Multi Robot Mapping Using Force Field Simulation

Rolf Lakaemper,
Nagesh Adluru, and Longin Jan Latecki
Temple University
Philadelphia, Pennsylvania, 19122
e-mail: lakamper@temple.edu
e-mail: nagesh@temple.edu
e-mail: latecki@temple.edu

Raj Madhavan
NIST
Gaithersburg, Marlyland, 20899
e-mail: raj.madhavan@nist.gov

Received 10 January 2007; accepted 16 July 2007

This paper describes a novel approach, called Force Field Simulation, to multi robot mapping that works under the constraints given in autonomous search and rescue robotics. Extremely poor prealignment, lack of landmarks, and minimal overlap between scans are the main challenges. The presented algorithm solves the alignment problem of such laser scans utilizing a gradient descent approach motivated by physics, namely simulation of movement of masses in gravitational fields, but exchanges laws of physics with constraints given by human perception. Experiments on different real world data sets show the successful application of the algorithm. We also provide an experimental comparison with classical ICP implementation and a Lu/Milios-type alignment algorithm.

1. INTRODUCTION

This paper focuses on multi robot mapping in the field of urban search and rescue robotics ("rescue robots"). The task of multi robot mapping in rescue environments imposes especially challenging constraints:

- no precise or reliable odometry can be assumed, which means especially that the robots' relative poses are unknown
- due to the nature of catastrophe scenarios no distinct landmarks are given
- the overlap between pairs of the robots' scans is minimal

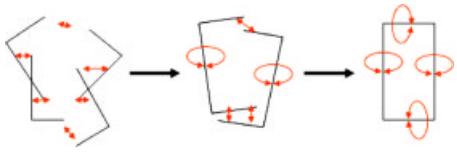

Figure 1. Basic principle of FFS. Forces are computed between four single scans. Red arrows illustrate the principle of forces. The scans are iteratively (here: two iterations) transformed by translation and rotation until a stable configuration is achieved.

There are two reasons for our selection of rescue robots: (1) Rescue robots belong to the class of most advanced robotic systems. Their goal is to search a catastrophic site (like a collapsed building) for survivors. This requires building a map and, once victims are found, localizing them in the constructed map. First responders can then use this map to identify the safest way to rescue the victims. (2) The performance of rescue robots is systematically evaluated in the RoboCupRescue competition series with performance metrics developed by NIST (Jacoff, Messina, & Weiss, 2003). The challenges involved in search and rescue applications provide objective evaluation of robotic implementations in representative environments and promote collaboration between researchers. The robots must demonstrate capabilities in mobility, sensory perception, planning, mapping, and practical operator interfaces, while searching for simulated victims in a maze of increasingly difficult obstacles. The competitions are hosted on Reference Test Arenas for Urban Search and Rescue Robots, which were developed by NIST and were proliferated globally. The arenas provide a continuum of increasingly difficult representations of collapsed buildings that have simulated victims scattered throughout.

This paper introduces a new process, called "Force Field Simulation" (FFS), which is tailored to align maps under the aforementioned constraints. It is motivated by simulation of dynamics of rigid bodies in gravitational fields, but replaces laws of physics with constraints derived from human perception. It is an approach of the family of gradient descent algorithms, applied to find an optimal transformation of local maps (in particular, laser range scans) to build a global map based on feature correspondences between the local maps. Figure 1 shows the basic principle: forces (red arrows) are computed between four single scans (the four corners). The scans are iteratively transformed by translation and rotation until a stable configuration is gained. The single scans are not merged but kept separated. As they are moved according to the laws of the motion of rigid bodies in a force field, single scans are not deformed. FFS has the following properties:

1. Low level correspondences (data point correspondences) are not made by a hard decision (an integral of forces between pairs of points defines the force field in place of hard 'nearest neighbor' correspondences).
2. FFS is a gradient approach; it does not commit to an optimal solution in each iteration step.
3. The iteration step towards an optimal solution is steered by a "cooling process" that allows it to jump the system out of local minima.
4. FFS transforms all scans simultaneously.
5. FFS easily incorporates structural similarity modeling human perception to emphasize/strengthen the correspondences.

2. RELATED WORK

The problem of aligning n scans has been treated as estimating sets of poses (Lu & Milios, 1997a). Since sets of poses and the associated structures (maps) are conditionally independent, this estimation is simultaneous localization and mapping (SLAM). The conditional independence is, e.g., the key for Rao-Blackwellization (factoring the posterior of maps) of particle filters for SLAM (Montemerlo, Thrun, Koller & Wegbreit, 2002).

There have been several algorithms to estimate the sets (Olson, Leonard & Teller, 2006; Frese, 2006; Frese, Larsson & Duckett, 2005; Thrun, Koller, Ghahramani, Durrant-Whyte & Ng, 2002; Minguez, Montesano & Lamiruix, 2006; Konolige, 2003). The underlying framework for all such techniques is to optimize a constraint graph, in which nodes are features, and poses and edges are constraints built using various observations and measurements like odometry and scan-matching of range scans. These techniques differ in

- how they represent graphs, e.g. (Frese, 2006) uses a sophisticated data structure called

Tree-map (Thrun et al., 2002) represents using sparse extended information filters (SEIF).

- how they build constraints, e.g. (Lu & Milios, 1997a) uses linearized constraints obtained from scan-matching and odometry (Olson et al., 2006) works with nonlinear constraints.
- how they optimize the graphs, e.g. (Olson et al., 2006) uses stochastic gradient descent for approximate optima, borrowing the ideas from learning theory (Lu & Milios, 1997a) solves for exact optima using brute-force, and (Frese et al., 2005) use Gauss-Seidel relaxation again for approximate optima.

All these approaches have performed well in many practical cases, but they have one drawback: they are sensitive to the behavior of error models of sensors because of several assumptions and approximations that might not hold with sparse sensing.

(Lu & Milios, 1997a) linearizes constraints by linearizing pose-relations, solving a linear equation of the form $AX=B$ to estimate X, the set of poses. This requires that A is invertible, so they conjecture that A is invertible if the constraint graph is fully connected and the errors of the observations behave in a Gaussian/normal way.

(Olson et al., 2006) present an approximate optimization of nonlinear constraints and demonstrate that their approach of approximating the optimization process in nonlinear state space yields superior results compared to finding exact optima by approximating a nonlinear state space (SLAM) to a linear state space.

Another strategy of attacking the problem is to treat the problem of SLAM from a perspective of aligning n scans simultaneously. The algorithms exploiting this perspective build from image registration techniques, the most famous being iterative closest point (ICP) (Besl & McKay, 1992; Chen & Medioni, 1992) and its numerous variants to improve speed and converge basins (Rusinkiewicz & Levoy, 2001; Lu & Milios, 1997b; Birk & Carpin, 2006). Basically all these techniques do search in transformation space trying to find the set of pair-wise transformations of scans by optimizing some function defined on transformation space. The techniques vary in defining the optimization functions that range from being error metrics like "sum of least square distances" to quality metrics like "image distance" as in (Birk, 1996). Their optimization process itself can be gradient descent or hill climbing or using genetic programming strategy as in (Robertson & Fisher, 2002). All of these techniques have one major limitation, which is they search in *pair-wise* transformation space. Though in some variants of ICP the error from all pair-wise transformations is spread across all transformations to simultaneously align all scans, the procedure can be highly sensitive to outliers (Rusinkiewicz, Brown & Kazhdan, 2005).

