STOCHASTIC PREDICTION OF WAYPOINTS FOR EXTENDED RANGE TELEPRESENCE APPLICATIONS

Patrick Rößler, Oliver C. Schrempf, and Uwe D. Hanebeck

Intelligent Sensor-Actuator-Systems Laboratory
Institute of Computer Science and Engineering
Universität Karlsruhe (TH)
Karlsruhe, Germany
{patrick.roessler, schrempf, uwe.hanebeck}@ieee.org

ABSTRACT

The Motion Compression framework for extended range telepresence applications consists of three functional modules: path prediction, path transformation, and user guidance. This paper presents a new path prediction module for known environments that exploits the property, that humans typically walk on straight paths toward discrete goal objects. In order to estimate the user's goal object out of a set of possible goals, we derived a Bayesian filter that gives this discrete estimate based on continuous measurements of the user's head pose.

1. INTRODUCTION

In the near future, humanoid robots will be omnipresent as household appliances. These robots are designed to help people with their everyday work. However, as household environments are typically unstructured and variable, those robots are likely to run into situations, where they cannot successfully finish their given tasks. In order to resolve these situations transparently for the owners of the robots, we proposed a service center for telepresent robot control [1] and exception handling.

In such a service center, also called the *user environment*, the user's head and hand motion is tracked. This motion data is transferred to the robot, that replicates this motion. On the other hand, the robot records sensory feedback of his environment, the *target environment*. This sensor information, e. g. camera images, is transferred back to the user and displayed on immersive displays like head-mounted displays. As a result, the user feels present in the target environment and can now control the robot intuitively.

In order to complete the given tasks, the robot typically needs to move freely in a large area. Most other systems use additional input devices [2] for wide area motion, however this leads to a loss of immersion as the user has no proprioceptive feedback of this motion.

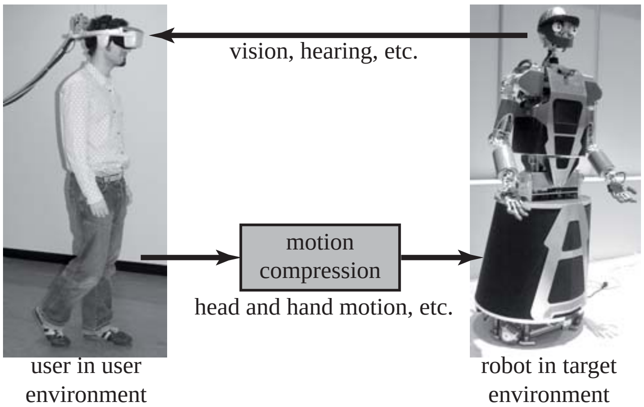

Fig. 1. A user telepresently operating the humanoid robot of the Collaborative Research Center SFB 588.

For this reason, the proposed system allows the user to control the robot by walking naturally and thus providing him with proprioception. However, the target environment is typically much larger than the user environment. Thus, the user's motion cannot be directly mapped into the target environment. The algorithmic framework of Motion Compression [3] provides an optimal nonlinear transformation between the robot's path in the target environment and the user's path in the user environment in such a way, that the user does not leave the user environment while controlling the robot through the target environment.

The Motion Compression framework consists of three modules, which are briefly explained in the following.

The first module is *path prediction*. Path prediction aims at predicting the user's desired path in the target environment based on his motion and, if available, additional information on the target environment. Currently, path prediction is only based on the user's view direction. This paper presents a

Supported by the German Research Foundation (DFG) within the Collaborative Research Center SFB 588 on "Humanoid robots–learning and cooperating multimodal robots".

new method for predicting the user's desired path in a known environment with a number of given goal objects.

Path transformation then transforms the predicted path in such a way that it fits into the target environment. As this is achieved by only modifying the path's curvature, path length and turning angles are preserved. This is important to guarantee a high degree of immersion. Path transformation is formulated as a dynamic optimization problem, that minimizes the curvature difference between the transformed path and the predicted path, under the given geometric constraints.

The final module *user guidance* guides the user on the transformed path, while he has the impression of walking on the original path. This is achieved by introducing small deviations into the robot's motion. The user unknowingly compensates for these deviations and, as a result, follows the transformed path.

One of the biggest challenges in this system is to give a good prediction of the user's desired path. However, there is only few work on prediction of human walking paths. The authors of [4] focus on realistic looking paths, which can be used in simulations. However, they do not give an analytical descriptions of these paths, which renders them useless for Motion Compression. In addition, the user's goal still has to be known.

