Tracking Multiple Objects Using the Condensation Algorithm

Esther B. Koller-Meier[a]*, Frank Ade[a]

[a]Communication Technology Lab, Image Science,
Swiss Federal Institute of Technology (ETH),
CH-8092 Zurich, Switzerland

Some years ago a new tracker, the CONDENSATION algorithm, came to be known in the computer vision community. It describes a stochastic approach that has neither restrictions on the system and measurement models used nor on the distributions of the error sources, but it can not track an arbitrary, changing number of objects. In this paper an extension of the CONDENSATION algorithm is introduced that relies on a single probability distribution to describe the likely states of multiple objects. By introducing an initialization density, observations can flow directly into the tracking process, such that newly appearing objects can be handled.

Keywords: Tracking; Recursive Bayesian filter; CONDENSATION algorithm; Monte Carlo filter; Kalman filter

1. Introduction

This paper is concerned with object tracking using a stochastic approach based on the CONDENSATION algorithm — conditional density propagation over time. We apply this new technique to mobile robot applications using range image sequences. Characteristic for these applications are the problems of tracking multiple, newly appearing or occluded objects which can not all be handled by the basic method.

The first step towards the CONDENSATION algorithm was the development of several resampling techniques [28, 9, 29]. A sample set from one distribution is resampled to form samples from another distribution. The idea of *recursive Bayesian filtering* based on sample sets was then independently discovered by several research groups. Our work has evolved from the CONDENSATION algorithm [13, 14], when it became known in the computer vision community. At the same time this filtering method was also studied for statistics and signal processing under the name *Bayesian bootstrap filter* [7] and *Monte Carlo Filter* [20].

Some extensions of the basic algorithm have already been proposed in the literature. The CONDENSATION algorithm was enhanced in [11, 15] to allow automatic switching between multiple motion models since the evolution of an object can often not be described with a single model. In [20, 16] *smoothing* methods are presented, which specify the statistical technique of conditioning the state distribution. The topic of *importance*

*Corresponding author. E-mail: ebmeier@vision.ee.ethz.ch

Table 1
Notation of vectors and probability distributions.

symbol	meaning
x_t	state vector at time t
z_t	measurement vector at time t
$\mathcal{Z}_t$	history of all observations $\{z_1, \ldots, z_t\}$ up to time t
$p(x_t \mid \mathcal{Z}_t)$	the *a posteriori* density
$p(x_t \mid \mathcal{Z}_{t-1})$	the *a priori* density
$p(x_t \mid x_{t-1})$	the process density describing the stochastic dynamics
$p(z_t \mid x_t)$	the observation density
$p(x_{t-1} \mid z_{t-1})$	the initialization density

sampling is discussed in [27, 17] where samples are placed according to a second information source without changing the original probability distribution. Other extensions for a similar purpose are described in [7, 4]. The authors in [17] also included a *reinitialization* in combination with importance sampling. Concepts related to an initialization are furthermore discussed in [26, 21, 23]. We introduced this idea [26, 21] to deal with newly appearing objects. Tracking multiple objects was recently considered by different research groups independently [8, 24, 26, 21]. We will describe our extension [26, 21] in details within this paper. The other two groups both tackle this problem by enlarging the dimension of a single state vector to include the states of all objects in the scene which results in tracking the joint probability.

The Condensation algorithm is already employed in several application areas. It was originally developed for contour tracking of a moving object in visual clutter. The authors in [4] have applied the Condensation method to localize a mobile robot by using a visual map of the ceiling. In [5] this method was further developed for multi-robot localization. In addition, the authors in [2] have proposed to use the Condensation algorithm for recognizing human gestures and facial expressions.

In the following section we first introduce the mathematical methods needed to formulate the Condensation algorithm. In Section 3 we explain how it can be extended to track multiple objects and to cope with newly appearing objects. A discussion in Section 4 shows then the differences to the basic scheme [13, 14] and also compares our approach to Kalman filtering [19, 10, 6]. Finally, in Section 5 we present results for mobile robot applications. The notation in this paper approximately follows [13, 14] and the used terminology is listed in Table 1.

2. Mathematical Methods

The quantities of an object that we want to track are described by an n-dimensional state vector $x \in \mathbb{X}$. Assuming that we are not able to know the exact state, we describe the knowledge about an object by a probability function $p(x)$.

Dynamics: As the observed scene changes over time, the probability function evolves to represent the altered object states. For computational purposes, the propagation is performed in discrete time steps. The dynamics of a single state vector is described by a stochastic difference equation

$$x_t = f(x_{t-1}, w_{t-1}) \tag{1}$$

where w_{t-1} is a noise vector with a known distribution that allows to model uncertainties in the *system function* f. The dynamics of the corresponding probability function is then described by a mapping that calculates $p(x_t)$ from $p(x_{t-1})$

$$p(x_t) = \int p(x_t|x_{t-1})\, p(x_{t-1})\, dx_{t-1} \tag{2}$$

where $p(x_t|x_{t-1})$ is the *process density*. This equation is used to estimate the probability distribution for the next time step. The density function $p(x_t)$ depends only on the immediately preceding distribution $p(x_{t-1})$, but not on any function prior to $t-1$, so it describes a *Markov process*.

Measurement: Let z_t be the measurement at time step t with history $\mathcal{Z}_t = \{z_1, \ldots, z_t\}$. We assume that the measurement is related to the state vector by

$$z_t = h(x_t, v_t) \tag{3}$$

where v_t is another noise vector and h is called the *measurement function*. For the tracking, we want to update our state estimate at each time step by incorporating the new measurement z_t. So instead of the density $p(x_t)$, we distinguish between the *a priori* density $p(x_t|\mathcal{Z}_{t-1})$ and the *a posteriori* density $p(x_t|\mathcal{Z}_t)$.