FFS also adapted the perspective of aligning n scans; it treats the alignment problem as an optimization problem. Rather than using a least squares solution to compute intermediate motions, FFS uses an iterative gradient technique to solve for (local) optima. Here FFS is similar to the approach proposed by (Eggert, Fitzgibbon & Fisher, 1998), which simulates a dynamic spring system to register multiple range scans simultaneously. They describe the advantages of such a gradient descent system as follows: "*The reason* [*not to use a least square solution*] *is that the effects of any significantly incorrect correspondences are compounded when the best alignment is computed* (…) *With a dynamic system it is possible to move in the direction of an intermediate solution without being totally committed to it*." (Eggert et al., 1998) differ in the choice of the registration function, which in contrast to FFS is based on one to one correspondences between points, as well as in the optimization technique.

FFS uses a gradient method with decreasing step width Δ_t. The registration function (target function) of FFS is based on Gaussian fields, similar to (Boughorbel, Koschan, Abidi & Abidi, 2004). In contrast to (Boughorbel et al., 2004), FFS uses a variable, decreasing σ for each iteration step t. Additionally (Boughorbel et al., 2004) solve the optimum of the registration using a quasi-Newton method, hence they do not steer the system with a step width parameter.

Since we keep the single scans separated, our search space is high dimensional, in the 2D case it is $3n$-dimensional (3D: $6n$-dimensional), with n being the number of scans. For example, our experiment described in Section 4.2 uses 60 scans: our search space is therefore 180-dimensional. Birk and Carpin use a random walk technique to reach the optimal solution. Since random walk techniques tend to become critical in high dimensions, we do not utilize this technique in our approach but decide in favor of a guided (gradient) walk.

This search in high dimensional space at first sight seems very complicated, demanding computa-

tion of a high dimensional gradient, but, fortunately, using potential field simulation for various computer vision tasks like contour detection, and segmentation, and registration has been empirically successful (Yang, Mermehdi & Xie, 2006; Jalba, Wilkinson & Roerdink, 2004; Xu & Prince, 1998; Ayyagari, Boughorbel, Koschan & Abidi, 2005; Paragios, Rousson & Ramesh, 2003; Veltkamp & Hagedoorn, 1999). Since mapping is closely related to registration, the approaches whose motivations are closely related to our approach are (Biber & Strasser, 2006; Ayyagari et al., 2005; Eggert et al., 1998). In (Eggert et al., 1998) they align range scans by moving them simultaneously. The movements are not just based on the minimizing error of transformation computed using correspondences but on the simulated fields generated by imaginary springs attached to the corresponding points. Our technique differs from (Eggert et al., 1998) in that the force field is generated not just by closest point correspondences but using perceptual principles and Gaussian fields similar to (Boughorbel et al., 2004). (Biber & Strasser, 2006) also performs a search in $3n$-dimensional space. For each configuration they compute energy as the sum of the normal distribution transforms (NDTs) (Biber & Strasser, 2003) of all the scans in the configuration and update the configuration using Newton's optimization algorithm that involves the first and second derivatives of the energy. Their approach is very closely related to ours but does not use perceptual features and rigid body dynamics and hence, in principle, can be more sensitive to outliers.

3. FORCE FIELD BASED MAPPING

The following motivates and describes the FFS algorithm. A pseudo code representation can be found at the end of this section.

3.1. Basic Principle

To draw the analogy to Newtonian physics, each scan s_i can be seen as a rigid body of masses: the scan points represent the masses, rigidly connected by massless rods. A global map g defines the transformation of all scans; it therewith defines the distribution of all masses (the union of all scan points). In the framework of Newtonian physics the gravitational forces between these masses forms a gradient field. The FFS algorithm is motivated by simulation of the movement of bodies in a gradient field. In contrast to pure physics it replaces physical principles of masses and forces by principles that correspond to human visual perception, i.e., gravitation is replaced by "strength of correspondence." Also, to achieve convergence to a stable state of minimal total energy, the kinetic energy is not taken into account, i.e., the velocity of each rigid body after each iteration step is set to 0. Also FFS uses a "cooling" strategy in its step width parameter that initially adds energy to the system to allow for escape from local minima (see Section 3.2).

Let $\mathcal{S}=s_1,\ldots,s_n$ be a set of n scans gained from laser range scan devices. A scan $s_i=(p_1^i,\ldots,p_j^i)$ consists of j data points. Data points are the coordinates of reflection points of the laser range scanner in a local coordinate frame defined by a single robot pose. We also assign a scalar value, a *mass* m_j^i, to each data point, which can be interpreted as the perceptual importance. For the purpose of multi robot mapping we assume that each scan is possibly gained from a different robot, while the robots' relative poses are unknown or poorly estimated. The task of the algorithm is an optimization over the set of the robots' poses; hence the goal is to find transformations for all n scans $s_{i=1,\ldots,n}$ to register the scans, such that similar features in different scans match "perceptually consistently" when they are superimposed on top of each other.

Observe that the order of local maps is irrelevant in our framework since we transform the scans simultaneously, which is an important property of the algorithm to be applicable for multi robot mapping. For single robot mapping, FFS is canonically extendable to online FFS (see Section 3.6).

The transformations performed are rotation θ_i and translation x_i, y_i of each scan s_i. Superimposing the transformed maps builds a *global map* g as shown in Figure 2. During the global map building process single maps are not merged but kept separated. Therefore a global map is defined by the vector of the transformation parameters of all scans: a *global map* $g=(t_1,\ldots,t_n)$ is a $3n$-dimensional vector of transformation parameters $t_i=(x_i,y_i,\theta_i)$ of all n scans $s_{1,\ldots,n}$. The space of all global maps is denoted by $\mathcal{G}$. To register the scans, we define a *fitness measure* P_g to evaluate the "perceptual consistency" of a global map g. Finding the global map g_k that minimizes P_g is clearly an optimization problem. FFS solves this optimization problem with a gradient technique that

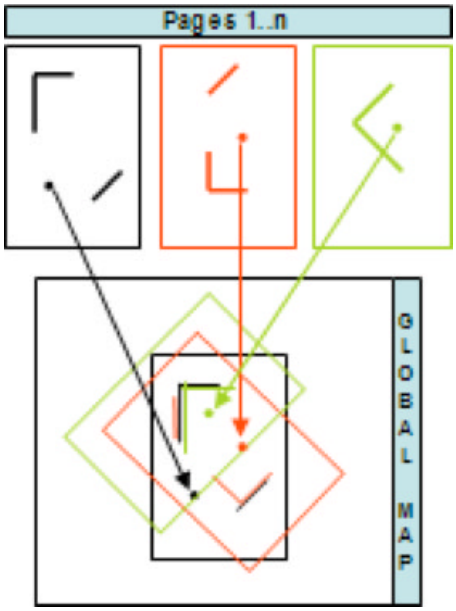

Figure 2. Single scans are transformed to build a global map.

iteratively transforms all scans simultaneously until a stable configuration (local minimum) is reached. The following section will motivate and define the fitness measure P_g as well as the implementation of the gradient approach.

3.1.1. Correspondence Function

As in (Boughorbel et al., 2004), the basic idea of our registration method is to use a Gaussian field to define a strength of correspondence between data points, i.e., a measure for both spatial proximity and visual similarity of two points belonging to different scans.

A *correspondence* between data point p_1 and a data point p_2 is defined as a vector

$$V(p_1,p_2) = C(p_1,p_2)\frac{p_2 - p_1}{\|p_2 - p_1\|}. \quad (1)$$

Its magnitude $\|V(p_1,p_2)\| = C(p_1,p_2)$ describes the strength of correspondence, defined as

$$C(p_1,p_2) = \frac{1}{\sigma_t\sqrt{2\pi}} e^{-\|p_2 - p_1\|^2/2\sigma_t^2} m_1 m_2 \cos(\angle(p_1,p_2)) \quad (2)$$

with m_i being the mass assigned to p_i (see Section 3.4), and the angle $\angle(p_1,p_2)$ being the angle between the *directions of points* p_1, p_2, which will be defined in Section 3.3. Intuitively, the direction of a point is the direction of an underlying model of a linear structure (a line segment). Expression (2) can be interpreted as a force field whose sources are located in the data points. It has the following properties:

1. The strength of correspondence decays with Euclidean distance; the influence of distance is controlled by the parameter σ_t.
2. The strength of correspondence is weighted by the mass of each data point and depends on the angle between point directions, i.e., it is 0 for orthogonal directions, 1 for parallel directions.

