Novel Planning-based Algorithms for Human Motion Prediction

Dizan Vasquez

Novel Planning-based Algorithms for Human Motion Prediction

Dizan Vasquez
Team CHROMA, Inria Grenoble/CITI
655 Av. de l'Europe, Montbonnot
38334 Saint Ismier Cedex, France
dizan.vasquez@inria.fr

***Abstract*— Human motion prediction from visual tracking is a challenging problem with a wide array of applications such as robotics, video surveillance and situation understanding. Recently, planning-based methods –which assume that people move by planning over a cost function– have emerged as one of the most promising alternatives. Nevertheless, state of the art planning based algorithms have shortcomings regarding their computational complexity and ability to predict for arbitrary time intervals. This paper addresses these shortcomings by leveraging alternative planning techniques (Fast Marching Method) and formulating efficient algorithms for goal estimation and full spatiotemporal prediction with lower complexity than comparable approaches. In preliminary experiments, the proposed method significantly outperforms the accuracy of the current state-of-the-art approach while reducing the computation time by a factor of 30 using a parallel version of our algorithm.**

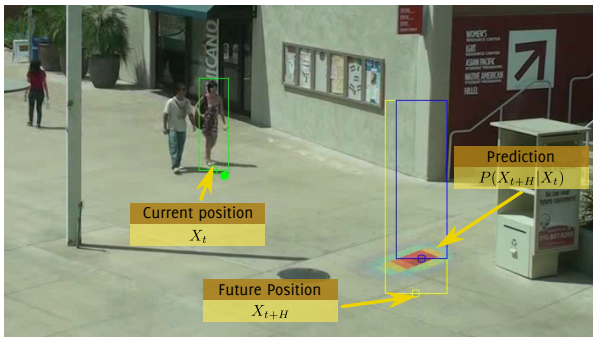

Fig. 1. Example of spatiotemporal state prediction for a single agent, indicating its current and future position after 8 seconds (green and yellow boxes) as well as the predicted position probability and its maximum value (blue box).

I. Introduction

In recent years, mobile robotics has seen an important shift in the way it considers humans. People are no longer seen as mere inanimate obstacles to avoid, but as essential environmental elements to interact and collaborate with. Moreover, in most cases, the safety, comfort and, in general, the well-being of people should take precedence over the robot's task. Hence, in order to successfully cohabit with humans, robots require models to estimate people's future position and velocity and plan their own motion accordingly. This has motivated the emergence of a significant number of motion prediction approaches in the last ten years [1]. The problem has also been widely studied by the computer vision community in areas such as video surveillance [2] and multiple object tracking [3].

A closer look at the literature shows that proposed methods actually address different sub-problems. For instance, *Goal Estimation* [4] focuses on determining the final place in the environment that an agent aims to reach. In this paper, we address the problem of estimating the agent position at specific future times, which we will henceforth call *Spatiotemporal State Prediction* (Fig. 1). Previous approaches belonging to this category differ on the kind of variables which they take into account: while early work was primarily based on kinodynamic variables [5], later approaches introduced further elements such as: interaction with other dynamic agents [6], [7]; typical motion patterns [8], [9]; static features of the environment such as walls [10], [11]; or all of the previous factors [12], [13].

This paper is close in spirit to the planning-based approach of Ziebart and Kitani [10], [11] which extends Markov Decision Processes to predict motion assuming a static environment with known destinations. Our paper addresses two drawbacks of those works: their high computational complexity and their limited capability to predict for a specific time. Our general contribution is a motion prediction framework enabling any cost-to-go planning algorithm to represent the uncertainty related to human motion. Our specific contributions are:

1) A simple and efficient goal estimation procedure based on the gradient of the local cost-to-go function.
2) A sound mechanism to fuse the output of different spatiotemporal state predictors based on "Bayesian Fusion with Diagnosis" [14].
3) Two spatiotemporal motion models: one based on the agent's velocity and the other on its estimated goal.
4) Experimental validation of our approach against a state-of-the-art algorithm using tracking data from the VIRAT data set [15].