We can use Bayes' rule to determine the *a posteriori* density $p(x_t|\mathcal{Z}_t) = p(x_t|z_t, \mathcal{Z}_{t-1})$ from the *a priori* density $p(x_t|\mathcal{Z}_{t-1})$

$$\begin{aligned} p(x_t|\mathcal{Z}_t) &= \frac{p(z_t|x_t, \mathcal{Z}_{t-1})\, p(x_t|\mathcal{Z}_{t-1})}{p(z_t|\mathcal{Z}_{t-1})} \\ &= k\, p(z_t|x_t, \mathcal{Z}_{t-1})\, p(x_t|\mathcal{Z}_{t-1}) \\ &= k\, p(z_t|x_t)\, p(x_t|\mathcal{Z}_{t-1}) \end{aligned} \tag{4}$$

where k is a normalization factor. The simplifications can be made using the assumption that the measurements are independent, a proof can be found in [14]. The *observation density* $p(z_t|x_t)$ defines the likelihood that a state x_t causes the measurement z_t.

The complete tracking scheme, known as the *recursive Bayesian filter* first calculates the *a priori* density $p(x_t|\mathcal{Z}_{t-1})$ using the system model and then evaluates the *a posteriori* density $p(x_t|\mathcal{Z}_t)$ given the new measurement:

$$p(x_{t-1}|\mathcal{Z}_{t-1}) \overset{dynamics}{\longrightarrow} p(x_t|\mathcal{Z}_{t-1}) \overset{measurement}{\longrightarrow} p(x_t|\mathcal{Z}_t). \tag{5}$$

Factored Sampling: Analytic solutions to this problem are only available for a relatively small and restrictive choice of system and measurement models. In general, the recursive Bayesian filter is too complex and can not be evaluated simply in closed form. Since the space $\mathbb{X}$ on which x is defined is multi-dimensional and large, we can not just sample the probability densities at regular intervals. Hence, we use a stochastic sampling method.

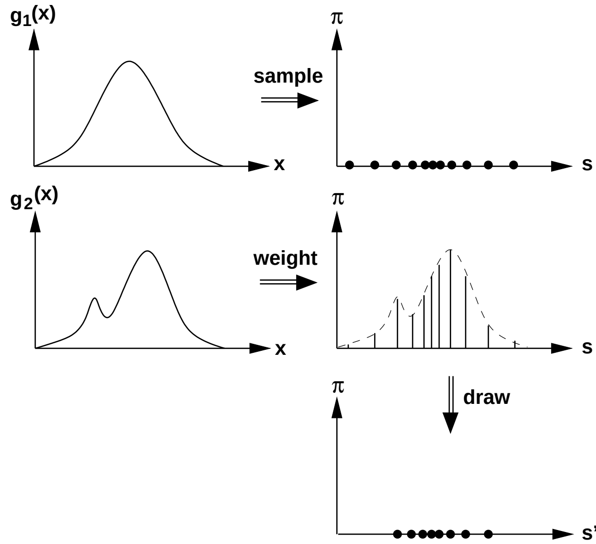


Figure 1. Factored sampling. A set s of N elements is sampled randomly from the density $g_1(x)$. Each sample is assigned a weight $\pi^{(j)}$ in proportion to the density $g_2(x)$. A new sample set s' is generated by choosing N elements according to their weights.

For the calculation of Eq. 4 we introduce the *factored sampling* method [9] which is used to find an approximation to a probability density

$$g(x) = g_2(x)\, g_1(x). \tag{6}$$

A set of samples $s = \{s^{(1)}, \ldots, s^{(N)}\}$ with $s^{(n)} \in \mathbb{X}$ is drawn randomly from $g_1(x)$. By then choosing a sample $s^{(j)}$ with probability

$$\pi^{(j)} = \frac{g_2(s^{(j)})}{\sum_1^N g_2(s^{(j)})}, \quad j = 1 \ldots N \tag{7}$$

from the sample set s, we obtain a new sample set s'. Its distribution tends to that of $g(x)$ as $N \to \infty$ (see Fig. 1).

The Condensation Algorithm: The CONDENSATION algorithm describes the sample-based representation of the recursive Bayesian filter and applies factored sampling iteratively to calculate the *a posteriori* density $p(x_t|\mathcal{Z}_t)$. We always have a sampled distribution of the *a priori* probability $p(x_t|\mathcal{Z}_{t-1})$ ($g_1(x)$ in Eq. 6), so the initial creation of a sample set can be omitted.

An iteration step of the CONDENSATION algorithm starts with a sample set s representing the *a posteriori* density $p(x_{t-1}|\mathcal{Z}_{t-1})$ from the previous time step. We propagate s to obtain a new sample set s' according to the system model, s' then represents the *a priori* density $p(x_t|\mathcal{Z}_{t-1})$. Applying factored sampling, a set s'' is drawn from s', where each element of the new set is chosen with probability $p(z_t|x_t)$, so that s'' represents the new *a posteriori* density $p(x_t|\mathcal{Z}_t)$.

A more detailed treatment of the basic CONDENSATION algorithm can be found in [13, 14].

3. Extending the Condensation Algorithm

The basic CONDENSATION algorithm was not designed to track an arbitrary, changing number of objects. However, in many applications it is necessary to track several objects at the same time. Additionally, in a typical dynamic scene objects will enter and leave the sensor's field of view. We are then faced with the following problems:

1. how to track *multiple* objects and
2. how to handle *newly* appearing objects.