We propose this model for the following reasons:

1. Distance likelihood and parallelism follow the basic principles of Gestalt psychology (Wertheimer, 1958), modeling low level cognition.
2. The scale parameter σ_t gives additional freedom to adjust the process (see Section 3.2); it enables the correspondence process to work on different visual scales.
3. Assigning a mass to a data point can be seen as assigning a visual importance to it. Data points in regions of interest, computed by mid level cognitive modules, will be assigned higher masses. See Section 3.4 for further details.

In terms of the physical framework, the correspondence $V(p_1,p_2)$ [Eq. (1)] describes a force on p_1 towards p_2 with strength $C(p_1,p_2)$. Embedding the scans s_i into $\mathbb{R}^2$ using the transformations defined by a global map g, we can define a vector field $F:\mathbb{R}^2 \supset \mathcal{P} \rightarrow \mathbb{R}^2$, the *force field* on the set of all points $\mathcal{P} = \{p \,|\, p \in \cup_{i=1..n} s_i\}$ by summing the correspondences.

$$F(p_i) = \sum_{p_j \in \mathcal{P} \setminus p_i} V(p_i, p_j). \tag{3}$$

By definition of the strength of correspondence F is radial and hence a gradient field. With

$$A = m_1 m_2 \cos(\angle(p_1, p_2))$$

the overlying potential is defined by

$$P(p_i) = \frac{1}{2} \sum_{p_j \in \mathcal{P} \setminus p_i} \int_{\infty}^{r} \frac{A}{\sigma_t \sqrt{2\pi}} e^{-z^2/2\sigma_t^2} \, dz \tag{4}$$

with $r = \sqrt{(X-x)^2 + (Y-y)^2}$, $p_i = (X, Y)$, $p_j = (x, y) \in \mathcal{P}$.

Note: $P(p_i)$ is the potential over F since

$$F(p_i) = -\nabla P = \begin{bmatrix} -\frac{\partial \mathcal{P}}{\partial x} \\ -\frac{\partial \mathcal{P}}{\partial y} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{p_j \in \mathcal{P} \setminus p_i} \frac{A}{\sigma_t \sqrt{2\pi}} e^{-r^2/2\sigma_t^2} \cdot \frac{X - x}{r} \\ \sum_{p_j \in \mathcal{P} \setminus p_i} \frac{A}{\sigma_t \sqrt{2\pi}} e^{-r^2/2\sigma_t^2} \cdot \frac{Y - y}{r} \end{bmatrix}$$

$$= \sum_{p_j \in \mathcal{P} \setminus p_i} \frac{A}{\sigma_t \sqrt{2\pi}} e^{-r^2/2\sigma_t^2} \cdot \mathbf{u} = \sum_{p_j \in \mathcal{P} \setminus p_i} V(p_i, p_j),$$

where $\mathbf{u} = p_i - p_j / \|p_i - p_j\|$.

Finally, we define the *fitness measure* or *potential of a global map* $g \in \mathcal{G}$ as the sum of potentials of all data points $p \in (\mathcal{P})$:

$$P(g) = \sum_{p_i \in \mathcal{P}} P(p_i). \tag{5}$$

In this framework, the potential $P(g)$ can be interpreted as the weighted average distance of all point pairs of different scans, the weight being the strength of correspondence. This potential can be seen as the quality of registration achieved by the transformations defined by g. To minimize the potential we apply an iterative gradient descent approach, the gradients in each data point given by $F(p_i)$ [Eq. (3)]. Computing the correspondences explicitly gives us these gradients, hence in the implementation of the algorithm there's no need to explicitly compute and derive the potential $P(g)$ [Eq. (5)] for the actual gradient descent.

In FFS, the computation of the transformation of each scan is determined assuming movement of rigid bodies in the given gradient field, i.e., all data points $p_i^j \subset \mathcal{P}$ of a single scan s_j share the same transformation, consisting of rotation and translation. However, Eq. (3) does assume a nonrigid, independent movement of the data points; also the points' potential $P(p_i)$ [Eq. (4)] is defined over the space of all single (not rigidly connected) point configurations, which is a $2m$-dimensional space with $m = |\mathcal{P}|$. This means the gradient $F(p_i)$ is defined under the assumption of unrestricted freedom of movement in $\mathbb{R}^2$. To implement rigid body movement, we have to impose a restriction on the movement. The restriction is defined by the possible point configurations that are allowed by the transformations $g \in \mathcal{G}$. The laws of rigid body dynamics define these constraints: computing the gradient $F(p_i)$ in each data point as in Eq. (3) results in a $2m$-dimensional gradient vector; the laws of rigid body dynamics map this vector to a $3n$-dimensional vector ∇g. ∇g describes the transformation of all n scans such that each data point moves in the direction of maximum descent of $P(g)$ in Eq. (5), i.e., under the rigid body constraints. We therefore achieve a movement in the direction of the gradient gained in the $2m$-dimensional single point space projected onto the restricted $3n$-dimensional rigid movement subspace.

The basic laws of dynamics of rigid bodies in force fields accumulate the translation of all masses of a single scan into a single translation and a single rotation around a defined center. For each scan s_i, the translational and rotational acceleration has to be determined. The translational acceleration $a_T(s_i)$ of a scan s_i is defined by

$$a_T(s_i) = \frac{\sum_{p \in s_i} F(p)}{\sum_{p \in s_i} m_p}. \tag{6}$$

The rotational acceleration a_R is computed by torque and moment of inertia. Torque and inertia play the role of force and mass, respectively, but take into account the distance to the rotational center c_R:

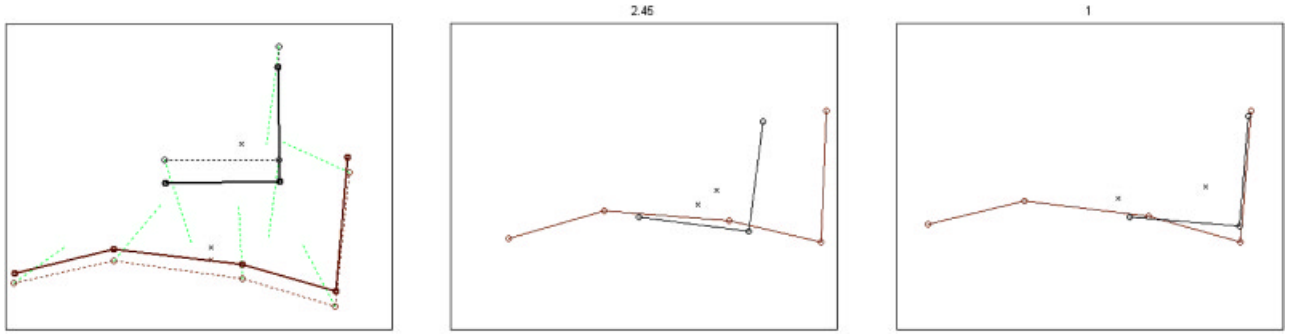

Figure 3. Left: two scans (black/brown) superimposed. Dotted lines: scans at time t, green lines: forces on the data points at time t. Solid lines: scans after one iteration, time $t+\Delta t$. Both scans are translated/rotated according to the forces. Center: after iteration 5. Right: iteration 10.

$$\text{inertia} = \sum_{p \in s_i} m_i \|p_i - c_R\|^2,$$

$$\text{torque} = \sum_{p \in s_i} \|p_i - c_R\| \times F(p).$$

a_R is defined as

$$a_R = \frac{\text{torque}}{\text{inertia}}. \tag{7}$$

The rotational center c_R is either defined as the robot's position or by the center of mass. Experiments show that in the first iteration steps it is useful to set the rotational center to the center of mass, while in later steps the robot's position is preferable. The first choice enables easier rotation, the second is modeling the actual scan setting more precisely. Hence, the closer the global map is to the solution, the more preferable is the robot's position as rotational center.

