3D Reconstruction from Accidental Motion

Project Report - Computer Vision

AadilMehdi Sanchawala
aadilmehdi.s@students.iiit.ac.in

Rahul Sajnani
rahul.sajnani@research.iiit.ac.in

Rohan Chacko
rohan.chacko@students.iiit.ac.in

Abstract

We tackle the problem of 3D Reconstruction from *accidental motion* of the photographer. Given the initial few frames of a video or series of burst photos, we aim to reconstruct a dense depth map of the scene from bundle adjustment. We further apply a CRF model to regularize the depth to provide a smooth depth map from a reference view. We demonstrate the results of bundle adjustment for a few scenes.

***K*eywords** 3D Reconstruction · Bundle Adjustment · Multi-View Stereo · CRFs

1 Brief Overview

We implement the paper *3D Reconstruction from accidental motion*[1] which aims to reconstruct a 3D scene from the accidental motions of a photographer. Accidental motion is defined as the inevitable motion that occurs when trying to hold a camera still. Given a series of N images, we consider the first image N_0 as the reference image. The final depth map is given w.r.t. the reference view. The paper implements the following pipeline to perform the above task :

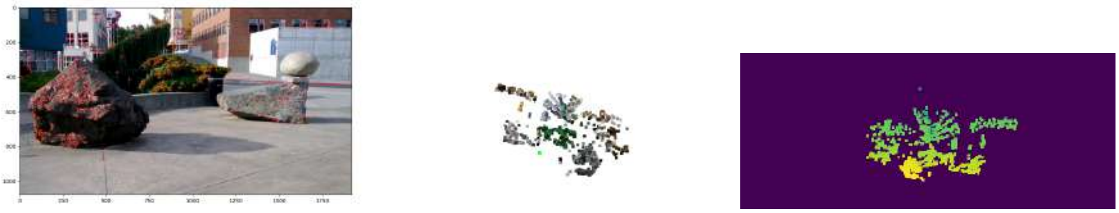
- Extract good features using the *Shi-Tomasi*[2] method.
- Track the detected features using the *Lucas-Kanade*[3] method w.r.t the reference image N_0
- We use these tracked features to perform bundle adjustment on the set of N frames to estimate the 3D structure of the scene
- A dense map is reconstructed from the sparse 3D structure using a CRF model [4] which incorporates a *photo-consistency* loss and a *smoothness* loss. The final output is the depth map from the reference view.

2 KLT Tracking

We use the KLT Tracker to track features across the $N - 1$ frames w.r.t to the reference frame. This step can be broken into two steps: (i) Feature Detection using [2] (ii) Feature Tracking using [3].

Shi-Tomasi method for feature detection uses the eigenvalues (λ_i's) of the Hessian matrix. The Hessian matrix considers the image intensities around a small square patch in the image. Based on the values of λ_1 and λ_2, we choose whether to consider the point or not as $\min(\lambda_1, \lambda_2) > \lambda$.

Lucas-Kanade method for feature tracking uses optical flow to estimate the displacement of the point from the reference image to the other frames. The KLT algorithm requires that all features be tracked to all the non-reference images. Another method to filter the patches is to choose only those patches that have a maximum color gradient difference per pixel below a threshold w.r.t. the reference image.



(a) (b) (c)

Figure 1: The point-cloud and the corresponding depth map of the image. The detected features are tracked across all non-reference images. We show here the trajectory of each feature across the set of frames.

3 Structure from motion

Given a set of images depicting a number of 3D points from different viewpoints, bundle adjustment can be defined as the problem of simultaneously refining the 3D coordinates describing the scene geometry, the parameters of the relative motion, and the optical characteristics of the camera(s) employed to acquire the images, according to an optimality criterion involving the corresponding image projections of all points.

3.1 Modeling bundle adjustment as an optimisation problem

Bundle adjustment boils down to minimizing the re-projection error between the image locations of observed and predicted image points, which is expressed as the sum of squares of a large number of nonlinear, real-valued functions. Thus, the minimization is achieved using nonlinear least-squares algorithms. Of these, Levenberg–Marquardt has proven to be one of the most successful due to its ease of implementation and its use of an effective damping strategy that lends it the ability to converge quickly from a wide range of initial guesses.

By iteratively linearizing the function to be minimized in the neighborhood of the current estimate, the Levenberg–Marquardt algorithm involves the solution of linear systems termed the normal equations. When solving the minimization problems arising in the framework of bundle adjustment, the normal equations have a sparse block structure owing to the lack of interaction among parameters for different 3D points and cameras.

3.2 Mathematical Modeling

The reprojection error minimisation for the problem can be written as shown in Figure 2. The Jacobian for the residuals is taken be differentiating the residuals with the rotation, and the translation parameters of the camera and the 3D position of the point. The Jacobian has the following structures as shown in Figure 3.

