April 10, 2014

Robot Navigation in Dense Human Crowds
Final Report

Neil Traft
University of British Columbia

Overview

In their 2010 IROS publication [1], Trautman and Krause develop a path planning algorithm that is safe and yet does not suffer from the "freezing robot problem" (FRP). Their method consists of a model of crowd interaction combined with a particle-based inference method to predict where the crowd (and the robot) should be at some time $t+1$ in the future. The idea is that if one can develop a reliable model of intelligent agents in a crowd, and include the robot as just another of those intelligent agents, then the predictions of the model yield the robot's future path.

The goal of this project is to reproduce their results in simulation on the original dataset. Given some annotated video of pedestrians in a crowd, we can choose one of the pedestrians to represent the robot, and compare the path planned by the robot to the actual path taken by the pedestrian.

Interacting Gaussian Processes

The crowd interaction model is a nonparametric stochastic model based on Gaussian processes, dubbed *Interacting Gaussian Processes* (IGP). In IGP, the actions of all agents, including the robot, are modeled as a joint distribution:

$$p(\mathbf{f}^{(R)}, \mathbf{f} | \mathbf{z}_{1:t})$$

where $\mathbf{f}^{(R)}$ is the robot's trajectory over T timesteps, $\mathbf{f}$ is the set of all human trajectories, and $\mathbf{z}_{1:t}$ is the set of all observations up to the current time point. For the purposes of this algorithm, observations of human and robot position are taken to be more or less perfect, since we are only trying to solve the navigation problem, not situational awareness.

At each timestep, each agent's new position is represented as a random variable from a probability distribution. It is important to note that this distribution is *not* Gaussian, due to two major additions to avoid the uncertainty explosion which leads to the FRP (see Figure 1).

First, goal information is given as a final "observation" at time T, resulting in the full set of observations $\mathbf{z}_{1:t,T}$. The robot's goal, $y_T^{(R)}$, is known and can be added with good confidence. The goals of other agents can be omitted or can be added with a high variance, to encode how uncertain we are about the goal.

The second addition IGP makes to standard Gaussian processes is the inclusion of an "interaction potential:"

$$\psi(\mathbf{f}^{(R)}, \mathbf{f}) = \prod_{i=1}^{n} \prod_{j=i+1}^{n} \prod_{\tau=t}^{T} \left(1 - \alpha \exp\left(-\frac{1}{2h^2} |\mathbf{f}_\tau^{(i)} - \mathbf{f}_\tau^{(j)}|\right)\right)$$

In essence, this potential grows very small whenever two agents i and j become very close at any time τ. This has the result that any set of paths where agents become too close is treated as very unlikely. The parameter h controls the desired "safety distance" and $\alpha \in [0,1]$ controls the "repelling force". Thus, the final posterior is given as:

$$p_{IGP}(\mathbf{f}^{(R)}, \mathbf{f}|\mathbf{z}_{1:t}) = \frac{1}{Z}\,\psi(\mathbf{f}^{(R)}, \mathbf{f}) \prod_{i=1}^{n} p(\mathbf{f}^{(i)}|\mathbf{z}_{1:t})$$

The above is a nonlinear, multimodal distribution, so it can't be sampled directly. Instead, we sample from Gaussian priors $p(\mathbf{f}^{(i)}|\mathbf{z}_{1:t})$ and resample weighted by our desired distribution (a particle filter). This is described in the next section.

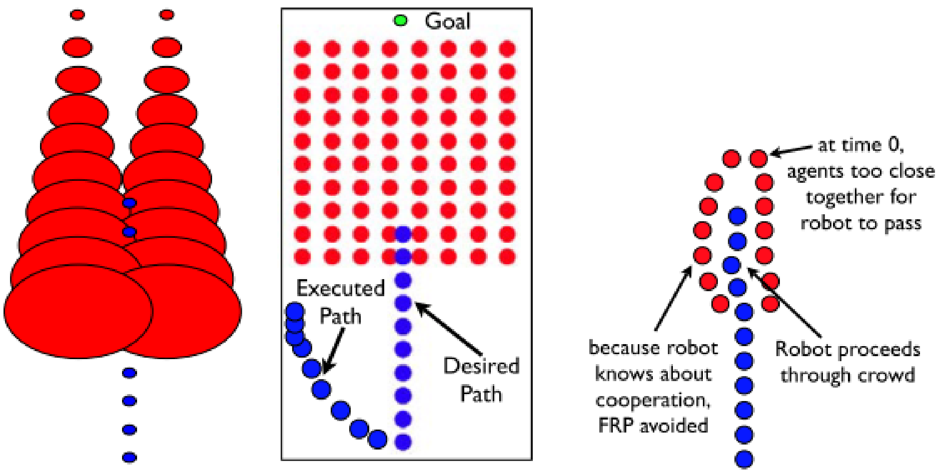


Figure 1 Diagrams taken from [1] describing the uncertainty explosion which leads to the Freezing Robot Problem. **Left:** Depicts the uncertainty explosion in standard motion models, where each agent's trajectory is *independent* from the others. **Middle:** A demonstration of why even *perfect* prediction, devoid of uncertainty, can still lead to the FRP. In crowded environments, *all* paths can have a high cost function, leading to extreme evasive maneuvers or freezing. **Right:** The ideal model, based on the insight that intelligent agents engage in *cooperative* collision avoidance.