Multiple Objects: We use a single CONDENSATION tracker for multiple objects. The probability distribution now represents several object states simultaneously. The tracker calculates the same, as if we would use an individual CONDENSATION tracker $p^{(i)}(x_t)$ for each object i and then combine the results, namely

$$p(x_t) = \sum_i \alpha^{(i)} p^{(i)}(x_t) \tag{8}$$

where the $\alpha^{(i)}$ weight the individual density functions.

During tracking we assume a fair segmentation, i.e. all objects are detected with the same accuracy. If this is not the case, the sample set may degenerate as the weights are larger for some objects than for others.

Newly Appearing Objects: In the basic CONDENSATION algorithm newly appearing objects are not considered. If we nevertheless apply the method in our applications, the results are not satisfactory because the observations are only utilized to calculate the weights, but do not affect the states directly. So, unless the state of a new object is sufficiently close to an already existing state, the object can not be tracked. As a solution we calculate an *initialization density* $p(x_{t-1}|z_{t-1})$ directly from the measurement at *every* iteration step. This function describes the probability of having an object with state x_{t-1} when only the measurement z_{t-1} is used. We then combine the initialization density with the *a posteriori* density that is obtained from the previous time step using

$$p'(x_{t-1}|\mathcal{Z}_{t-1}) = \gamma\, p(x_{t-1}|z_{t-1}) + (1-\gamma)\, p(x_{t-1}|\mathcal{Z}_{t-1}) \tag{9}$$

where γ weights the distributions. In the CONDENSATION algorithm the combination of the initialization density and the *a posteriori* density results in the following change: During the factored sampling we select only $N-M$ samples instead of N, but we draw M new samples from the initialization density. So, the weight for combining the densities is chosen as $\gamma = \frac{M}{N}$.

The introduction of an initialization step has apart from the possibility to track newly appearing objects also the effect of counteracting the problem of an unfair segmentation.

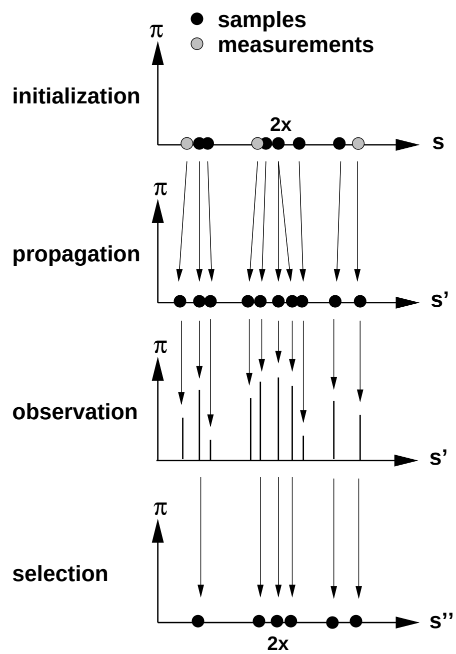


Figure 2. One iteration step of the extended CONDENSATION algorithm.

The initial samples will namely be concentrated in areas of the state-space where they will get a high weight. This focusing allows to choose the size of the sample set relatively low.

3.1. Implementation of the Extended Condensation Tracker

In mobile robot applications we are interested in localizing objects to evade potentially harmful situations. Our recorded range image sequences are given in spherical coordinates, so we use this coordinate system for the state representation. The state vector for any object at time t contains the distance d_t, the relative velocity $\dot{d}_t$, the horizontal angle ψ_t and its change $\dot{\psi}_t$, the vertical angle θ_t and the corresponding change $\dot{\theta}_t$, as well as the dimensions e_t^1 and e_t^2 of the bounding box around the object:

$$x_t^T = [d_t \;\; \dot{d}_t \;\; \psi_t \;\; \dot{\psi}_t \;\; \theta_t \;\; \dot{\theta}_t \;\; e_t^1 \;\; e_t^2]. \tag{10}$$

At each time step a segmentation [25, 30] of the image is used to detect objects and provides the observations:

$$z_t^T = [\bar{d}_t \;\; \bar{\psi}_t \;\; \bar{\theta}_t \;\; \bar{e}_t^1 \;\; \bar{e}_t^2]. \tag{11}$$

As can be seen, in the state vector x_t we also manage elements which can not be measured directly, but are computed from the other elements during tracking.

The extended algorithm used for our applications is charted in Fig. 2. Each iteration step can be divided into four parts, as described below.

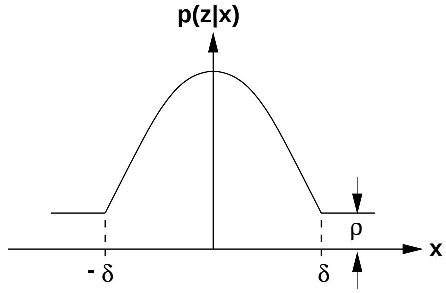


Figure 3. The observation density $p(z_t|x_t)$ for our applications is described by a truncated Gaussian.

Initialization: A new sample set s is constructed from $N - M$ samples that represent the *a posteriori* density $p(x_{t-1}|\mathcal{Z}_{t-1})$ from the previous time step and from M samples that are added directly based on the measurements at time $t-1$. In the first iteration no results from a previous time step are available and therefore, the whole set s is initialized with values based on the observations. Elements of the state vectors which can not be measured directly $(\dot{d}_t, \dot{\psi}_t, \dot{\theta}_t)$ are drawn from a Gaussian distribution with zero-mean or a reasonable mean value based on the ego-motion of the mobile robot.

