ELSEVIER

Robotics and Autonomous Systems 34 (2001) 131–143

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Multisensor on-the-fly localization: Precision and reliability for applications

Kai O. Arras*, Nicola Tomatis, Björn T. Jensen, Roland Siegwart

Autonomous Systems Lab, Swiss Federal Institute of Technology Lausanne (EPFL), CH-1015 Lausanne, Switzerland

Abstract

This paper presents an approach for localization using geometric features from a 360° laser range finder and a monocular vision system. Its practicability under conditions of continuous localization during motion in real time (referred to as on-the-fly localization) is investigated in large-scale experiments. The features are infinite horizontal lines for the laser and vertical lines for the camera. They are extracted using physically well-grounded models for all sensors and passed to a Kalman filter for fusion and position estimation. Positioning accuracy close to subcentimeter has been achieved with an environment model requiring 30 bytes/m^2. Already with a moderate number of matched features, the vision information was found to further increase this precision, particularly in the orientation. The results were obtained with a fully self-contained system where extensive tests with an overall length of more than 6.4 km and 150,000 localization cycles have been conducted. The final testbed for this localization system was the Computer 2000 event, an annual computer tradeshow in Lausanne, Switzerland, where during 4 days visitors could give high-level navigation commands to the robot via a web interface. This gave us the opportunity to obtain results on long-term reliability and verify the practicability of the approach under application-like conditions. Furthermore, general aspects and limitations of multisensor on-the-fly localization are discussed.

Keywords: Mobile robot localization; On-the-fly localization; Position tracking; Multisensor data fusion; Kalman filtering

1. Introduction

Localization in unmodified environments belongs to the basic skills of a mobile robot. Dead-reckoning based techniques are impracticable since systematic and non-systematic measurement errors grow without bound over time. Therefore, additional techniques are required to compensate these errors by periodically sensing the environment. In many potential applications of mobile robots, the vehicle is operating in structured or semi-structured surroundings. This property can be exploited by modelling these structures as geometric primitives and using them as reliably recognizable features for navigation. As it will be shown in this work, this approach leads to very compact environment descriptions which allow for accurate navigation with the limited computational resources of fully autonomous systems. Furthermore, due to the extraction step, which is essentially an abstraction from the type and amount of raw data, information from sensors of any kind can directly be integrated in the same way, leading to versatile and easily extensible environment representations for navigation.

In this paper we take advantage of this representation by simultaneously employing geometric features from different sensors with complementary properties. We consider localization by means of infinite lines extracted from range data of a 360° laser scanner and vertical lines extracted from images of an embarked CCD camera. An extended Kalman filter (EKF) is used for fusion and position estimation.

* Corresponding author.
E-mail address: kai-oliver.arras@epfl.ch (K.O. Arras).

0921-8890/01/$ – see front matter
PII: S0921-8890(00)00117-2

Navigation in a step-by-step manner where localization is performed only at standstill is unsatisfactory for several reasons: The vehicle advances slowly and has a non-continuous movement which is important in certain applications like cleaning tasks. Also, the position update rate is typically low with respect to the distance travelled and on a aesthetical level the robot behaves 'unnaturally'. Therefore, continuous localization during motion in real-time — henceforth referred to as *on-the-fly localization* — is desirable but confronts the researcher with difficulties which are present but only hidden at low speeds or step-by-step navigation. This includes resolution and uncertainties of time stamps the system can provide for sensory inputs and the need for compensating the distortion of sensory data caused by the vehicle movement. Time stamp quality imposes bounds on localization precision and feature matching rates whose influence is to be studied when a localization method shall prove its relevance for real-world applications.

Kalman filter localization with line segments from range data has been done early [9,13]. Vertical edges in combination with an EKF have been employed in [6,15]. The combination of these features is used in [16,18]. In [16], a laser sensor with an opening angle of 60° providing both range and intensity images was utilized, and in a recent work [18], the absolute localization accuracy of laser, monocular and trinocular vision was examined. Similar precision has been found for the three cases.

The Kalman filter acts as a position tracker whose performance is dependent on how fast per time or distance travelled the true probability distributions deviate from the idealized Gaussians. Since only a single distribution is maintained, an incorrect matching of the local map onto the global map can lead to irreversible filter divergence. In such a case, the robot has to be globally re-localized. Feature-based global localization methods are for example [5,20]. On the other hand, if the tracker operates fast enough and obtains precise and discriminant sensory information, it can reliably keep the robot localized because the deviation from the idealized statistical assumptions remains bounded to an uncritical extent.

Other probabilistic localization techniques maintain a probability distribution over the space of possible robot positions in the map. They represent the Markov localization paradigm and have been used with purely topological maps and landmarks [17], topologic–metric maps [21], and grid-based maps with raw range data [10]. Since a distribution over all possible robot poses is always maintained, these approaches are inherently global and can re-localize the robot after a lost situation or at start-up. For the same reason, however, they need incomparable more computational resources in terms of memory and calculation power if a localization accuracy is required similar to the one in this work (see e.g. [11] for an experimental comparison).

Raw range data are employed also in [8] and matched against an a priori map composed of line segments by minimizing a least-square criterion. Similar methods are used for scan matching where the best alignment of the local and the global map is sought, both maps being a set of sensed points [12,14]. The robot pose is also represented as a single Gaussian which makes scan matching a local method, and a simplified Kalman filter is used to calculate the posterior state. Compared to feature-based approaches, scan matching techniques usually operate with memory-intensive maps since the environment model consists in raw range data recorded from a set of reference positions.

The contribution of this paper is threefold. First, we present the multisensor setup, the features with their extraction and error models. We conduct experiments to determine the localization precision which is attainable with this approach. Second, in contrast to most of the contributions in the domain of mobile robot localization, this paper presents results from extensive experiments where practicability is verified under conditions which do not differ from those of typical applications. We further explore the feature-based paradigm and, ultimately, try to answer the question whether a robot localization system can be realized as a position tracker for its integration into a navigation framework, ready for its 'blind use' as a black-box. Finally, the paper introduces infinite lines as features, opposed to range-only information in [13], and segments of finite length in [5,9,16,18].