With a_T and a_R the transformation $t_k=(x_k, y_k, \theta_k)$ for scan s_k is defined by

$$(x_k, y_k) = \tfrac{1}{2} a_T \Delta_t^2, \tag{8}$$

$$\theta_k = \tfrac{1}{2} a_R \Delta_t^2, \tag{9}$$

Δ_t being the step width of the gradient descent, as described in Section 3.2.

With these constraints, the gradient in each iteration is computed by the following steps:

1. For each pair of points $p_i, p_j \in \mathcal{P}$, compute $V(p_i, p_j)$.
2. For each point $p_i \in \mathcal{P}$, compute $F(p_i)$.
3. For each scan $s_k \in \mathcal{S}$, compute the transformation $t_k=(x_k, y_k, \theta_k)$ using the points $p_i^k \in s_k$. This step results in a $3n$-dimensional gradient vector ∇g.

Computing all correspondences V in step one is an $O(n^2)$ process; Section 3.5.1 will deal with the necessary reduction of computational complexity.

Figure 3 shows two iteration steps of FFS using two simple scans, consisting of three and five data points. In the left figure, the forces $F(p_i)$ in each data point p_i are plotted as green dotted lines. The two scans are transformed until they are superimposed, i.e., a stable configuration [local minimum of $P(g)$] is reached. As in all gradient descent methods, the determination of the step width Δ_t is crucial. Also, gradient methods imply the danger of being trapped in local minima. We tackle both problems with the determination of step with Δ_t and σ_t as described in the following section.

3.2. Cooling Down the Motion: Time Stepping Δ_t and Parameter σ_t

The determination of step width parameter Δ_t in any gradient descent approach is a well known problem. Δ_t chosen too small results in inapplicably slow convergence behavior and is not robust to noise; Δ_t cho-

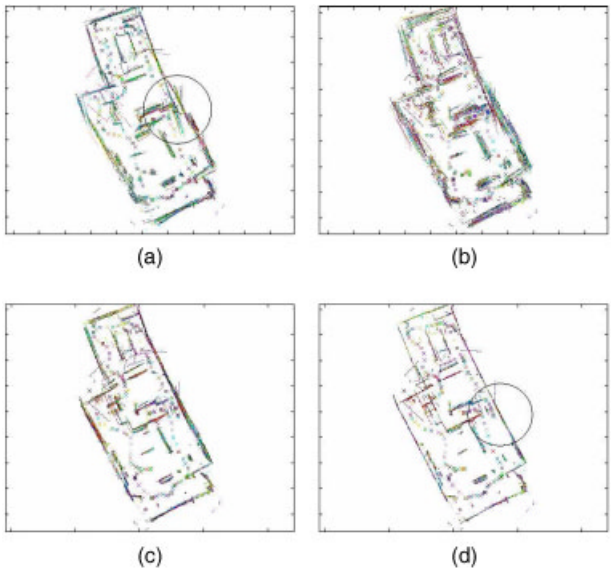

Figure 4. Apartment data set. (a) Initial configuration. The circled area shows an error due to incorrect loop closing. (b) A large step parameter dT blurs the map in the first iteration step to escape from the local minimum. (c) Iteration 50. (d) Iteration 150. FFS has not only contracted the edges given in (a), but also has realigned the entire global map to fix the error (circled area).

sen too big might miss the optimum. In FFS, the step width Δ_t is used as a steering parameter of the algorithm in connection with the parameter σ_t, which determines the influence of distance in the correspondence function. We designed Δ_t as exponentially decreasing, σ_t linearly decreasing.

A large Δ_t allows the scans to be massively relocated (shuffled); they overshoot their correct position in the direction of the correspondence gradient. Naturally, a small Δ_t moves the scans less (the amount of replacement is directly proportional to Δ_t^2, as defined by the laws of movement). We chose the strategy of decreasing Δ_t and σ_t experimentally, having analogies of the cooling behavior of algorithms like simulated annealing in mind. The imprecise, nonoptimal large Δ_t at the beginning allows the system to possibly escape from local minima. Observe that in contrast to a technique like simulated annealing we cool down a gradient guided process, not a random state change or a random walk technique that would not be applicable in our high dimensional search space. We therefore avoid the problems with a high computational load (high number of iteration steps) that tend to appear in simulated annealing due to unguided selection of the next state.

The parameter σ_t in Eq. (2) steers the influence of distance in the computation of point correspondences. A large σ_t enhances the relative influence of data correspondences with greater distances and, since it equalizes this spatial proximity property, favors the influence of visual similarity. A small σ_t emphasizes local proximity, which is useful if the global map is already close to an optimum.

The effect of cooling is shown in Figure 4 showing our experimental results on the "apartment data set." Observe that the potential function (the fitness measure) as shown in Figure 5 is not monotonically decreasing in the first iteration steps. This shows an "overshooting" of the system due to a large Δ_t; in the data registration this can indicate an escape from a local minimum.

It is important to mention that after each iteration the system resets the velocity of each scan to zero. This guarantees that the system converges to a stable state (assuming $\Delta_t \to 0$).

3.3. Point Direction and Optional Resampling

The "point direction" is used in the correspondence Eq. (2) to assign to points the direction of an underlying linear structure. It is derived by modeling the point set with line segments using the extended EM algorithm described in (Latecki & Lakaemper, 2006). Utilizing a segment split and merge approach, the extended EM algorithm automatically adjusts number and location of the line segments in a way such that linear structures are represented (by a single, possibly long line segment) as well as round structures (by multiple short segments). Hence, even scenarios not being rich in linear structures are robustly represented. The algorithm was already successfully applied to model indoor and outdoor rescue scenarios. A 3*D* version of this algorithm for approximation of scan points with planar patches is described in (Lakaemper & Latecki, 2006).

The data of each scan s_i are approximated by a set of line segments L_i. The direction of a point p in s_i is the angular direction (in the scan's coordinate frame) of the closest line segment in L_i. The approximation of the data set with line segments results in a very stable and intuitive estimation of point directions. Figure 6 shows the influence of point directions for the correspondences. The closest line segment l_j to the point p is called the *supporting line*

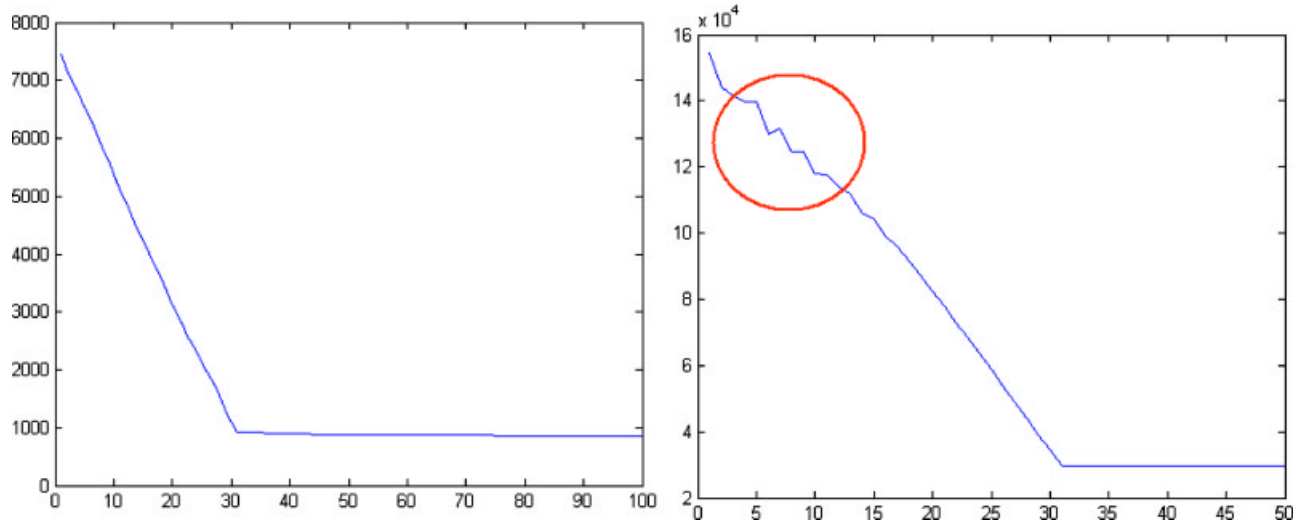

Figure 5. Left: potential vs. iterations of FFS for disaster data. Right: potential for apartment data set. The potential (encircled) of the apartment data is not monotonically decreasing, indicating a possible escape from a local minimum.

