Upsampling Range Data in Dynamic Environments

Jennifer Dolson, Jongmin Baek, Christian Plagemann, and Sebastian Thrun

Department of Computer Science, Stanford University

Abstract

We present a flexible method for fusing information from optical and range sensors based on an accelerated high-dimensional filtering approach. Our system takes as input a sequence of monocular camera images as well as a stream of sparse range measurements as obtained from a laser or other sensor system. In contrast with existing approaches, we do not assume that the depth and color data streams have the same data rates or that the observed scene is fully static. Our method produces a dense, high-resolution depth map of the scene, automatically generating confidence values for every interpolated depth point. We describe how to integrate priors on object motion and appearance and how to achieve an efficient implementation using parallel processing hardware such as GPUs, resulting in real-time frame rates.

GPU Implementation

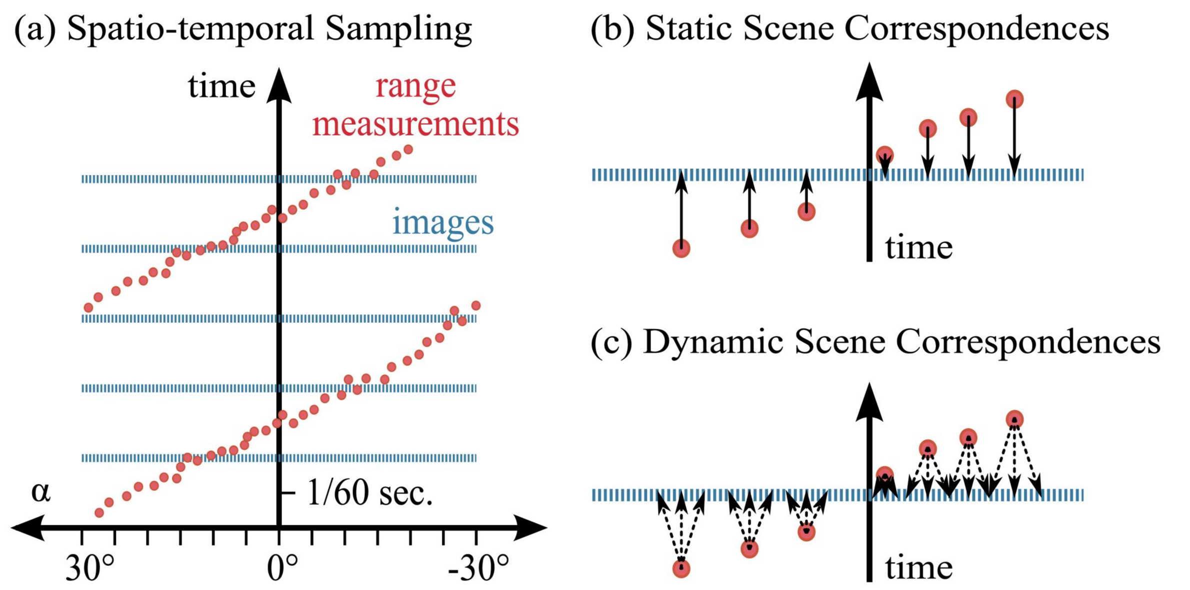
We implemented our d-dimensional Gaussian framework in CUDA on the GPU by modifying the permutohedral lattice data structure of Adams et. al.

The algorithm operates on a stream of color images each of which is coupled with a its relevant, sparse depth information. For each pair, we construct a d-dimensional data structure using the pair itself and two previous and subsequent pairs, for a total of five pairs. We then blur the d-dimensional space and begin the query stage. For our data, this amounts to roughly 27 000 depth points in our buffer and 1024x768 queries made per frame (one query per pixel). We use a desktop graphics card: NVIDIA GTX260 Core 216; copying the data onto the graphics card device memory and applying the GPU algorithm runs at 29.2 fps.

Code and data sets available:

http : //graphics.stanford.edu/papers/upsampling_cvpr10/

Data Fusion



The fusion of image and range information is difficult because different parts of the scene are observed at different points in time. (a) visualizes the spatio-temporal sampling behavior of cameras and typical range scanning devices. (b) shows that correspondences between measurements are easy to establish for static scenes and (c) shows that this is not the case for dynamic scenes, since a measured object may have moved in space.