The rest of this paper is structured as follows: Sec. II describes the ideas behind planning based approaches and the current bottlenecks motivating our work; Sec. III describes our general probabilistic model and other theoretical contributions; experiments and their results are presented in Sec. IV; finally, we present our conclusions and outlook in Sec. V.

II. Planning-based prediction approaches

Planning-based approaches are built around the assumption that people and other intelligent agents, when they move, minimize a cost function which depends on the context. Thus, if the cost function and the context are known, future motion may be predicted using the vast arsenal of existing motion planning algorithms.

Besides its intuitive appeal, one of the main advantages of this assumption is that it lays down a sound theoretical framework to deal with the different variables which may be considered relevant to human motion. Nonetheless, this formulation implies at least two underlying challenges: (a) the trade-off between the computational cost of motion planning algorithms and the real-time requirements of many applications; and (b) the difficult problem of expressing the cost function and tunning its parameters.

This paper focuses on the first challenge and assumes that the cost function is known *a priori*[1]. This choice is motivated by our intended application, autonomous robot navigation, where opportune predictions are critical to plan safe motion. Such real-time requirements are also important for other applications such as motion tracking.

We will focus our discussion on the work of Kitani *et al.* [10], [11] which, from our point of view, summarizes the state of the art on planning-based prediction. The interested reader is referred to [2], [1] for comprehensive reviews of motion prediction approaches from the computer vision and robotics perspectives.

The works of Ziebart and Kitani are based on the use of a Markov Decision Process (MDP) defined over a regular discretization (*i.e.* grid) of the agent's workspace in order to infer goals and predict motion. They model cost as a linear combination of features (presence of obstacles, kind of terrain, etc.) which encode the context and may be extracted from camera images using semantic segmentation approaches. They apply Inverse Reinforcement Learning to estimate the combination weights which, in theory, can then be used to compute the cost for a different environment without further learning. Despite its generality, this approach has the following important drawbacks concerning prediction: (a) its high computational complexity due to the use of an iterative inference algorithm requiring an unbounded, but usually consequent, number of iterations; (b) its limited ability to predict the state for a specific time: temporal predictions are only addressed in [10], where they propose a method that, essentially, assumes that all agents move at the same speed; and (c) they assume that, while moving, agents are not able to change their intended destination.

Despite these drawbacks, this approach –which we will henceforth denote as MDPMP (Markov Decision Process Motion Prediction)– is very attractive because of its use of a unified formalism for both planning and uncertainty modeling. With this paper, we aim to propose an alternative which keeps these advantages while addressing the corresponding weaknesses.

[1]For an example of cost function modeling and learning, we refer the reader to [10].

III. Proposed Approach

Our approach aims to propose solutions to the three stated weaknesses of MDPMP, namely:

1) *High computational complexity*. This is dealt with by using the Fast Marching Method (FMM) [16], [17] an efficient deterministic planning algorithm which computes the cost-to-go to a given location for every cell of a grid representing the agent's workspace. This grid is then used in a novel goal prediction algorithm (Sec. III-B) and to produce a path-like prediction equivalent to the output of MDPMP (Sec. III-D).
2) *Limited ability to model the temporal evolution along the predicted path*: this is addressed through the use of a velocity-dependent probabilistic motion model which is used to estimate a probability distribution of the future agent's position (Sec. III-C). This is then fused with the cost-based model proposed in Sec. III-D to produce a full spatiotemporal prediction (Sec. III-E).
3) *Constant-goal assumption*. We propose a gradient-based goal prediction approach (Sec. III-B) which does not rely on filtering, making it capable of quickly recognizing intended destination changes as they happen.

In the rest of this section, we describe our global probabilistic model and the notation used throughout this document.