Propagation: The evolution is described by the application of the system model

$$x_t = A\,x_{t-1} + B\,w_{t-1} \tag{12}$$

where A defines the deterministic and $B\,w_{t-1}$ the stochastic component of this difference equation. The deterministic part of the equation models the system knowledge while the stochastic part allows us to model uncertainties. In our applications we use a first order model for A describing an object moving with constant velocity. Expanding this model to second order, by additionally incorporating the acceleration into the state vector, is straightforward.

w_{t-1} is a vector of normal random variables scaled by B so that BB^T is the covariance of the process noise. The process density $p(x_t|x_{t-1})$ is therefore a Gaussian distribution. This is the same model as would be used for a Kalman filter [19, 10, 6].

Observation: In the observation step we weight each element of the set s' in terms of the measurements. For our applications we use a truncated Gaussian, which has a minimal constant value ρ. This residual probability permits that hypotheses of disappeared or occluded objects may survive the next time step. As we expect that an object with state x_t causes a measurement z_t in its vicinity, we only look at the closest observation. Thus, as observation density $p(z_t|x_t)$ (see Fig. 3) we use

$$\pi_t^{(n)} = \begin{cases} e^{-\frac{u^2}{2\,\sigma^2}} & u < \delta \\ \rho & \text{otherwise} \end{cases} \tag{13}$$

$$u = \min_m G(Hx_t^{(n)}, z_t^{(m)}) \tag{14}$$

where H is called the *measurement sensitivity matrix*. The function G calculates the difference between two z vectors using the position $p = (p_x, p_y, p_z)$ in Cartesian coordinates and the dimensions e^1 and e^2

$$G(Hx_t, z_t) = \sqrt{(p_t - \bar{p}_t)^2 + (e_t^1 - \bar{e}_t^1)^2 + (e_t^2 - \bar{e}_t^2)^2}. \tag{15}$$

As an object is represented by several samples, we use the *nearest neighbor* technique [1, 3] to calculate the observation density. Another approach is the *probabilistic data association filter* (PDAF) [1, 3] which weights all measurements. We do not expect significantly better results by applying the idea of the PDAF as long as an object is represented by an adequate number of samples. Using the concept of the *joint probabilistic data association filter* (JPDAF) [1, 3], which weights all measurements with all objects, is not straight forward as we do not know the exact number of objects. An area of further research could be the use of local clustering methods in order to incorporate the JPDAF into the extended CONDENSATION algorithm.

Selection: A fixed size of $N - M$ samples can now be selected from the set s'. A particular $s'^{(j)}$ is drawn with replacement, by choosing it with probability $\pi^{(j)}$. Some elements, especially those with high weights, may be chosen several times, leading to identical elements in the new set s''. In the propagation step these samples will be split due to the stochastic component of the system model. Others with relatively low weights may not be chosen at all.

The programming details for one iteration step are given in Fig. 4.

4. Comparisons

We would like to point out some differences and similarities of our approach to the other methods which are based on the recursive Bayesian filter, especially the basic CONDENSATION algorithm and the Kalman filter. An overview of the advantages and disadvantages of these different tracking methods is shown in Table 2.

4.1. Comparisons to the Basic Condensation Algorithm

To our knowledge this is the first application that uses the CONDENSATION algorithm to track an arbitrary, changing number of objects. The challenge in mobile robot systems is their dynamic character as objects constantly enter or leave the sensors' field of view. In order to cope with this situation the basic approach had to be extended. We propose to use a single CONDENSATION tracker for multiple objects and to apply an initialization scheme to deal with newly appearing objects.

As the basic method was designed to track only one object, the mean of the sample set serves as an estimator of the distribution mean. In our extension we can for example look for the closest sample as this indicates the nearest potential obstacle. To determine the mean of each object, local clustering methods have to be used.

4.2. Comparisons to Kalman Filtering

In contrast to the CONDENSATION tracker the density function used by a Kalman filter [19, 10, 6] is unimodal and evolves as a Gaussian. The great advantage of the

1. **Initialize** the sample set s_{t-1}:
   (a) calculate the initialization density $p(x_{t-1}|z_{t-1})$ and draw M elements $i_{t-1}^{(m)}$ from it
   (b) combine the samples from the previous time step with the initial samples
   $s_{t-1}^{(n)} = s_{t-1}^{\prime\prime(l)} \cup i_{t-1}^{(m)} \qquad n = 1 \ldots N,\ l = 1 \ldots N - M$ and $m = 1 \ldots M$
2. **Propagate** each sample from the set s_{t-1} by a stochastic difference equation:
   $s_t^{\prime(n)} = A\, s_{t-1}^{(n)} + B\, w_{t-1}^{(n)}$
   where $w_{t-1}^{(n)}$ is a vector of standard random variables and BB^T is the process noise covariance
3. **Observe** the measurements:
   (a) weight each sample of the set s_t' with
   $$\pi_t^{(n)} = \begin{cases} e^{-\frac{u^2}{2\,\sigma^2}} & u < \delta \\ \rho & \text{otherwise} \end{cases} \qquad u = \min_m G(Hx_t^{(n)}, z_t^{(m)})$$
   (b) calculate the normalized cumulative probabilities
   $c_t^{(0)} = 0$
   $c_t^{(n)} = c_t^{(n-1)} + \pi_t^{(n)} \qquad$ for $n = 1 \ldots N$
   $c_t^{\prime(n)} = \frac{c_t^{(n)}}{c_t^{(N)}} \qquad$ for $n = 1 \ldots N$
4. **Select** $N - M$ samples from the set s_t' with probability $\pi_t^{(n)}$:
   (a) generate a uniformly distributed random number $r \in [0, 1]$
   (b) find, by binary search, the smallest j for which $c_t^{\prime(j)} \geq r$
   (c) set $s_t^{\prime\prime(l)} = s_t^{\prime(j)}$

Figure 4. An iteration step of the extended CONDENSATION algorithm.