3.3 Problem formulation for our use case

As per the authors, we model the bundle adjustment problem by the following initialisation,

- The Rotation parameters for each view is set to Identity matrix.
- The Translation parameters for each view is set to the origin that is $\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ in homogeneous coordinates.
- The 3D world points are initialised with their X and Y coordinates as the reference view's (In our case the first image) pixel coordinates u and v respectively. The depth for the points is randomly initialised between 2 to 4 meters. The points are parameterised by their inverse depth as this results in a convex optimization problem (proved in section 7).

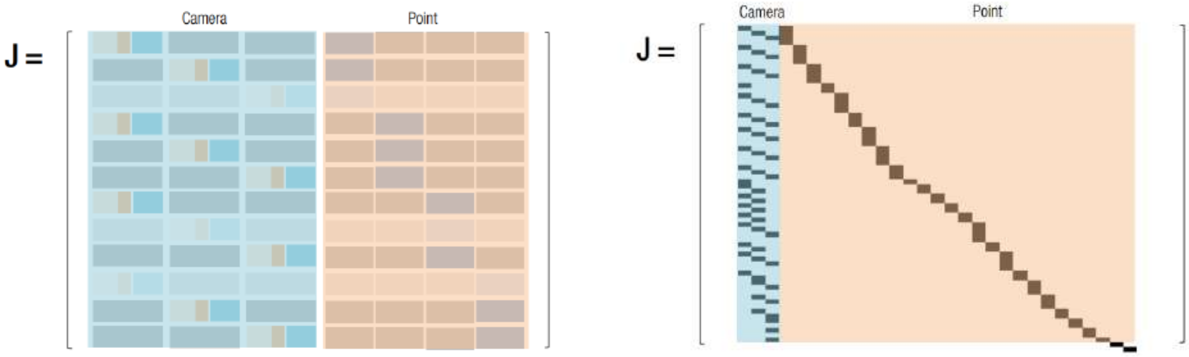
As for the residuals, we obtain them by tracking the optical flow of the obtained feature points as described in Section 2. Therefore we obtain, the location of the feature points in all the views. Using that we initialise the Jacobian's residuals.

$$
\begin{aligned}
F &= \sum_{i=1}^{N_c}\sum_{j=1}^{N_p} ||p_{ij} - \pi(R_i P_j + T_i)||^2,\\
&= \sum_{i=1}^{N_c}\sum_{j=1}^{N_p} (\frac{e^x_{ij} + f^x_{ij} w_j}{c_{ij} + d_{ij} w_j})^2 + (\frac{e^y_{ij} + f^y_{ij} w_j}{c_{ij} + d_{ij} w_j})^2,
\end{aligned}
$$

where

$$
\begin{aligned}
a^x_{ij} &= x_j - \theta^z_i y_j + \theta^y_i,\\
b^x_{ij} &= T^x_i,\\
a^y_{ij} &= y_j - \theta^x_i + \theta^z_i x_j,\\
b^y_{ij} &= T^y_i,\\
c_{ij} &= -\theta^y_i x_j + \theta^x_i y_j + 1,\\
d_{ij} &= T^z_i,\\
e^x_{ij} &= p^x_{ij} c_{ij} - a^x_{ij},\\
f^x_{ij} &= p^x_{ij} d_{ij} - b^x_{ij},\\
e^y_{ij} &= p^y_{ij} c_{ij} - a^y_{ij},\\
f^y_{ij} &= p^y_{ij} d_{ij} - b^y_{ij}.
\end{aligned}
$$

Figure 2: Cost function for Bundle Adjustment



(a)

(b)

Figure 3: Jacobian structure

$$
\mathbf{R}_i = \begin{bmatrix} 1 & -\theta^z_i & \theta^y_i \\ \theta^z_i & 1 & -\theta^x_i \\ -\theta^y_i & \theta^x_i & 1 \end{bmatrix}
$$

$\mathbf{P}_j = \frac{1}{w_j}[x_j, y_j, 1]^T$, where (x_j, y_j) is the projection

(a) (b) (c)

Figure 4: (a) Initialization of Rotation matrix (b) Inverse depth point initialization

3.4 Solving the Bundle Adjustment problem

The cost function of bundle adjustment with the assumption of small motion (both rotation and translation) yields a convex problem. When the camera poses are fixed, it is convex to get the depth of a feature relative to a reference view. Also, it is convex to optimize the rotation for the points at infinity when an approximation is used.

We optimize the cost function of bundle adjustment in Figure 2 with Ceres Solver. In the following subsection we show the results obtained after solving the BA problem.

4 DenseCRF model

After estimating the sparse 3D structure of the scene along with camera extrinsics, we want to construct a dense depth map of the 3D scene using a conditional random field formulation.