Human users typically walk toward a goal based on visual cues [5]. This fact is used in the approach presented in [3]. This work assumes, that the user always walks toward temporary goal objects in straight lines, an assumption that was shown to work very well for Motion Compression. However, it uses heuristic methods for goal prediction.

The performance of the goal prediction can be enhanced by using a more sophisticated approach, originally designed for intention recognition [6]. As this system is designed to deal with many different sensor inputs, which are not available in the given application, this approach cannot be directly used here. However, the key idea of using hybrid probability densities, i. e., mixed discrete and continuous densities, is used to systematically derive a new method for goal prediction. The path is then assumed to be a straight line toward the estimated goal.

The remainder of this paper is structured as follows. Section 2 formulates the problem. A Bayesian filter algorithm, for the specific problem is derived in section 3. Section 4 looks into finding the models needed for the filter. An experimental evaluation of the system is given in section 5. Finally, section 6 draws conclusions.

2. PROBLEM FORMULATION

Humans moving in a goal directed way, walk in almost straight line paths toward goal objects. Thus, it is sufficient for path prediction to recognize the user's current goal object. This object is then assumed to be the next waypoint. The path is now predicted as a straight line from the user's current position to this goal object. Arbitrary paths are by constantly recalculating the goal estimation.

As we focus on the household scenario, we deal with a known environment. In these environments, there are typically a number of a priori known, discrete goal objects like doorways, furniture, or other landmarks. The objective is now to chose the actual goal object out of those possible goals, based on the head motion in the target environment.

In a Bayesian filter setup, the posterior probability density f^e for the state of the goal x has to be calculated based on a measurement of the head pose in the target environment $\underline{\hat{y}}$ and a prior probability density f^p for the goal as

$$f_k^e(x) = c_k f(\underline{\hat{y}}|x) f_k^p(x) \ , \qquad (1)$$

where c_k is a normalization factor. The actual goal position can then be estimated by maximizing the posterior density.

The conditional density $f_k(\underline{y}|x)$ describes the stochastic measurement model. This density is fixed, but unknown. Thus it has to be identified.

With a system model $x_{k+1} = a(x_k) + v_k$ the prediction step of the Bayesian filter is given as

$$f_{k+1}^p(x_{k+1}) = \int f(x_{k+1}|x_k) f_k^e(x_k) \mathrm{d}x_k \ , \qquad (2)$$

where the conditional density $f(x_{k+1}|x_k)$ is the stochastic system model, which is also to be identified.

3. BAYESIAN FILTER

In order to use such a Bayesian approach, a filter step and a prediction step specialized for the given application, have to be derived from the generic filter shown in figure 2.

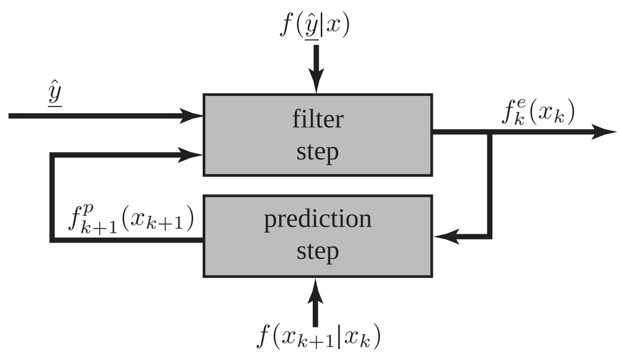

Fig. 2. Structure of the Bayesian filter as used for estimation of the user's goal.

3.1. Filter Step

This section derives a filter step for the given setup, based on the generic filter from above.

In the setup described above, the goal $x \in \{1, \ldots, n\}$ is a discrete variable, where $P(x = i)$ gives the probability that $x = i$, i. e., the user's current goal is the goal object number i. Thus, $f(x)$ is a discrete probability density, which can be written as dirac mixture

$$f(x) = \sum_{j=1}^{n} \delta(x - j) P(x = j) \ . \tag{3}$$

As the conditional density $f(\underline{y}|x)$ describes the density of a continuous variable $\underline{y}$ depending on a discrete variable, it is a hybrid conditional density. Assuming the conditional density to be time-invariant, it can be written as

$$\begin{aligned} f(\underline{y}|x) &= \sum_{j=1}^{n} \delta(x - j) f(\underline{y}|x = j) \\ &=: \sum_{j=1}^{n} \delta(x - j) f^{(j)}(\underline{y}) \ . \end{aligned} \tag{4}$$

