Sparse Depth Super Resolution

Jiajun Lu[1], David Forsyth[2]
[1]University of Illinois at Urbana Champaign. [2]University of Illinois at Urbana Champaign.

Recent work in HCI has demonstrated a variety of potential applications for depth sensors on mobile devices (gesture interfaces [2]; deictic references in augmented reality [3]), and several such sensors are in production. However, the relatively high power consumption of current depth sensors even at relatively low resolutions presents major difficulties in making these applications practical. Besides that, further improving the resolution of current depth sensors has high cost, while high resolution RGB images are comparatively cheap.

This paper describes methods to reconstruct high-resolution depth measurements accurately from sparse scattered samples, see Figure 1. We propose to exploit image (resp. video) data obtained at the same time as the depth samples to produce a spatial model that governs our smoothing of depth samples.

Contributions: 1) We demonstrate methods that can produce good depth from aggressive subsampling (for example, one depth sample per 4096 pixels) on both images and videos, outperforming recent strong methods and pushing forward upscale ratio. 2) Our experimental work is conducted on three different widely used publicly available datasets, and one novel dataset that we collected, and previous methods only work on limited data. 3) Our depths with big upscale ratio are successfully used by applications such as hand trackers, while other methods have problems. 4) Even though our methods are primarily aimed at improving either resolution or power use of active depth sensors by upsampling sensed depth maps, our spatial model is powerful that it can be used to improve the results of existing depth-from-image methods.

The segmenter of [1] is a form of agglomerative cluster, and so produces a tree of region merges. Each image segmentation is a choice of the region merges in the segmentation tree. We have depth samples, which is useful in identifying segmentations, so we use depth samples to guide the process of region merging. For each image, we first build the segmentation tree with fixed number of levels and minimum area sizes for each dataset. Then, we start with the level that contains the largest segments, and recursively for each segment use a consistency function $C_s(\mathbf{c}_i, \mathbf{c}_j)$ to decide whether go down to the next level of the segmentation tree.

$$C_s(\mathbf{c}_i,\mathbf{c}_j) = \begin{cases} \mu(e - e^{\gamma(\mathbf{c}_i,\mathbf{c}_j;\delta)}) & \mathbf{c}_i, \mathbf{c}_j \text{ share a segment} \\ e^{\gamma(\mathbf{c}_i,\mathbf{c}_j;\delta)} & \text{otherwise} \end{cases}.$$

Now we have a set of image segments, and a collection of depth samples. For each segment, we will smooth the samples inside the segment to form a dense depth map. By using only the samples inside the segment, we can obtain sharp depth boundaries at image segment boundaries. Once each segment has a depth map, we compute an overall depth map by copying the depths from each segment to the image plane. Our **simple depth smoothing** algorithm is a form of scattered data interpolation. Our **advanced depth smoothing** algorithm allows some large derivatives of depth. Here is a brief description of the advanced depth smoothing.

Two basis functions are used in the advanced depth smoothing. $\phi(\mathbf{x};\mathbf{c}_i) = f(\|\mathbf{x}-\mathbf{c}_i\|)$ is the radial basis function centered at $\mathbf{c}_i$. and $f(u) = \max(1 - \frac{u}{d\max}, 0)^2$.

$$\beta(\mathbf{x};\mathbf{c}_i) = \frac{\phi(\mathbf{x};\mathbf{c}_i)}{\sum_j \phi(\mathbf{x};\mathbf{c}_j)}$$

$$\psi(\mathbf{x};\mathbf{c}_i,\mathbf{u}_i) = \phi(\mathbf{x};\mathbf{c}_i)\,(\mathbf{u}\cdot(\mathbf{x}-\mathbf{c}_i))$$

Our smoothed depth model becomes

$$z(\mathbf{x};\mathbf{a},\mathbf{b},\mathbf{u}) = \left(\sum_i a_i\beta(\mathbf{x};\mathbf{c}_i)\right) + \left(\sum_i b_i\psi(\mathbf{x};\mathbf{c}_i,\mathbf{u}_i)\right) + z_\pi(\mathbf{x}).$$

