and merge steps can be incorporated into a statistical formalism, we introduce in Section 2 a new target function to be estimated in the EM framework, and reformulate the E and M steps in Section 3. Then we introduce statistical tests for the proposed split and merge steps in Section 4. Finally in Section 5, we describe the geometric parts of the split and merge steps.

2 Optimizing Kullback-Leibler Divergence

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:

$$\begin{aligned} D(q(x)||p_\Theta(x)) &= \int \log \frac{q(x)}{p_\Theta(x)} q(x) dx \\ &= \int \log q(x) q(x) dx - \int \log p_\Theta(x) q(x) dx \end{aligned} \tag{1}$$

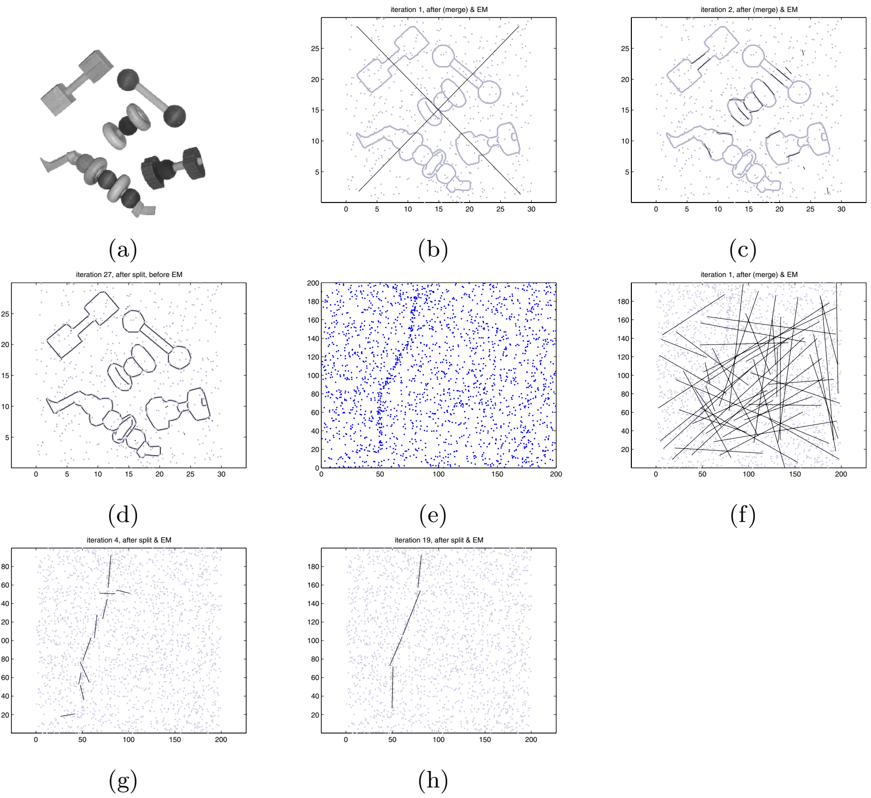


Fig. 3. (a) An original input image. (b) The edges obtained by Canny edge detector, and two initial line segments. (c) We see the polygonal approximation of the edge pixels obtained after (d) The final polygonal approximation obtained after 27 iterations is shown in (e)-(g) Illustrate our approach on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

Observe that KLD is able to determine the optimal number of model components of p_Θ. 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.

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 distribution $q(x)$.

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

However, as we derive below (equation (6)), if some proportion of the observations $x_1, ..., x_n$ are noisy, a more accurate estimator of Θ in (2) is given by:

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

where *sdd* is called the **smoothed data density** and is defined in (7) below.

