IEEE Int. Conf. on Robotics and Automation, Orlando, Florida, May 2006

Polygonal Approximation of Laser Range Data Based on Perceptual Grouping and EM

Longin Jan Latecki
Computer and Information Sciences Dept.
Temple University
Philadelphia, PA 19122, USA
Email: latecki@temple.com

Rolf Lakaemper
Computer and Information Sciences Dept.
Temple University
Philadelphia, PA 19122, USA
Email: lakamper@temple.edu

***Abstract*— Our goal is polygonal approximation of laser range data points obtained by a mobile robot. The proposed approach provides a precise estimation of the number of model components (line segments) and their initial parameters independent of their initial values. We use principles of perceptual grouping to evaluate the approximation quality obtained in each Expectation Maximization (EM) step. By evaluating EM approximation quality we are able to recognize a locally optimal solution, and modify the number of model components and their parameters. Consequently, EM can converge only to a globally optimal solution independent of the initial number of model components and their initial parameters.**

I. INTRODUCTION

Our domain of interest is polygonal approximation of data points. The existing solutions make assumptions about the number of fitted line segments, extent of noise, and/or the order of data points. 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 [1]. An overview of approaches to obtain polygonal maps from laser range data can be found in [2], [3]. We do not make any assumptions about the order of data points and extent of noise. On contrary to the existing approaches, 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.

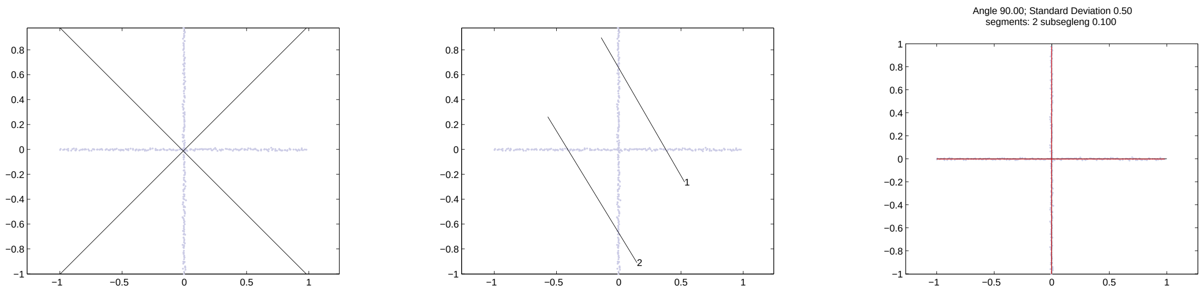
The proposed approach adds two new steps that are well integrated with the standard E and M steps of EM. In the first new step, the model components obtained by a previous EM iteration are examined for support of the data points. Parts of the components that do not have sufficient support are removed, which leads to component splitting and removal. 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. This results in a new set of model components for the next EM iteration. The second new step is merging similar model components. It prevents generating statistical models that over fit the data, i.e., fit noise in the data. This step requires a similarity measure of statistical model components. Since the similarity measure of model components requires domain specific knowledge, we present the proposed methodology in a context of a particular domain. However, the proposed framework provides a domain independent extension of EM.

We use line segment similarity based on principles of perceptual grouping that date back to the German school of Gestalt psychology in the beginning of 20th century [4] in merging similar line segments. Perceptual grouping is rooted in human perception and is an active research topic in computer vision. In our approach perceptual grouping principles, based on [5], are used to merge pairs of line segments, visually belonging together, to a single longer line segment.

Assuming that the initial number of model components (line segments) is well estimated, the main difficulty of fitting line segments to point data is that the correspondence of data points to line segments is unknown. The Expectation Maximization (EM) algorithm [6] provides an iterative solution to the correspondence problem. Our departing point is an EM line fitting algorithm. In fact EM applied to line fitting is known as the Healy-Westmacott procedure in statistics, and predates EM by many years [7]. However, since our goal is to fit polylines (polygonal curves) to point data, we trim lines to line segments (Section II-A). The main contribution of this paper is the introduction of nonreversible split and merge steps. While we base merging on principles of perceptual grouping, we relate split directly to support in the data points. Both split and merge steps in the proposed approach require only local evaluation. Thus, we use local optimization to provide a better model for EM. Due to the integration of these operations in the EM framework, we are able to obtain a globally optimal solution after just a few iterations (between 5 and 20) in all our experiments.