A. Probabilistic model and Notation

In its most basic form, state prediction aims to compute the probability distribution of the future agent's position knowing the current one (Fig. 1). This may be expressed as:

$$P(\vec{X}_{t+H}|\vec{X}_t), H > 0 \tag{1}$$

where $\vec{X}_t$ and $\vec{X}_{t+H}$ are vectors in $\mathbb{R}^2$ representing the agent's positions at times t and $t+H$ (*i.e.* current and future positions). H is often called the *prediction lookahead* or *time horizon*.

The proposed approach augments this basic model with the following variables:

- **Velocity** $\vec{V}_t \in \mathbb{R}^2$: The agent's velocity at time t.
- **Goal** $G_t \in 1, \cdots, |G|$: The agent's intended goal at time t. The time sub-index indicates that we assume that the agent may change its intentions while it is moving. $|G|$ is the total number of goals.
- **Velocity model consistency variable** $I_V \in 0, 1$: A binary variable indicating whether our velocity model is consistent with the real velocity of the agent. As explained in Sec. III-E, this kind of variable is part of the "Bayesian Fusion with Diagnosis" formalism.
- **Cost model consistency variable** $I_C \in 0, 1$: Another consistency variable for model fusion indicating whether our path cost model is consistent with the real cost incurred by the agent.

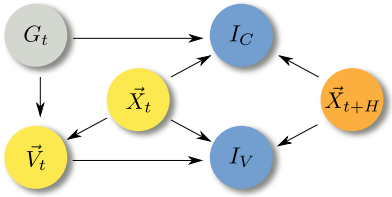

Fig. 2. Graphical model of the proposed JPD. For prediction, the observed variables are the current position and velocity (yellow nodes) the output variable is the predicted position (orange node), latent variables include the goal (gray node) and the consistency variables (blue nodes).

Having defined our six variables (including $\vec{X}_t$, $\vec{X}_{t+H}$) we may now describe the Joint Probability Distribution (JPD) of our global model (Fig. 2):

$$P(\vec{X}_t,\vec{X}_{t+H},\vec{V}_t,G_t,I_V,I_C) = \tag{2}$$
$$P(\vec{X}_t)P(\vec{X}_{t+H}) \tag{3}$$
$$\times P(G_t)P(\vec{V}_t|\vec{X}_t,G_t) \tag{4}$$
$$\times P(I_V|\vec{X}_t,\vec{X}_{t+H},\vec{V}_t) \tag{5}$$
$$\times P(I_C|\vec{X}_t,\vec{X}_{t+H},G_t) \tag{6}$$

The JPD can be seen as the composition of four different models:

- The *position priors* (3), modeled as uniform probability distributions over the agent's workspace.
- *Goal Prediction* (4), described in Sec. III-B.
- The *Velocity-based prediction model* (5), described in Sec. III-C
- The *Cost-based prediction model* (6), described in Sec. III-D

It is straightforward to apply Bayesian inference to predict the future state from the current state and velocity (setting $I_V = I_C = 1$):

$$P(\vec{X}_{t+H}|\vec{X}_t,\vec{V}_t,I_V,I_C) = \frac{1}{Z}P(I_V=1|\vec{X}_t,\vec{X}_{t+H},\vec{V}_t) \times \sum_{g=1}^{|G|} P(G_t=g)P(\vec{V}_t|\vec{X}_t,G_t=g)P(I_C=1|\vec{X}_t,\vec{X}_{t+H},G_t=g) \tag{7}$$

where Z is a normalization variable.

The remaining subsections describe the individual sub-models of Eq. (2).

B. Goal Prediction

Our Goal Prediction model is based on the use of a planning approach to compute the minimum cost-to-go $C_{\vec{X}_i \to G_g}$ from a given position $\vec{X}_i$ to a given goal region G_g in the environment. For grid-based representations of the agent's workspace and non-uniform cost functions, Dijkstra's algorithm is often used to compute the cost-to-go from every cell to a given region. Such approach is very efficient, having run-time complexity of $O(n \log n)$, where n is the number of cells in the grid. However, Dijkstra's algorithm suffers from severe aliasing problems (*i.e.* digitization bias) due to its coarse discretization on the number of directions (usually four or eight). Instead, we use the Fast Marching Method [16], [17] which is much more accurate while keeping the same run-time complexity.