In a first set of experiments under controlled conditions, we examine the improvement in precision when the vision information is added to the laser range finder. Alike [16] the posterior uncertainty bounds are considered which, under the assumption of realistic uncertainty models, permit inference about the

first moments of the robot position. For this, throughout this work, it was attempted to employ physically well-grounded uncertainty models for odometry, the laser range finder and the vision system. The second experiment is the Computer 2000 event, an annual computer tradeshow in Lausanne, where during 4 days visitors could give high-level navigation commands to the robot via a web interface. With a system up-time of 28 h, an overall travel distance of 5 km and more than 140 000 localization cycles, long-term reliability under application-like conditions was the main concern.

2. Sensor modelling

2.1. Odometry

Non-systematic odometry errors occur in two spaces: the joint space and the Cartesian space. With a differential drive kinematics the joint space is two-dimensional and includes the left and right wheel. Effects of wheel slippage, uneven ground and limited encoder resolution appear in this space. In [7] a physically well-grounded model for these kind of errors is presented starting from the uncertain input $u(k+1) = [\Delta d_L, \Delta d_R]^T$ with $\Delta d_L, \Delta d_R$ as the distances travelled by each wheel, and the diagonal input covariance matrix

$$U(k+1) = \begin{bmatrix} k_L|\Delta d_L| & 0 \\ 0 & k_R|\Delta d_R| \end{bmatrix}, \quad (1)$$

which relies on the assumption of proportionally growing variances per Δd_L, Δd_R travelled. The odometry model for the first and second moment of the state vector $x = (x, y, \theta)^T$ is then

$$\hat{x}(k+1|k) = f(x(k|k), u(k+1)), \quad (2)$$

$$P(k+1|k) = \nabla f_x P(k|k) \nabla f_x^T + \nabla f_u U(k+1) \nabla f_u^T, \quad (3)$$

where $f(\cdot)$ uses a piecewise linear approximation, $P(k|k)$ is the state covariance matrix of the last step, and ∇f_x and ∇f_u are the Jacobians of $f(\cdot)$ with respect to the uncertain vectors $\hat{x}(k|k)$ and $u(k+1)$, k_L and k_R are constants with unit meter.

The Cartesian space is spanned by the vector x encoding position and orientation of the vehicle. Effects of external forces (mainly collisions) occur in this space. Non-systematic Cartesian errors could be additionally modelled in Eq. (3) by a 3×3 covariance matrix $Q(k+1)$ being a function of the robot displacement Δx, $\Delta\theta$ in the robot frame. Such a model has been used in [6]. In any case it is difficult to identify these models, i.e. to obtain rigorous values for k_L, k_R and the coefficients in $Q(k+1)$ which are valid for a range of floor types. In this work we use only the joint space model with $k_L = k_R = 5 \times 10^{-6}$ m which have been identified by observation.

2.2. Laser range finder

The laser range finder which was used in the experiments is the Acuity AccuRange 4000 LIR. The rotation frequency of the mirror is 2.78 Hz, yielding a 1° angular resolution with its maximal sampling frequency in calibrated mode of 1 kHz. It delivers range ρ and intensity i as analogue signals. The latter is the signal strength of the reflected beam and predominantly affects range variance. In order to have a good uncertainty model of range variability accounting not only for the distance to the target but also for its surface properties, a relationship $\sigma_\rho = f(i)$ is sought. Identification experiments with a Kodak gray-scale patch performed in [2] yielded a simple relationship describable by two parameters: i_{min} allows to reject too uncertain range readings with $i < i_{min}$ and for measurement with $i > i_{min}$ a constant value for range variance $\sigma^2_{\rho_{const}}$ could have been found.

2.3. Camera

The vision system consists of a Pulnix TM-9701 full-frame, EIA (640×480), gray-scale camera with a 90° objective and a Bt848 based frame grabber which delivers the images directly to the main CPU memory. There is no dedicated hardware for image processing.

The camera system is calibrated by combining method [19] with spatial knowledge from a test field. This provides a coherent set of extrinsic, intrinsic and distortion parameters. Since the visual features are vertical lines, only horizontal calibration

is needed, yielding the simplified model of Eq. (4) for parameter-fitting:

$$C\frac{y_r - \beta(x_r - O_x)}{\beta y_r + (x_r - O_x)} = S[x_c + x_c(k_1 r^2 + k_2 r^4 + k_3 r^6 + k_4 r^8)], \quad (4)$$

(x_r, y_r, z_r) is the position of a point in the robot frame, $x_c = x - H_x$, $y_c = y - H_y$, and $r^2 = x_c^2 + y_c^2$, where the coordinates (x, y) refer to the distorted location of the point in the uncorrected image. Focal length C, scale factor S and image center (H_x, H_y) are instrinsic parameters, β and O_x are extrinsic parameters defining the robot to sensor transformation and k_1, k_2, k_3 and k_4 are the parameters of radial distortion.

Uncertainties from the test field geometry and those caused by noise in the camera and acquisition electronics are propagated through the camera calibration procedure onto the level of camera parameters, yielding a 10×10 parameter covariance matrix.

3. Feature representation and extraction

3.1. Laser range finder

The algorithm for line extraction has been described in [1]. The method delivers lines and segments with their first order covariance estimate using polar coordinates. The line model is

$$\rho \cos(\varphi - \alpha) - r = 0, \quad (5)$$

where (ρ, φ) is the raw measurement and (α, r) the model parameters, α the angle of the perpendicular to the line, and r its length. The method differs from the widely used recursive split-and-merge technique which is also applied in [9,18] in the segmentation criterion: Instead of using a line specific decision on a single point, it decides on a model independent criterion on a group of points. Multiple segments which lie on the same physical object are merged for particular precise re-estimates of the line position. This is realized by a clustering algorithm with a Mahalanobis distance matrix. It merges lines until their distance in the (α, r)-model space is greater than a threshold from a χ^2-distribution. Fig. 1 shows an extraction example where six lines have been found.

