EFFICIENT COST VOLUME SAMPLING FOR PLANE SWEEPING BASED MULTIVIEW DEPTH ESTIMATION

Olli Suominen, Atanas Gotchev

Department of Signal Processing
Tampere University of Technology, Finland

ABSTRACT

Plane sweeping is an increasingly popular algorithm for generating depth estimates from multiview images. It avoids image rectification, and can align the matching process with slanted surfaces, improving accuracy and robustness. However, the size of the search space increases significantly when different surface orientations are considered. We present an efficient way to perform plane sweeping without individually computing reprojection and similarity metrics on image pixels for all cameras, all orientations and all distances. The procedure truly excels when the amount of views is increased and scales efficiently with the number of different plane orientations. It relies on approximation to generate the costs, but the differences are shown to be small. In practice, it provides results equivalent to conventional matching but faster, making it suitable for applying in many existing implementations.

Index Terms— *Depth estimation, plane sweeping, cost volume, multi-view*

1. INTRODUCTION

Depth reconstruction has long been the focus of extensive work, and still continues as an active field of research. In addition to traditional computer vision applications, depth and scene geometry are also increasingly needed in 3D media related uses – free viewpoint TV, multiview rendering, light-field sampling etc. Depth estimation based on tracking similarities between images is fully passive, differentiating it from the alternatives: active range sensing devices which require specific hardware, power and are vulnerable to ambient signals or other devices.

Initially, the method of choice was stereo matching [1], where a pair of images is taken, post-processed (rectified) [2] to make the search feasible and then compared in order to identify corresponding image points. Recently, a shift from pure epipolar constrained stereo matching algorithms has been made to more freely structured multiview approaches, such as the method referred to as *plane sweeping*, which can handle multiple images without strict rectification.

In this paper, we describe a simple, yet highly efficient method of performing multi-directional plane sweeps. The concept is very straightforward, but provides significant gains in the processing efficiency of the algorithm.

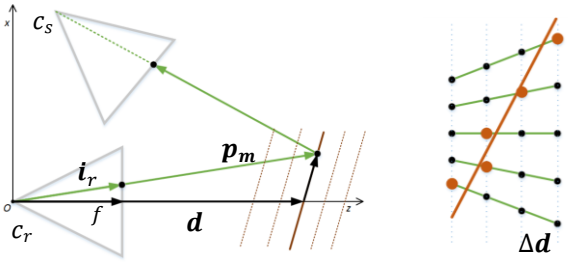

Figure 1: Left: The basic concept of plane sweeping and the vectors used to describe it demonstrated with a single light ray. Right: a plane approximated by known cost volume samples along rays.

It is based on constructing a pixel-wise cost volume and consequently sampling the volume to achieve different plane orientations, thus avoiding the repeated per-plane reprojection, image interpolation and cost calculation steps present in other methods. The complexity is reduced further by decreasing the depth-wise sampling density, which leads to small approximation errors, but their magnitude is shown to be negligible.

2. PREVIOUS WORK

Originating from the need to perform matching between multiple images, Collins presented work that made it possible to find feature correspondences between non-rectified images [3]. Advancing the case for plane sweeping was the realization that many scenes commonly input to depth reconstruction algorithms often contain man-made surfaces, i.e. buildings, streets etc. which mostly consist of slanted planes [4],[5]. The same concept was used by Zabulis et al. to increase the accuracy of the matching by sweeping with spherical surfaces, adapting to the oblique viewing angles along the peripheral vision [6].

Due to the large amount of plane hypotheses stemming from the consideration of different orientations, much research has gone into improving the computational aspects of plane sweeping. Gallup et al. showed how planes with multiple orientations could be processed in real time [7], while many have gone into reducing the disparity search space. PatchMatch stereo is one of the latest approaches, which aims at speeding up the algorithm through propagation of estimates with high confidence [8]. Following the similar trend is also [9], where feature matching is used to limit the amount of depth hypothesis.

2.1 Plane Sweeping Depth Estimation

We follow closely [7], in which the process is formulated as a homography between the image planes of the different views. In practice, the cost volume construction stage in our approach can be done using the homography for increased efficiency, but a slightly more detailed formulation is useful for describing the proposed method.