Importance Sampling

Now that we have a model, we wish to sample from it and take the mean as the desired path. Since we can't sample from it directly, we instead use the *importance sampling* technique which is widely used in particle filters. Each sample is weighted by the ratio of the IGP to the basic GP (i.e. the Gaussian distribution, without the interaction potential):

$$w_i = \frac{p_{IGP}}{p_{GP}} = \frac{p_{IGP}((\mathbf{f}^{(R)}, \mathbf{f})|\mathbf{z}_{1:t})}{\prod_{j=R}^{n} p((\mathbf{f}^{(j)})_i|\mathbf{z}_{1:t})} = \frac{\psi((\mathbf{f}^{(R)}, \mathbf{f})_i)\prod_{j=R}^{n} p((\mathbf{f}^{(j)})_i|\mathbf{z}_{1:t})}{\prod_{j=R}^{n} p((\mathbf{f}^{(j)})_i|\mathbf{z}_{1:t})} = \psi((\mathbf{f}^{(j)})_i)$$

where $(\mathbf{f}^{(j)})_i$ is a single sample from the trajectory of agent j.

Given this formulation for p_{IGP} and an appropriate weighting w_i for each sample, the ideal paths can now be expressed as:

$$(\mathbf{f}^{(R)}, \mathbf{f})^* = \arg\max\left(\sum_{i=1}^{N} w_i(\mathbf{f}^{(R)}, \mathbf{f})_i\right)$$

where $(\mathbf{f}^{(R)}, \mathbf{f})_i$ is a set of samples from the Gaussian processes $(\mathbf{f}^{(j)})_i \sim \mathrm{GP}(\mathbf{f}^{(j)}, m_t^{(j)}, k_t^{(j)})$ (detailed equations for the mean and covariance matrix can be found in the original paper). The total number of samples is N and we take the robot's next position to be $\mathbf{f}_{t+1}^{(R)}$. To approximate the optimal robot path $\mathbf{f}^{(R)}$, we take the mean path over all samples after importance resampling:

$$\mathrm{f}_t^{(R)*} = \frac{\sum_{i=1}^{N} \mathrm{f}_t^{(R)}}{N}$$

Implementation

The project is implemented in Python, using the OpenCV library [2] for visualization and video playback (but not for any of the vision algorithms it provides).

The Dataset

The dataset to be used is the ETH Walking Pedestrians (EWAP) dataset from [3]. It can be obtained from [4].

The dataset contains two annotated videos of pedestrian interaction captured from a bird's eye view. The one used depicts pedestrians entering and exiting a building.

The main annotations are a matrix where each row has the format:

```
[frame_number pedestrian_ID pos_x pos_z pos_y v_x v_z v_y]
```

Thus for each frame t we have potentially multiple pedestrian observations $\mathrm{pos}_t^{(i)}$, and this forms our observation at time t:

$$\mathbf{z}_t = \mathrm{pos}_t^{(1:n)}$$

where one of the n pedestrians is chosen to represent the robot R. The velocities are not used in the present IGP formulation.

The positions and velocities are in meters and were obtained with a homography matrix H, which is also provided with the annotations. To transform the positions back to image coordinates, it is necessary to apply the inverse homography transform:

$$^{m}\mathrm{pos}_t^{(i)} = H_{mw}^{-1} \cdot {}^{w}\mathrm{pos}_t^{(i)} \qquad m = \mathrm{image}, w = \mathrm{world}$$

The intrinsic camera parameters are not needed; it is presumed that they are included in the camera matrix provided by the dataset. There is also no translation needed; the origin of the world can be taken to be the (transformed) origin of the image without loss of generality. Thus, pixel coordinates can be expressed as

$$r = -f_x \frac{x}{z} = \frac{u}{z} \qquad c = -f_y \frac{y}{z} = \frac{v}{z}$$

where $f_x,\ f_y$ are the intrinsic camera parameters and $x,\ y$ are coordinates in the image frame. So to obtain pixel coordinates from image coordinates it is necessary only to normalize so that $z = 1$:

$$\begin{pmatrix} r \\ c \\ 1 \end{pmatrix} = \begin{pmatrix} {}^{m}x/z \\ {}^{m}y/z \\ {}^{m}z/z \end{pmatrix}$$