3.2. Camera

Vertical lines are extracted in four steps:

- Vertical edge enhancement: Specialized Sobel filter approximating the image gradient in the horizontal direction.
- Non-maxima suppression with dynamic thresholding: The most relevant edge pixels (maximal gradi-

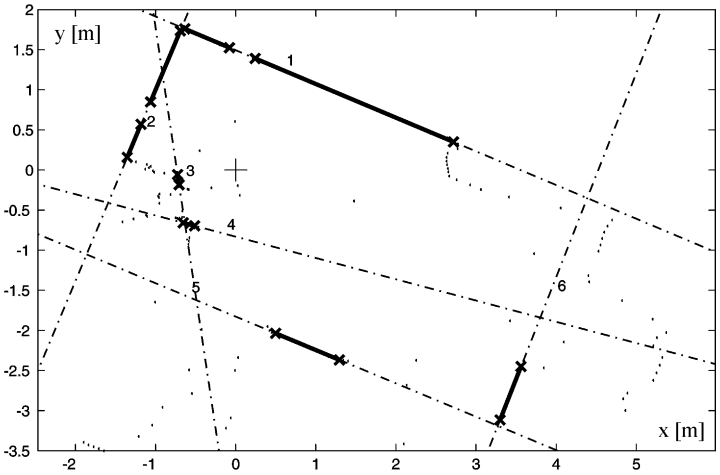


Fig. 1. A scan of the Acuity sensor and the extraction result. Eight segments on six lines have been found. Two closely situated objects produced evidence for the two outlier segments. Thus, the local map contains six (α, r)-pairs which are passed to the EKF matching step.

ent) are extracted and thinned by using a standard method.

- Edge image calibration: The horizontal position of each edge pixel is corrected yielding a new position $\bar{x}$ with

$$\bar{x} = S[x_c + x_c(k_1 r^2 + k_2 r^4 + k_3 r^6 + k_4 r^8)], \quad (6)$$

resulting from the camera model.

- Line fitting: Columns with a predefined number of edge pixels are labelled as vertical lines. Line fitting reduces to a one-dimensional problem. The resulting angle is $\varphi = \text{atan}(x/C)$, where C is the focal length and x the weighted mean of the position of the pixels in the extracted line.

Uncertainty from the camera electronics is modelled on the level of the uncalibrated edge image. Together with the uncertainty of the calibration parameters, it is propagated through calibration and line fit, yielding the first two moments $(\varphi, \sigma_\varphi^2)$ of the vertical edges.

3.3. Map

The a priori map (Fig. 2) contains 191 infinite lines and 172 vertical edges for the 50 m × 30 m portion of the institute building shown in Fig. 3. This is an environment model of extreme compactness with a memory requirement of about 30 bytes/m^2.

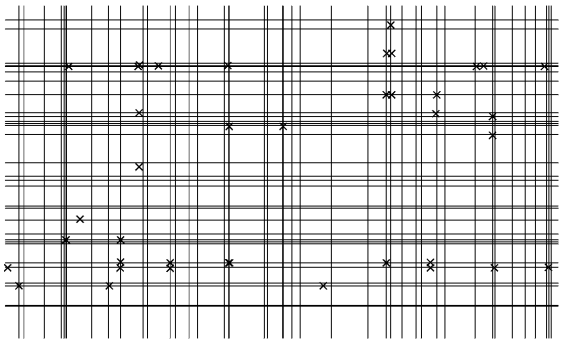


Fig. 2. A 12 m × 6 m section of the environment model. Crosses indicate vertical edges. The complete map consisting of 191 infinite lines and 172 vertical edges describes 12 offices, two corridors, the seminar and the mail room.

4. Multisensor EKF on-the-fly localization

After a brief summary of a Kalman filter localization cycle, aspects particular for this work are considered in more detail.

A localization cycle consists of five steps [3,13]:

State prediction. The state $\hat{x}(k+1|k)$ and its associated covariance $P(k+1|k)$ is determined from odometry based on the previous state moments $\hat{x}(k|k)$ and $P(k|k)$. This has been described in Section 2.1.

Observation. The feature parameters $(\alpha\, r)$ of lines and φ of vertical edges constitute the vector of observation $z(k+1)$. Their associated covariance estimates constitute the observation covariance matrix $R(k+1)$. Since measurement errors of sensors and features are independent, $R(k+1)$ is blockwise diagonal. This means that all subsequent equations operate with 2×2-matrices for lines and scalars for vertical edges.

Measurement prediction. The modelled features in the map, ${}^{W}m$, get transformed into the frame of the observations. The first moments are computed by $\hat{z}(k+1) = h(\hat{x}(k+1|k),\ {}^{W}m)$, where $h(\cdot)$ is the non-linear measurement model (the global-to-local transform). Error propagation is done by a first-order approximation which requires the Jacobian ∇h with respect to the state prediction $\hat{x}(k+1|k)$.

Matching. Since the Kalman filter represents and propagates a single Gaussian distribution for the robot pose, false pairings can lead to irreversible filter divergence, i.e. a lost situation, which requires manual intervention. This step is therefore of high importance and is explained below in more detail.

Estimation. Successfully matched observation and predictions yield the innovations