CONDENSATION algorithm is, that neither restrictions on the system and measurement models nor on the noise distributions are given. On the other hand, if the models *are* linear and the error sources *are* Gaussian, the Kalman filter calculates the exact solution of the recursive Bayesian filter [12, 22, 18] while the solution of the CONDENSATION algorithm has a discretization error.

A single Kalman filter is able to track only one object. Accordingly, the tracking of several objects can be done by using a Kalman filter for each object. In the case of multiple hypotheses tracking caused by occlusion or disappeared objects it is even necessary to keep several trackers for each object or to use methods like the joint probabilistic data association filter [1, 3]. As the extended CONDENSATION algorithm has the ability to deal with multi-modal distributions, multiple objects as well as multiple hypotheses can easily

Table 2
Advantages and disadvantages of the different tracking methods.

method	advantages	disadvantages
basic CONDENSATION algorithm	unrestricted tracking models unrestricted error sources multiple hypotheses simple easy implementation automatic aging	only one object discretization error importance sampling
extended CONDENSATION algorithm	unrestricted tracking models unrestricted error sources multiple hypotheses arbitrary number of objects newly appearing objects simple easy implementation fixed running time for multiple objects automatic aging	discretization error complete segmentation of the objects necessary interpretation
Kalman filter	linear tracking models Gaussian error sources exact solution easy interpretation	only one object management system for multiple hypotheses and multiple objects necessary running time depends on the number of hypotheses and objects no automatic aging

be tracked simultaneously.

The evolution of hypotheses of disappeared objects is directly regulated in the CONDENSATION algorithm by the selection of samples according to their weights. Kalman filtering may require an additional control mechanism that keeps or removes unattractive hypotheses from the management system. In [30] we present such an *aging* approach where hypotheses survive until their age exceeds a predefined time.

Using a sample-based representation of the probability density has many advantages. The calculations for each sample are simple and easy to program. The fixed number of samples leads to a constant running time per iteration step. Of course, for larger N we are able to achieve a better approximation of the density function, although this has its limit in accordance with the available processing time. A study of the computational complexity can be found in [21]. Furthermore, a parallel implementation of the filter can be readily realized.

5. Results

To show the ability of the extended CONDENSATION algorithm we present results on different data sources. We illustrate the method on range image sequences of a line and a matrix sensor. In both applications we are interested in evading potential obstacles for autonomous navigation. For safety reasons we are dealing with the worst case and will already start an evasive action when some samples are critical. We will demonstrate especially that we can cope with new or unseen objects.

5.1. Range Images from a Line Sensor

We tested the extended CONDENSATION algorithm on an image sequence recorded by a line sensor. This SICK LMS 200 sensor was mounted on a mobile robot at a height of $60cm$ above ground for indoor applications. The sequence was produced at the Institute of Robotics at the ETH with a frame rate of 1.5 images per second, an image size of 181 pixels and a field of view of $180°$. The maximal range of the line sensor was $8m$.

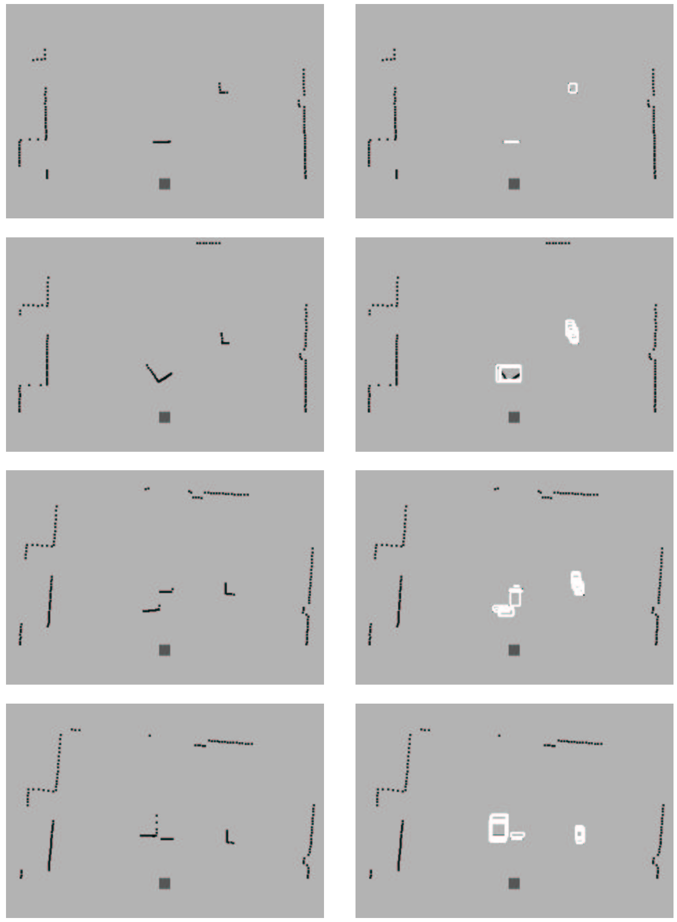
In Fig. 5 the left column indicates the range data plotted as points as seen from the top. The walls are visible as lines along the border where the back wall is not in the range of the sensor at the beginning. The position of the sensor is shown by a grey square in each image. In the right column, the results of the extended CONDENSATION tracker are overlaid to the range information. Each sample which represents an object state is illustrated by a bounding box which can be calculated from the object center (d_t, ψ_t) and the object dimensions (e_t^1, e_t^2) (see Section 3.1). We are interested in tracking objects within a certain safety zone, which we have chosen as $5m$. In this example, an indoor scene with three obstacles is shown where one object is occluded at the beginning. Two of the obstacles are stationary while the robot and one object move with different velocities. The moving object starts a left turn in the second image from the top. For this experiment we have used only $N = 20$ and $M = 3$ samples for each iteration step.

