Learning 3D Models of Dynamic Objects: Preliminaries

Sebastian

July 28, 2002

1 Introduction

We are interested in learning models of dynamic objects from multiple color and range images. A robot scans a static scene using cameras and a 3D laser range scanner, and analyzes the scene for differences in depth. A segmentation routine is used to extract scans that correspond to objects that moved between different scans. The problem now is to recover a full 3D model of those objects. For simplicity, we assume that there is just one dynamic object in the world, which is present in every scan (see [1, 2] for an extension to multiple objects).

2 Model

The **model** consists of two parts: 3D occupancy grid m and a 3D color map θ. Each occupancy grid cell m_j is a probability that this grid cell might show up in a range scan. Each color map pixel θ_j is a three-dimensional RGB vector that describes the expected measurement should this pixel be detected by camera. Clearly, θ_j is only meaningful if $m_j > 0$; otherwise it will never be detected. The index j is three-dimensional, that is, our model is volumetric.

3 Measurements

At each point in time t the robot selects a subsets of scans and corresponding image pixels to correspond to the dynamic objects. The set of all those range scans at time t will be denoted z_t. The corresponding pixels from the camera image are denoted ϕ. The variable n will be used to refer to a specific range value $z_{t,n}$ and a specific RGB vector $\phi_{t,n}$. The variable n is a two-dimensional index, since camera and range images are two-dimensional.

4 Alignment

To reconstruct the model, it will be necessary to align the measurement data z_t and ϕ_t with the coordinate frame of the model m and θ. The alignment of the t-the measurement will be denoted x_t. If we assume that each object is rigid and always stands upright on a planar surface—which is a reasonable assumption for most office furniture—each x_t is a three-dimensional displacement. The assumption that x_t is inessential for the mathematical framework, but will be essential to attain computation efficiency when calculating various quantities below.

The alignment variables x_t enable us to map measurement coordinates into model coordinates. In particular, this will be expressed by the function

$$f(x_t, n, k) \quad = \quad j \tag{1}$$

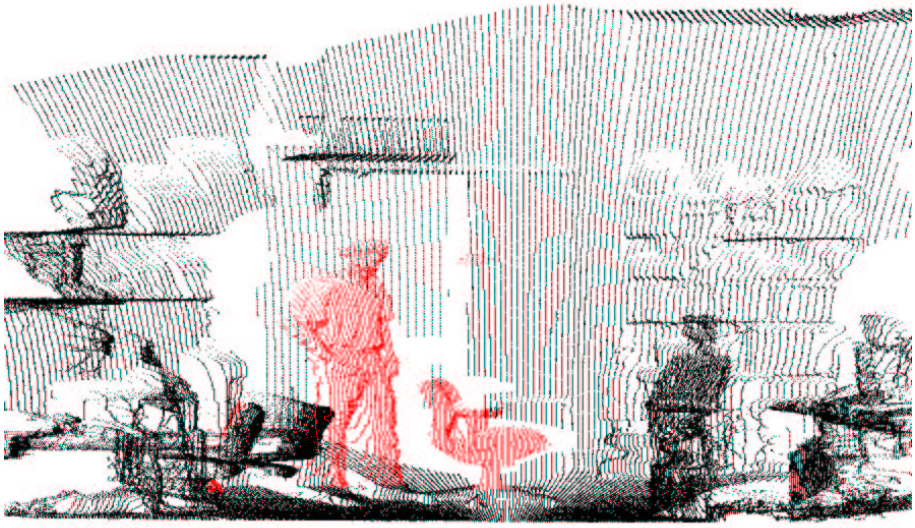

Figure 1: 3D range image. Shown in red is a dynamic object, segmented through range scan differencing. The scans and corresponding camera pixels that correspond to dynamic objects form the input to our algorithm.

where x_t is an alignment vector, n the index of a pixel int he range/camera image recorded at time t, and k a specific range value.

5 Measurement Model

Measurements z_t and ϕ_t are probabilistic projection of the model, and as such depend on the parameters m, θ, and x_t. In particular, we assume a model in which range measurements are generated by a beam traveling through space, and being reflected with probability m_j by doing so, for appropriate indexes j. The camera pixel is generated by the pixel in the object that is finally detected by the range sensor, but we assume that it corrupted by Gaussian noise with zero mean and covariance Σ.