The starting point is a set of S images from different viewpoints with the corresponding intrinsic and extrinsic pinhole camera parameters [2]. One of the cameras is designated as the reference camera (origin of the coordinate system, rotation matrix R identity). The parameter guiding the depth estimation is the set of candidate planes $\Pi = \{\Pi_m = (\boldsymbol{n_m}, \boldsymbol{d_m}) | m = 1, \dots, M\}$, which are defined through their normal $\boldsymbol{n}$ and distance $\boldsymbol{d}$ from the reference camera center. The distance parameter is bound by some constraints from both sides, i.e. $d \in [d_{min}\ d_{max}]$, to limit the search space.

First, a ray vector $\boldsymbol{i_r} = [x_c, y_c, f]^{\boldsymbol{T}}$ is defined for each pixel in the reference image, where x_c and y_c are the global space coordinates of the pixels. The 2D pixel coordinates $\boldsymbol{u} = [u\ v]^{\boldsymbol{T}}$ are converted into global image plane vector $\boldsymbol{i} = [x\ y\ z]^{\boldsymbol{T}}$ through

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = s\left(\begin{bmatrix} u_s \\ v_s \end{bmatrix} - \begin{bmatrix} c_x \\ c_y \end{bmatrix}\right), \tag{1}$$

where $\boldsymbol{c} = \left[c_x, c_y\right]^{\boldsymbol{T}}$ is the principal point of the target camera and s the physical (square) pixel size of the camera.

The intersection point $\boldsymbol{p_m}$ between the ray and the depth candidate plane is found by the scaling multiplier which extends $\boldsymbol{i_r}$ to reach the plane (Figure 1). This leads to

$$\boldsymbol{p_m} = \frac{\langle \boldsymbol{n}, \boldsymbol{d} \rangle}{\langle \boldsymbol{n}, \boldsymbol{i}_r \rangle} \boldsymbol{i}_r \tag{2}$$

for the global space coordinate $\boldsymbol{p_m}$ of the back-projected pixel at $\boldsymbol{i_r}$. Next step is the projection of $\boldsymbol{p_m}$ to a secondary camera $c_s, s \in S$. This is the familiar projection equation

$$\boldsymbol{i}_s = \frac{f}{z_{.s}} \boldsymbol{p}_s = \frac{f}{z_{.s}} R^T(\boldsymbol{p_m} - \boldsymbol{t}). \tag{3}$$

The relation between the vectors $\boldsymbol{i_r}$ and $\boldsymbol{i_s}$ is then utilized in computing the matching cost value $C(x, y, \Pi_m)$ associated with the depth plane hypothesis. The global image coordinates are converted to image plane coordinates by reversing Equation 1.

There are numerous ways of computing the matching cost, and its typical formulations also include the cost aggregation. If e.g. the L1 norm and block based aggregation is selected (such as in [7]) and the system contains S cameras, it becomes

$$C_a(\boldsymbol{u_r}, \Pi_m) = \sum_{s=1}^{S} \sum_{\boldsymbol{u} \in W} |I_r(\boldsymbol{u_r} + \boldsymbol{u}) - I_s(\boldsymbol{u_s} + \boldsymbol{u})|, \tag{4}$$

where W is the local aggregation window centered on $\boldsymbol{u_r}$.

To get a depth estimate from the cost volume C, a winner-takes-all procedure is applied for each reference pixel, where the plane with the lowest matching cost is selected as the winning hypothesis. From this, the depth for reference pixel $\mathbf{u_r}$ is found at the intersection between the corresponding image plane ray $\boldsymbol{i_r}$ and the winning plane as in Equation (2).

3. PROPOSED COST VOLUME CONSTRUCTION AND SAMPLING

The proposed method consists of two steps – cost volume construction and cost volume sampling. It is applicable to both stereo and multiview algorithms, and is independent of the processing steps taken further down the reconstruction pipeline. Thus, it can replace the cost computation of any plane sweeping based algorithm where the cost metric is based on aggregation of per-pixel costs. This includes methods based on L1 and L2 norms and normalized cross-correlation, but leaves out cases where aggregation is performed before computing the cost, e.g. matching based on descriptors.