In addition to the cost-to-go for every goal, our approach also computes, for every cell in the grid, the unitary gradient vector $\hat{\nabla}_{G_t}$ of the cost-to-go function using second order differences and normalizing the result. Since we assume that neither the costmap nor the set of existing goals change, it is possible to compute all these values during the initialization stage of the algorithm, instead of doing it at every time step.

Our goal prediction approach is based on the assumption that the orientation of an agent aiming to reach goal G_t, is normally distributed around the *optimal* direction, given by the negative gradient vector. This yields the following model:

$$P(\vec{V}_t|\vec{X}_t,G_t) = \mathcal{N}\left(\vec{V}_t;\mu_G(\vec{X}_t,\vec{V}_t),\Sigma_G^2\right) \tag{8}$$

where $\mathcal{N}$ denotes the normal distribution, the covariance Σ_G^2 is a system parameter expressing the sensibility to misalignments, and the mean value:

$$\mu_G(\vec{X}_t,\vec{V}_t) = -\hat{\nabla}_{G_t}(\vec{X}_t)\left\|\vec{V}_t\right\| \tag{9}$$

is the optimal direction multiplied by the velocity's norm as a way of penalizing higher speeds which will lead to the object getting farther away from the goal when the vectors are misaligned. This model may be seen as a predictive version of the approach proposed in [18] for crowd simulation.

An advantage of using the cost-to-go gradient is that, as opposed to MPDMP, it is possible to represent time-varying goals. On the other hand, this means that our approach does not rely on filtering, potentially making it less robust to noise. In practice, however, velocities are often computed upstream (*i.e.* by the tracking algorithm) using some sort of filter, considerably limiting this problem.

C. Velocity-Based Model

As its name indicates, the velocity-based model predicts the future position on the basis of the current position and velocity. It assumes that the agent will keep its current speed and that the total distance $D(\vec{X}_t,\vec{X}_{t+H})$ it will travel during the time horizon is normally distributed, with mean:

$$\mu_D(\vec{V}_t) = H\left\|\vec{V}_t\right\| \tag{10}$$

So that the full model is:

$$P(\vec{X}_{t+H}|\vec{X}_t,\vec{V}_t) = \mathcal{N}\left(D(\vec{X}_t,\vec{X}_{t+H});\mu_D(\vec{V}_t),\sigma_D^2\right) \tag{11}$$

where σ_D^2 is a model parameter expressing the uncertainty on the predicted distance and $D(\cdot)$ is the *cost-corrected* distance, which accounts for the fact that minimum cost paths avoid high-cost regions of the environment.

For a homogeneous cost function, the corrected distance is just the Euclidean distance, and the distribution is shaped as a circle centered in the current position $\vec{X}_t$ (Fig. 3b).

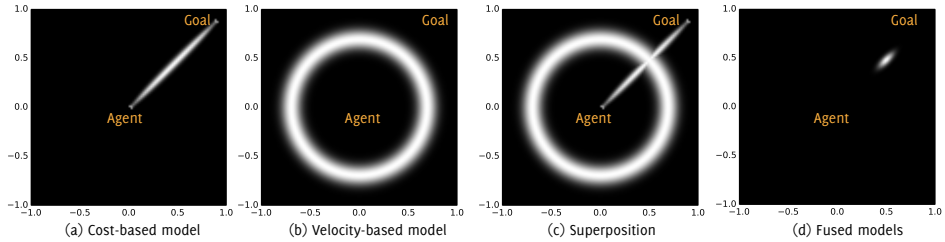

Fig. 3. Illustration of the fusion of a cost-based model predicting the minimum cost path to the goal (a); a velocity-based model predicting the traveled distance (b); their superposition (c); and the fusion results (d). A uniform costmap has been used for the sake of clarity.