segment if its distance to p is below a certain threshold. Points without supporting line segments are removed from the data set. Due to the nature of the extended EM algorithm, these removed points are points in areas with low point density. Low point density results from objects that are hit less than others. This is the result of either erroneous scanning, nonstatic objects, or low scanning density, which by itself results from either long distance to an object or simply the fact that a certain location was only scanned a few times. All of these topics include uncertainty about the existence of the object, hence we disregard such points. The behavior of the extended EM segment fitting guarantees that safe, distinct objects are not removed. The removal of uncertain data increases the stability of the FFS algorithm.

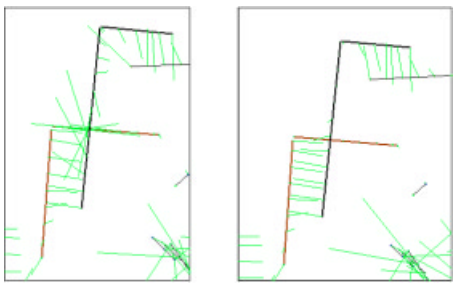

Figure 6. Left: forces between two scans (red lines belong to first scan, black lines to second scan) computed without direction information. Right: forces computed using L and K as described in example 2 (using equal masses). Correspondences between nonparallel structures are weakened.

Having the segments, the data can optionally be resampled along the supported line segments with an equal sampling distance. Such a point set has a more homogeneous distribution of points, which tends to be advantageous: experiments showed that homogeneous distributions are helpful to avoid local minima if the configuration of scans is still far from the optimal solution, since overrepresented areas (e.g., features with unusually high scanning density due to multiple scans in a single location) are equalized. Additionally, the optional resampling can significantly speed up the computation if the number of data points reduces drastically due to the chosen sampling resolution (see Section 3.5.1). If the data is resampled, only the line segments (two endpoints) are stored, also resulting in a significant data compression (typically about 1:100).

3.4. Regions of Interest

A major difference from the pure physics simulation is that the mass values assigned to the data points are not assumed to be constant. The mass m_i for a point p_i is used to compute the force as in Eq. (3), yet it can be reassigned a different value for the compu-

tation of movement of the scan (we are not modeling physics but perception, hence freedom from Newton's law is given). Steering the mass enables the algorithm to react better to perceptual properties: there is no perceptual reason for an "important point," e.g., a corner point, assigned a high mass for force computation, to be less mobile than other points during movement computation (caused by its high mass). This observation suggests using different masses during the computation of forces than during the computation of the movement.

To compute the mass distribution, we focus on cognitively interesting features in the global map by defining regions of interest (ROI). ROI are regions around certain features detected in the global map. Since the global map is analyzed, not single scans, features are detected even if they are not present in single scans but emerge from the overlap of scans, a case that is very likely in our assumed setting of sparse scans with little overlap (e.g., a corner that consists of one segment of scan 1 and one segment of scan 2 will be detected, although it might not be present in either scan). In general, once an interesting feature is detected, the ROI is defined as a region around the feature. To let FFS focus on these regions, the correspondences (forces) in these regions are emphasized. The emphasis is gained by the assignment of mass values: data points in ROI are assigned a higher mass during the force computation. Hence forces in regions of interest are stronger, i.e., the correspondences and therefore the rigid body movement are based on forces with focus on perceptually interesting features. There are many possible ROIs, in particular higher level objects as described in the spatial semantic hierarchy (Kuipers, 2000) could be used, e.g., corridor junctions represented with line segments. In the current implementation, ROI are defined by automatic corner detection as an example for ROI. Though such an implementation restricts the applicability to the presence of corners, it can be easily exchanged to focus on different features. In this implementation we benefit again from the supporting line segments, as described in Section 3.3: corners are gained from intersections between supporting line segments. Each intersection between two supporting line segments (extended to infinite lines) in the global map with a distance below a certain threshold to both segments and an intersection angle between 80 and 100 deg counts as a corner. The ROI are the union of discs with a given radius centered at all corners [see Figure 7(a)]. Data points

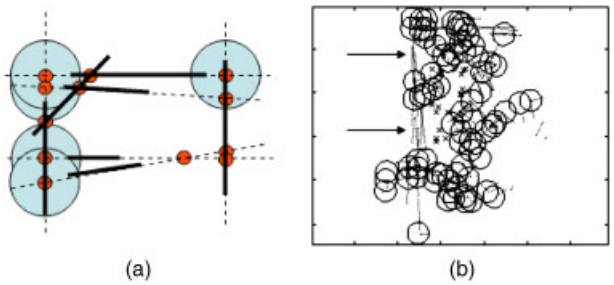

Figure 7. Regions of interest (ROI). (a) Line segments (black) and intersection points (red). Intersection points satisfying the constraints on angle of intersection and distance to segments define ROI (blue discs). (b) ROI of an early global map (upper left corner of the NIST data set (Figure 9)). Arrows mark areas of lower interest ('wall-areas'), see text.

in these regions will be assigned higher mass values [Figure 7(a) does not show the data points but only the supporting lines]. Figure 7(b) shows a global map with ROI at an early iteration stage. Since the scans are still poorly arranged, the dislocation of superimposed scans leads to detection of many corners. However, long walls, represented by parallel lines [in Figure 7(b) the vertical lines on the left side, marked by arrows], are not emphasized. This allows for easier relocation in these "wall areas" in favor of corner correspondences. Without ROI, point correspondences between data points of the wall areas (then equal in strength to correspondences between corners) would fix the scans' positions in a local minimum, not giving the necessary freedom to be relocated (intuitively: to slide along each other) with regard to the perceptually more important correspondences. In early iteration stages, the ROI are still widespread around the correct feature positions, as seen in Figure 7(b). Since in later iteration steps the scans become better aligned, ROI become better focused and mark the positions around features in higher precision.