We have tested the capability of tracking the multi-modal distribution on the shown sequence. The results for the first 30 images are plotted in Fig. 6. To simplify matters only the distribution of the horizontal image coordinate of the object centers is shown instead of the multi-dimensional object states. The first initialization of the density function was only based on the measurements and therefore sharp peaks characterize our objects. After some time the distribution blurs as several hypotheses of an object develop.

5.2. Range Images from a Matrix Sensor

Results of a matrix sensor are presented for a public domain image sequence, which can be found at *http://marathon.csee.usf.edu/range/DataBase.html* in the database of the University of South Florida. This range image sequence was recorded by an Odetics LADAR sensor, which was mounted on the HERMIES III robot at the CESAR lab at Oak Ridge National Labs. The sequence has an image size of 128×128 pixels and a field of view of $60° \times 60°$.

In Fig. 7 we track three different objects in an indoor scene. To simplify matters we focus only on objects which have at least a dimension of $5cm \times 5cm$ and which are closer to the sensor than $1m$. From the observed scene a reflectance image (left column) and a corresponding registered range image (other columns) have been recorded. The object states are indicated by their bounding boxes and overlaid on the corresponding range



a b

Figure 5. Tracking multiple obstacles which are closer than $5m$ with a SICK LMS 200 sensor. a: Top view of the range data plotted as points. b: Result of the extended Condensation tracker. Each hypothesis of an object state is represented by a bounding box and overlaid to the range data.

image. In the middle column, the results of tracking multiple objects with an extended Condensation tracker are illustrated. The bounding boxes represent all the samples of the corresponding distribution. New objects which enter the safety zone of $1m$ are automatically incorporated into the tracking process. For example, the pyramid in the second image from the top and the cupboard in the third image appear subsequently as new potential obstacles. We can observe that new objects which are initialized through the observations develop multiple state hypotheses over time.

For comparison, the results of employing a Kalman filter are shown in the right column. As a Kalman filter can track only one object, a management system is required to handle all obstacles in the safety zone. Each object is marked by an identification number. When a match for the tracked object has been found, the identification number

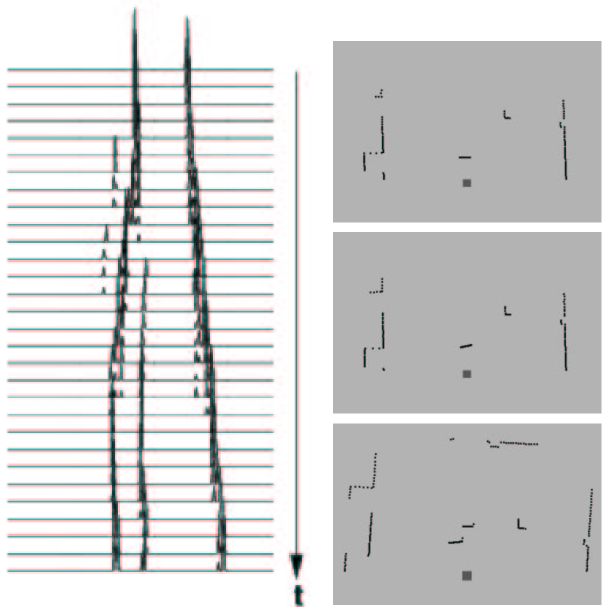


Figure 6. Propagation of a multi-modal density function at discrete time steps. On the left, the distribution of the object centers is plotted and on the right, the corresponding range images are shown.

remains unchanged, otherwise a new object is detected. An insufficient segmentation — as can be seen for example in the second image from the top — does not always allow a matching between successive frames and leads to an increase in the number of managed objects. In comparison the extended CONDENSATION algorithm automatically handles several hypotheses simultaneously.

The effect of incorporating the observations into the tracking process is presented in Fig. 8. As in the previous figure we see a reflectance and a registered range image of the indoor scene. We show the tracked samples as white bounding boxes and in black we show the incorporated observations. This example illustrates the survival of an object hypothesis even if it was not detected in the current frame. The cupboard for example is not marked as an observation and was therefore not in the safety zone. Nevertheless, this object hypothesis is still known in the tracking process and represented by a white bounding box. As we are dealing with disappeared and occluded objects, such samples should remain temporarily in the tracking process for safety reasons.

6. Conclusion

A general stochastic tracking approach with real-time capability has been presented which is neither limited to linear models nor requires the noise to be Gaussian. This method is based on a recursive Bayesian filter that calculates probability density functions of the likely object states at discrete time steps. Because analytical solutions are only