Equation (4) is the basis of the proposed approach. We demonstrate theoretically and experimentally that maximization in (4) yields substantially better results than the classical EM maximization in (3). 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 Θ. The parametric family $\{p_\Theta(x)\}$, being a family of Gaussian mixture distributions, is convex. It follows that there is a unique member $p_{\widehat{\Theta}}(x)$ of the Gaussian mixture family with minimum KLD from q. This minimizing mixture must have the correct number of model components. However, 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:

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

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. Due to the noise, the following equation provides a substantially better estimate

$$\int f(x)q(x)dx \approx \sum_i f(x_i) sdd(x_i). \tag{6}$$

Finally equation (4) clearly follows from (6) and (2).

The smoothed data density *sdd* is defined as

$$sdd(x) \propto \sum_{i=1}^{n} K(\frac{d(x, x_i)}{h}) = \frac{1}{nh} \sum_{i=1}^{n} G(d(x, x_i), 0, h), \tag{7}$$

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 *sdd* is as follows:

- If a given data point x_j is sampled from the true distribution $q(x)$, then x_j lies in a dense region of the observed sample points and consequently $sdd(x_j)$ is large.
- If a given data point x_j is sampled from the noise distribution, then x_j is likely to lie in a sparse region of the sample space, and consequently $sdd(x_j)$ is small.

To estimate the bandwidth parameter h, we can draw from a large literature on nonparametric density estimation [9, 10]. As we show in the presented experimental results, an accurate bandwidth estimation in not crucial in our approach.

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 $i = 1, \dots, n$ 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

$$D(q(x,z)\|p_\Theta(x,z)) = D(q(x)q(z|x)\|p_\Theta(x)p_\Theta(z|x))$$
$$= \int_x \int_z \left\{ \log\left[\frac{q(x)}{p_\Theta(x)}\right] + \log\left[\frac{q(z|x)}{p_\Theta(z|x)}\right]\right\} q(x)q(z|x)dzdx$$
$$= \int_x \log\left[\frac{q(x)}{p_\Theta(x)}\right] q(x)dx + \int_x q(x) \int_z \log\left[\frac{q(z|x)}{p_\Theta(z|x)}\right] q(z|x)dz \tag{8}$$

We are now ready to introduce the expectation (E) and maximization (M) steps. Both steps aim at minimizing the same target function (8) 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 Θ, 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 (8) is minimized for $q(z|x) = p_\Theta(z|x)$ (in which case it is equal to zero), we obtain from (8) that

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

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

$$q(z_i = l|x_i) = p_\Theta(z_i = l|x_i) = p(z_i = l|x_i, \Theta) \tag{9}$$
$$= \frac{p(x_i|z_i = l, \Theta)p(z_i = l|\Theta)}{p(x_i|\Theta)} \tag{10}$$
$$= \frac{p(x_i|z_i = l, \Theta)p(z_i = l|\Theta)}{\sum_{j=1}^k p(x_i|z_i = j, \Theta)p(z_i = j|\Theta)} = \frac{p(x_i|z_i = l, \Theta)\pi_l}{\sum_{j=1}^k p(x_i|z_i = j, \Theta)\pi_j}, \tag{11}$$

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 Θ that maximizes (8). Substituting $q(z|x) = p_\Theta(z|x)$ in (8), we obtain

$$D(q(x,z)||p_\Theta(x,z)) = \int \log(\frac{q(x)}{p_\Theta(x)})q(x)dx = D(q(x)||p_\Theta(x)) \tag{12}$$

Thus, minimizing $D(q(x,z)||p_\Theta(x,z))$ in Θ is equivalent to minimizing $D(q(x)|| p_\Theta(x))$ in Θ. Using the estimate derived in equation (4), minimizing (12) in Θ is equivalent (in the MC setting discussed above) to maximizing the weighted marginal density

$$\begin{aligned} WM(\Theta) &= \sum sdd(x_i) \log p_\Theta(x_i) = \sum sdd(x_i) \log p(x_i|\Theta) \\ &= \sum_{i=1}^{n} sdd(x_i) \log[\sum_{l=1}^{k} p(x_i|z_i = l, \Theta) p(z_i = l|\Theta)] \\ &= \sum_{i=1}^{n} sdd(x_i) \log[\sum_{l=1}^{k} p(x_i|z_i = l, \Theta)\pi_l] \end{aligned} \tag{13}$$

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 (13) that an update of $WM(\Theta)$ is estimated by maximizing

$$WM(\Theta; \Theta^{(\mathrm{t})}) = \sum_{i=1}^{n} sdd(x_i) \log[\sum_{l=1}^{k} p(x_i|z_i = l, \Theta)\pi_l^{(t)}] \tag{14}$$

in Θ with $\Theta^{(t)}$ denoting the value of Θ 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 (14) does not influence the convergence.