Inserting the discrete density from (3) and the hybrid likelihood as in (4) into the filter step as given in equation (1) results in

$$\begin{aligned} &\sum_{j=1}^{n} \delta(x - j) P_k^e(x_k = j) = \\ &c_k \sum_{l=1}^{n} \delta(x - l) f^{(l)}(\hat{\underline{y}}) \sum_{m=1}^{n} \delta(x - m) P_k^p(x_k = m) \ . \end{aligned} \tag{5}$$

It can be easily seen, that the posterior probability for each goal i can be calculated separately as

$$P_k^e(x_k = i) = c_k \cdot f^{(i)}(\hat{\underline{y}}) \cdot P_k^p(x_k = i) \ . \tag{6}$$

A maximum a posteriori estimation of the user's goal is then given as the goal object with the highest posterior probability P_k^e.

3.2. Prediction Step

This section now derives the prediction step for the given setup. Inserting the discrete density over x from (3) into the prediction step of equation (2) leads to

$$\begin{aligned} &\sum_{j=1}^{n} \delta(x_{k+1} - j) P^p(x_{k+1} = j) = \\ &\sum_{l=1}^{n} \sum_{m=1}^{n} \delta(x_{k+1} - l) \delta(x_k - m) P(x_{k+1} = l | x_k = m) \cdot \\ &\sum_{i=1}^{n} \delta(x_k - i) P^e(x_k = i) \ . \end{aligned} \tag{7}$$

As $f(x)$ is a discrete density, it is also possible to use its vector notation

$$\underline{P}(x) = \begin{bmatrix} P(x = 1) \\ \vdots \\ P(x = n) \end{bmatrix} \ . \tag{8}$$

This allows to write the prediction step as vector-matrix-multiplication

$$\underline{P}_{k+1}^p(x_{k+1}) = \mathbf{A} \underline{P}_k^e(x_k) \ , \tag{9}$$

where the elements $\alpha_{i,j}$ of the transition matrix $\mathbf{A}$ are given as

$$\alpha_{i,j} = P(x_{k+1} = i | x_k = j) \ . \tag{10}$$

4. MODEL BUILDING

The stochastic measurement model given as the conditional density $f(\underline{y}|x)$ models the relation between the user's goal and the observed user data. In order to simplify this model, features derived from the observed user data are used rather than raw tracking information. These features include user position, view direction, and walking direction, as a discrete derivative of the user position.

However, the relationship between those features and the actual goal is still unknown. For this reason, we assume a parametric model for the conditional density and identify the parameters based on training data gathered from annotated data of real users.

As shown above, it is sufficient to model each component $f^{(j)}(\underline{y})$ of the conditional density separately, in order to obtain the complete conditional density $f(\underline{y}|x)$. Modeling the components by means of a Gaussian mixture model as

$$f^{(j)}(\underline{y}) = \sum_{i=1}^{L} w_i \mathcal{N}(\underline{y} - \underline{\mu}_i, C_i) \ , \tag{11}$$

where L is the number of mixture components, results in a general parametric model. The parameters to be estimated are the weights w_i, the means $\underline{\mu}_i$, and covariance matrices C_i of each Gaussian.

However, it is hard to acquire enough training data to parameterize a separate model for each goal. In addition, such a procedure would not allow to add new goals at run time, as for those goals there is no training data available.

Under the assumption that the human's behavior is the same independent of the goal object, it is possible to use relative, goal-independent features, rather than the absolute features given above. That means, a new feature vector

$$\tilde{\underline{y}} = \underline{g}(\underline{y}, x) \tag{12}$$

is calculated as a function of the absolute features $\underline{y}$ and each goal x. The new features are now distance to the goal, relative view direction, and relative walking direction as shown

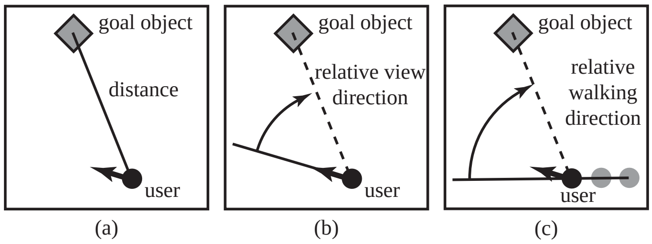

Fig. 3. Relative features used for goal estimation. Distance to goal object (a), relative view direction (b), and relative walking direction (c).

in figure 3. The conditional density

$$f^{(j)}(\underline{\tilde{y}}) = f(\underline{\tilde{y}}) \tag{13}$$

is now the same for every goal $x = j$. However, the relative features $\underline{\tilde{y}}$ are now goal-dependent.