$$\upsilon(k+1) = z(k+1) - \hat{z}(k+1)$$

and their innovation covariance

$$S(k+1) = \nabla h\, P(k+1|k) \nabla h^{T} + R(k+1).$$

Finally, with the filter equations

$$W(k+1) = P(k+1|k) \nabla h^{T} S^{-1}(k+1), \quad (7)$$

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)\upsilon(k+1), \quad (8)$$

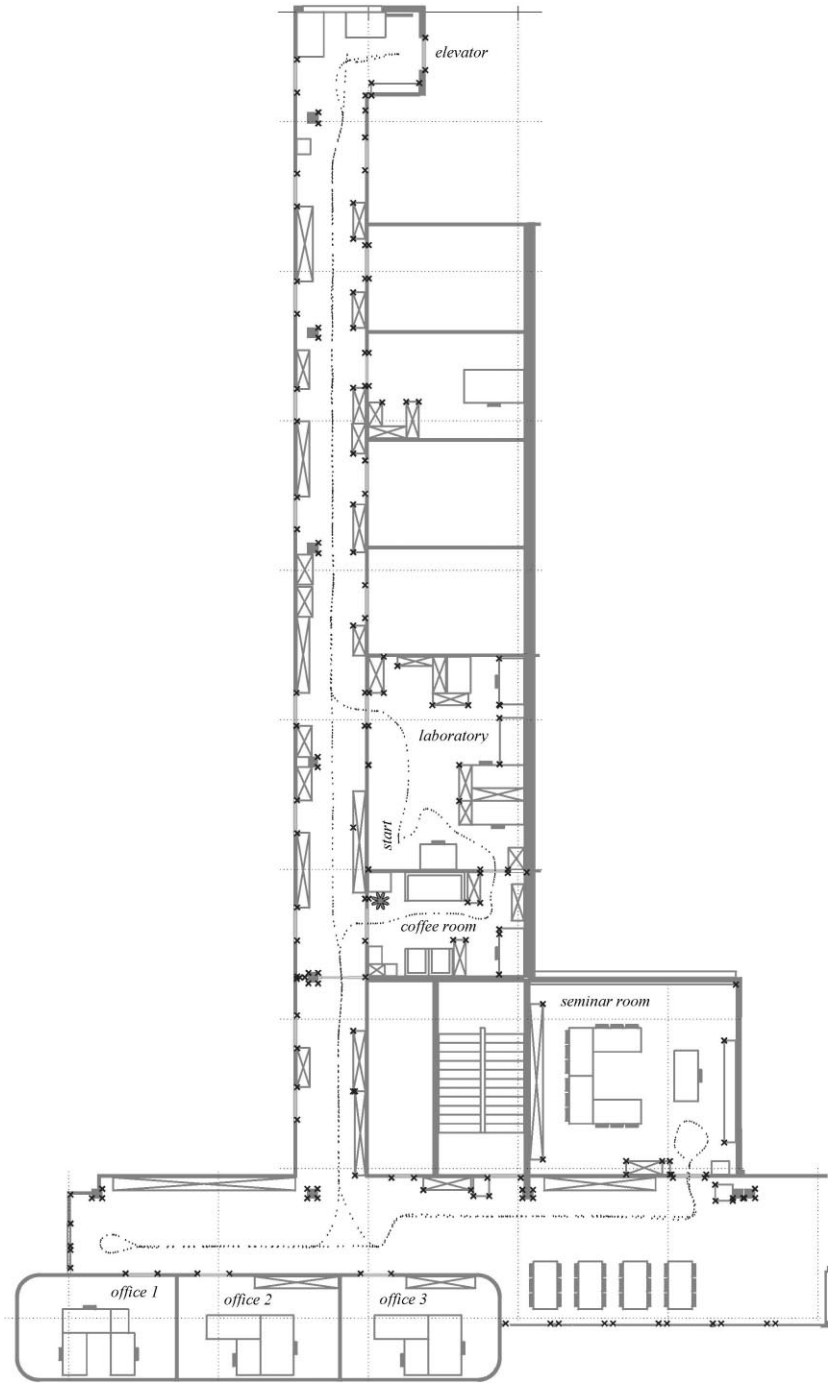


Fig. 3. Floor plan of the 50 m × 30 m environment where the experiments have been conducted. One of the test trajectories is shown with a point at each location where the robot localized itself. Crosses indicate the modelled vertical edges. The robot starts in the laboratory, goes to the elevator, then passes through the corridor to offices 1, 2 and 3, continues to the seminar room and returns to the laboratory via the coffee room. The trajectory length is 140 m and has been driven 10 times with about 950 localization cycles per run. The average speed was 0.3 m/s, maximal speed 0.6 m/s, resulting in about 7′45″ for the whole path. In order to compare the multisensor setup and the laser-only setup with respect to localization accuracy, five runs have been made with laser-only, five with laser and vision. The resulting uncertainty of the posterior position estimates are illustrated in Fig. 5, overall averages are given in Table 1.

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W^{\mathrm{T}}(k+1), \quad (9)$$

the posterior estimates of the robot pose and associated covariance are computed.

4.1. Matching

Since the observation covariance matrix $R(k+1)$ is blockwise diagonal we have the freedom to integrate matched pairings in a manner which is advantageous for filter convergence:

The laser observations are integrated first since they typically exhibit far better mutual discriminance making their matching less error-prone. They are followed by the vertical edges from the camera, where often ambiguous matching situations occur. Starting from the same idea, each pairing is integrated according to its quality in an iterative procedure for each sensor: (i) matching of the current best pairing, (ii) estimation and (iii) re-prediction of features not associated so far. This procedure has also been used in [16,18] where the same observations concerning feature discriminance have been reported.

The quality of a pairing of prediction $\hat{z}_{\mathrm{l,v}}^{[i]}$ and observation $z_{\mathrm{l,v}}^{[j]}$ is different for both sensors:

- For the lines the quality criterion of a pairing is *smallest observational uncertainty* — not smallest Mahalanobis distance like in [16,18]. This renders the matching robust against small spurious and uncertain segments which have small Mahalanobis distances (see Fig. 1). The 'current best' pairing $p_{ij} = (z_{\mathrm{l}}^{[j]}, \hat{z}_{\mathrm{l}}^{[i]})$ is therefore that of observation $z_{\mathrm{l}}^{[j]}$ with $\mathrm{trace}(R_{\mathrm{l}}^{[j]}) = \min_i$ which satisfies the validation test

$$(z_{\mathrm{l}}^{[j]} - \hat{z}_{\mathrm{l}}^{[i]}) S_{ij}^{-1} (z_{\mathrm{l}}^{[j]} - \hat{z}_{\mathrm{l}}^{[i]})^{\mathrm{T}} \leqslant \chi_{\alpha,n}^2, \quad (10)$$

  where S_{ij} is the innovation covariance matrix of the pairing, and $\chi_{\alpha,n}^2$ a number taken from a χ^2-distribution with $n = 2$ degrees of freedom and α the probability level on which the hypothesis of pairing correctness is rejected.
- The criterion for vertical edges is uniqueness. Predictions $\hat{z}_{\mathrm{v}}^{[i]}$ with a single observation $z_{\mathrm{v}}^{[j]}$ in their validation gate are preferred and integrated according to their smallest Mahalanobis distance provided that they satisfy Eq. (10) with $n = 1$ (subscripts l become v). When there is no unique pairing anymore, candidates with multiple observations in the validation region or observations in multiple validation regions are accepted and chosen according to the smallest Mahalanobis distance.