4 Split and Merge

The proposed split and merge steps adjust the number of model components by performing compnent split and merge steps only if they increase the value of our target function (14). 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 5.

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 (14), we simply need to check whether replacing l with l_1, l_2 increases WM, where $j \in \{1, \ldots, k\}$:

$$\begin{aligned} WM(\Theta; \Theta^{(t)}) &= \sum_{i=1}^{n} sdd(x_i) \log[\sum_{j} p(x_i|z_i = j, \Theta)\pi_j^{(t)}] \\ < \sum_{i=1}^{n} sdd(x_i) \log[\sum_{j \neq l} p(x_i|z_i = l, \Theta)\pi_l^{(t)} \\ + p(x_i|z_i = l_1, \Theta)\pi_{l_1}^{(t)} + p(x_i|z_i = l_2, \Theta)\pi_{l_2}^{(t)}] \end{aligned} \tag{15}$$

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 [11], (see equation (15)). In accordance with the results of [11] this local computation guarantees that the target function increases after each iteration (if (15) 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 $j \in \{1, \ldots, k\}$

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

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 (16) holds and we replace l_1, l_2 with l, the convergence of our algorithm follows from the results of [11].

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 (16) 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 (15) 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. 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 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.
(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 5.2.

Now we describe the specific details related to line segments for steps (1) and (2). In order to derive the solution of (14) 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 x_i corresponds to segment s_l for $l = 1, \ldots, k$. We use the notation θ_l for the parameters of the line segment s_l itself. In our framework

$$w_{il}^{(t)} \propto sdd^{(t)}(x_i) \cdot p(z_i = l|x_i, \Theta^{(t)}), \tag{17}$$

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 $\mathcal{L}_l$, constructed below, is constructed to go through the point $(\bar{x}, \bar{y})$. To obtain the solution of (14), we perform an orthogonal regression weighted with the matrix (w_{il}). The solution is given as the normal vector to line $\mathcal{L}_l$, which is the vector corresponding to the smallest eigenvalue of the matrix M_l defined as

$$\begin{bmatrix} \sum_{i=1}^{n} w_{il}(x_{ix} - \bar{x})^2 & \sum_{i=1}^{n} w_{il}(x_{ix} - \bar{x})(x_{iy} - \bar{y}) \\ \sum_{i=1}^{n} w_{il}(x_{ix} - \bar{x})(x_{iy} - \bar{y}) & \sum_{i=1}^{n} w_{il}(x_{iy} - \bar{y})^2 \end{bmatrix} \tag{18}$$

Finally the parameters $\theta_l^{(t+1)}$ are given as parameters of the line segment $s_l^{(t+1)}$ obtained by trimming the line $\mathcal{L}_l$ 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 three steps

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

Split step is presented in detail in Section 5.2 while Merge step is described in Section 5.1. 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.

5.1 Merging

If inequality (16) 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 hot 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 (16) 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 5.2.

5.2 Line Split (LS)

A classical case of EM local optimum problem is illustrated in Fig. 2(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. 2(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. 2(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 a presence of a linear structure in the data points around the segment. Each line or line segment is examined on having sufficient support in data points measured as point density around it. Only parts of segments that have sufficient support of the data points remain. This leads to split of existing lines or segments allowing us to adjust the number of the line segments (i.e., the number of EM model components) to better fit the input data points.