As a result, it is only necessary to identify the parameters of one single conditional density. These parameters can be estimated with standard procedures, like expectation maximization.

As the work presented here concentrates on the measurement step, the prediction matrix $\mathbf{A}$ was set with a simple heuristic. It is assumed, that the user stays at his current goal with a probability p at every time step. If he changes the goal, all other goals are equally possible. Thus, the matrix $\mathbf{A}$ is given by

$$\begin{aligned} \alpha_{i,i} &= p \\ \alpha_{\substack{i,j \\ i \neq j}} &= \frac{1-p}{n-1} \end{aligned} \quad , \tag{14}$$

where n is the number of possible goals.

5. EXPERIMENTAL RESULTS

In order to obtain experimental data, unaffected by issues arising from robot control or robot localization, all experiments were conducted with a real user in a real user environment and a virtual target environment. In such a scenario, the user navigates an idealized robot through a virtual reality.

We designed one large room with a size of $20 \times 30\,\mathrm{m}^2$. The room includes several walls and has some images on the walls. As shown in figure 4, two types of possible goals were identified: wall corners and images.

In order to gather training data for the measurement model, three different users each walked about the virtual environment for several minutes and stated their current goal verbally. This goal was protocoled and fused with the tracking information. Of course, this procedure leads to lots of wrongly classified data points, especially when users switch from one goal to another.

Using the three relative features, described earlier, and planning for good generalization of the results, we trained the

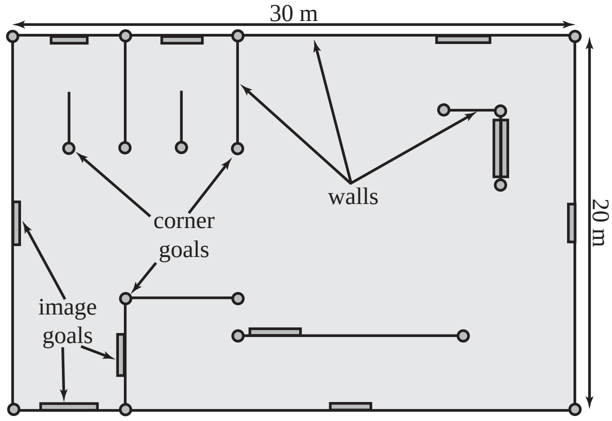

Fig. 4. Virtual target environment with all possible goal objects, including eleven images and 18 corners.

conditional density as a single Gaussian. The training was conducted with an expectation maximization algorithm [7].

As humans typically only walk toward visible goals, the current goal is not chosen from the complete list of possible goals, but only from a smaller number of active goals as shown in figure 5. Those active goals are the goals visible from the user's current position. This visibility relation is precalculated. The current goal is then estimated from the active goals with the method proposed in this paper.

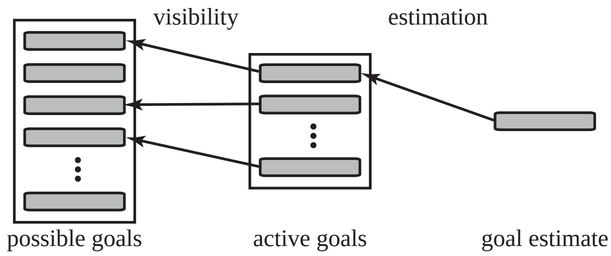

Fig. 5. The current goal is estimated from the active goals.

In order to verify the proposed method, we conducted an experiment where a user was walking in the virtual environment from above. During this test run the goal estimation was calculated at an update rate of approximately $150\,\frac{1}{\mathrm{s}}$ and p was set to $p = 0.9$. Figure 6 shows the user's positions during the experiment and the corresponding estimated goals. Note, that for the clarity of presentation only every 150th data point is shown.

The predicted paths for the user are shown as blue lines. It is clear to see, that these are sound estimates for the user path. Swapping between several goal objects, leads to a predicted path directed in between those goals objects. A comparison with the user's real goal object is not necessary, as this work did not aim at estimating the real goal, rather than finding a good path prediction for Motion Compression.

Figure 7 shows the user's path in the user environment during the experiment. The user stays well inside the bounds of the environment. This shows, that Motion Compression works well together with the new path prediction module.

The most important criterion for the quality of this algorithm is immersion. Unfortunately, there is no objective measure for immersion. However, the test person stated, that the immersion was high during the whole experiment.