The proposed approach also provides a solution to the well-known problem of local optimum in EM. A classical case of EM local optimum problem is illustrated in Fig. 1. Fig. 1(a) shows data points and the initial configuration of two straight lines. Fig. 1(b) shows the final result obtained by the classical EM algorithm. Finally, Fig. 1(c) shows a globally optimal approximation obtained by the proposed method on the same input.

An example application of our approach is outlined in Fig.



(a) (b) (c)

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.

2. (a) shows an original data set of laser range scan points aligned with the algorithm presented in [8]. 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. (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. Some of the data points are not approximated by line segments. These points are very likely to represent noise artifacts due to their low density. (Due to the limited size of the figures, it is hard to perceive the point densities.) Therefore, it is necessary to approximate only point structures with sufficient point density. The required point density is determined automatically in our approach (Section II-C). Further results obtained by our approach for several other indoor and outdoor maps can be viewed on our home page.

The proposed approach is presented in Section II, and it is related to the existing approaches in Section IV.

II. Split and Merge in the EM Framework

We begin with a short overview of EM applied to line fitting (a detailed presentation can be found in [9]). As mentioned before, EM applied to line fitting is known as the Healy-Westmacott procedure in statistics, and predates EM by many years [7]. The following two steps are alternated until the algorithm converges, and the algorithm is guaranteed to converge to some local optimum. The input is a set of data points on the plane and an initial set of straight lines.

- **E-Step** (Expectation Step): Given a current set of lines, for each point the probabilities of its correspondences to all lines are estimated based on its distances to lines.
- **M-step** (Maximization Step): Given the probabilities computed in the E-step, the new positions of the lines are computed using a regression weighted with these probabilities.

A. Expectation Maximization Segment Fitting (EMSF)

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 only difference of EMSF in comparison to EM line fitting is that it starts and finishes with line segments. The input is a set of line segments and a set of data points. EMSF is composed of the following three steps:

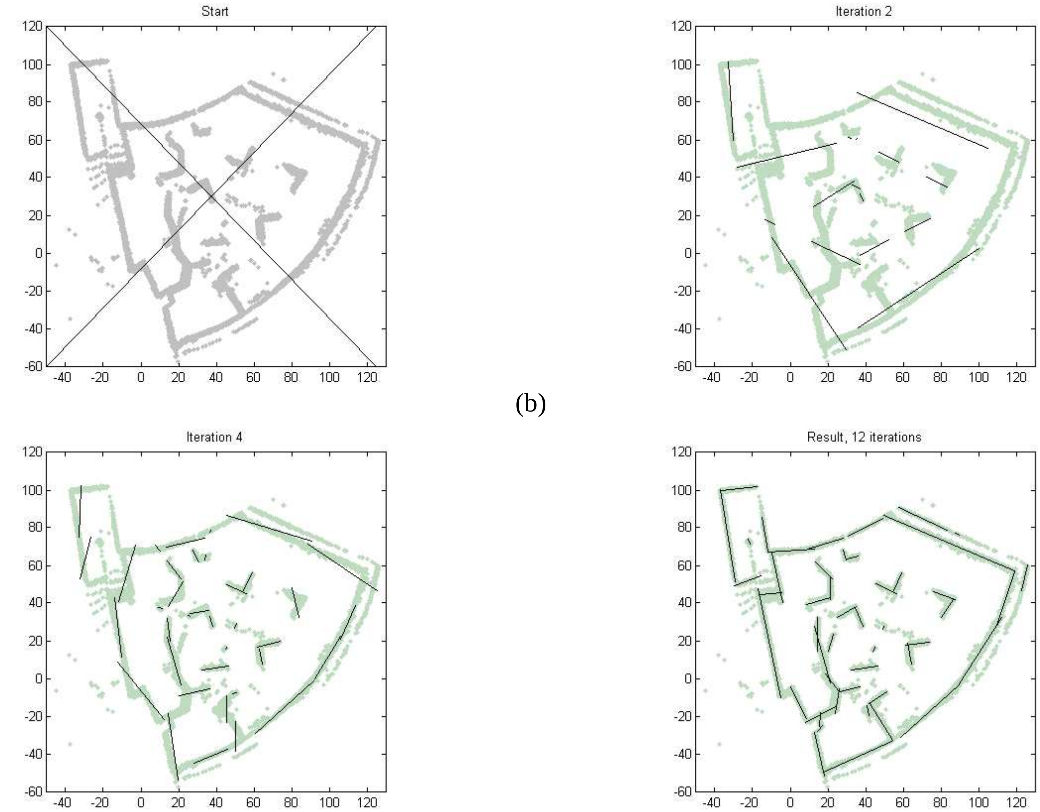
(1) E-step with line segments (the EM probabilities are computed based on the point distances to line segments instead of lines, see below)
(2) M-step with the probabilities computed in the E-step and lines extending the line segments
(3) Trimming lines to line segments. The new lines computed in the M-step are trimmed to line segments.