In [12] the number of points in discretely enumerated rectangular strips (e.g., neighborhoods of all possible segments with endpoints in some finite set) is counted. Then signal strips are selected based on the ratio of the number of points to the area. Then strips are grouped to polylines based on proximity of their endpoints and their angles. The main difference of our approach to the approach in [12] is that we do not select the signal segments, but evaluate the existing structures selected by EM. This makes our computation more efficient, since we do not need to numerate all possible strips, and more accurate, since the line segments are optimally fitted to the data points in our approach.

Line Split (LS) is composed of the following steps:

(2.1) Subsegment support computation.

(2.2) Removal of subsegments with insufficient support that satisfy inequality (15).

The input to LS are segments $s_1, \dots, s_k$ obtained by clipping the lines $l_1, \dots, l_k$ created in EMSF to the image rectangle. We divide each segment $s_j \in \{s_1, \dots, s_k\}$ into subsegments of a predefined length $2r$, i.e., $s_j = I_1^j \cup \dots \cup I_l^j$, so that two consecutive subsegments overlap in their common endpoint, where l is the number of subsegments. (For simplicity we assume that the length of s_j is exactly multiple of $2r$.) For each subsegment I_k^j, we define its support as the number of data points in the square $S(I_k^j)$ whose two sides are parallel to subsegment I_k^j and whose center is contained in I_k^j, i.e.,

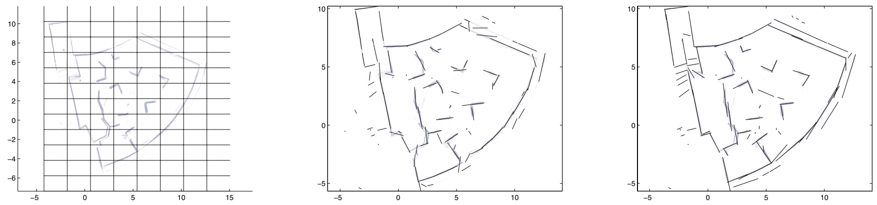
$$support(I_k^j) = \#(\{x_i\} \cap S(I_k^j)).$$

A few such squares are illustrated in Fig. 2(b).

In each iteration a support threshold C is computed from the statistics of $support(I_k^j)$ values over all subsegments of all line segments. Finally subsegments I_k^j with $support(I_k^j) \leq C$ are removed. The subsegments to be removed are marked with crosses in Fig. 2(b). New segments are created as connected components of remaining subsegments of segment s_j. If inequality (15) holds, then the original input segment s_j is removed, and the newly created segments are added to the list of original segments for the next iteration of EMSF. If all its subsegments are removed, then a given segment is removed.

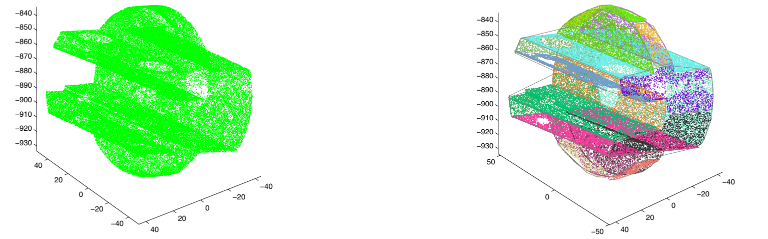
6 Applications

Two examples of approximations of point sets in digital images are illustrated in Fig. 3. An example application of our approach in robot mapping is outlined in Fig. 4. (a) shows an original data set of laser range scan points aligned with the algorithm presented in [13]. 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 192 segments, the grid segments, as model components. (b) shows the output with 96 segments after the first iteration of our algorithm. The final polygonal map in (c), obtained after 6 iterations, is composed of 86 segments, i.e., of 172 points. Thus, the proposed approach yields the data compression ratio of 829:1. The mean distance of scan points to the closest line segments is 3.5cm. We selected this map, since it contains surfaces of curved objects. The obtained polylines in (c) illustrate that the proposed approach is



(a) (b) (c)

Fig. 4. (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 192 line segments that form the grid. (b) shows the output after the first iteration of our algorithm with 96 segments. (c) The final polygonal map obtained after 6 iterations is composed of only 86 segments. The obtained compression rate is 829:1, and the approximation accuracy is 3.5cm.