4.2. Time stamps

The main difference from the viewpoint of multisensor localization between step-by-step and on-the-fly navigation is that temporal relations of observations, predictions and estimations of all involved sensors have to be maintained and related to the present. This is done by assigning time stamps to all sensory inputs. Time stamps are delivered by odometry which continuously updates robot position and uncertainty according to Eqs. (2) and (3). Each update receives a time stamp and is written to a circular buffer. When sensor A performs its data acquisition, the data receive a time stamp T_{A} and, after feature extraction is completed, the corresponding state prediction is read out from the odometry buffer. When the position estimate arrives from the Kalman filter, it is valid at time stamp T_{A}, which is now in the past. Based on the odometry model, a means is then needed to relate this old position estimate to the current position of time t. This is done by forward simulation of Eqs. (2) and (3) from T_{A} to t.

For a multisensor system, care has to be taken that prediction and estimation results of one sensor are not overwritten by those of another sensor. This would be the case if each sensor would have its own EKF running independently from the others with its own cycle time, yielding temporally nested updates. Nested updates are unfavourable since a slow outer update cycle (e.g. vision) overwrites the estimation results of faster running inner cycles (e.g. laser). A sequential scheme of EKFs for each sensor is therefore required with the constraint that the estimates get integrated in exactly the succession of their respective observation.

4.3. Scan compensation

The vehicle movement imposes a distortion on the raw data of the laser scans. This deformity depends

on the ratio of robot speed to mirror rotation velocity which in our case is non-negligible (mirror speed is 2.78 Hz). It is important to note that scans have to be compensated on the raw data level and not on the level of extracted features. Since in the latter case the extraction method would operate with an artificially modified feature evidence.

We compensate for the vehicle displacement during a scan by transforming each range reading ${}^{S'}P$ acquired in the sensor frame S′ into the non-stationary robot frame R′ and then into the world frame W, followed by a re-transform into the stationary robot frame R and finally into the desired reference frame of the scan S. For a compensation on-the-fly, S must be the sensor frame at the start position of a new scan. By reading out odometry each time when a new range reading arrives, it gets immediately transformed by the expression

$$ {}^{S}P = {}^{R}_{S}T^{-1}\,{}^{W}_{R}T^{-1}\,{}^{W}_{R'}T\,{}^{R'}_{S'}T\,{}^{S'}P, \tag{11} $$

where ${}^{R}_{S}T = {}^{R'}_{S'}T$. The 4×4 matrices T are homogeneous transforms casting the rotation and translation of the general transform into a single matrix. ${}^{R}_{S}T$ is the sensor-to-robot frame transform and ${}^{W}_{R}T$ the world-to-robot transform given by the actual robot pose vector x. The compensated scan receives the time stamp of S, i.e. the time when the scan has started recording.

5. Implementation and experiments

5.1. The robot

Our experimental platform is the robot Pygmalion which has been built in our lab (Fig. 4). Its design principles are oriented towards an application as service or personal robot. Long-term autonomy, safety, extensibility and friendly appearance were the main objectives for design. With its dimensions of 45 cm × 45 cm × 78 cm and its weight of 55 kg, it is of moderate size and danger opposed to many robots in its performance class. The robot is operating autonomously without any off-board infrastructure: sensor data acquisition, feature extraction, localization, global and local path planning, obstacle avoidance, security monitoring and the web-server are all running on the VME card. The code, the environment map and our hard real-time operating system XO/2 [4] fit easily into 32 MB of RAM. No hard disk is required.

Fig. 4. Pygmalion, the robot which was used in the experiments. It is a VME based system carrying currently a PowerPC card at 300 MHz with 32 MB of RAM. Besides wheel encoders and bumpers, the sensory system includes a 360° laser range finder and a gray-level CCD camera discussed in Section 2. During all experiments it ran in a fully autonomous mode.

5.2. Experiments under controlled conditions

In the first set of experiments, the robot was driving under controlled conditions: all runs have been performed in the evening which allowed to limit environment dynamics and to control the illumination conditions. Almost all corridors, offices and rooms in Fig. 3 are subject to direct daylight which otherwise would influence the performance of the vision system. In order to assure trajectories as reproducible as possible, obstacle avoidance has been turned off and replaced by a position controller for robots with non-holonomic kinematics. The floor plan of the environment and the test trajectory are shown in Fig. 3. In the laser-only mode and in the multisensor mode the trajectory has been driven five times. Care has been taken that both experiments had the same localization cycle time by limiting the implementation to 2 Hz resulting in about 950 cycles on the 140 m test trajectory. The average speed was 0.3 m/s, maximal speed 0.6 m/s.

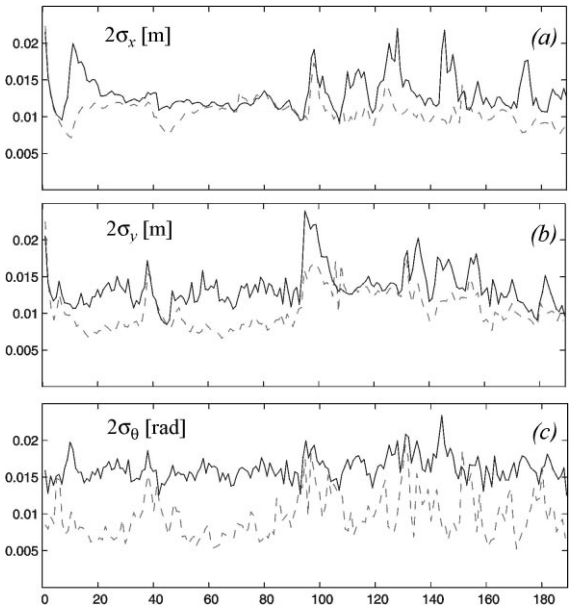


Fig. 5. Averaged 2σ-error bounds of global x (a), y (b), and θ (c) a posteriori uncertainty during the test trajectory (showing only each fifth step). In each mode, five runs have been made. Solid lines: laser range finder only, dashed lines: laser and vision. In some cases the uncertainty in the multisensor mode is greater than for the single-sensor setup. This is possible since the values are averaged over five runs containing noise on the level of matched features.