We use the plane-sweeping along with a CRF framework to solve for a dense depth map. The plane-sweeping method generates unary potentials for the CRF model. Using a fully-connected CRF model allows pixel connections with longer range so that the photo-consistency measurement can be effectively aggregated from an area to a pixel in it

We formulate this problem of obtaining a dense depth map as a labelling problem where the depth values are the labels and each pixel is a node. The unary potential of every node gives an initial estimate of the depth locations. We minimize the below energy to obtain a smooth out depth map:

$$E(D) = E_p(D) + \alpha E_s(D) \tag{1}$$

Here, $E_p(D)$ is the photo-consistency term (unary) and $E_s(D)$ is the smoothness term (pairwise). D here is the dense depth map we are minimizing over.

4.1 Fully Connected CRF Model

Consider a random field X defined over a set of variables $X_1, \ldots, X_N$. The domain is a set of labels $\mathcal{L} = l_1, \ldots, l_k$. Consider a random field $I = I_1, \ldots, I_N$. I ranges over possible input images of size N and X ranges from the possible pixel-level image labelings. I_j is the colour vector of the pixel j and X_j is the label assigned to pixel j.

In the fully connected pairwise CRF model, G is the complete graph on X and C_G is the set of all unary and pairwise cliques. The unary potential is computed independently for each pixel using the plane-sweeping method which produces a distribution over the label assignment x_i given image features and extrinsic matrices of the cameras. For pairwise potentials, we use a pairwise bilateral term which is a linear combination of Gaussian kernels that incorporate spatial and color dependencies over a defined region.

4.2 Unary Potentials

To calculate the unary potential, we use a photo-consistency term using the plane sweep algorithm. The algorithm sweeps a plane at different depths from each viewpoint and calculates the color intensity loss at each pixel location w.r.t the reference image.

We warp every viewpoint to the reference viewpoint by computing the homography transform between the two views. The homography matrix $H_j^{ref}(D)$ between the reference image and j^{th} camera frame at depth D is computed as follows:

$$\begin{gathered} H_j^{ref}(D) = D * K * {}^{C}_{W}R_{ref} * {}^{C}_{W}R_j^{-1} * K^{-1} \\ H_j^{ref}(D)[:,2] = K * {}^{C}_{W}R_{ref} * (C_j - C_{ref}) \end{gathered} \tag{2}$$

Here d, ${}^{C}_{W}R_j$, K, and C are the depth, rotation matrix, camera intrinsic matrix, and camera centers respectively. E_p is the photo-consistency term defined as an L1 Loss between small patches in the reference image and the warped image from another viewpoint. It can be expressed as:

$$E_p(D) = \sum_j \sum_i \left| p_{i,ref} - H_j^{ref}(D) * p_{i,j} \right| \tag{3}$$

$p_{i,ref}$ and $p_{i,j}$ are the gray-scale patches of the reference camera and j^{th} camera respectively.

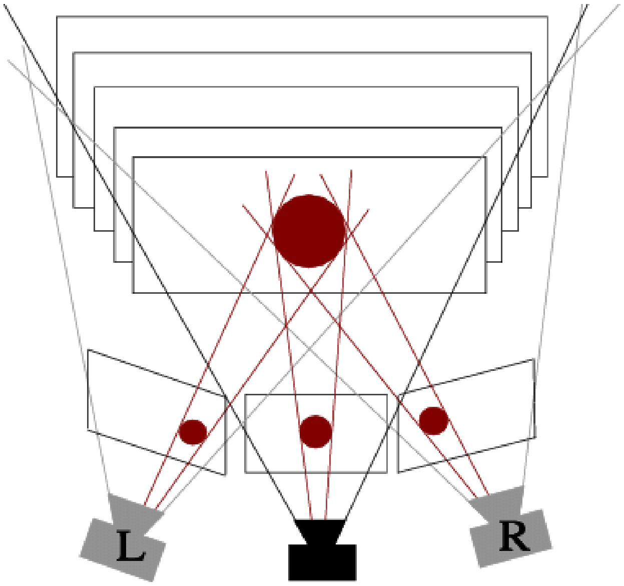


Figure 5: Illustration of the Plane-sweep algorithm

4.3 Pairwise potential

The pairwise potential is a linear combination of Gaussian kernels. Let I be the index set of the pixels in a reference view, and $I(i)$, $i \in I$, be the color of the i-th pixel. The goal is to determine a dense depth map, D of the reference view. Let L map each pixel index $i \in I$ to a 2D location in the image. The pairwise potential $E_s(D)$ is computed as :

$$E_s(D) = \sum_{i \in \mathcal{I}, j \in \mathcal{I}, i \neq j} C(i, j, I, L, D) \tag{4}$$

and

$$C(i, j, I, L, D) = \rho_c(D(i), D(j)) \times \exp\Big(- \underbrace{\frac{||I(i) - I(j)||^2}{\theta_c}}_{\text{Intensity term}} - \underbrace{\frac{||L(i) - L(j)||^2}{\theta_p}}_{\text{Spatial term}} \Big) \tag{5}$$