Put mathematically, we have

$$\begin{aligned} p(z_{t,n}, \phi_{t,n} \mid x_t, m, \theta) \quad &\propto \quad m_{f(x_t,n,z_{t,n})} \left[\prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t,n,k)}) \right] \\ &\exp \left\{ -\tfrac{1}{2} (\phi_{t,n} - \theta_{f(x_t,n,z_{t,n})})^T \Sigma^{-1} (\phi_{t,n} - \theta_{f(x_t,n,z_{t,n})}) \right\} \end{aligned} \tag{2}$$

Notice that the joint over the measurements and the alignment parameters is given by

$$p(z_{t,n}, \phi_{t,n}, x_t \mid m, \theta) \quad \propto \quad p(z_{t,n}, \phi_{t,n} \mid x_t, m, \theta)\; p(x_t \mid m, \theta) \tag{3}$$

$$= \quad p(z_{t,n}, \phi_{t,n} \mid x_t, m, \theta)\; p(x_t) \tag{4}$$

Assuming uniform prior over x_t, we notice that (2) is proportional to the joint $p(z_{t,n}, \phi_{t,n}, x_t \mid m, \theta)$.

6 Expected Log Likelihood

The log likelihood of the data conditioned on the alignments $x = \{x_t\}$ and the model m and θ is now obtained as follows. Here we assume conditional independence of the measurements at different points

in time:

$$\log p(z,\phi,x \mid m,\theta) = \sum_t \sum_n \log p(z_{t,n},\phi_{t,n} \mid x_t,m,\theta) \quad (5)$$

$$= \text{const.} + \sum_t \sum_n \log m_{f(x_t,n,z_{t,n})} + \left[\sum_{k=0}^{z_{t,n}-1} \log(1-m_{f(x_t,n,k)}) \right] - \tfrac{1}{2}(\phi_{t,n}-\theta_{f(x_t,n,z_{t,n})})^T \Sigma^{-1} (\phi_{t,n}-\theta_{f(x_t,n,z_{t,n})}) \quad (6)$$

The expectation over the alignment variables thus gives us the following expression:

$$E_x\left[\log p(z,\phi,x \mid m,\theta) \mid z,\phi,m,\theta\right]$$

$$= \int p(x \mid z,\phi,m,\theta) \left\{ \text{const.} + \sum_t \sum_n \log m_{f(x_t,n,z_{t,n})} + \sum_{k=0}^{z_{t,n}-1} \log(1-m_{f(x_t,n,k)}) - \tfrac{1}{2}\,(\phi_{t,n}-\theta_{f(x_t,n,z_{t,n})})^T \Sigma^{-1} (\phi_{t,n}-\theta_{f(x_t,n,z_{t,n})}) \right\} dx \quad (7)$$

$$= \text{const.} + \sum_t \int p(x_t \mid z_t,\phi_t,m,\theta) \sum_n \left\{ \log m_{f(x_t,n,z_{t,n})} + \sum_{k=0}^{z_{t,n}-1} \log(1-m_{f(x_t,n,k)}) - \tfrac{1}{2}(\phi_{t,n}-\theta_{f(x_t,n,z_{t,n})})^T \Sigma^{-1} (\phi_{t,n}-\theta_{f(x_t,n,z_{t,n})}) \right\} dx_t \quad (8)$$

Optimizing this expected log likelihood function shall provide us with a model m and θ, along with expectations over our latent alignment parameters x_t.

7 Expectation Maximization

We optimize (8) using Dempster's EM algorithm [**?**]. This algorithm iterates two steps, an E-step and an M-step, in which expectations over the latent variables are calculated, and the model is optimized under these expectations, respectively. Iteratively applying EM leads to a sequence of models

$$\langle m^{[0]},\theta^{[0]}\rangle, \langle m^{[1]},\theta^{[1]}\rangle, \langle m^{[2]},\theta^{[2]}\rangle, \ldots \quad (9)$$

that gradually increase the likelihood of the data.