5.2.1. Results

The resulting 2σ-uncertainty bounds of the a posteriori position estimates are illustrated in Fig. 5. For both cases they generally reflect a very high localization accuracy in all the three state variables. Subcentimeter precision is approached. Table 1 shows the overall means of error bounds $2\bar{\sigma}_x, 2\bar{\sigma}_y, 2\bar{\sigma}_\theta$, number of matched lines $\bar{n}_\mathrm{l}$, number of matched vertical edges $\bar{n}_\mathrm{v}$, and execution times $\overline{t_\mathrm{exe}}$. The error bounds say that, based on the uncertainty model, the robot is with a 95% probability within twice this value. The vision information contributes to a reduction of this uncertainty in x and y equally (-20%), but particularly in the orientation (-40%), although the average number of matched vertical edges is moderate. The cycle times $\overline{t_\mathrm{exe}}$ include sensor acquisition and the mean duration for calculating an iteration under full CPU load on the currently used hardware.

Table 1
Overall mean values of the error bounds, the number of matched line segments n_l and matched vertical edges n_v per cycle, and the average localization cycle time $\overline{t_\mathrm{exe}}$ under full CPU load

	Laser	Laser and vision
$2\bar{\sigma}_x$ (cm)	1.31	1.07
$2\bar{\sigma}_y$ (cm)	1.35	1.05
$2\bar{\sigma}_\theta$	0.92°	0.56°
$\bar{n}_\mathrm{l}/\bar{n}_\mathrm{v}$	2.73/–	2.66/2.00
$\overline{t_\mathrm{exe}}$ (ms)	64	411

5.2.2. Discussion

Even carefully derived uncertainty bounds do not necessarily permit inference about the sought first moments, since the estimation error could be arbitrarily big without being noticed (estimator inconsistency). We argue that the simple fact that the robot always succeeded in returning to its start point is compelling evidence for the correctness of these bounds. In fact, they are even conservative since the true bounds could be better. Otherwise the robot would have gone lost due to a lack of matches caused by first moments drifted away from the true values. Ground truth information like in [18] would be preferable but is impractical and expensive to obtain for experiments of this kind and extent. Positioning accuracy of the vehicle in the endpoint has been measured and further confirm the values in Table 1. These results are very similar to the accuracy reported in [18]. In [11] the experiments yielded a maximal precision of about 5 cm for Markov localization, whereas scan matching produced, in the best case, estimates of millimeter accuracy.

Matching vertical edges is, due to their lack of depth information and their frequent appearance in compact groups, particularly error-prone. For example, door frames commonly have multiple vertical borders which, dependent on the illumination conditions, produce evidence for several closely situated vertical edges. In the matching stage, they might be confronted with a large validation region, position bias from odometry or time stamp uncertainty, making the predicted model edge difficult to identify. In such ambiguous matching situations, incorrect pairings are likely to occur and, in fact, they have been occasionally produced in the multisensor experiments. But their effect remains weak since these groups are typically very compact.

However, this lack of discriminance in the presence of time stamp uncertainty is the main cause of

reproducible failure of vision-only navigation (i.e. robot went lost). With the frame grabber in use, it is difficult to identify the precise instant when the image is taken. Also limited time stamp resolution, in our case 5 ms given by odometry, became noticeable particularly during fast turns (the camera of Pygmalion is not mounted on a turret which keeps a constant orientation). This furthermore makes the matching more difficult since the limited temporal accuracy of the state prediction causes a reduced metric accuracy of the measurement predictions.

5.3. Experiments under controlled conditions: The Computer 2000 event

The 'Computer' tradeshow is an annual fair for computer hard- and software at the Palais de Beaulieu exposition center in Lausanne, Switzerland. Our laboratory (Fig. 6) was present during the 4 days, 2–5 May 2000, giving visitors the opportunity to control Pygmalion by means of the web-interface shown in Fig. 7. The robot itself was at EPFL, in the environment illustrated in Figs. 3 and 7.

The Computer 2000 event was the final testbed for our localization system where we were mainly interested in long-term reliability under application-like conditions. The setup was active during normal office hours with an average of about 7 h system up-time per day. The environment exhibited typical dynamics from people, doors, chairs and other robots, as well as daylight illumination. Several doors open into the corridor (see Fig. 6, right image). Travel speed has been limited to 0.4 m/s since the robot shares its environment with persons, some of them not implied into robotics. An obstacle avoidance strategy based on the laser range data was active during the event. The web-interface allows to give global navigation commands (e.g. an office) and local navigation commands (a (x, y)-point or an orientation) to the robot.

Fig. 6. During the 4 days of the Computer 2000 event, visitors could teleoperate Pygmalion in the corridors and offices of our institute building at EPFL. The environment which was attainable to the robot is 50 m × 30 m in size and contains 12 offices, two corridors, the seminar and the mail room. The robot was navigating 7 h each day during normal office hours with typical environment dynamics.