In the cost volume construction step, a pixel-wise cost volume is created following the procedure of a single orientation plane sweep through the volume without cost aggregation. The process for the most part follows the one described in section 2.1 until the cost aggregation step in Equation 4. The outcome of this is then effectively a volume $\hat{C}(\boldsymbol{u_r}, d)$, indexed by reference camera pixel coordinate and depth, much like in conventional stereo matching.

3.1 Cost volume sampling

Once the ray-wise cost volume has been constructed, performing any additional sweeps with any plane orientation simply requires sampling the volume in correct places. As the volume is formed based on the rays of the reference camera, the known cost samples do not form a uniformly spaced regular grid. However, the structure of the grid is still known, and the samples can be retrieved without having to resort to inefficient nearest neighbor searches within the volume.

For retrieving the values from the volume, it is only necessary to compute the intersection of the ray with whatever plane Π_m is currently being used for sweeping, which leads to the index of the correct cost sample. Assuming the sampling density of the cost values is uniform in depth with a step of Δd, the index $\hat{k}$ comes easily from the scaling coefficient of $\boldsymbol{i_r}$ from Equation (2), i.e.

$$\hat{k} = [k] = \left[\frac{n_3(d - d_{min})f}{\langle \boldsymbol{n}, \boldsymbol{i}_r \rangle \Delta d}\right], \tag{5}$$

where [ ] is the rounding operator and n_3 the third component of the normal vector $\boldsymbol{n}$. With this in mind, the approximated cost value for that particular pixel is

$$\hat{C}(\boldsymbol{u_r}, \Pi_m) = \hat{C}(\boldsymbol{u_r}, d_m, \boldsymbol{n}_m) = \hat{C}\big(\boldsymbol{u_r}, \hat{k}\big), \quad (6)$$

and furthermore, the aggregated cost is

$$\hat{C}_a(\boldsymbol{u_r}, d_m, \boldsymbol{n}_m) = \sum_{\boldsymbol{u}\in W} \hat{C}\big(\boldsymbol{u}, \hat{k}_{\boldsymbol{u}}\big)\,. \quad (7)$$

Compared to Equation (5), this form of the aggregated cost notably lacks reprojection of the reference pixel coordinate, calculation of the norm (similarity metric) and summing over the S different cameras. This simplified cost computation is the source of the speed gain of the proposed method.

With nearest neighbor (NN) interpolation within the volume, our approach produces near exact replicas of the values which result from full ray tracing and NN interpolating on the image planes. This happens with the assumption that the maximum reprojection error stays below 0.5 pixels on all cameras involved. Due to the fact that we are avoiding the rest of the ray traces to the other image planes beyond the first construction stage, some differences (Section 4) arise between the proposed method and simply tracing the ray bounces for each plane separately and performing some more sophisticated interpolation on the image planes.

One way of introducing finer approximation of the cost function is to perform interpolation along the cost ray. This is similar to the approach sometimes used e.g. in stereo matching when aiming for sub-pixel precision matching [1]. It will not produce the same results as interpolation on the image planes of secondary cameras, but nevertheless can increase the precision of the estimate.

4. SAMPLING DENSITY AND ERRORS

The most important parameter in the process is the sampling density of the first estimation stage. It determines the magnitude of the error induced into the estimate and as such, should be selected carefully. As a reference, we use the lower limit described in [7], where planes are selected such that the secondary image plane locations of a reference pixel reprojected via two consecutive planes differ at greatest by one pixel. Anything sparser than that would not use all information available in the images.

In the global space position of the points, the maximum error comes from points which lie exactly in the middle of two known planes. Thus the global positioning error is in practice at most half of the inter-plane distance Δd. The reprojection error is a more interesting measure, as it depends not only on Δd, but also the orientation and location of the candidate planes and the cameras. The reprojection error from the approximation points is roughly proportional to the reprojected difference between two consecutive sampling planes. To decouple this, the cost volume can be constructed with a different density than with what the actual plane sweeps are performed. Thus the quality of the approximation can be adjusted independently from the sweeping planes. Increasing the density also increases the processing load in the cost volume construction stage, but does not significantly affect the sampling step. We utilize two different parameters for the plane distances, one for constructing the volume and one for deciding which planes are generated by sampling it.