d-dimensional Interpolation

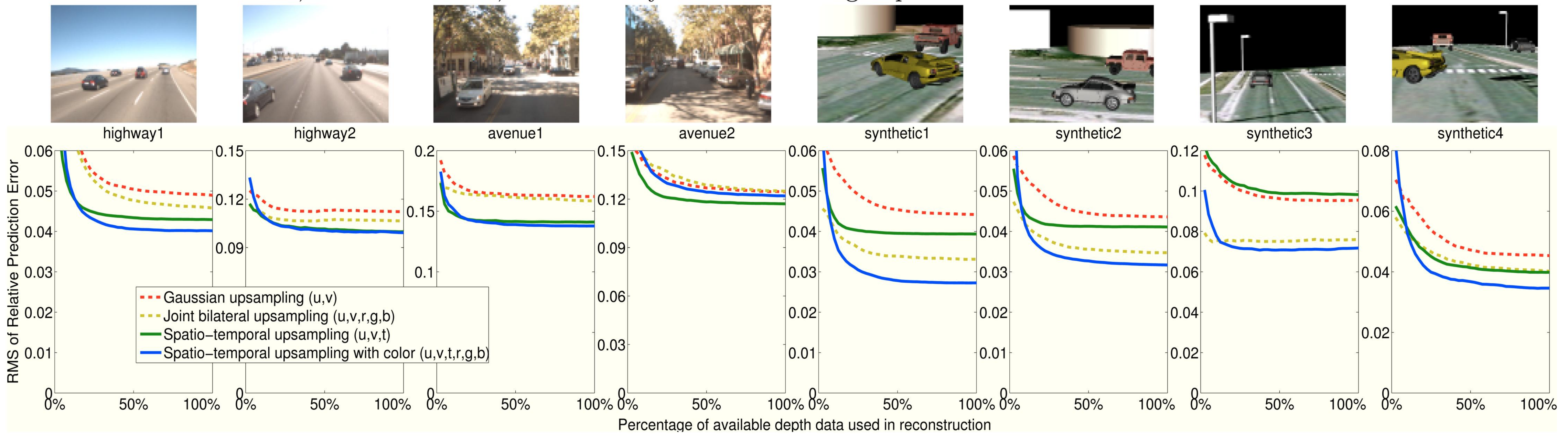
Many image operators such as *blurring*, *bilateral filtering*, and *non-local means denoising* can be grouped into a general class of d-dimensional filters, formalized as:

$$\hat{v}_i = \sum_{j=1}^{n} f(|p_i - p_j|) \cdot v_j$$

To create a high-resolution depth map, the general d-dimensional filtering formulation given above remains the same, but v_i and v_j = depth values, position vectors $= (\rho, u, v, t)$; ρ = descriptor based on color information, (u, v) = the 2D position of the descriptor on the image plane, and t = time of the data capture. We use a high-dimensional data structure to efficiently solve for v_i at pixels where depth information does not exist, accessing neighboring values in (ρ, u, v, t) space.