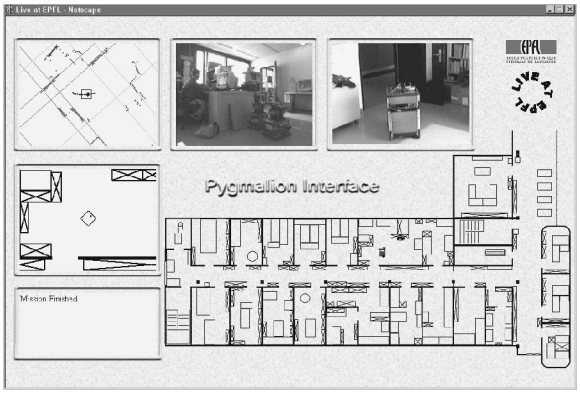


Fig. 7. The Pygmalion web-interface, a plug-in-free Netscape application. It provides context-sensitive menus on the map and all subwindows with intuitive click-and-move-there commands for robot teleoperation. Four different real-time streams constitute the visual feedback on the current robot state: an external web-cam (top-right), an embarked camera (top-middle), raw data from the laser range finder together with predicted, observed and matched features (top-left) and the robot animated in its model map (left-middle). By clicking onto the map an office is defined as destination, clicking onto the camera image turns the robot and clicking on the laser scanner image defines a goal in the (x, y)-plane.

5.3.1. Results

The event statistics of Computer 2000 is shown in Table 2. A mission is either a global or a local navigation command from the user interface. A lost situation is defined as a mission whose goal could not be attained due to a localization error and which required manual intervention to re-localize the robot. We do not count missions where the robot went lost due to a collision with an invisible object (e.g. glass door or object

Table 2
Overall statistics for the Computer 2000 event

Hours of operation	28
Environment size	50 m × 30 m
Environment type	Office, unmodified
Overall travel distance	5013 m
Average speed	0.2 m/s
Travel speed	0.4 m/s
Number of missions	724
Number of localization cycles	145,433
Number of lost situations	0
Number of unknown collisions	∼10

lower than the beam height of the scanner) and where the robot was already lost (after such a collision).

It happens that there are no matched features during a certain period, when the vehicle is surrounded by people, is in less structured terrain or when odometry delivers inconsistent position estimates (e.g. from driving over uneven floors). We counted 14 of 724 missions where the robot had no matches during 10 s, 21 missions where it had no matches during 5 s. None of them required manual intervention during or after the mission.

5.3.2. Discussion

These positive results were further underlined by the feedback we got from the big number of visitors during Computer 2000. In particular, they enjoyed the easy-to-use interface which allowed anybody to control a mobile robot and discover our institute building over the internet.

Another important result of the Computer 2000 event is that under these experimental conditions, vertical lines performed poorly. In addition to the shortcomings already stated in the first set of experiments, they are:

- *Environment dynamics*: When navigating with a robot in a populated office environment, its free space and hence its vision sensor are often blocked by obstacles (e.g. co-workers, doors; see Fig. 6). This results in two independent problems. First, a blocked sensor cannot contribute to the localization update. Second, when avoiding these obstacles the robot turns typically with high angular velocities which in combination with the time stamp uncertainties and the low feature discriminance is likely to produce false matches.
- *Illumination conditions*: It is obvious that the illumination in an office environment cannot be controlled to suit best the need of a vision sensor. In particular both corridors have big windows at their end. The camera is heading to these windows from a relatively dark corridor when navigating in this direction.

This leads us to the conclusion that, in our setup, vertical lines are features of insufficient robustness with respect to environment dynamics, time stamp uncertainty, changing illumination conditions and mutual discriminance. In general, a trade-off between the robustness of a feature and the computational effort to obtain it has to be found. This in turn influences how many features can be used for localization per time or distance travelled. Vertical edges appeal by their simple availability and the little requirements of computational power compared to many other visual features.

We finally emphasize the necessity of these types of experiments. They reflect the ‘real case’ in a one-to-one manner and are therefore indispensable when a navigation approach shall prove its real-world performance. ‘Laboratory experiments’ whose conditions have been carefully controlled only yield optimistic bounds for the practicability of a method. It is clear that these kinds of experiments always remain in a certain dependency upon the specific robot setting. Especially a different vision/frame grabber system or a turret keeping a constant camera orientation could have performed better under the same experimental conditions. However, we rate it as not surprising that during this work an approach which in the beginning seems promising, and which has been successively applied by several researchers before, turns out to be partially incompatible with the requirements of application-like conditions.

6. Conclusions

In this paper an approach for localization using geometric features from a 360° laser range finder and a monocular vision system has been presented. The features allow for an extremely compact environment model of only 30 bytes/m^2 memory requirement. This is contrasted by a positioning accuracy close to sub-

centimeter and small localization cycle times. These results were obtained with a fully self-contained system where long-term tests with an overall length of more than 6.4 km and 150,000 localization cycles have been conducted.

Already with a moderate number of matched features, the vision information was found to further increase this precision, particularly in the orientation. However, the limitations encountered with this feature motivate the use of constraint-based matching strategies as utilized for feature-based global localization [5,20]. In spite of the reliability of the presented position tracking approach, collisions can never be excluded completely, especially when the robot is employed, e.g. in public places. A method to globally relocalize the vehicle after collisions and its integration in real-time into a feature-based localization system is the subject of on-going work [20].

The experiments show finally that the feature-based approach for localization with infinite lines does an excellent job concerning reliability, efficiency and precision and marks the closure of our efforts in local, metric mobile robot localization. With this work, the requirement claimed in the beginning, that localization should be an easy-to-use black-box within a navigation framework, could be met.

Acknowledgements

The authors would like to thank Jan Persson and Benoit Moreau for their important contribution to the Computer 2000 event. Also, we thank Illah Nourbakhsh from CMU for many fruitful discussions.

References

[1] K.O. Arras, R.Y. Siegwart, Feature extraction and scene interpretation for map-based navigation and map building, in: Proceedings of the SPIE, Mobile Robotics XII, Vol. 3210, 1997.

[2] K.O. Arras, N. Tomatis, Improving robustness and precision in mobile robot localization by using laser range finding and monocular vision, in: Proceedings of the Third European Workshop on Advanced Mobile Robots, Eurobot'99, Zurich, Switzerland, 1999.

[3] Y. Bar-Shalom, T.E. Fortmann, Tracking and data association, Mathematics in Science and Engineering, Vol. 179, Academic Press, New York, 1988.