4.1 Memory consumption

Storing the constructed cost volume for the duration of the depth estimation requires some additional memory. For instance, a Full HD reference image with a 10 meters deep volume of interest and a 1 cm plane stepping would consume close to 2 GB of memory in single precision floating point. While perfectly feasible for a modern computer, this can be considered somewhat excessive and e.g. in embedded solutions, impossible. However, the minimal amount of information that has to be stored at once is the volume limited by a single aggregation window on the image plane. The volume can thus be easily divided into partitions to conserve memory at the expense of some added control structures. Performance implications from dealing with the overlaps between the partitions are small as long as the number of partitions remains reasonable. E.g. 10 horizontal partitions will already bring the memory requirement down from 2 GB to 200 MB.

5. EXPERIMENTAL RESULTS

In the following error analysis experiments, we define our simulated cameras to have focal length 35mm, pixel size 10 micrometers and resolution 640x480. We generate a random set of extrinsic camera parameters (rotation, translation) and sweeping plane directions to investigate the associated errors. For determining the density of sweeping planes, we adopt the metric from [7] to find such a step size Δd_{sample} which leads to a maximum disparity just under 1. Regardless, the behavior of the errors appears to stay the same even when these parameters change. The experiments are done on 10 different randomly rotated and translated camera pairs with random sweeping directions with a double density volume.

In Figure 2, the position error is given relative to the sweeping distance step Δd_{sample} to account for the different step sizes of different camera configurations. As could be expected, the global position error is evenly distributed over the range from 0% to 25%, i.e. half of the distance between the double density sample planes. The reprojection error is mostly concentrated around small values. It reaches its maximum level at one quarter of the reprojection difference between two consecutive sweeping planes, which in this case is enforced to be ~1 pixel.

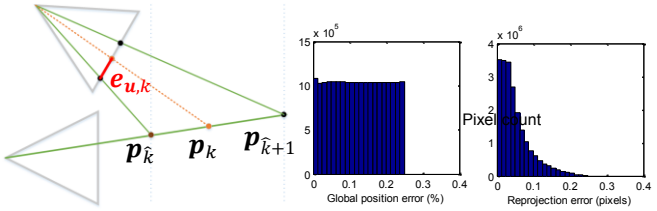

Figure 2: Left: Illustration of the source of error $\boldsymbol{e_{u,k}}$, i.e. reprojection error, which follows from approximating the requested point $\boldsymbol{p_k}$ with $\boldsymbol{p_{\hat{k}}}$. Right: histograms of the error in practice respectively in relative global positions and projected pixel positions.

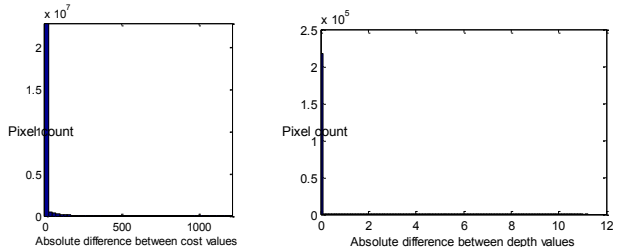

Figure 3: Histogram of absolute differences between the cost volumes (left) and depth maps (right) generated with the conventional approach and the proposed method. Non-zero values are mainly explained by occlusions.

In Figure 3, the differences between two cost volumes generated respectively with the conventional plane sweep described in Section 2.1, and our method are presented. The volumes exhibit only minor differences. Also the resulting depth maps are almost identical (Figure 4), with most of the differences appearing on volatile areas such as occlusions and low texture. Beyond block based aggregation, no post processing has been applied. It is crucial to note that the images are to show that the two methods produce same or similar results, not to claim that the quality of those particular estimates would be anything special. If results at this level are the same, they will behave in the same way in any additional processing. Furthermore, most differences in the images appear at the occluded areas, which do not have a correct matching result at all.

If only a single plane orientation is used for sweeping, the conventional method performs faster as the extra effort spent on the double density volume is wasted. However, most of time is spent in constructing the cost volume, after which each consecutive plane orientation takes only ~30% of the time required by the reference algorithm. Thus the scaling (as shown in Figure 5) with number of plane orientations is good, and when considering numbers like 40 different orientations [10], the impact is significant.