For non-homogeneous cost-functions, we assume that the agent will cover a shorter distance if the average cost of its path is higher than that of its current position, but it will keep its speed otherwise:

$$D(\vec{X}_i, \vec{X}_j) = \max\left(\left\|\vec{X}_i - \vec{X}_j\right\|, L\frac{C_{\vec{X}_i \to \vec{X}_j}}{\mathcal{M}(\vec{X}_i)}\right) \quad (12)$$

where L is the side of a grid's cell, $C_{\vec{X}_i \to \vec{X}_j}$ is the cost to go from $\vec{X}_i$to $\vec{X}_j$, and $\mathcal{M}(\vec{X}_i)$ is the value of the costmap at $\vec{X}_i$.

D. Cost-Based Model

Our cost-based model aims to compute the probability that an agent passes through position $\vec{X}_{t+H}$ when traveling from its current position $\vec{X}_t$ to a goal G_t. The key hypothesis is that this probability depends on the *total cost* of the trajectory $C_{\vec{X}_t \to \vec{X}_{t+H} \to G_t}$. In particular, we assume trajectories to be normally distributed around the optimal cost[2]:

$$\mu_C(\vec{X}_t, G_t) = C_{\vec{X}_t \to G_t} \quad (13)$$

according to the following expression:

$$P(\vec{X}_{t+H}|\vec{X}_t, G_t) = \mathcal{N}\left(C_{\vec{X}_t \to \vec{X}_{t+H} \to G_t}; \mu_C(\vec{X}_t, G_t), \sigma_C^2\right) \quad (14)$$

where σ_C^2 is a model parameter which regulates the width of the path towards the goal.

The cost of the path that goes from $\vec{X}_t$ to G_t *passing through* $\vec{X}_{t+H}$ may be efficiently computed thanks to the triangle's inequality as the sum of the two partial costs:

$$C_{\vec{X}_t \to \vec{X}_{t+H} \to G_t} = C_{\vec{X}_{t+H} \to G_t} + C_{\vec{X}_t \to \vec{X}_{t+H}} \quad (15)$$

Since the first term has been precomputed (*c.f.* Sec. III-B), we only need to compute the second one. Assuming that costs are symmetrical (*i.e.* $C_{\vec{X}_i \to \vec{X}_j} = C_{\vec{X}_j \to \vec{X}_i}$), this only requires executing the planning algorithm once per prediction step for the entire grid.

E. Model Fusion

As they are formulated, both the cost-based model of Sec. III-D and the velocity-based one of Sec. III-C have the same dependent (*i.e.* left-side) variable, making it impossible to put them together in a joint probability decomposition. Bayesian Fusion with Diagnosis [14], provides an elegant and sound solution to this precise problem, which may be summarized as follows: for every fused model, we introduce a binary *consistency variable* with the following semantics: a value of one means that our model is *consistent with the observed data* and a value of zero means that it is not.

What is more relevant in our case, is that it provides a straightforward way of fusing the models by simply multiplying them together, as illustrated by Fig. 3.

In order to apply it to our particular model, we simply recast (11) as:

$$P(I_V = 1|\vec{X}_t, \vec{V}_t, \vec{X}_{t+H}) = \mathcal{N}\left(D(\vec{X}_t - \vec{X}_{t+H}); \mu_D(\vec{V}_t), \sigma_D^2\right) \quad (16)$$

which is the parametric form for (5).

Similarly, for (14):

$$P(I_C = 1|\vec{X}_t, G_t, \vec{X}_{t+H}) = \mathcal{N}\left(C_{\vec{X}_t \to \vec{X}_{t+H} \to G_t}; \mu_C(\vec{X}_t, G_t), \sigma_C^2\right) \quad (17)$$

which is the parameteric form for (6).

Given that both I_V and I_C are binary variables, the corresponding expressions for $I_V = 0$ and $I_C = 0$ may be obtained by subtracting the above probabilities from one. In our case, however, this is not necessary, since we simply set both variables to one querying the model to indicate that the model is consistent (which we believe it is) as indicated in Eq. (7).