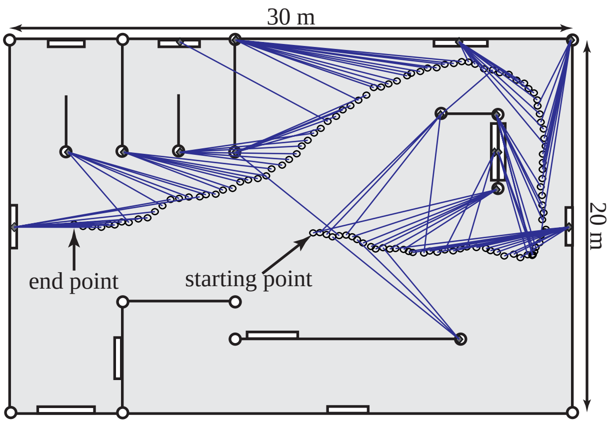

Fig. 6. Overlay of the map of the target environment, the user path (circles), estimated goals (diamonds), and temporarily predicted paths (lines).

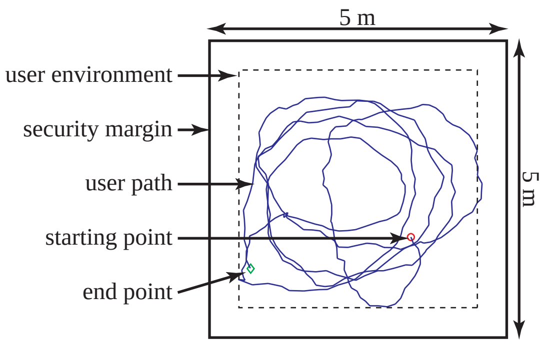

Fig. 7. The path of the user in the user environment.

6. CONCLUSIONS

In order to handle exceptions in household robots, a service center for telepresent robot control has been implemented. By using Motion Compression, a nonlinear path transformation, the user is able to navigate the robot through vast target environments, even though he is located in a limited user environment.

One important step of Motion Compression is path prediction. This paper presented a new path prediction model based on a hybrid Bayesian filter approach. A filter, that estimates the current goal based on position and orientation of the robot was derived systematically from the generic Bayesian estimator. The conditional density of the measurement step was modeled with a bank of Gaussian mixture models, which was parameterized by means of expectation maximization. In order to generalize this model, goal-independent features were used.

First experiments show the feasibility of the new approach as the new algorithm provides sound estimates of the user's current goal. A high degree of immersion was stated.

Future work will include investigation of more complex measurement models. For example it could be useful to add more features and train a Gaussian mixture model with more components. Another interesting topic is the identification of more sophisticated prediction models, that give a more detailed model of the behavior of switching goals.

The proposed path prediction module is an important improvement to Motion Compression, that will eventually lead to a system that allows intuitive robot teleoperation.

7. REFERENCES

[1] Patrick Rößler and Uwe D. Hanebeck, "Telepresence Techniques for Exception Handling in Household Robots," in *IEEE Intl. Conference on Systems, Man, and Cybernetics (SMC'04)*, The Hague, Netherlands, 2004, pp. 53–58.

[2] Terrence Fong and Charles Thorpe, "Vehicle Teleoperation Interfaces," *Autonomous Robots*, vol. 11, no. 1, pp. 9–18, 2001.

[3] Norbert Nitzsche, Uwe D. Hanebeck, and Günther Schmidt, "Motion Compression for Telepresent Walking in Large Target Environments," *Presence*, vol. 13, no. 1, pp. 44–60, 2004.

[4] David C. Brogan and Nicholas L. Johnson, "Realistic Human Walking Paths," in *Computer Animation and Social Agents (CASA)*, New Brunswick, NJ, USA, 2003, pp. 94–101.

[5] Simon K. Rushton, Julie M. Harris, Maugan R. Lloyd, and John P. Wann, "Guidance of Locomotion on Foot Uses Perceived Target Location Rather than Optic Flow," *Current Biology*, vol. 22, no. 8, pp. 1191–1194, 1998.

[6] Oliver C. Schrempf and Uwe D. Hanebeck, "A Generic Model for Estimating User Intentions in Human-Robot Cooperation," in *2nd Intl. Conference on Informatics in Control, Automation and Robotics (ICINCO 2005)*, Barcelona, Spain, 2005, vol. 3, pp. 250–256.

[7] Radford M. Neal and Geoffrey E. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants," in *Learning in Graphical Models*, Michael I. Jordan, Ed., pp. 355–368. MIT Press, Cambridge, MA, USA, 1998.