Now we describe these steps in detail. First we need to recall the computation of EM probabilities in the E-step. Let $x_1, \ldots, x_m$ be a set of data points on the plane, and let $s_1, \ldots, s_n$ be a set of line segments. Usually m is significantly larger than n. For each point x_i, the probability w_{ij} that x_i corresponds to segment s_j is computed for $j = 1, \ldots, n$. Formally, $w_{ij} = p(z_i = j)$, where z_i is the hidden variable associated with point x_i whose values range over the segment indices. Analog to EM line fitting, this probability is computed based on the distance $d(x_i, s_i)$ from point x_i to segment s_j:

$$w_{ij} \propto e^{-d(x_i,s_i)^2/2},$$

and normalized so that $\sum_{j=1}^{n} w_{ij} = 1$ for each i. The only difference to the standard EM line fitting is that we have substituted the distance point to line with the distance point to segment. After every E-step we obtain a new matrix (w_{ij}), where each row i represents the probabilities for point x_i, and each column j can be viewed as weights representing the influence of each point on the computation of a new line position in the M-step.

The output of the M-step, which performs an orthogonal regression weighted with (w_{ij}), is a set of lines $l_1, \ldots, l_n$ corresponding to the input segments $s_1, \ldots, s_n$. The normal



(a) (b) (c) (d)

Fig. 2. (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.

vector to line l_j is the vector corresponding to the smallest eigenvalue of the matrix M_j defined as

$$\begin{bmatrix} \sum_{i=1}^{m} w_{ij}(x_{ix}-\bar{x})^2 & \sum_{i=1}^{m} w_{ij}(x_{ix}-\bar{x})(x_{iy}-\bar{y}) \\ \sum_{i=1}^{m} w_{ij}(x_{ix}-\bar{x})(x_{iy}-\bar{y}) & \sum_{i=1}^{m} w_{ij}(x_{iy}-\bar{y})^2 \end{bmatrix} \quad (1)$$

where $x_i = (x_{ix}, x_{iy})$ are the coordinates of the data points, and $(\bar{x}, \bar{y})$ is their average weighted with w_{ij} for $i = 1 \dots m$, and line l_j goes through the point $(\bar{x}, \bar{y})$.

The step (3) is composed of two substeps:

(3.1) Assignment of supporting data points to lines.
(3.2) Trimming the lines to segments.

In order to trim the lines to line segments (3.1), we first need to assign supporting data points to lines. This assignment is based on the probabilities computed in the E-step for the input segments. A **support set** $Sup(s_j)$ for a given line segment s_j is defined as set of points whose probability of supporting segment s_j is the largest, i.e.,

$$Sup(s_j) = \{x_i : w_{ij} = \max(w_{i1}, \dots, w_{in})\}.$$

Thus, we map each data point to a segment using the Maximum A Posteriori principle. The **support set** for a given output line l_j is the same as for the corresponding input segment, i.e., $Sup(l_j) = Sup(s_j)$ for $j = 1, \dots, n$.

Trimming the lines to segments (3.2) is a simple step now. The straight lines $l_1, \dots, l_n$ computed in the M-step are trimmed using the support sets $Sup(l_j)$. For each j, we project the set of points in $Sup(l_j)$ onto the line l_j. Then a new segment $s_j^{(new)}$ is defined as the smallest segment contained in line l_j that contains all points in $P_{l_j}(Sup(l_j))$, where P_{l_j} is the orthogonal projection to the line l_j. We obtain a set of new segments $s_1^{(new)}, \dots, s_n^{(new)}$ such that $s_j^{(new)} \subset l_j$.