3.5. Computational Complexity

3.5.1. Time Complexity

The definition of C [Eq. (2)] on pairs of data points leads to an algorithm with $O(n^2)$ time complexity where n is the number of points. This is certainly

prohibitive for real applications. Different techniques can be used to reduce the complexity by taking advantage of two main properties of Eq. (2):

1. For each data point only its local neighborhood must be examined, since the forces between points rapidly decrease with distance. Hence some techniques successfully built into ICP implementations (which suffers from the same complexity problem) can be used to reduce the complexity, e.g., KD-trees. In the current implementation, we take advantage of the line segment representation of the data. We use a bounding box intersection approach on axis aligned bounding boxes around the line segments: bounding boxes around all line segments are computed and extended by $2\sigma_t$ in each direction. The force between two data points is computed only if the two corresponding lines' bounding boxes overlap. Hence we first reach a computational complexity based on the number $m \ll n$ of line segments, which is significantly lower than the number n of data points. Secondly, though bounding box intersection is an $O(m \log m)$ computation (note again: m=number of segments), update techniques as reported in (Cohen, Lin, Manocha & Ponamgi, 1995) reduce the expected complexity to $O(m)$. This linear complexity is reported under the constraint of "relatively small" movements of objects, such that the $O(m \log m)$ sorting in the sweep and prune step reduces to $O(m)$ on a nearly presorted list. The constraint of small movements is met for most of the iteration steps in FFS. To give an idea of the order of magnitude of reduction that is achieved, some numbers for the NIST data set (see Section 4.2) should be mentioned:
   - 60 scans contain a total of 21,420 points, represented by 332 line segments (on average: 65 points per segment)
   - average number of colliding pairs of segments per iteration: 1500; hence we have $65 \times 65 \times 1500$(=6,337,500) computations, compared to $21,420^2$(460,000,000).
2. The data points have to be evaluated with a certain accuracy only. By approximating the evaluation of force field we can achieve computational reduction in the following two manners:
   - The current FFS implementation subsamples each segment equally with some sampling distance. For the NIST data set (sampling distance 10 cm), we achieve in average 7 data points per segment; the force computation is therefore reduced to $7 \times 7 \times 1500$(=73,500) computations.
   - Greengard and Strain introduced fast Gauss transform (FGT) (Greengard & Strain, 1991), which is in turn based on fast multipole methods introduced for high speed simulation of particle dynamics in potential fields (Greengard & Rokhlin, 1987). The main advantage with FGT is that the force field can be computed in linear time with a constant factor depending on the precision required in computation of the field. Details can be found in (Greengard & Strain, 1991). The main idea is to compute the force field using a divide and conquer strategy and exploiting Hermite and Taylor expansions. FGT was first introduced in (Elgammal, Duraiswami & Davis, 2003; Ayyagari et al., 2005).

3.5.2. Space Complexity

Since we approximate the scans by segments, it is not necessary to keep the original data. For each scan, only the segments' endpoints have to be stored. Experiments with the extended EM algorithm (Latecki & Lakaemper, 2006) on 2D laser data sets show an average compression rate of 1:100 (200 data points per segment).

3.6. Online FFS

The described algorithm easily can be extended to online SLAM, i.e., scans are recorded and processed subsequently, as they arrive from the laser device. The extension is canonical: each additional scan is pre aligned, then FFS runs on the previously aligned data set plus the new scan. The current FFS system targets the application of multi robot mapping, hence the sequential processing is not implemented yet.

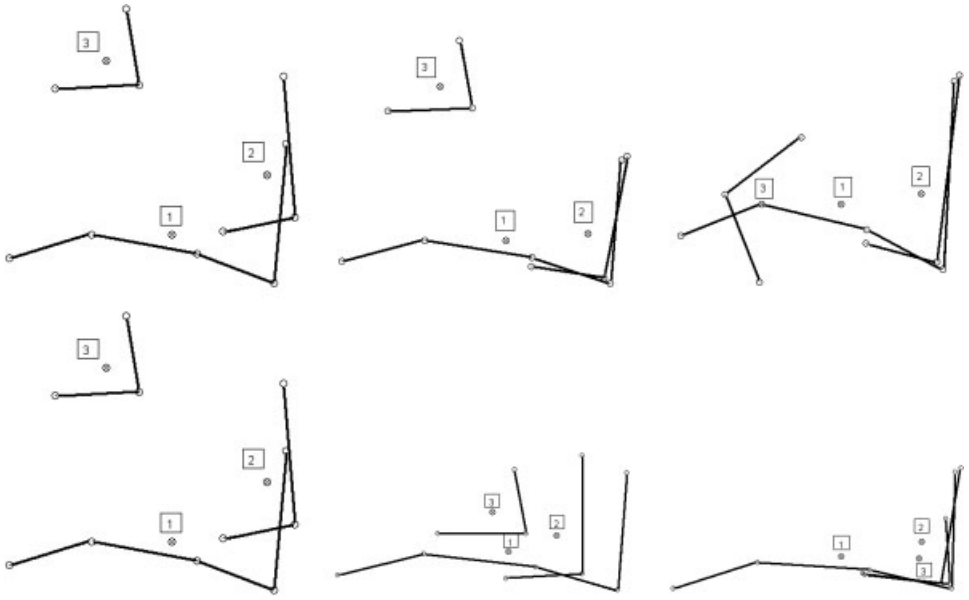

Figure 8. The top row shows three steps of the alignment of three scans (each scan consists of a single corner only) using classical ICP, the bottom-row shows the results of the proposed approach. The alignment progress can be seen from left to right in both cases. The square boxes show the robot poses of the scans.

Algorithm 1 Force field based mapping.

1: Compute supporting line segments (Section 3.3)
2: Resample data set (Section 3.3)
3: $S \leftarrow$ initial state of transformations
4: Initialize step width Δ_t and σ_t (Section 3.2)
5: **repeat**
6: Compute regions of interest (ROI) on global map (Section 3.4)
7: Assign masses to data points with respect to ROI
8: Compute forces using σ_t (Eq. (3))
9: Assign constant mass values to data points
10: Compute rotational and translational acceleration (Eq. (7) and Eq. (6))
11: Compute transformations ∇g
12: $g \leftarrow g + \nabla g$
13: Set velocity of all points to zero
14: u\Update σ_t and Δ_t (Section 3.2)
15: **until** average of relocations > Threshold

3.7. Prealignment

The prealignment does not make use of odometric sensor data, but is based on shape similarity. It finds distinct shape features in single scans and tries to find an optimal overlap based on the shape similarity of these features. For further details see (Adluru, Latecki, Lakaemper & Madhavan, 2006).

4. EXPERIMENTAL RESULTS

4.1. Performance Comparison to Classical ICP

Figure 8 shows the difference between the results of aligning a hypothetical set of three simple scans using classical ICP and our approach. Due to the hard constraints of using the nearest point correspondence, only ICP (top row) ends in a nonperceptually optimal configuration.

FFS takes into account the correspondences between all points first; a decreasing σ_t finally guarantees the correct positioning of the scans, decreasing

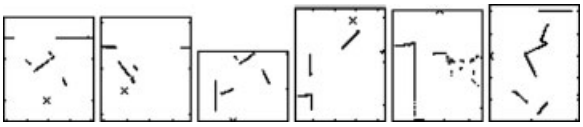

Figure 9. Six out of 60 scans of the NIST rescue scenario data set. The scans in this data set are very sparse and have minimal overlap.

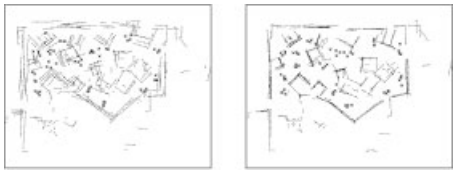

Figure 10. Left: 60 scans superimposed building a global map using a rough initial transformation estimation. Right: after 20 iterations of FFS. The crosses show the robots' positions. (The reader might try to find the six single scans of Figure 9 in the global map. 1,2,3 and 6 are part of the upper left corner, 4 and 5 are located in the lower left corner.) The final result of FFS is shown in Figure 11, left.

the influence of points being too far away. The bottom right image shows the result of FFS after 12 iteration steps.

4.2. NIST Disaster Area

The NIST data set used in this experiment simulates a typical data set of multi robot mapping in rescue scenarios. It is especially complicated, as it matches the complicated constraints imposed by these settings, which contain only imprecise odometry, no landmarks, and very little overlap.