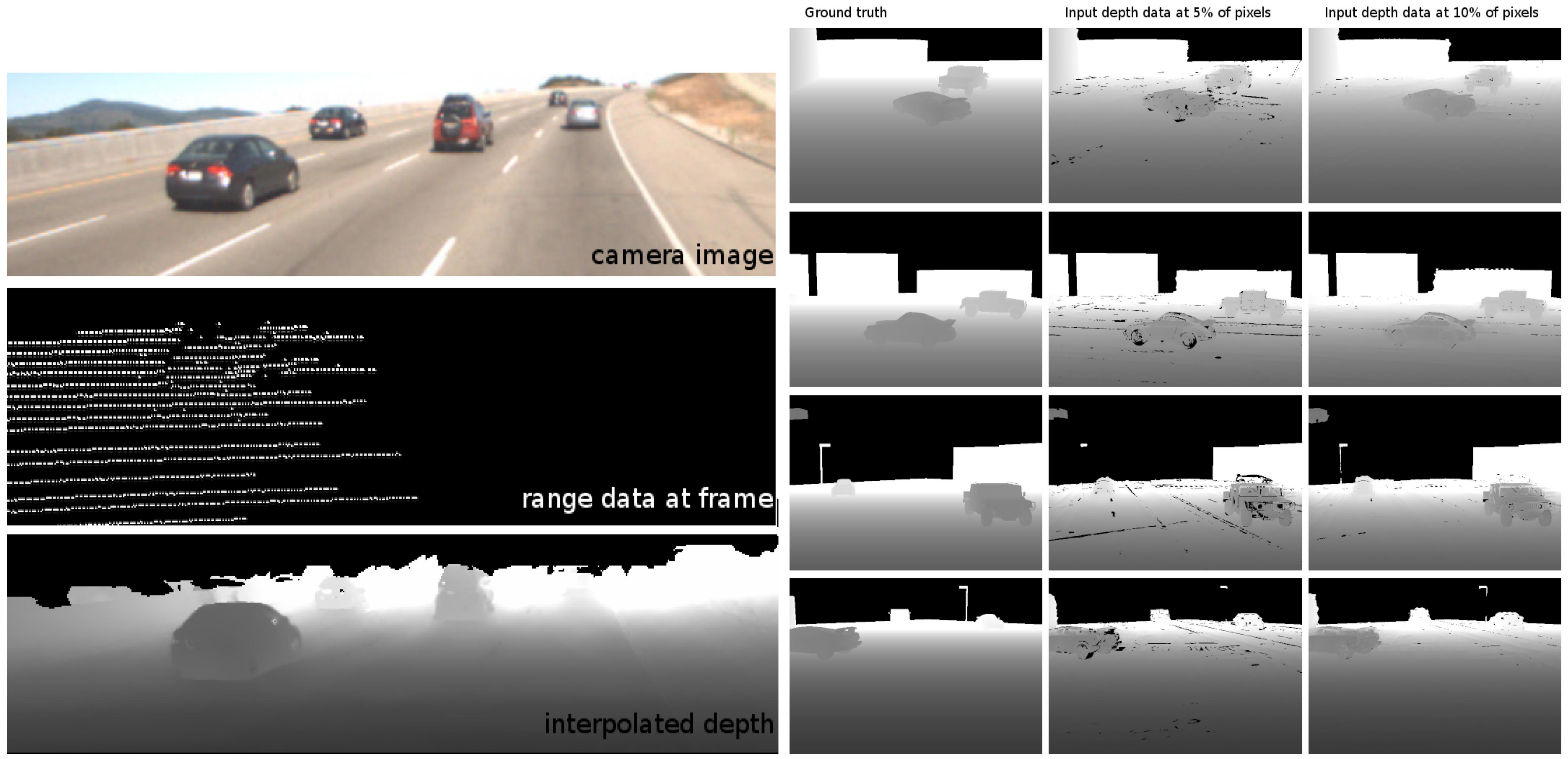
Evaluation and Results

Here we show the input buffer used in our non-synthetic evaluations. The white points indicate pixels where both depth and color information is available. We hold out depth information from frame 4, bordered in red, and use only the surrounding depth information



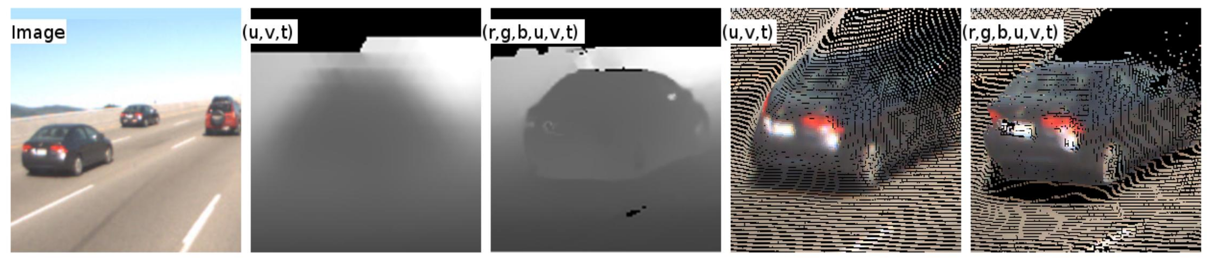
Above we show the convergence behavior of the different algorithms with respect to available range measurement densities by additionally subsampling the range measurements. Range points were randomly sampled from a uniform distribution. The horizontal axes of each diagram gives the amount of range measurements visible to the algorithms. Note that even a value of 1, that is, the case in which all range measurements are available, means that only 5% of all image pixels were within one pixel of a range measurement. The experiments were run 10 times for each sequence and the average relative RMS prediction errors are plotted. The standard deviations of the errors over the 10 trials were between 1% and 5% of each average, assuring that the average error shown is indicative of the performance of a typical execution.

Upsampling Temporally and Spatially



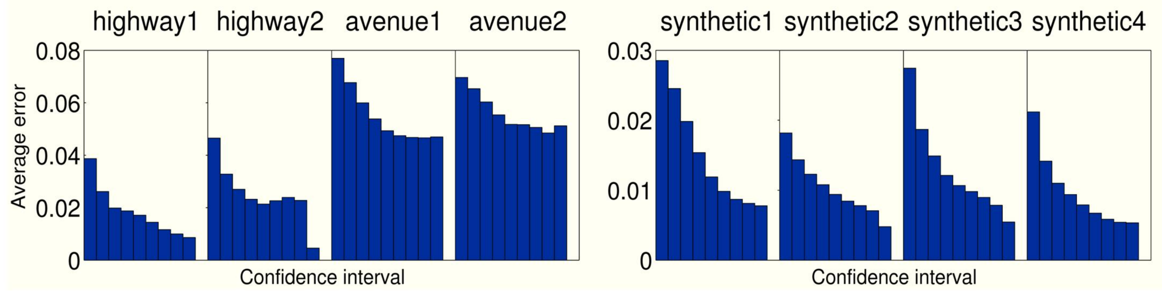
For any given camera frame, we can recover an accurate, dense depth map. The depth map shown in the lower left panel corresponds to a single camera frame from a sequence of highway images, recorded from a mobile platform. The right images are depth maps generated from synthetic data of increasing density. We can generate a depth map for an arbitrary camera frame even if the coincident depth data is sparse, or missing, using depth data from neighboring frames.

The figure below compares depth maps generated by our algorithm with color information and without. Without color information, blending occurs across depth discontinuities.



Confidence Weighting

The confidence value at each pixel is equal to the sum of weights returned from a query at that pixel.



Each plot above shows how the depth interpolation error correlates with the confidence values. For each data set, the reconstructed depth values were binned according to their corresponding confidence values, and the average error was calculated for each bin. Higher confidence estimates lead to lower actual errors on average.