[4] R. Brega, N. Tomatis, K.O. Arras, The need for autonomy and real-time in mobile robotics: Case study of Pygmalion and XO/2, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Takamatsu, Japan, 2000.

[5] J.A. Castellanos, J.D. Tardos, J. Neira, Constraint-based mobile robot localization, in: Proceedings of the International Workshop on Advanced Robotics and Intelligent Machines, Salford, UK, 1996.

[6] F. Chenavier, J.L. Crowley, Position estimation for a mobile robot using vision and odometry, in: Proceedings of the IEEE International Conference on Robotics and Automation, Nice, France, 1992.

[7] K.S. Chong, L. Kleeman, Accurate odometry and error modelling for a mobile robot, in: Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, NM, 1997.

[8] I.J. Cox, Blanche — An experiment in guidance and navigation of an autonomous robot vehicle, IEEE Transactions on Robotics and Automation 7 (1991) 193–204.

[9] J.L. Crowley, World modeling and position estimation for a mobile robot using ultrasonic ranging, in: Proceedings of the IEEE International Conference on Robotics and Automation, Scottsdale, AZ, 1989.

[10] D. Fox, W. Burgard, S. Thrun, Markov localization for mobile robots in dynamic environments, Journal of Artificial Intelligence Research 11 (1999) 391–427.

[11] J.-S. Gutmann, W. Burgard, D. Fox, K. Konolige, An experimental comparison of localization methods, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Victoria, BC, 1998.

[12] J.-S. Gutmann, C. Schlegel, Amos: Comparison of scan matching approaches for self-localization in indoor environments, in: Proceedings of the First European Workshop on Advanced Mobile Robots, Eurobot'96, Kaiserslautern, 1996.

[13] J.J. Leonard, H.F. Durrant-Whyte, Directed Sonar Sensing for Mobile Robot Navigation, Kluwer Academic Publishers, Dordrecht, 1992.

[14] F. Lu, E. Milios, Robot pose estimation in unknown environments by matching 2D range scans, in: Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR), 1994.

[15] A.J. Muñoz, J. Gonzales, Two-dimensional landmark-based position estimation from a single image, in: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, 1998.

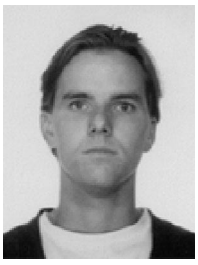
[16] J. Neira, J.D. Tardos, J. Horn, G. Schmidt, Fusing range and intensity images for mobile robot localization, IEEE Transactions on Robotics and Automation 15 (1) (1999) 76–84.

[17] I. Nourbakhsh, R. Powers, S. Birchfield, DERVISH, an office-navigating robot, AI Magazine 16 (2) (1995) 53–60.

[18] J.A. Pérez, J.A. Castellanos, J.M.M. Montiel, J. Neira, J.D. Tardós, Continuous mobile robot localization: Vision vs. laser, in: Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, MI, 1999.

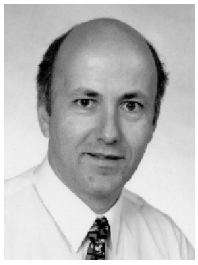
[19] B. Prescott, G.F. McLean, Line-based correction of radial lens distortion, Graphical Models and Image Processing 59 (1) (1997) 39–47.
[20] M. Schilt, Matching techniques for global localization with infinite lines, Project Thesis, Autonomous Systems Lab, EPFL, June 2000.
[21] R. Simmons, S. Koenig, Probabilistic navigation in partially observable environments, in: Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, Que., Vol. 2, 1995, pp. 1660–1667.

Kai Oliver Arras is a Ph.D. student with the Autonomous Systems Lab at the Swiss Federal Institute of Technology, Lausanne (EPFL). He received his Masters in Electrical Engineering from the Swiss Federal Institute of Technology, Zurich (ETHZ) in 1995 and worked as a Research Assistant in Nanorobotics at the Institute of Robotics in Zurich. In 1996 he joined Professor Siegwart to co-establish the Autonomous Systems Lab at EPFL where he is working on several aspects of mobile robotics including hardware design, feature extraction, local and global localization, data association and SLAM.



Nicola Tomatis is a Ph.D. student with the Autonomous Systems Lab at the Swiss Federal Institute of Technology, Lausanne (EPFL). He received his M.S. in Computer Science from the Swiss Federal Institute of Technology, Zurich (ETHZ) in 1998. His research covered mobile robot navigation, computer vision, sensor data fusion, human–robot interaction and multimodal web interfacing. His current focus includes hybrid (metric–topological) mobile robot navigation, simultaneous localization and map building and adaptive unsupervised exploration.

Björn T. Jensen is a Ph.D. student with the Autonomous Systems Lab at the Swiss Federal Institute of Technology. He received his Masters in Electrical Engineering and business administration from TU Darmstadt in 1999. In addition to robot navigation and image processing his research covers man–machine interaction.



Roland Siegwart (1959) received his M.Sc. ME in 1983 and his Doctoral degree in 1989 at the Swiss Federal Institute of Technology, Zurich (ETHZ). After his Ph.D. studies, he spent 1 year as a Postdoc at Stanford University, where he was involved in micro-robots and tactile gripping. From 1991 to 1996 he worked part time as R&D Director at MECOS Traxler AG and as a Lecturer and Deputy Head at the Institute of Robotics, ETH. During this time he was mainly involved in magnetic bearings, mechatronics and micro-robotics. Since 1997 he is a full-time Professor for Autonomous Systems and Robots at the Swiss Federal Institute of Technology, Lausanne (EPFL). His current research interests are robotics and mechatronics, namely high precision navigation, network base robotics (Internet, space exploration), all terrain locomotion and micro-robotics. He lectures various courses in robotics, mechatronics and smart product design at the two Swiss Federal Institutes of Technology and is cofounder of several spin-off companies. Roland Siegwart published more than 60 papers and is member of various scientific committees. He namely represents Switzerland in the International Federation of Robotics (IFR) and the Advisory Group for Automation and Robotics (AGAR) of the European Space Agency (ESA).