B. Split and merge EM segment fitting

Now we introduce the outline of the proposed algorithm. The proposed **split and merge EM segment fitting (SMEMSF)** algorithm iterates the following steps (described in detail below):

1) EMSF (Expectation Maximization Segment Fitting)
2) LSS (Line Segment Split): data support evaluation of segment obtained by EMSF (Section II-C)
3) EMSF
4) Line segment merge (Sections II-D and III)

Thus, we alternate line segment splitting and merging between the steps of the segment fitting EM algorithm.

The main goal of LSS is to evaluate the quality of the EMSF output, i.e., how well the EM weights positioned the new segments. The subsegments of the new segments that do not have sufficient support in the data points will be removed leading to splits into two or more smaller segments. Hence LSS creates a sufficient number of segments in order to optimally fit the input data points. This way we overcome the problem of locally optimal solutions, since such a solution will not have a good global support in the data points. The initial position of the line segments also does not matter, since the following EMSF will reposition the split segments to better fit the data. This is illustrated in Fig. 3.

If a pair of line segments is supported by a nearly the same set of collinear data points, EMSF (in step 3) maps them to two similar segments. In the merging step (4), pairs of similar segments are merged to single segments. Due to merging, the number of segments cannot grow to infinity. Therefore, in the EM framework extended by the merging step, the number and position of the new segments introduced by the split is not critical. Iterating split and merge in the EM framework is a powerful tool to adjust the number and position of line segments to better fit the data points.

A few iterations of the proposed algorithm are illustrated in Fig. 2. The proposed algorithm converges, since EM converges and the LSS procedure (Section II-C) stops splitting if a certain goodness of fit criterion is met. Usually just a few iterations of steps (1)-(4) are required. Our stop condition is the stability of distances of data points to the closest line segments. By partially ordering the line segments into polylines, we obtain a global polygonal approximation (2(d)).

C. Line segment split (LSS)

A classical case of EM local optimum problem is illustrated in Fig. 3(a). Clearly, the problem here is that the model consists of one line segment, while two line segments are needed. Fig. 3(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. 3(c) shows a globally optimal approximation of the data points obtained by EM applied to the two segments.

The main observation is that higher point density along a segment indicates a presence of a linear structure in the data points around the segment. The approach in [10] uses this idea to find polygonal structures in point data sets. The computation is based on counting the points in all possible rectangular strips (i.e., neighborhoods of all possible segments), and then selecting the strips representing polygonal structures based on statistics of the count. This approach works only if the noise points have uniform distribution, which is an unrealistic assumption for our application.

The main difference of our approach to the approach in [10] is that we do not select the structures based on the data point density, 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.

Each segment obtained by an EMSF 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 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.

Line Segment Split (LSS) is composed of the steps:

(2.1) Subsegment support computation.
(2.2) Removal of subsegments with insufficient support.

The input to LSS are segments $s_1, \ldots, s_n$ created in EMSF. We divide each segment $s_j \in \{s_1, \ldots, s_n\}$ into subsegments of a predefined length $2r$, i.e., $s_j = I_1^j \cup \cdots \cup I_l^j$, so that two consecutive subsegments overlap in a single point, where l is the number of subsegments. (For simplicity we assume that the length of s_j is exactly multiple of $2r$.) For each iteration the subsegment length $2r$ is defined based on the distribution of the distances of data points to segments they support (see below). 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.,

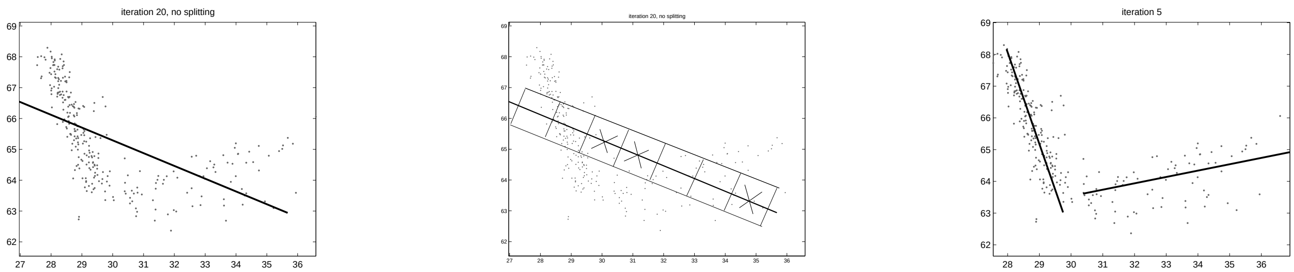
$$support(I_k^j) = \#(\{x_i\} \cap S(I_k^j)).$$

A few such squares are illustrated in Fig. 3(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 (see below). 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. 3(b). New segments are created as connected components of remaining subsegments of segment s_j. 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 the segment s_j is removed. The parameters $2r$ and C are computed dynamically each time the subsegment removal step is called.