(a) (b)

Fig. 5. (a) An input surface with 744,450 sample points. (b) Approximation with 27 planar patches.

well suited to approximate linear as well as curved surfaces. For more details on the application of the proposed method in robot mapping see [14].

Examples illustrating fitting planar patches to 3D range data are given in [15]. Here we show only one example in Fig. 5. Fig. 5(a) shows a 3D projection of the surface of some industrial part composed of 744,450 laser range scan points, obtained from `http://edge.cs.drexel.edu/Dmitriy/Scanned.tar.gz`. Fig. 5(b) shows our approximation with 27 planar patches. The mean distance of each point to the closest surface patch is 0.49 with the original object size of $100 \times 90 \times 100$.

7 Conclusions

The combination of Expectation Maximization Segment Fitting with alternating Segment Splitting and Merging was proven to be a powerful tool to gain a polyline representation of edge points in digital images, leading to a geometrically higher representation and an excellent data compression rate. The newly introduced, perceptual grouping based merging step balances the number of segments, created by partitioning and splitting, in a visually natural way and therefore allows for the number of starting segments for the EM step to be imprecise. The extended EM algorithm is proven to yield globally optimal results.

Acknowledgment

This work was supported in part by the National Science Foundation under grant NSF IIS-0534929. We also would like to acknowledge the support from the National Institute of Standards and Technology, award Nr. NIST 70NANB5H11119. We thank Giorgio Grisetti (Univ. La Sapienza, Rome) for providing us the scan data.

References

1. Beal, M.J., Ghahramani, Z.: The variational bayesian em algorithm for incomplete data. In: BAYESIAN STATISTICS 7. Oxford Univ. Press (2003)
2. Wertheimer, M.: Untersuchungen zur lehre von der gestalt ii. Psycologische Forschung **4** (1923) 301–350
3. Rosin, P.L.: Techniques for assessing polygonal approximations of curves. IEEE Trans. PAMI **19(3)** (1997) 659–666
4. Ramer, U.: An iterative procedure for the polygonal approximation of plane curves. Computer Graphics and Image Processing **1** (1972) 244–256
5. Latecki, L.J., Lakämper, R.: Shape similarity measure based on correspondence of visual parts. IEEE Trans. Pattern Analysis and Machine Intelligence **22(10)** (2000) 1185–1190
6. Latecki, L.J., Lakämper, R., Eckhardt, U.: Shape descriptors for non-rigid shapes with a single closed contour. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition. (South Carolina, June 2000) 424–429

7. Sack, D., Burgard, W.: A comparison of methods for line extraction from range data. In: Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV). (2004)
8. Veeck, M., Burgard, W.: 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)
9. Scott, D.W.: Multivariate Density Estimation: Theory, Practice, and Visualization. Wiley and Sons (1992)
10. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall (1986)
11. Neal, R., Hinton, G.: A view of the em algorithm that justifies incremental, sparse, and other variants. In Jordan, M.I., ed.: Learning in Graphical Models, Kluwer (1998)
12. Arias-Castro, E., Donoho, D.L., Huo, X.: Adaptive multiscale detection of filamentary structures embedded in a background of uniform random points. Annals of Statistics (to appear)
13. Grisetti, G., Stachniss, C., Burgard, W.: Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In: ICRA. (2005)
14. Latecki, L.J., Lakaemper, R.: Polygonal approximation of laser range data based on perceptual grouping and em. In: IEEE Int. Conf. on Robotics and Automation (ICRA). (2006)
15. Lakaemper, R., Latecki, L.J.: Decomposition of 3d laser range data using planar patches. In: IEEE Int. Conf. on Robotics and Automation (ICRA). (2006)