The data set consists of 60 scans taken from 15 different positions in directions E,N,W,S with an overlap of 5 deg (i.e., overlap between E and N, N and W, etc.). The area has a size of $\sim 10 \times 15$ m, the 15 locations differ ~ 2 m from each other (see Figure 9 for sample scans). The distance between the positions of the four scans taken from an assumed single position differs up to 30 cm, with an angular error of up to 20 deg to the assumed direction. The size of the arena was $\sim 12 \times 10$ m. This data set can be seen as a multi robot mapping scenario using 15 robots, with four scans gained from each robot. Although the prealignment assumes this setting, FFS actually treats the 60 scans as independent scans without the help of any further information, e.g., constraints on the groups of four. The single scans have very little pair wise overlap. Figure 9 shows six example scans, all located on the left side of the global map; the overlapping pairs among these scans are (1,2),(1,3),(1,4),(2,3),(3,4),(3,6),(4,5).

The test performed on this complicated data set demonstrates the robustness of the FFS system. The initial, prealignment map is gained by the shape-based algorithm described in Section 3.7. Figure 10 shows the initial global map as well as iteration step 20, Figure 11 the final global map, after 50 iterations. The data set was resampled as described in Section 3.3. The radius of the ROI was set to 5 cm; the parameters for the motion cooling were set as

- Δ_t decreases from 5 to 1 with step factor of 0.96
- σ_t decreases from 15 to 4 with step size of 0.25

Although the data is poorly prealigned and the overlap between the single scans is minimal, FFS successfully reconstructs the global map, which proves its applicability for this multi robot setting. The mean translation/rotation of the scans (translation/rotation between initial and final global map) is 16 cm/4 deg, the maximum translation/rotation is 50 cm/10.5 deg.

The alignment can also be seen as a movie at http://knight.cis.temple.edu/~lakaemper/FFS/FFSTheMovie.wmv

The movie especially makes clear the effect of the motion cooling.

Figure 11 shows the result after 50 iteration steps. The computational time for each iteration is

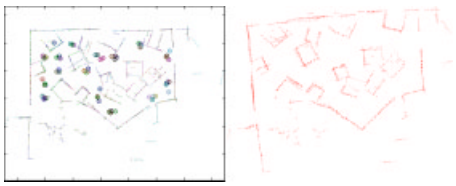

Figure 11. Left: final map (after 50 iterations) of intialization in Figure 10 with FFS. Right: the final map obtained by the Lu and Milios technique as reported in (Lu & Milios, 1997a). The systems lead to results of comparable quality.

1 s on a 1.5 GHz desktop computer in the current MATLAB implementation, using the bounding box approach as described in Section 3.5.1 and resampling with a distance of 10 cm.

The sensitivity to initial conditions was tested as follows: using the result of the previously described experiment (Figure 11, left), we distorted this map by random transformations of the single scans in the range of ±30 cm (translation) and ±20 deg (rotation). The visual appearance of such maps is similar to the appearance of the map shown in Figure 10, left. Taking these distorted maps as initial configurations, FFS achieved results that differed from the source map by a maximum of 0.9 cm and 0.7 deg, i.e., they were visually identical.

In order to compare the proposed FFS approach to the state of the art of existing robot mapping approaches, we applied three influential approaches to the NIST data set illustrated in Figure 11. We applied the particle filter based DP-SLAM (Eliazar & Parr, 2004), the ICP based VASCO robot mapping module of CARMEN, and improved grid-based SLAM with Rao-Blackwellized particle filters (Grisetti, Stachniss & Burgard, 2005). All three approaches failed to produce any reasonable results, since they are based on sequential processing of data (online SLAM), which cannot be applied on this data set due to the extremely minimal overlap of consecutive scans (even if the order is known).

However, we compared to a recent implementation of Lu and Milios type SLAM (Lu & Milios, 1997a). The results are shown in Figure 11. Both algorithms show an overall comparable performance, although local differences can be seen: the Lu and Milios type SLAM reconstructs the top right corner better, while FFS performs better on the left side.

Figure 5, left, shows the potential $P(g)$ vs. iteration curve for this data set. The potential is monotonically decreasing, hence in this case FFS steers directly towards a (local) minimum, which is reasonable due to the initialization. The next experiment will show a different case.

4.3. Apartment

This experiment demonstrates the benefits and applicability of FFS in data sets which are incorrectly prealigned, e.g., due to effects of wrong loop closing. We used the IROS 2006 test data set taken from http://staff.science.uva.nl/~zivkovic/FS2HSC/dataset.html. The data set consists of about 2000 scans from which we select every tenth scan. Thus, our test data set consists of 200 scans taken from a single robot in an apartment of size about 16×8 m. As shown in Figure 4(a), the prealignment gained shows a huge error, additionally the alignment is very imprecise (blurred features). The experiment shows the power of FFS to escape local minima: starting with a large stepping parameter Δ_t, the first transformation blurs the data set and therefore weakens the wrong correspondences, giving space for new connections, Figure 4(b). Transforming all scans in parallel eventually results in a version of the map, which not only shows the misaligned parallel walls correctly contracted but also corrected the huge error as shown in Figure 4(d). The values of the parameters are equal to the experiment in Section 4.2. Figure 4 also shows a limit of the algorithm: one single initially strongly misaligned scan does not find consistent correspondences and therefore cannot be correctly repositioned by the algorithm. It stays in its incorrect position. We assume that no algorithm working only on low level perceptual features is able to handle such a strong error correctly; mid level cognitive correspondences are needed. However, mid level perceptual features can easily be integrated into the system using correspondence functions modeling these perceptual forces, which will be part of the future work on the system.

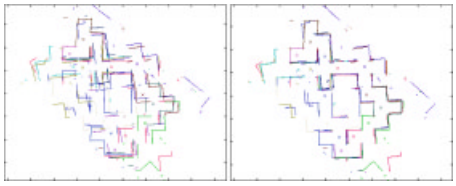

Figure 12. Left: initial configuration of NIST's maze data (16 scans). Right: after five iterations of FFS.

4.4. NIST Maze

This data set consists of 16 scans with similar structures, a typical indoor environment, yet again scanned with minimal overlap. See Figures 12 and 13 for this experiment. The final result was achieved after 20 iterations, transforming the single scans up to 25 cm and 20 deg. The size of the maze is ≈10 ×10 m. FFS was able to align the scans correctly.

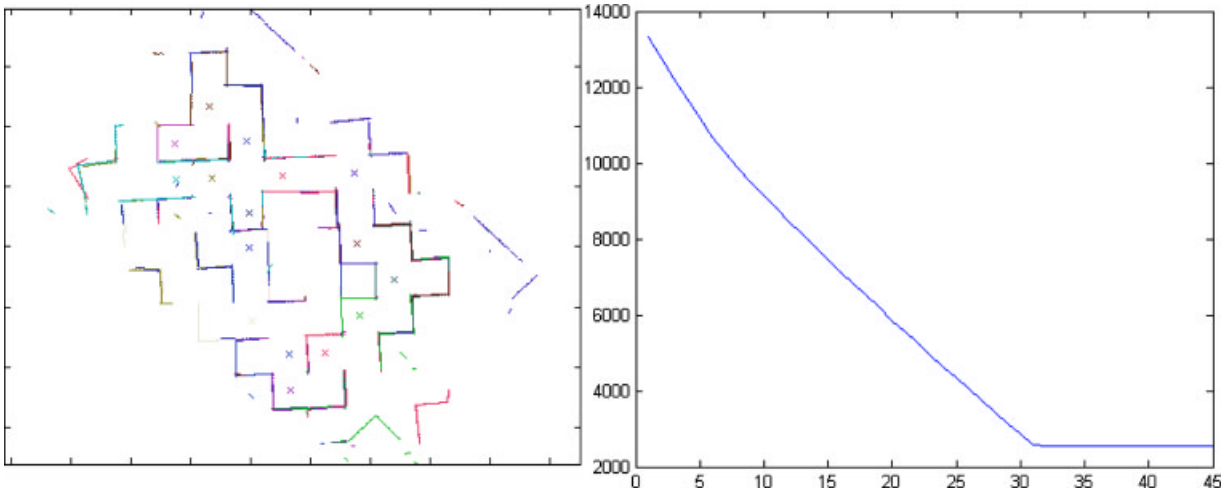

Figure 13. Left: final map obtained with FFS. Right: the potential vs. iterations plot.