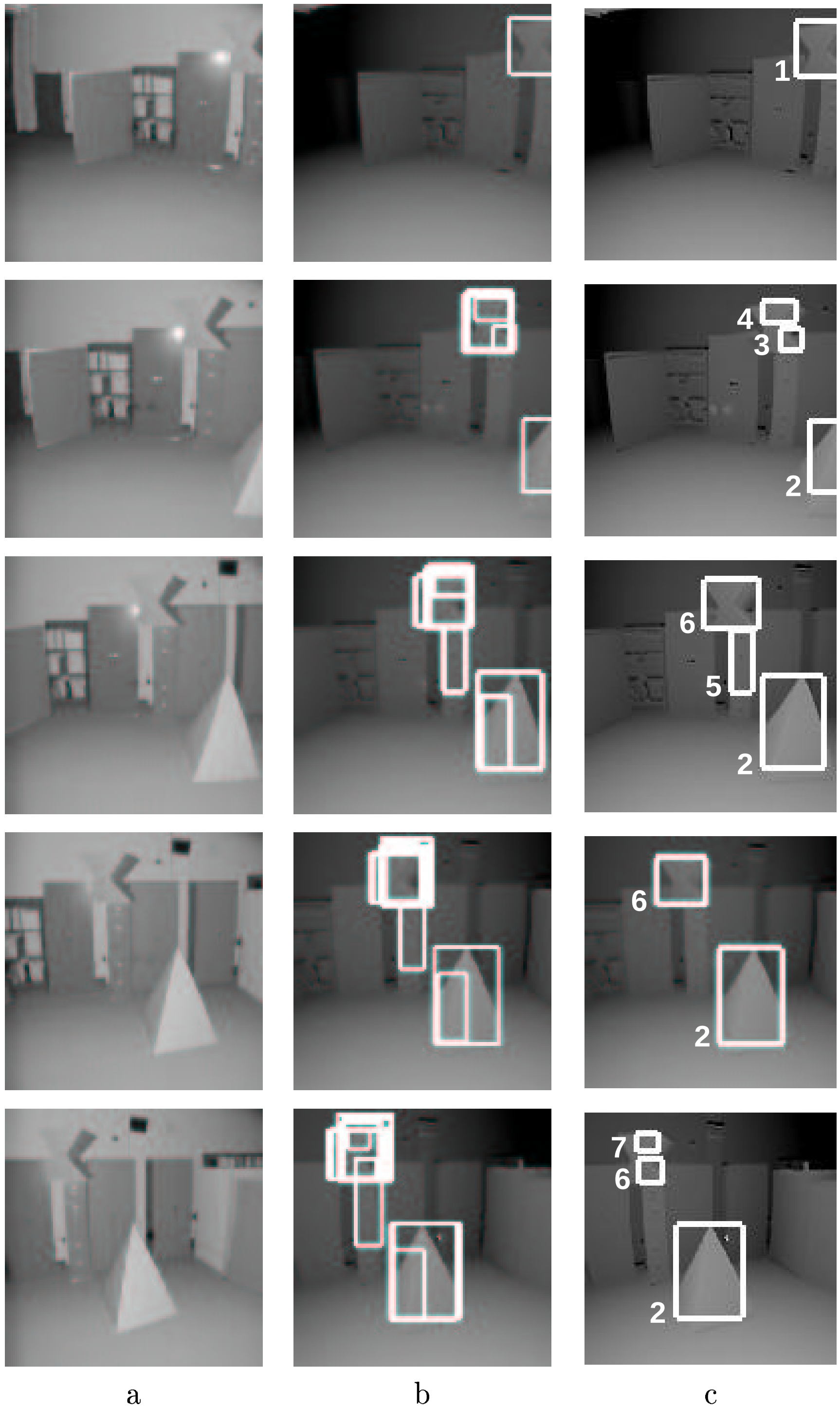


Figure 7. Tracking multiple obstacles with an Odetics LADAR sensor. Only objects closer than $1m$ and larger than $5cm \times 5cm$ are considered. a: The reflectance images recorded by a mobile robot. b: The distributions of the sample set as calculated by the extended Condensation algorithm are overlaid on the range images. c: Results of applying a Kalman filter for each object.

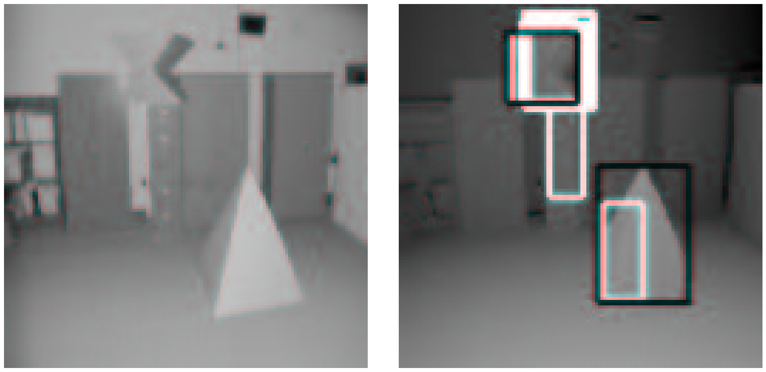


Figure 8. The incorporation of the observations. The white bounding boxes show the tracked object states and the black boxes represent the observations flowing into the tracking process. Even an object hypothesis which was not detected in the current frame has a change to survive.

available for a relatively small and restrictive set of system and measurement models, stochastic sampling techniques have to be applied. This leads to the CONDENSATION algorithm which is the sample-based solution of the recursive Bayesian filter.

As application we concentrated on mobile robot systems based on line and matrix range image sequences. Evading potential obstacles permits an effective path planning. Characteristic for such images is the ambiguity caused by several objects within the scene and the incomplete data information by disappeared or occluded objects. Furthermore, the dynamic character of the scenes implies that objects constantly enter or leave the sensors' field of view.

The simplicity and generality of the CONDENSATION algorithm allows a variety of straightforward extensions. Our proposed extension of the basic method aims at tracking an arbitrary number of objects simultaneously. As a consequence we represent multiple objects — instead of only one — with a single probability density function. In comparison to other methods, no extra management system is required to handle additional trackers.

The introduction of an initialization at each iteration step ensures that newly appearing objects can be tracked. It also guarantees that a sufficient number of samples are always available in the neighborhood of likely object states. This eliminates the problem that the sample distribution degenerates into only a few different states.

The extended CONDENSATION algorithm can deal with many of the problems that are typically encountered in tracking, such as unrestricted tracking models and error sources, arbitrary number of objects, newly appearing, disappearing or occluded objects.

Using a single distribution for describing the states of all objects is less accurate than using for example the joint distribution, but it leads to a simple system that can be used for real-time applications. A further limitation of the scheme may be the form of the output data. Depending on the application, the probability distribution respectively the sample set may have to be interpreted in different ways. This interpretation should be the subject of future research, for example local clustering methods could be studied to determine the number of tracked objects in the scene.