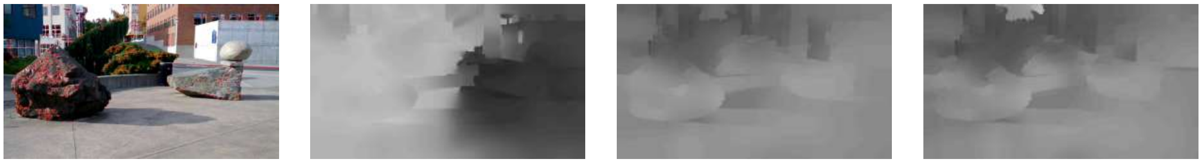
where $\rho_c(.)$ is the truncated linear function defined as $\rho_c = min(t, |D(i) - D(j)|)$ with some threshold t. The pairwise term has a spatial term such that depth within a small neighborhood is consistent as well as a intensity term such that pixels within an area with similar colors have consistent depth, since they are more likely to belong to the same object.

The energy term in Equation 1 is solved using a dense CRF model as proposed in [4] which uses the mean field approximation to give a smooth depth map. This approximation yields an iterative message passing algorithm for approximate inference. Message passing in the CRF model is performed using Gaussian filtering in some arbitrary feature space. The truncated linear function is implemented as two convolutions of 1D box filtering. This allows the running time to be linear to the number of depth labels thus allowing fast inference. The output of the energy minimization step is the final dense depth map as required.

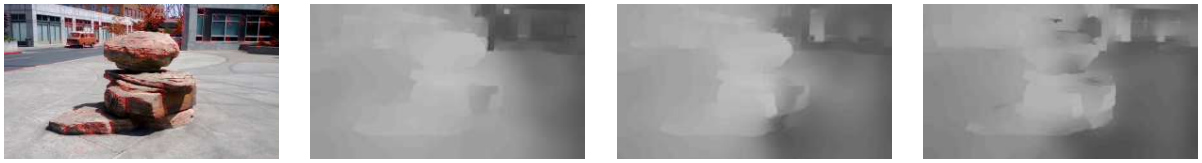
5 Experiments

We perform 6 experiments involving varying the hyperparameters. We study the effect of each hyperparameter and provide our own inferences on the observed variations.