5. CONCLUSION AND FUTURE WORK

We presented a new approach to the problem of multi robot mapping under the constraints given in rescue scenarios. It does not rely on odometry, i.e., the relative pose between the robots is unknown. It also can deal with the problem of extremely minimal overlap. Experiments conducted on a real data set of a disaster area from NIST showed the performance of the FFS approach under these complicated constraints and proved its applicability to the problem of multi robot mapping; they also proved the excellent performance of the algorithm correcting effects from wrong prealignment. The future work will mainly focus on detection of higher level features: the modeling of the correspondence function with respect to the masses opens different ways to interface to mid level modules. The approach is easily extendable to 3D; a report about the performance of an implementation of the 3D FFS is the topic of a forthcoming paper.

ACKNOWLEDGMENT

We thank Andreas Nüchter from University of Osnabrück, Germany, for his helpful comments and the code for the experiments on 3D-Lu/Milios-SLAM.

REFERENCES

Adluru, N., Latecki, L.J., Lakaemper, R., & Madhavan, R. (2006). Robot mapping for rescue robots. In Proc. of the IEEE Int. Workshop on Safety, Security and Rescue Robotics (SSRR), Gaithersburg, MD.

Ayyagari, V.R., Boughorbel, F., Koschan, A., & Abidi, M.A. (2005). A new method for automatic 3d face registration. In CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, Washington, DC, p. 119. New York: IEEE Computer Society.

Besl, P., & McKay, N. (1992). A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), 239–256.

Biber, P., & Strasser, W. (2003). The normal distributions transform: A new approach to laser scan matching. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV.

Biber, P., & Strasser, W. (2006). nScan-matching: simultaneous matching of multiple scans and application to slam. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Orlando, FL, pp. 2270–2276.

Birk, A. (1996). Learning geometric concepts with an evolutionary algorithm. In Proceedings of The Fifth Annual Conference on Evolutionary Programming. Cambridge, MA: The MIT Press.

Birk, A., & Carpin, S. (2006). Merging occupancy grid maps from multiple robots. Proceedings of the IEEE, 94, 1384–1397.

Boughorbel, F., Koschan, A., Abidi, B., & Abidi, M. (2004). Gaussian fields: a new criterion for 3d rigid registration. Pattern Recognition, 37, 1567–1571.

Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. Image Vision Comput., 10(3), 145–155.

Cohen, J.D., Lin, M.C., Manocha, D., & Ponamgi, M.K. (1995). I-COLLIDE: An interactive and exact collision detection system for large-scale environments. In Symposium on Interactive 3D Graphics, pp. 189–196, 218.

Eggert, D.W., Fitzgibbon, A.W., & Fisher, R.B. (1998). Simultaneous registration of multiple range views for use in reverse engineering of CAD models. Computer Vision and Image Understanding: CVIU, 69(3), 253–272.

Elgammal, A., Duraiswami, R., & Davis, L.S. (2003). Efficient kernel density estimation using the fast gauss transform with applications to color modeling and tracking. IEEE Trans. Pattern Anal. Mach. Intell., 25(11), 1499–1504.

Eliazar, A., & Parr, R. (2004). DP-SLAM 2.0. In Proceedings of IEEE International Conference on Robotics and Automation, New Orleans, LA.

Frese, U. (2006). Treemap: An o(log n) algorithm for indoor simultaneous localization and mapping. Auton. Robots, 21(2), 103–122.

Frese, U., Larsson, P., & Duckett, T. (2005). A multilevel relaxation algorithm for simultaneous localization and mapping. IEEE Transactions on Robotics and Automation, 21, 196–207.

Greengard, L., & Rokhlin, V. (1987). A fast algorithm for particle simulations. Journal of Computational Physics, 73, 325–348.

Greengard, L., & Strain, J. (1991). The fast gauss transform. SIAM J. Sci. Stat. Comput., 12(1), 79–94.

Grisetti, G., Stachniss, C., & Burgard, W. (2005). Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In Proceedings of the IEEE Int. Conference on Robotics & Automation (ICRA), pp. 2443–2448.

Jacoff, A., Messina, E., & Weiss, B. (2003). Evolution of a performance metric for urban search and rescue robots. In Proc. of the Performance Metrics for Intelligent Systems (PerMIS) Workshop, Gaithersburg, MD.

Jalba, A.C., Wilkinson, M.H., & Roerdink, J.B. (2004). CPM: A deformable model for shape recovery and segmentation based on charged particles. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(10), 1320–1335.

Konolige, K. (2003). Map merging for distributed robot navigation. In Proceedings of the 2003 IEEE International Conference on Intelligent Robots and Systems, pp. 212–217, Las Vegas, NV.

Kuipers, B.J. (2000). The spatial semantic hierarchy. Artificial Intelligence, 119, 191–233. http://www.cs.utexas.edu/users/qr/papers/Kuipers-aij-00.html

Lakaemper, R., & Latecki, L.J. (2006). Decomposition of 3D laser range data using planar patches. In IEEE Int. Conf. on Robotics and Automation (ICRA), New Orleans, LA.

Latecki, L.J. & Lakaemper, R. (2006). Polygonal approximation of laser range data based on perceptual grouping and EM. In IEEE Int. Conf. on Robotics and Automation (ICRA), Orlando, FL.

Lu, F., & Milios, E. (1997a). Globally consistent range scan alignment for environment mapping. Auton. Robots, 4(4), 333–349.

Lu, F., & Milios, E. (1997b). Robot pose estimation in unknown environments by matching 2D range scans. Journal of Intelligent and Robotic Systems, Vol. 18(3), 249–275.

Minguez, J., Montesano, L., & Lamiraux, F. (2006). Metric-based iterative closest point scan matching for sensor displacement estimation. IEEE Transactions on Robotics, Vol. 22(5), 1047–1054.

Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. In 18th National Conference on Artificial Intelligence, Menlo Park, CA: American Association for Artificial Intelligence, pp. 593–598.

Olson, E., Leonard, J., & Teller, S. (2006). Fast iterative optimization of pose graphs with poor initial estimates. In Proceedings of ICRA, pp. 2262–2269.

Paragios, N., Rousson, M., & Ramesh, V. (2003). Non-rigid registration using distance functions. Computer Vision and Image Understanding, 89, 142–165.

Robertson, C., & Fisher, R. (2002). Parallel evolutionary registration of range data. Computer Vision and Image Understanding, 87, 39–50.

Rusinkiewicz, S., Brown, B., & Kazhdan, M. (2005). 3D scan matching and registration, ICCV Short Course.

Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. In Proceedings of the Third International Conference on 3D Digital Imaging and Modeling, pp. 145–152.

Thrun, S., Koller, D., Ghahramani, Z., Durrant-Whyte, H., & Ng, A. (2002). Simultaneous mapping and localization with sparse extended information filters. In J.-D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson (Eds.), Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics, Nice, France.

Veltkamp, R., & Hagedoorn, M. (1999). State-of-the-art in shape matching. Technical Report UU-CS-1999-27, Utrecht University, The Netherlands.

Wertheimer, M. (1958). Principles of perceptual organization, In D. Beardslee and M. Wertheimer (Eds.), Readings in Perception. Princeton, NJ: Princeton University Press.

Xu, C., & Prince, J. (1998). Snakes, shapes, and gradient vector flow. IEEE Transactions on Image Processing, Vol. 7(3), 359–369.

Yang, R., Mirmehdi, M., & Xie, X. (2006). A charged active contour based on electrostatics. In Advanced Concepts for Intelligent Vision Systems, ACIVS, Lecture Notes in Computer Science, 4179, 173–184.