Acknowledgments

This research was partially supported by MINORA, a project of the Swiss priority program OPTIQUE II, funded by the ETH Council. We thank Sjur Vestli and Felix Wullschleger from the Institute of Robotics, ETH Zurich, for having helped us greatly to acquire range image sequences with their SICK LMS 200 sensor. Further, we would also like to acknowledge the CESAR lab at Oak Ridge National Labs, especially Judd Jones, Ole Henry Dorum, Fank Sweeney and Reinhold Mann, to have made it possible for us to use the range image sequences recorded by the Odetics LRF sensor mounted on the HERMIES III robot system.

References

[1] Y. Bar-Shalom and T. E. Fortmann, Tracking and Data Association, Academic Press Inc., (1988).

[2] M. J. Black and A. D. Jepson, A Probabilistic Framework for Matching Temporal Trajectories: Condensation-Based Recognition of Gestures and Expressions, 5th European Conference on Computer Vision 1 (1998), pp. 909-924.

[3] I. J. Cox, A Review of Statistical Data Association Techniques for Motion Correspondence, International Journal of Computer Vision 10(1) (1993), pp. 53-66.

[4] D. Fox, W. Burgard, F. Dellaert and S. Thrun, Monte Carlo Localization: Efficient Position Estimation for Mobile Robots, Sixteenth National Conference on Artificial Intelligence (1999).

[5] D. Fox, W. Burgard, H. Kruppa and S. Thrun, Collaborative Multi-Robot Localization, German Conference on Artificial Intelligence (1999).

[6] A. Gelb, Applied Optimal Estimation, MIT Press, (1996).

[7] N. Gordon and D. Salmond, Bayesian State Estimation for Tracking and Guidance Using the Bootstrap Filter, Journal of Guidance, Control and Dynamics 18(6) November-December (1995), pp. 1434-1443.

[8] N. Gordon, A Hybrid Bootstrap Filter for Target Tracking in Clutter, IEEE Transactions on Aerospace and Electronic Systems 33 (1997), pp. 353-358.

[9] U. Grenander, Y. Chow and D. M. Keenan, HANDS, A Pattern Theoretic Study of Biological Shapes, Springer-Verlag, (1991).

[10] M. S. Grewal and A. P. Andrews, Kalman Filtering, Prentice Hall, (1993).

[11] T. Heap and D. Hogg, Wormholes in Shape Space: Tracking through Discontinuous Changes in Shape, 6th International Conference on Computer Vision (1998), pp. 344-349.

[12] Y. C. Ho and R. C. K. Lee, A Bayesian Approach to Problems in Stochastic Estimation and Control, IEEE Transactions on Automatic Control AC-9 October (1964), pp. 333-339.

[13] M. Isard and A. Blake, Contour Tracking by Stochastic Propagation of Conditional Density, 4th European Conference on Computer Vision 1 (1996), pp. 343-356.

[14] M. Isard and A. Blake, CONDENSATION – conditional density propagation for visual tracking, International Journal on Computer Vision 29(1) (1998), pp. 5-28.

[15] M. Isard and A. Blake, A mixed-state Condensation tracker with automatic model-switching, 6th International Conference on Computer Vision (1998), pp. 107-112.

[16] M. Isard and A. Blake, A Smoothing Filter for Condensation, 5th European Conference on Computer Vision 1 (1998), pp. 767-781.

[17] M. Isard and A. Blake, ICONDENSATION: Unifying Low-Level and High-Level Tracking in a Stochastic Framework, 5th European Conference on Computer Vision 1 (1998), pp. 893-908.

[18] A. H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, (1970).

[19] R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, Transactions of the ASME, Journal of Basic Engineering, Series D 82(1) March (1960), pp. 35-45.

[20] G. Kitagawa, Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models, Journal of Computational and Graphical Statistics 5(1) (1996), pp. 1-25.

[21] E. B. Koller-Meier, Extending the CONDENSATION Algorithm for Tracking Multiple Objects in Range Image Sequence, Ph.D. Thesis, Hartung-Gorre, (2000).

[22] R. C. K. Lee, Optimal Estimation, Identification, and Control, MIT Press, (1964).

[23] S. Lenser and M. Veloso, Sensor Resetting Localization for Poorly Modelled Mobile Robots, IEEE International Conference on Robotics and Automation (2000), pp. 1225-1232.

[24] J. MacCormick and A. Blake, A probabilistic exclusion principle for tracking multiple objects, 7th International Conference on Computer Vision (1999).

[25] E. B. Meier and F. Ade, Object Detection and Tracking in Range Image Sequences by Separation of Image Features, IEEE International Conference on Intelligent Vehicles (1998), pp. 176-181.

[26] E. B. Meier and F. Ade, Using the Condensation Algorithm to Implement Tracking for Mobile Robots, 3rd European Workshop on Advanced Mobile Robots (1999), pp. 73-80.

[27] B. D. Ripley, Stochastic Simulation, John Wiley & Sons, (1987).

[28] D. B. Rubin, Using the SIR Algorithm to Simulate Posterior Distributions, Bayesian Statistics 3, Oxford University Press, (1988), pp. 395-402.

[29] A. F. M. Smith and A. E. Gelfand, Bayesian Statistics Without Tears: A Sampling-Resampling Perspective, The American Statistician 46(2) May (1992), pp. 84-88.

[30] K. Sobottka, E. Meier, F. Ade and H. Bunke, Toward Smarter Cars, in Sensor-Based Intelligent Robots, H. Christensen and H. Bunke and H. Noltemeier, eds., Springer-Verlag (1999).