5.1 Varying number of images for Bundle Adjustment

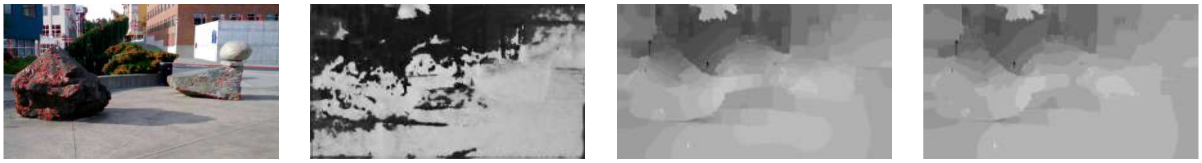


(a) Optical flow (b) 30 frames (c) 50 frames (d) 100 frames

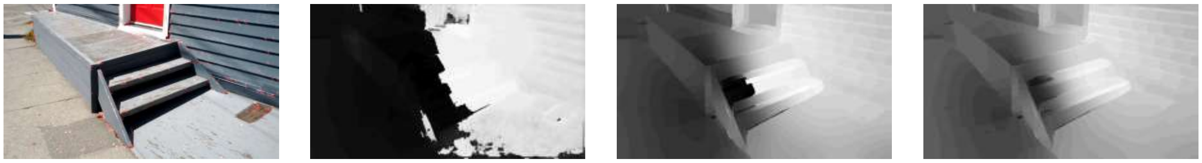


(a) Optical flow (b) 30 frames (c) 50 frames (d) 100 frames

5.2 Varying maximum penalty for DenseCRF

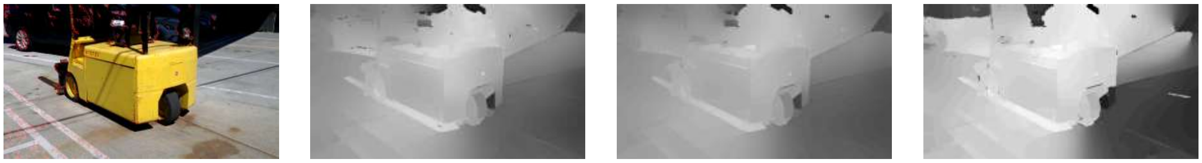


(a) Optical flow (b) t = 0.1 (c) t = 0.25 (d) t = 0.35

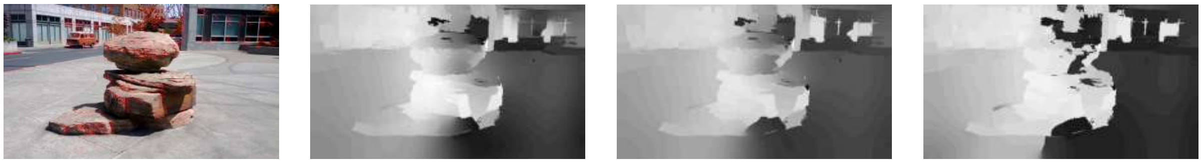


(a) Optical flow (b) t = 0.1 (c) t = 0.25 (d) t = 0.35

5.3 Varying weight for DenseCRF

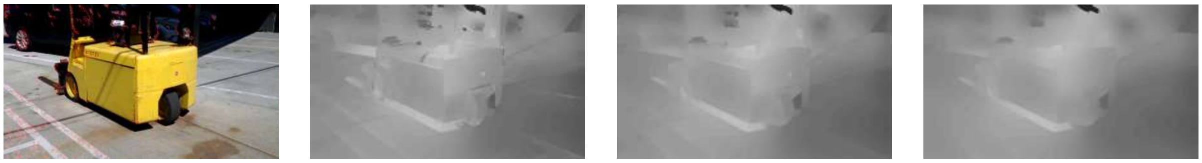


(a) Optical flow (b) w = 0.5 (c) w = 1.0 (d) $\theta_c = 2.5$

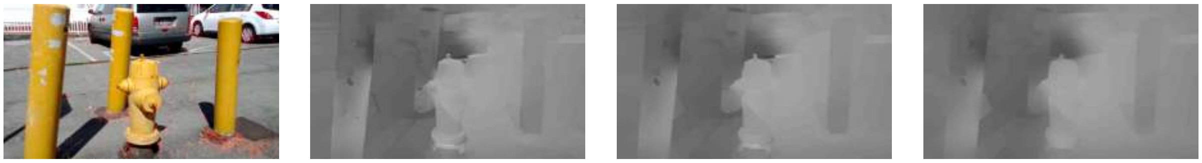


(a) Optical flow (b) w = 0.5 (c) w = 1.0 (d) $\theta_c = 2.5$

5.4 Varying intensity standard deviation for pairwise potentials

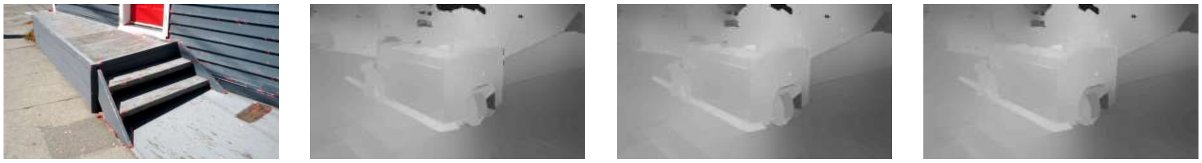


(a) Optical flow (b) $\theta_c = 10$ (c) $\theta_c = 20$ (d) $\theta_c = 35$