D. Merging

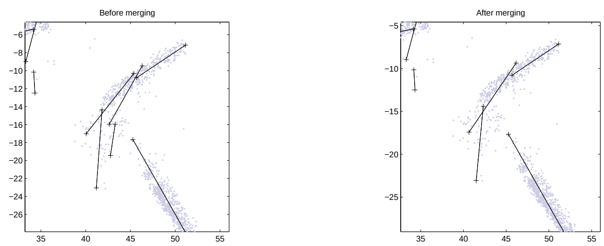
Before we introduce merging, we elaborate on its role in the EM framework. The main idea is that if a given segment is split correctly to two subsegments, then EMSF will reposition the two segments to better fit the data points. Consequently, the two segments will move and turn away from each other, and therefore, will not be similar segments, e.g., Fig. 3(c). If a segment is unnecessarily split to two subsegments, the two segments remain very similar after an EMSF iteration, where similar means that they will be nearly collinear and close to each other. This elaboration suggests that merging should combine two perceptually similar segments to a single segment, and leave unchanged pairs of perceptually dissimilar segments. Without merging the model may end up with too many components, which could mean fitting the noise in the data. The segment similarity measure used in merging is responsible for the accuracy of the statistical model. The



(a) (b) (c)

Fig. 3. (a) shows the best possible approximation of the data points obtained by EM. (b) illustrates the line segment split (LSS) based on subsegment removal. The subsegments to be removed are marked with crosses. (c) shows the final approximation result obtained by EM after the split.

proposed merging operation is introduced in Section III, where also the perceptually motivated segment similarity measure is introduced.



(a) (b)

Fig. 4. (a) shows line segments introduced by split and repositioned by EM. (b) shows line segments obtained by merging similar segments in (a).

III. Line segment merging based on perceptual grouping

The goal of this section is to define a similarity measure of EM model components so that we can merge similar components. Merging in connection with split allows us to automatically determine the number of model components. Since in our case the model components are line segments, we will use human visual perception to define the similarity of line segments.

Given a pair of line segments, L_1 and L_2, the objective of the merging process is to compute a merged segment $ms(L_1, L_2)$. The main idea is to only merge line segments that are sufficiently similar. Therefore, we will define a cost function $C(L_1, L_2)$ that measures the similarity of L_1 and L_2. The geometric intuition of the presented merging process and, in particular, the definition of merging cost $C(L_1, L_2)$ is based on cognitively motivated principles of perceptual grouping that date back to [4]. Lowe [5] states that proximity of endpoints, parallelism, and collinearity are the main geometric relations that influence the perceptual grouping of line segments. However, Lowe did not consider merging but rule based grouping, i.e., he did not reduce the number of segments but grouped them to some predefined spatial structures. Therefore, we developed a new cost function that integrates these geometric relations with a goal of segment merging.

We begin with a construction of the merged segment $ms(L_1, L_2)$. Let $L_1 = AB$ and $L_2 = CD$ be oriented in the same direction so that $||AB - CD|| \leq ||AB + CD||$, i.e., the scalar product $AB \cdot CD \geq 0$, then the weighted average direction ad of L_1 and L_2 is obtained as the direction of vector $AB + CD$. The line lad with direction ad is positioned between L_1 and L_2 so that the following equation $d_1 \cdot l_1 = d_2 \cdot l_2$ is satisfied, where l_i is the length of segment L_i and d_i is the distance of the midpoint of L_i to line ld for $i = 1, 2$. This has the effect of positioning line lad closer to the larger of two segments. Finally, the segment $ms(L_1, L_2)$ obtained by merging L_1 and L_2, called the merged segment, is defined as the shortest segment contained in ld that contains the projections of L_1 and L_2 on line ld.