Technical Hurdles

Experimental Results

Next Steps

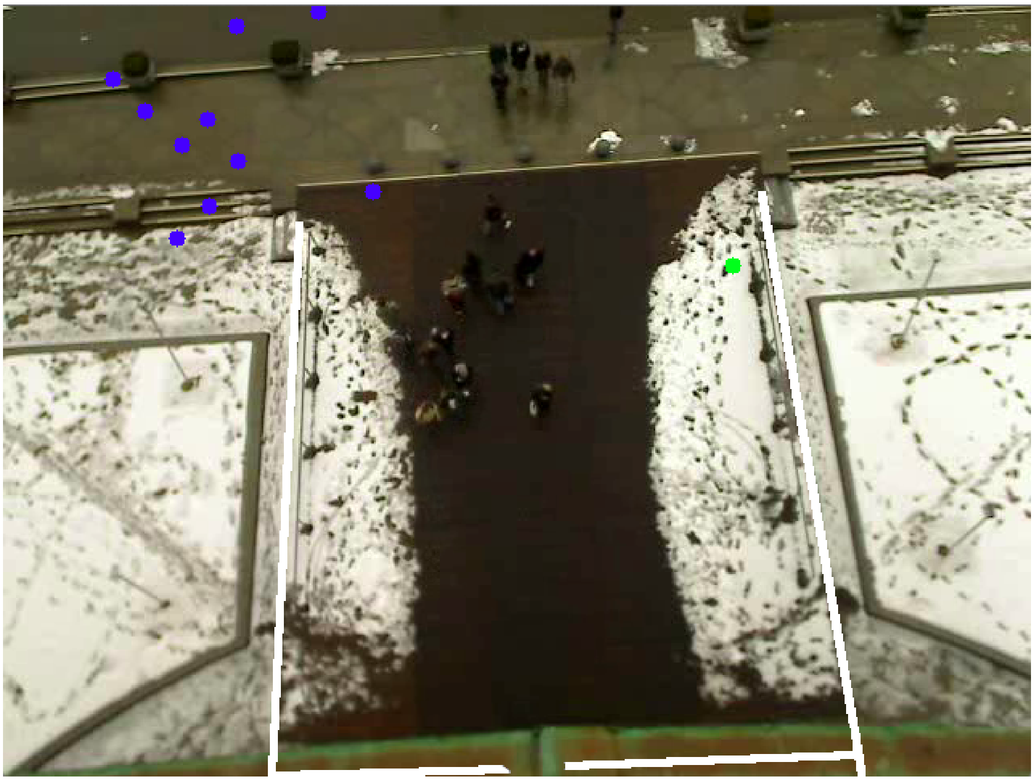
References

[1] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 797–803, Oct. 2010.

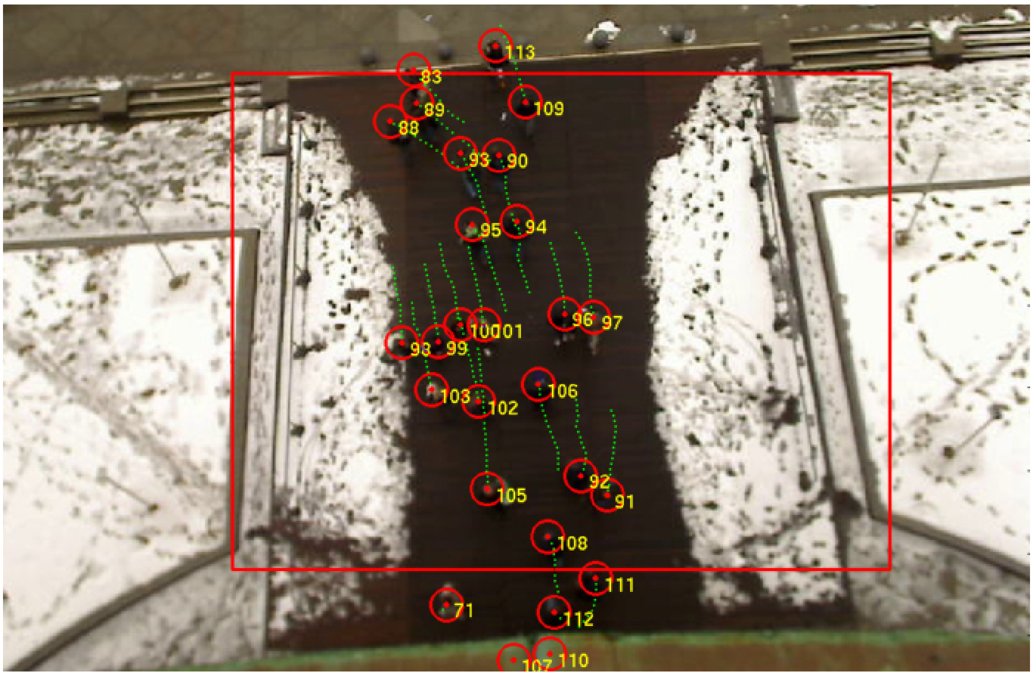
[2] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[3] S. Pellegrini, a. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," *2009 IEEE 12th International Conference on Computer Vision*, pp. 261–268, Sept. 2009.

[4] "Ethz - computer vision lab: Datasets." `http://www.vision.ee.ethz.ch/datasets/index.en.html`. Accessed: 2014-03-01.



(a) A sample frame from my current visualization.



(b) A sample frame from the ETH paper. [3]

Figure 2 A sample frame from my (buggy) implementation (2a) compared with a correct implementation (2b). In the top image, blue dots mark pedestrians, white lines mark static obstacles, and green dots mark possible destinations. If examined closely, one can see the pedestrian markings in my video are correct in relation to each other, but appear scaled and translated within the image plane. The destinations appear to have the same issue (only one destination is visible but there are about 4 total).