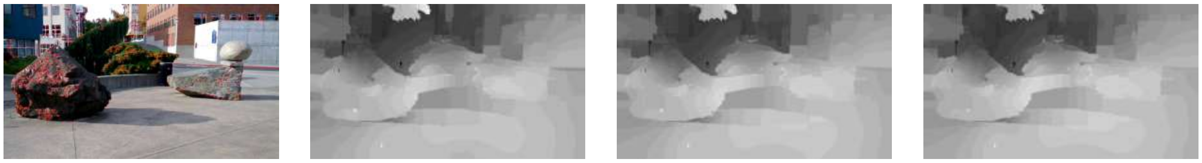


(a) Optical flow (b) $\theta_c = 10$ (c) $\theta_c = 20$ (d) $\theta_c = 35$

5.5 Varying patch size for calculating unary potential

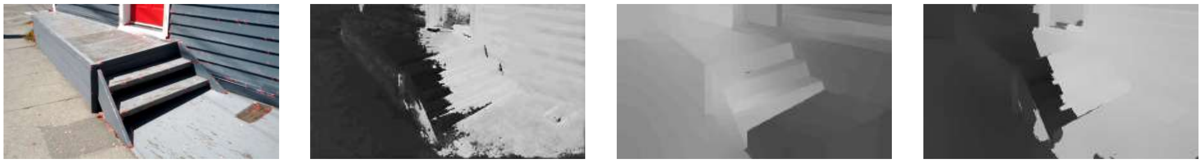


(a) Optical flow (b) patch radius = 1 (c) patch radius = 2 (d) patch radius = 3

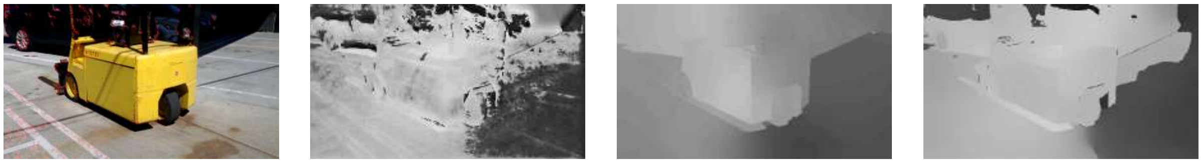


(a) Optical flow (b) patch radius = 1 (c) patch radius = 2 (d) patch radius = 3

5.6 Varying number of depth samples for CRF



(a) Optical flow (b) samples = 32 (c) samples = 64 (d) samples = 128



(a) Optical flow (b) samples = 32 (c) samples = 64 (d) samples = 128

6 Results

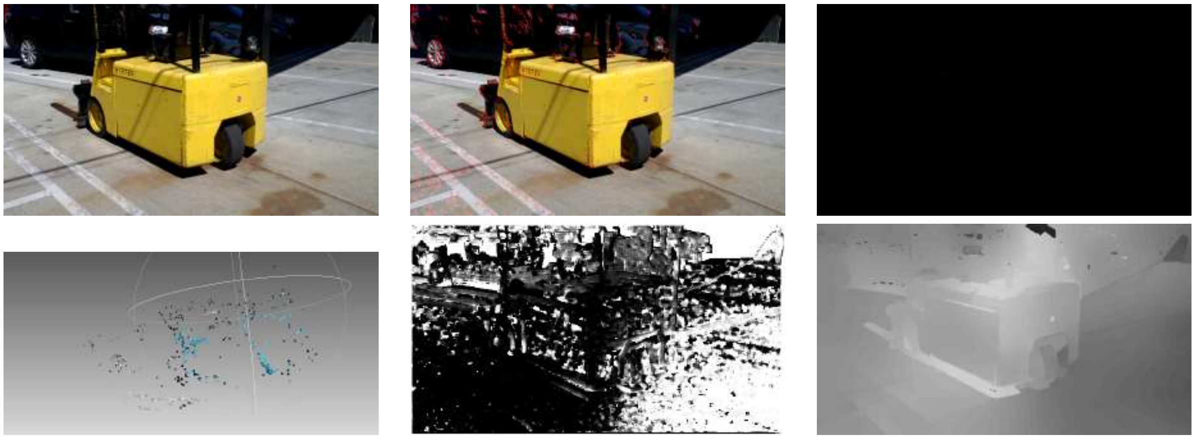


Table 1: (a) Reference Image (b) Optical Flow (c) Sparse depth map (d) Sparse pointcloud (e) WTA (f) Dense depth map

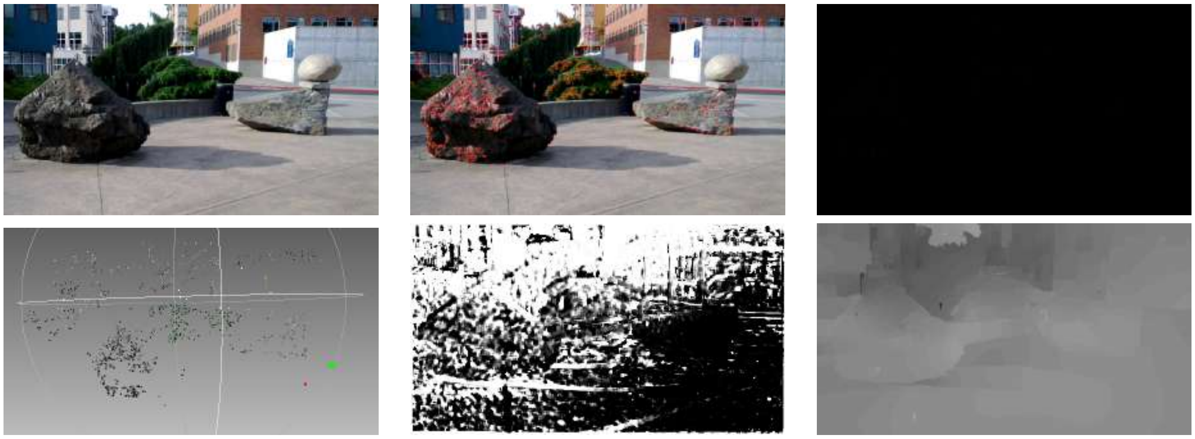


Table 2: (a) Reference Image (b) Optical Flow (c) Sparse depth map (d) Sparse pointcloud (e) WTA (f) Dense depth map