The main intuition for the segment merging cost is that we want to measure how visually significant is to replace L_1 and L_2 with $ms(L_1, L_2)$. The cost of merging segments L_1 and L_2 is defined by:

$$C(L_1, L_2) = w_1 \cdot parC(L_1, L_2) + \\ w_2 \cdot colC(L_1, L_2) + w_3 \cdot proxC(L_1, L_2),$$

where $parC$, $colC$, $proxC$ are measures of parallelism, collinearity, and proximity between line segments L_1 and L_2 correspondingly, defined in [11] using elementary geometric relations. The weights are used to obtain an adequate balance between of the geometric relations of parallelism, collinearity and proximity. In our approach they were determined with cognitive experiments and set to $w_1 = 2, w_2 = \frac{1}{4}, w_3 = \frac{1}{2}$.

IV. Relation to the Existing Approaches

A. Statistics

We provide a domain specific solution to one of the most challenging problems in statistical reasoning which is the estimation of the number of components of a statistical model. A general discussion of this problem in statistical framework can be found in Richardson and Green [12]. The importance of statistical models is computer vision is discussed in Mumford [13].

In [14] the usage of Bayesian Information Criterion (BIC) to estimate the number of model components is discussed. BIC is equivalent to Minimum Description Length (MDL).

For a fix number of data points, which is the case in our application at each given time t, BIC calculates a trade-off between the model complexity and the likelihood of the data points. Typically a model that has the highest BIC value is selected by repeatedly executing EM for all possible numbers of model components. The problem with this approach is that it works only if EM converges to the global optimum for the evaluated numbers of model components. If for some number of components EM gets stuck in a local optimum, the BIC estimate may be wrong. For example, the correct number of model components cannot be determined for the situation in Fig. 1(b). Due to the fact that EM got stuck in a local optimum, the likelihood of the model with two components will be very low. To our best knowledge, this problem is not addressed by any existing approach 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 the BIC this standard deviation realizes a weighing factor between the likelihood of the data points and the model complexity.

In addition, as determined experimentally on ground-truth data in [14] BIC tends to over penalize the complexity, which leads to a too small number of model components.

It is also important to mention that the proposed method yields a quicker convergence of the EM since it adopts the number of model components and model parameters to the given environment after every EM operation while BIC requires EM convergence for each number of model components.

Green [15] 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 fully Bayesian mixture analysis that is making use of reversible jump Markov chain Monte Carlo methods, which are capable of jumping between the parameter subspaces corresponding to different numbers of components in the mixture. The jumps are realized by split and merge moves that are reversible. In the proposed approach, the split and merge steps are not reversible. While spit is based on the goodness of fit measure of model components to the data points, the merge is based on similarity of model components that is not directly related to the data points. The merge allows us to estimate the number of model components based on perceptual grouping of segments. Since Green's merge move is evaluated based on the data points, it cannot be used to estimate the number of model components. Hence Green's approach requires an additional penalty for the number of models. Consequently, the number of models heavily depends on this penalty, which is not directly related to the model quality assessment, which is the case in our approach.

Green's approach is used to fit polygons to contours in digital images in [16], where split corresponds to inserting a new vertex to the polygon and merge corresponds to removing a vertex. The split and merge moves are based on a random selection that is then evaluated and executed with a probability corresponding to the outcome of the evaluation, e.g., a vertex to be removed is randomly selected, and the probability of its removal depends on the goodness of fit of the polygon with this vertex removed. Not only has the algorithm required a huge number of iterations (Green reported 20000 in some experiments), but also the random selection seems to be unintuitive from point of view of human visual perception. Humans are able to identify good and bad fitting parts of a given polygon by visual inspection. Thus, it seems to be more intuitive to base the moves on local visual inspection rather than on random selection.

In general, it seems to be reasonable to assume that a decision whether to merge two mixture components to one can be based on a similarity measure, i.e., we merge two components to one if they are sufficiently similar. Clearly, similarity of two mixture components can be expressed in statistical framework. However, it is unclear whether the definition of similarity must be restricted to a statistical framework only. In our domain specific solution, we base the definition of similarity on principles of perceptual grouping. It is an open question whether it is possible to express this definition in a pure statistical framework.

B. Robot Mapping

Since polygonal maps are very attractive means to represent range scan data, particularly due to their very compact size and simplicity, several approaches have been proposed to obtain such maps, the most recent ones being [3] and [17]. We do not provide an overview of approaches to obtain polygonal maps, since an excellent overview can be found in [3].