A major benefit is that our approach precomputes the combination of cost values from each camera (view). The increase in cameras only increases the processing load while constructing the original volume (the constant component in Figure 5), and it has no effect on the sampling stage. Therefore while the conventional approach starts a steep incline with 10 cameras, ours scales the same with any number of them.

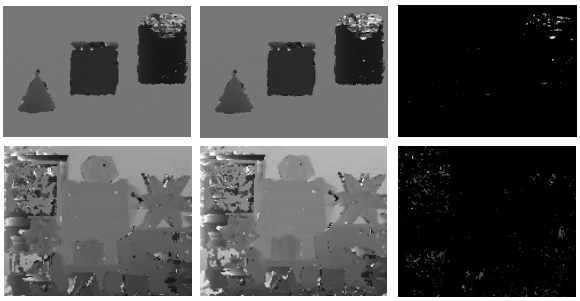

Figure 4: Depth maps generated with the simple version of conventional plane sweeping (left) and the proposed approximation (middle), and their absolute differences (right). The important aspect is not the quality (or lack thereof) of the estimate, but the relative similarity between the two methods.

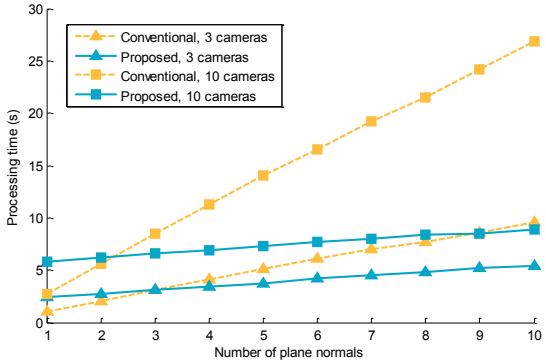

Figure 5: The effect of increasing the number of plane orientations on the processing time in an exhaustive search over the whole volume of interest when the construction density is double of the sampling density. Based on C++/OpenCV implementations.

6. CONCLUSIONS

We have presented a highly efficient way of performing multi-view, multi-directional plane sweeps through sampling a per-pixel cost volume. As the method is suitable for replacing the cost volume generation of various plane sweeping algorithms, it has many potential use cases especially in the context of multi-camera systems. The speedup gained is also complementary to efforts in reducing the search space. If there are several plane orientations involved, there is a benefit to using the proposed approach.

The experimental results show that the error induced by the approximation process behaves predictably and reliably with different parameters. The cost volumes and consequent results from the depth estimation are very similar between the conventional method and ours, and only exhibit notable differences in the occluded areas, which are in any case unreliable. The scaling with the amount of plane orientations is very good, and remarkably, unlike its conventional counterpart, the scaling of the method is invariant to the number cameras.

7. REFERENCES

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision,* vol. 47, pp. 7-42, 2002.

[2] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge university press, 2003.

[3] Collins, R.T., "A space-sweep approach to true multi-image matching," *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on* , pp.358,363, 18-20 Jun 1996

[4] T. Werner and A. Zisserman, "New techniques for automated architectural reconstruction from photographs," in *Computer Vision—ECCV 2002* Springer Berlin Heidelberg, 2002, pp. 541-555.

[5] N. Cornelis, K. Cornelis and L. Van Gool, "Fast Compact City Modeling for Navigation Pre-Visualization," *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* , vol.2, pp.1339,1344, 2006

[6] X. Zabulis, G. Kordelas, K. Mueller and A. Smolic, "Increasing the accuracy of the space-sweeping approach to stereo reconstruction, using spherical backprojection surfaces," in *Image Processing, 2006 IEEE International Conference on,* 2006, pp. 2965-2968.

[7] D. Gallup, J. Frahm, P. Mordohai, Q. Yang and M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on,* 2007, pp. 1-8.

[8] M. Bleyer, C. Rhemann and C. Rother, "PatchMatch stereo-stereo matching with slanted support windows." in *Bmvc,* 2011, pp. 1-11.

[9] S. N. Sinha, D. Scharstein and R. Szeliski, "Efficient high-resolution stereo matching using local plane sweeps," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on,* 2014, pp. 1582-1589.

[10] M. Bleyer, C. Rother and P. Kohli, "Surface stereo with soft segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on,* 2010, pp. 1570-1577.