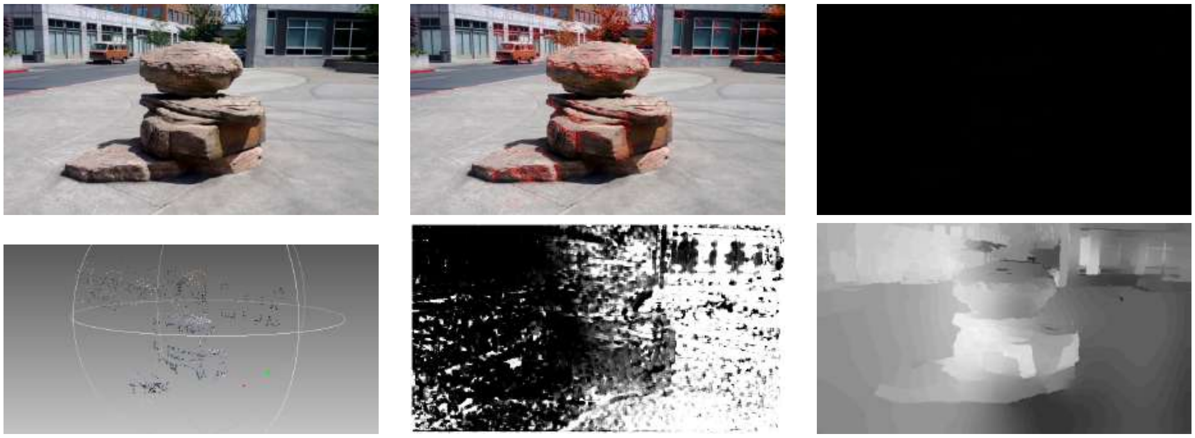


Table 3: (a) Reference Image (b) Optical Flow (c) Sparse depth map (d) Sparse pointcloud (e) WTA (f) Dense depth map

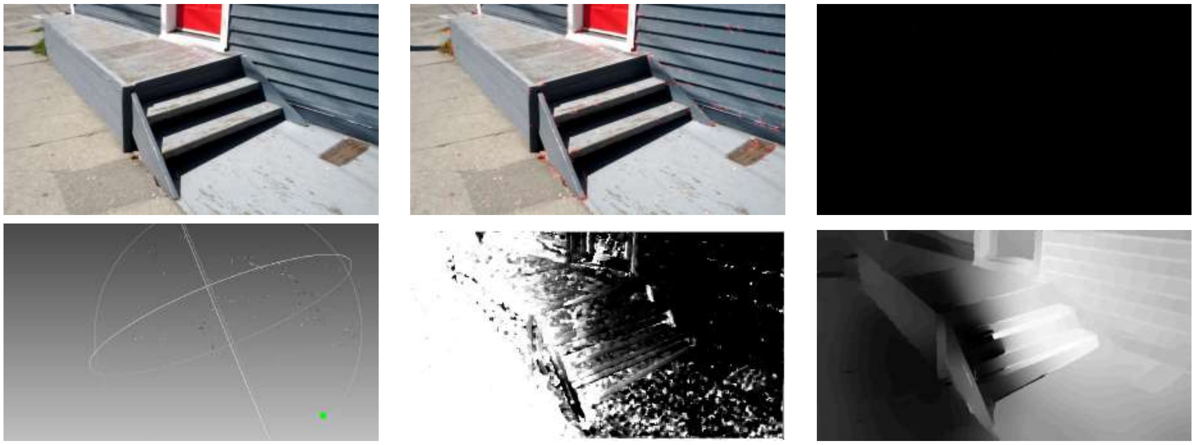


Table 4: (a) Reference Image (b) Optical Flow (c) Sparse depth map (d) Sparse pointcloud (e) WTA (f) Dense depth map

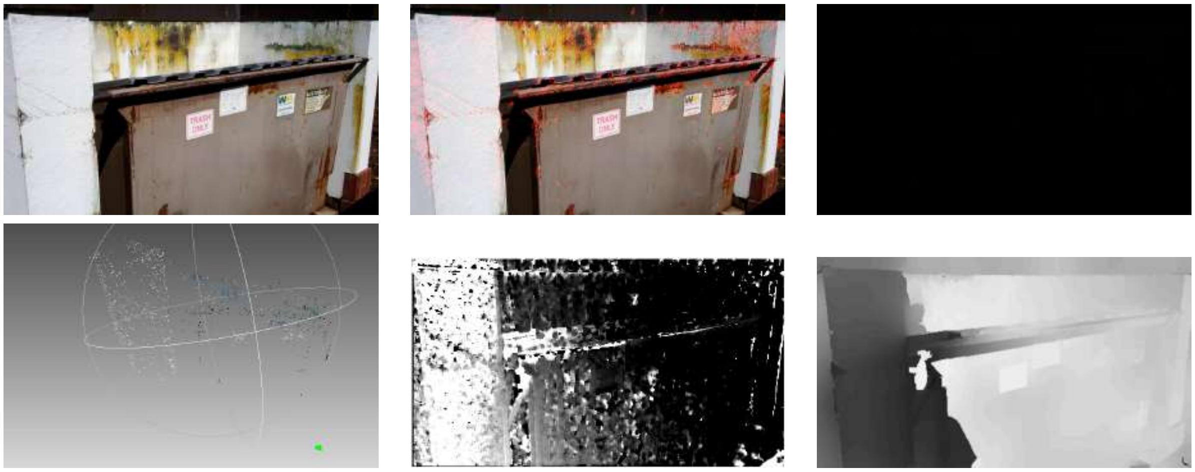


Table 5: (a) Reference Image (b) Optical Flow (c) Sparse depth map (d) Sparse pointcloud (e) WTA (f) Dense depth map