Sack and Burgard [2] use EM to obtain polygonal approximation of laser range data. As stated in [2] a crucial problem when applying EM is the number of model components. The algorithm in [2] applies the approach presented in Bennewitz et al. [18], which uses BIC to estimate the number of model components. In the proposed approach we use similarity of model components to adjust the number of model components in that similar components are merged. The other significant difference is that we remove only parts of lines that do not have sufficient support while the whole lines that have low utility measured as function of EM weights are removed in [2]. Additionally the experimental results in [2] demonstrated that shorter linear structures were not approximated well using the EM algorithm in [2]. Sack and Burgard [2] raised the following question related the their EM algorithm: *One important question is, whether the EM based approach can be extended to operate on line segments instead of lines.* Our approach provides a positive answer to this question. Our algorithm approximates extremely well short as well long linear structures as is demonstrated by our experimental results.

One of the main differences in comparison to the existing approaches, in particular to [17] and [3], is the fact that information about the robot pose is not required in our approach. In contrary, a robust estimation of robot pose is an essential information required for merging segments in [17]. The merging of polyline segments in [3] is based on the distance between their endpoints, using a threshold of 15cm, which is often not capable to master the challenges. Utilizing perceptual grouping overcomes these problems in our approach.

The other significant difference in comparison to the existing approaches is the fact that the proposed approach can be applied incrementally, which is mainly due to the excellent performance of the proposed merging method. This allows us to keep the size of data needed to incrementally construct a global map very small. A global map $G(t)$ at time t is represented with polylines, and we only need $G(t)$ and the data points obtained from the last few scans to obtain a new global map $G(t+1)$. Consequently, only a small amount of computational resources is needed for a real time implementation. Thus, our approach provides a solution to the main issues raised in [3]: "*The current system requires a huge amount of computational resources and can only be applied offline after the data has been recorded. In the future we therefore will investigate how to speed-up the learning process in order to obtain an on-line variant and to even further increase the accuracy.*

V. Relation to SLAM Approaches

The SLAM problem, the Simultaneous Localization and Mapping problem [19] is of high importance to mobile robotics. A solution to it is judged a prerequisite for true autonomy. Consequently, it has received considerable attention (see Thrun [20] for an overview). A relation between the SLAM problem and map building is given by the common task of place recognition, also termed the *correspondence problem*, which is part of localization. Odometry information and scan matching techniques provide good means for incremental updates to estimate robot pose in the EM framework.

In the classical EM based approach to SLAM, there is a fixed number of model components, which are robot poses. In our framework, the model components, which are line segments, are dynamically estimated. No information about the robot poses, and consequently, no odometry information is used in our framework. However, we can derive the robot poses in our approach at any time, while we incrementally build the global map.

VI. Conclusion

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 maps formerly consisting of laser range scanner reflection points, 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 and their positions to be imprecise.

Acknowledgment

We would like to thank Wolfram Burgard (Univ. Freiburg) for his helpful comments and discussion. This work was supported in part by the National Science Foundation under grants NSF INT-0331786 and NSF IIS-0534929. We also would like to acknowledge the support from the National Institute of Standards and Technology, award Nr. NIST 70NANB5H11119, and discussions with Elena Messina and Adam Jacoff. We thank Giorgio Grisetti (Univ. La Sapienza, Rome) for providing us the scan data.

References

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

[2] 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.

[3] 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.

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

[5] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence*, vol. 31, pp. 355–395, 1987.

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

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

[8] 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.

[9] A. Blake and B. Freeman, "Learning and vision: Generative methods," in *Lecture at ICCV*, 2003.

[10] E. Arias-Castro, D. L. Donoho, and X. Huo, "Adaptive multiscale detection of filamentary structures embedded in a background of uniform random points," *Annals of Statistics*, to appear.

[11] R. Lakaemper, L. J. Latecki, and D. Wolter, "Incremental multi-robot mapping," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005.

[12] S. Richardson and P. Green, "On bayesian analysis of mixtures with an unknown number of components," *Journal of the Royal Statistical Society*, vol. B, 59, pp. 731–792, 1997.

[13] D. Mumford, "Pattern theory: the mathematics of perception," in *ICM*, vol. 3, 2002.

[14] M. J. Beal and Z. Ghahramani, "The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures," in *BAYESIAN STATISTICS 7*, J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, Eds. Oxford Univ. Press, 2003.

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

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

[17] 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.

[18] M. Bennewitz, W. Burgard, and S. Thrun, "Using em to learn motion behaviors of persons with mobile robots," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.

[19] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. of Robotics and Automation*, 2001.

[20] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.