IV. Experiments

We have conducted a preliminary comparison of our approach against MDPMP using the data set provided by Kitani [19] which is a subset of the one used in [11], which comes from the VIRAT ground video surveillance data set [15]. The published data contains 14 trajectories extracted from video recorded in a parking lot (Fig. 4). It includes the following information:

- *Trajectories*, obtained from a visual tracker and consisting of sequences of agent positions. Positions have been projected into the floor plane. In order to smooth them and compute velocities, we have post-processed this data by applying a standard Kalman filter. The resulting sequences have the form:

$$D_i = \{(\vec{X}_1, \vec{V}_1), \cdots, (\vec{X}_{T_i}, \vec{V}_{T_i})\}$$

  where T_i is the total number of points in trajectory i.
- *Cost functions*, the current *reward values* of the static environment are computed as a linear combination of

[2]An alternative to a normal distribution is discussed in Sec. V

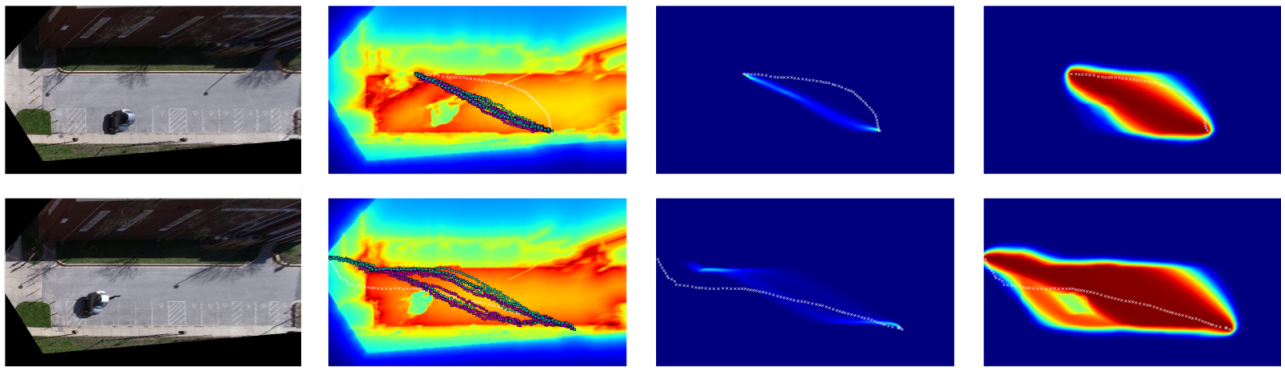

Fig. 4. Two examples of our results on the VIRAT data set(one per row). First column: video sample. Second column: reward heat map and five predicted trajectories sampled from each approach (blue for MPDMP, magenta for the proposed approach). Third column: MPDMP forecast distribution heat map. Fourth column: Proposed approach forecast distribution heat map. The white line in columns 2 through 4 correspond to the actual trajectory followed by the agent.

features which encode the context. The features have been obtained by post-processing the output of a semantic segmentation algorithm [20] on the floor-projected input images. The weights have been learned using an inverse reinforcement learning approach described in [11]. Finally, since our algorithm requires strictly positive cost values, the real-valued reward grid has been converted by applying the following identity:

$$C_{ij} = 1/(\epsilon + R_{ij} - \min_{l,m \in grid} R_{lm}) \tag{18}$$

Where C_{ij} and R_{ij} are the cost and the reward values of cell i, j and ϵ is a small regularization value (0.1 in our experiments) to avoid division by zero.

A. Implementation Details.

We have implemented the proposed approach in Python and C++. We have also developed an OpenCL-based GPU-accelerated version of the approach which, in our experiments, reduces the computation time by a factor of 5. For our algorithm, parameter values have been set by trial-and-error: $\Sigma_G^2 = 0.2$ and $\sigma_C^2 = 30$. For MPDMP, we have used the code and parameters provided by the authors [19].

We are currently cleaning up our code and it will be made publicly available at the time of the final version of this paper, if it is accepted.