References

[1] Fisher Yu and David Gallup. 3d reconstruction from accidental motion. In *27th IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[2] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. 1991.

[3] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, page 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[4] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 109–117, Red Hook, NY, USA, 2011. Curran Associates Inc.

7 Appendix

7.1 Inverse depth initialization results in convex optimization

P_i is the i^{th} 3D point initialized with $[x_i \quad y_i \quad 1] * \frac{1}{w_i}$
${}^{C}_{W}R_j$ is the rotation matrix and ${}^{C}_{W}T_j$ of the world with respect to j^{th} camera

$$ {}^{C}_{W}R_j = \begin{bmatrix} 1 & -\theta_j^z & \theta_j^y \\ \theta_j^z & 1 & -\theta_j^x \\ -\theta_j^y & \theta_j^x & 1 \end{bmatrix} \tag{6} $$

p_{ij}^x is the x coordinate of the image of P_i in the j^{th} camera .Projecting P_i to the j^{th} camera we get

$$ {}^{C}_{W}R_j P_i +^{C}_{W} T_j = \frac{1}{w_i} * \begin{bmatrix} 1 & -\theta_j^z & \theta_j^y \\ \theta_j^z & 1 & -\theta_j^x \\ -\theta_j^y & \theta_j^x & 1 \end{bmatrix} * \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} + \begin{bmatrix} T_j^x \\ T_j^y \\ T_j^z \end{bmatrix} = \frac{1}{w_i} \begin{bmatrix} a_{ij}^x \\ a_{ij}^y \\ c_{ij} \end{bmatrix} + \begin{bmatrix} b_{ij}^x \\ b_{ij}^y \\ d_{ij} \end{bmatrix} \tag{7} $$

To minimize the re-projection error we need to optimize the following cost function:

$$ \underset{w_i}{\operatorname{argmin}} \left(p_{ij}^x - \frac{\frac{a_{ij}^x}{w_i} + b_{ij}^x}{\frac{c_{ij}}{w_i} + d_{ij}} \right)^2 + \left(p_{ij}^y - \frac{\frac{a_{ij}^y}{w_i} + b_{ij}^y}{\frac{c_{ij}}{w_i} + d_{ij}} \right)^2 \tag{8} $$

$$ \underset{w_i}{\operatorname{argmin}} \left(\frac{(p_{ij}^x c_{ij} - a_{ij}^x) + (p_{ij}^x d_{ij} - b_{ij})w_i}{c_{ij} + d_{ij}w_i} \right)^2 + \left(\frac{(p_{ij}^y c_{ij} - a_{ij}^y) + (p_{ij}^y d_{ij} - b_{ij})w_i}{c_{ij} + d_{ij}w_i} \right)^2 \tag{9} $$

From equation 9 we get

$$ \underset{w_i}{\operatorname{argmin}} \left(\frac{e_{ij}^x + f_{ij}^x w_i}{c_{ij} + d_{ij}w_i} \right)^2 + \left(\frac{e_{ij}^y + f_{ij}^y w_i}{c_{ij} + d_{ij}w_i} \right)^2 \tag{10} $$

$$ \underset{w_i}{\operatorname{argmin}} \left(\frac{f_{ij}^x}{d_{ij}} \right)^2 \left(\frac{\frac{e_{ij}^x}{f_{ij}^x} + w_i}{\frac{c_{ij}}{d_{ij}} + w_i} \right)^2 + \left(\frac{f_{ij}^y}{d_{ij}} \right)^2 \left(\frac{\frac{e_{ij}^y}{f_{ij}^y} + w_i}{\frac{c_{ij}}{d_{ij}} + w_i} \right)^2 \tag{11} $$

Equation 11 is of the form $\left(\frac{x-a}{x-b}\right)^2$. Since $d_{ij} \approx 0$, $\frac{c_{ij}}{d_{ij}} > \frac{e_{ij}^x}{f_{ij}^x}$. The above equation is convex in the range $0 < w_i < \frac{c_{ij}}{d_{ij}}$.

$$ w_i \in \left(0, \min_j(w_{ij})\right) \tag{12} $$

The values of inverse depth that we choose by random initialization lies within this range.