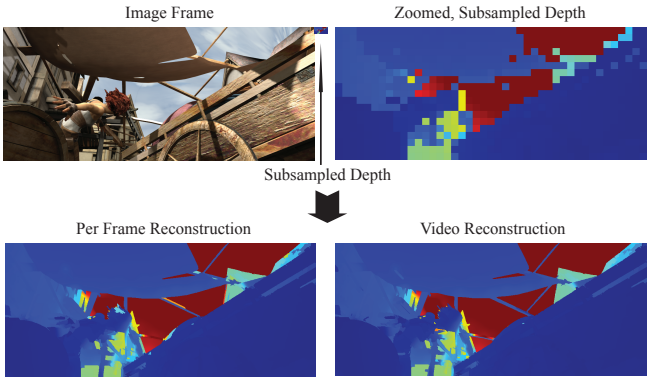

Figure 1: We describe a method to reconstruct high resolution depth maps from aggressively subsampled data, using a smoothing method that exploits image segment information to preserve depth boundaries. Our method is evaluated on four different datasets, and produces state of art results. This example shows a case where there is one depth sample per 24 × 24 block of image pixels (the tiny inset shows the depth map on the same scale as the image). Our method can exploit optic flow and space-time segmentation to produce improved reconstructions for video data.

The parameters are chosen by optimization over three energy terms, and the weights are chosen by cross-validation.

$$E_1(\mathbf{a},\mathbf{b},\mathbf{u}) = \sum_j \left(z(\mathbf{c}_j;\mathbf{a},\mathbf{b},\mathbf{u}) - z_s(\mathbf{c}_j)\right)^2$$

$$E_2(\mathbf{a},\mathbf{b},\mathbf{u}) = \|\mathbf{b}\|_1$$

$$E_3(\mathbf{a},\mathbf{b},\mathbf{u}) = \sum_k \sum_{c_i,c_j\in\mathcal{N}(x_k)} \begin{pmatrix} a_i\phi(\mathbf{x}_k;\mathbf{c}_i) + b_i\psi(\mathbf{x}_k;\mathbf{c}_i,\mathbf{u}_i) \\ -a_j\phi(\mathbf{x}_k;\mathbf{c}_j) - b_j\psi(\mathbf{x}_k;\mathbf{c}_j,\mathbf{u}_j) \end{pmatrix}^2,$$

$$\underset{\mathbf{a},\mathbf{b},\mathbf{u}}{\operatorname{argmin}} \ \lambda_1 E_1 + \lambda_2 E_2 + \lambda_3 E_3$$

Our method also applies to video data with some modifications. We use space time segments, because we expect depth to be fairly smooth within a space time segment, but change on its boundaries. We reconstruct from depth samples that are time-stamped. First, consider points nearby in space. We expect the depth at these points to be similar. But temporal smoothing is somewhat different. At each depth sample, we can compute optic flow, which is used to predicts the location $\mathbf{c}_i(t_i, T)$ of the sample forward and backward in time. For some time interval, we can trust these flow-based predictions, so we transport depth samples along the flow direction before interpolation. We allow the sample to have influence for times up to δt in the future and $-\delta t$ in the past, and the weighting $\omega(\Delta t;\delta t, C)$ of a sample decline as the inter-frame time interval increases.

$$\omega(\Delta t;\delta t,C) = \frac{\min(-\log(|\Delta t|)/\delta t, C)}{C}$$

$$\mathbf{c}_i(t_i,T) = \mathbf{c}_i(t_i,T-\Delta t) + \sum_{t_d=1}^{\Delta t} \mathbf{v}(\mathbf{c}_i(t_i,T-t_d)).$$

$$\sum_i \left[\begin{pmatrix} a_i\beta(\mathbf{x};\mathbf{c}_i(t_i,T)) + \\ b_i\psi(\mathbf{x};\mathbf{c}_i(t_i,T),\mathbf{u}_i) \end{pmatrix} \omega(T-t_i;\delta t,C) \right] + z_\pi(\mathbf{x}).$$

Our method yields state of the art results compared to strong recent methods in both image depth super resolution and video depth super resolution, and the simple smoothing version could be applied to real time system.

[1] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV, 2004*, 59(2):167–181.

[2] B. Jones, R. Sodhi, D. Forsyth, B. Bailey, and G. Maciocci. Around device interaction for multiscale navigation. In *MobileHCI*, 2012.

[3] R. Sodhi, B. Jones, D. Forsyth, B. Bailey, and G. Macciocci. Bethere: 3d mobile collaboration with spatial input. In *SIGCHI*, 2013.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.