B. Evaluation metrics.

The forecast distribution obtained in MPDMP is equivalent to the cost-based model of eq. 14 and both can be interpreted as representing the probability of an agent visiting a given cell when traveling from its current pose towards a destination. In our experiments, we aim to evaluate the accuracy of these distributions by comparing them against the actual observed motion. This is done by taking, for every trajectory in the data set, the start and end position and using the respective algorithms to compute the forecast distributions (third and fourth columns of Fig. 4), which are then compared against the complete input trajectory using the following metrics:

- *Negative log-loss (NLL)*: measures the expected log-likelihood of a trajectory given the model. In the case of MPDMP, it is computed as:

$$NLL(D_i) = E_{\pi(\vec{V}|\vec{X})}\left[-\log \prod_t \pi(\vec{V}_t|\vec{X}_t)\right] \tag{19}$$

  while, for our approach, it is computed from eq. 8 as:

$$NLL(D_i) = E_{P(\vec{V}|\vec{X},G)}\left[-\log \prod_t P(\vec{V}_t|\vec{X}_t, G_t)\right] \tag{20}$$

  in order to have a fair comparison with the 8-action MPDMP policy, we have discretized $P(\vec{V}_t|\vec{X}_t, G_t)$ into 8 values at intervals of $\frac{\pi}{4} rad$, with the initial angle aligned with the local gradient vector $\hat{\nabla}_{G_t}(\vec{X}_t)$.
- *Modified Hausdorff Distance (MHD) [21]*: used by Kitani *et al.* as an estimation of the distance between two trajectories.
- *Position negative log likelihood (PNLL)*: a distance-only spatial evaluation criterion like the MHD has the drawback that a high-cost trajectory and a low-cost one which are equidistant to the reference trajectory would have the same score. In order to address this problem, we propose to use the PNLL, defined as:

$$PNLL(D_i) = E_F\left[-log \prod_t F_{\vec{X}_t}\right] \tag{21}$$

  which is the expected value of the *normalized forecast distribution* F –obtained by normalizing the respective forecast/cost-based model grid– and $F_{\vec{X}_t}$ denotes the value of the cell containing $\vec{X}_t$.
- *Time*: the required time to compute the forecast distribution, excluding all precomputed values. Since we have no parallel version of MPDMP, we are using the sequential version of our algorithm in order to be fair.

C. Results.

Our results, summarized in Table I, show that, in our experiments, the proposed approach outperforms MDPMP's accuracy while requiring considerably less computation time. Moreover, the computed p-values indicate that our results are significant, with the possible exception of the MHD

metric, whose score is slightly higher than the our assumed significance level of 0.05.

TABLE I
METRIC SCORES (LOWER IS BETTER)

Metric	*MDPMP*	*Proposed*	*p-value*
NLL	1.38	**0.83**	6.8×10^{-5}
MHD	25.04	**20.62**	5.1×10^{-2}
PNLL	16.08	**9.82**	5.4×10^{-4}
Time	0.2 s	**0.03** s	1.5×10^{-11}

The accuracy difference may be better understood by looking at Fig. 4. We can see that our approach yields forecast distributions which are relatively wide, while MPDMP probabilities are much more less spread out. In consequence, they yield bad NLL and PLL scores when the actual trajectory is not in the narrow high-probability band. For instance, we can see that the actual agent trajectory of row one in Fig. 4 cannot be explained by the MPDMP forecast distribution, while in the proposed approach it has a low but significant probability. Similarly, in the second example we see that MPDMP completely excludes the possibility that the agent passes in front of the vehicle (lower part of the image), which is not the case for the proposed approach.

There is, however, at least one case in which our approach does not perform as well as MPDMP. It happens when the agent suddenly becomes partially obstructed or the other way around, leading to wrong velocity estimates which, given our algorithm's lack of filtering, are immediately reflected in the prediction. Yet, this problem is often corrected after a couple of frames.

Finally, we would like to stress out that these are preliminary results. Notably, we need to conduct further experiments to assess the respective differences concerning goal estimation.