7.1 The E-Step

Here we seek to calculate the probability

$$\epsilon_{x_t}^{[N]} := p(x_t \mid z_t,\phi_t,m^{[N]},\theta^{[N]}) \quad (10)$$

$$\propto p(z_t,\phi_t \mid x_t,m^{[N]},\theta^{[N]})\; p(x_t) \quad (11)$$

$$\propto p(z_t,\phi_t \mid x_t,m^{[N]},\theta^{[N]}) \quad (12)$$

Our simplifications exploited our uniform prior assumption on the alignments x_t. The resulting probability was already defined through (2):

$$p(z_t,\phi_t \mid x_t,m^{[N]},\theta^{[N]}) \propto \prod_n m^{[N]}_{f(x_t,n,z_{t,n})} \left[\prod_{k=0}^{z_{t,n}-1} (1-m^{[N]}_{f(x_t,n,k)}) \right] \exp\left\{ -\tfrac{1}{2}(\phi_{t,n}-\theta^{[N]}_{f(x_t,n,z_{t,n})})^T \Sigma^{-1} (\phi_{t,n}-\theta^{[N]}_{f(x_t,n,z_{t,n})}) \right\} \quad (13)$$

This expression is easily calculated by iterating through all range and camera pixels, and multiplying the resulting measurement probabilities.

7.2 The M-Step

For the M-step, we now calculate a new model $\langle m^{[N+1]}, \theta^{[N+1]} \rangle$ by maximizing our expected log likelihood (8), but using the fixed expectations $\epsilon_{x_t}^{[N]}$ calculated in the E-step:

$$\sum_t \int \epsilon_{x_t}^{[N]} \sum_n \Bigg\{ \log m_{f(x_t,n,z_{t,n})} + \sum_{k=0}^{z_{t,n}-1} \log(1 - m_{f(x_t,n,k)}) \\ -\tfrac{1}{2}(\phi_{t,n} - \theta_{f(x_t,n,z_{t,n})})^T \Sigma^{-1} (\phi_{t,n} - \theta_{f(x_t,n,z_{t,n})}) \Bigg\} \; dx_t \tag{14}$$

For the optimization, it shall prove convenient to rearrange this expression by the pixels j in the model. This is achieved by introducing an indicator variable I:

$$\begin{aligned} = \quad & \sum_j \Bigg[\log m_j \underbrace{\sum_t \int \epsilon_{x_t}^{[N]} \sum_n I(j = f(x_t, n, z_{t,n}))\; dx_t}_{=:g_j} \\ & + \log(1 - m_j) \underbrace{\sum_t \int \epsilon_{x_t}^{[N]} \sum_n \sum_{k=0}^{z_{t,n}-1} I(j = f(x_t, n, k))\; dx_t}_{=:h_j} \\ & - \tfrac{1}{2} \sum_t \sum_n (\phi_{t,n} - \theta_j)^T \Sigma^{-1} (\phi_{t,n} - \theta_j) \underbrace{\int I(j = f(x_t, n, z_{t,n})) \epsilon_{x_t}^{[N]}\; dx_t}_{=:q_{j,t,n}} \Bigg] \end{aligned} \tag{15}$$

$$= \quad \sum_j g_j \log m_j + h_j \log(1 - m_j) - \tfrac{1}{2} \sum_t \sum_n q_{j,t,n} (\phi_{t,n} - \theta_j)^T \Sigma^{-1} (\phi_{t,n} - \theta_j) \quad =: \quad L^{[N]} \tag{16}$$

The maximum of this expression is attained where the first derivative is zero. This can be done independently for each grid cell index j, and for the model components m and θ. For the occupancy grid cell m_j we obtain the following weighted average:

$$\frac{\partial L^{[N]}}{\partial m_j} \quad = \quad \frac{g_j}{m_j} - \frac{h_j}{1 - m_j} \quad \stackrel{!}{=} 0 \tag{17}$$

$$\iff \quad m_j^{[N+1]} \quad = \quad \frac{g_j}{g_j + h_j} \tag{18}$$

For the color vector θ_j we also obtain a weighted average:

$$\frac{\partial L^{[N]}}{\partial \theta_j} \quad = \quad \sum_t \sum_n q_{j,t,n} (\phi_{t,n} - \theta_j)^T \Sigma^{-1} \quad \stackrel{!}{=} 0 \tag{19}$$

$$\iff \quad \theta_j \quad = \quad \sum_t \sum_n q_{j,t,n} \phi_{t,n} \tag{20}$$

References

[1] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and S. Thrun. Learning hierarchical object maps of non-stationary environments with mobile robots. In *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, 2002.

[2] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun. Towards object mapping in dynamic environments with mobile robots. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, 2002.