V. CONCLUSIONS

This paper has introduced a novel approach for motion prediction allowing for full spatiotemporal prediction of human motion. Experimental results are very promising with our approach consistently outperforming the state-of-the-art alternative in terms of accuracy. As for computational cost, the sequential version of our algorithm ran 6 times faster than MPDMP while the speed up was of 30 times for the parallel version.

Current work focuses in two main aspects: (a) peforming more extensive experiments on real robots and evaluating goal prediction; and (b) improving the cost-based model, in particular by using a lognormal distribution instead of a normal one in order to take into acount the fact that, by definition, no trajectory may have a cost-to-go that is inferior to the optimal.

Future work will follow three main directions: (a) extending the framework to realize costmap learning *à la* Inverse Optimal Control; and (b) take advantage of the very fast prediction times ($10ms$ per prediction for the parallel version) in order to address the more complex and realistic case of agent interaction.

REFERENCES

[1] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech*, vol. 1, no. 1, July 2014.

[2] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1114–1127, 2008.

[3] H. Gong, J. Sim, M. Likhachev, and J. Shi, "Multi-hypothesis motion planning for visual object tracking," in *Proceedings of the 2011 International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 2011, pp. 619–626. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126296

[4] G. Ferrer and A. Sanfeliu, "Bayesian human motion intentionality prediction in urban environments," *Pattern Recognition Letters*, vol. 44, pp. 134–140, July 2014.

[5] Q. Zhu, "A stochastic algorithm for obstacle motion prediction in visual guidance of robot motion," in *IEEE International Conference on Systems Engineering*, 1990, pp. 216–219.

[6] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation," in *Robotics: Science and Systems*, Sydney, Australia, July 2012.

[7] S. Kim, S. J. Guy, W. Liu, R. W. Lau, M. C. Lin, and D. Manocha, "Predicting pedestrian trajectories using velocity-space reasoning," in *Tenth Workshop on the Algorithmic Foundations of Robotics*, ser. Springer Tracts in Advanced Robotics, E. Frazzoli, T. Lozano-Perez, N. Roy, and D. Rus, Eds., vol. 86. Springer Berlin Heidelberg, 2013, pp. 609–623.

[8] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *Internationl Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, January 2005.

[9] D. Vasquez, T. Fraichard, and C. Laugier, "Growing hidden markov models: a tool for incremental learning and prediction of motion," *International Journal of Robotics Research*, vol. 28, no. 11–12, pp. 1486–1506, 2009.

[10] B. D. Ziebart, N. Ratliff, G. Galagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *In Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, St. Louis (US), 2009, pp. 3931–3936.

[11] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity forecasting," in *European Conference on Computer Vision*, vol. 7575, 2012, pp. 201–214.

[12] K. Yamaguchi, A. Berg, L. E. Ortiz, and L. T. Berg, "Who are you with and where are you going?" in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1345–1352.

[13] J. Elfring, R. V. D. Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," *Journal of Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 591–602, April 2014.

[14] C. Pradalier, F. Colas, and P. Bessiere, "Expressing bayesian fusion as a product of distributions: Applications in robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2003, pp. 1851–1856.

[15] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3153–3160.

[16] J. N. Tsitsiklis, "Efficient algorithms for globally optimal trajectories," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1528–1538, 1995.

[17] J. A. Sethian, "Fast marching methods," *SIAM review*, vol. 41, no. 2, pp. 199–235, 1999.

[18] A. Treuille, S. Cooper, and Z. Popović, "Continuum crowds," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 1160–1168, 2006.

[19] K. Kitani, "Activity forecasting data set," http://www.cs.cmu.edu/~kkitani/datasets/, 2012, accessed: 2015/05/2015.

[20] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *European Conference on Computer Vision*. Springer, 2010, pp. 57–70.

[21] M.-P. Dubuisson and A. K. Jain, "A modified hausdorff distance for object matching," in *International Conference on Pattern Recognition*, vol. 1. IEEE, 1994, pp. 566–568.