15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

Plane labeling trinocular stereo matching with baseline recovery

Luis Horna
The University of Edinburgh
s1145121@sms.ed.ac.uk

Robert B. Fisher
The University of Edinburgh
rbf@inf.ed.ac.uk

Abstract

In this paper we present an algorithm which recovers the rigid transformation that describes the displacement of a binocular stereo rig in a scene, and uses this to include a third image to perform dense trinocular stereo matching and reduce some of the ambiguities inherent to binocular stereo. The core idea of the proposed algorithm is the assumption that the binocular baseline is projected to the third view, and thus can be used to constrain the transformation estimation of the stereo rig. Our approach shows improved performance over binocular stereo, and the accuracy of the recovered motion allows to compute optical flow from a single disparity map. These claims are validated with the KITTI 2012 data set.

1 Introduction

The problem of 3D plane labeling stereo matching using three images, two binocular and a third with displacement, can be described as finding the correspondences for each pixel from image I_l to I_r and I_u by assigning a 3D plane that encodes the 1D disparity that is used to recover a 3D point X. Using a 1D disparity implies that (I_l, I_r) are rectified, and a known projective transformation P (camera) maps X_i to x_i^u in I_u. Finding the optimal 3D disparity plane labeling D is modeled as an optimization problem where the objective is to minimize eq.1.

$$E(D) = \arg\min_{D} \sum_{p}^{NumP} \{C_p(D_p) + \sum_{q \in N(p)} V_{pq}(D_p, D_q)\} \quad (1)$$

$E(D)$ is the cost of the disparity assignment (energy), D is a set of planes and D_p encodes the plane, that gives the disparity of the pixel at p with respect to another rectified image. $D_p(q)$ is the disparity estimated using plane D_p evaluated at pixel q. $NumP$ is the number of pixels in the image. $N(p)$ is a neighborhood around p, and q is a neighbor of p. V_{pq} (*smoothness* term) is a function that evaluates how well the disparity at position p fits its neighbors. The plane D_p has two parameters: a 3D unit normal vector $\hat{n}_p = (n_p^x, n_p^y, n_p^z)$ and disparity d_p. The disparity of pixel $q = (x_q, y_q)$ using D_p is given by:

$$D_p(q) = a * x_q + b * y_q + c \quad (2)$$

where $a = -\hat{n}_p^x / \hat{n}_p^z$, $b = -\hat{n}_p^y / \hat{n}_p^z$ and $c = (\hat{n}_p^x * x_q + \hat{n}_p^y * y_q + \hat{n}_p^z * d_p) / \hat{n}_p^z$ as in [4]. C_p is a function that measures the similarity/dissimilarity of three pixels, e.g. $I_l(p)$ is compared to $I_r(p + D_p(p)))$ and $I_u^l(\phi(P \cdot X))$ with $\phi(x) = (x_1/x_3, x_2/x_3)$. In this paper the pairwise function in eq.1 is represented as a Markov Random Field and minimized using TRW-S[10]. Our algorithm works under the assumption that the binocular baseline T_r (fig.1) is projected to I_u^l , and thus can be used to constrain the recovery of the transformation P from the stereo rig. The proposed approach takes advantage of rectified binocular stereo pairs (i.e with fronto parallel cameras) with known intrinsic matrix K and baseline T_r.

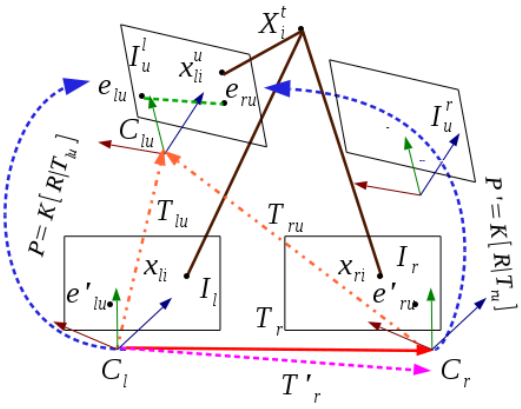


Figure 1. baseline T_r vs. recovered baseline T_r'

Fig.1 shows how the baseline endpoints (C_l, C_r) are projected to (e_{lu}, e_{ru}) by $[R|T_{lu}]$ and $[R|T_{ru}]$. The dotted green line connecting (e_{lu}, e_{ru}) shows how the points along T_r are projected by $P = K[R|T_{lu}]$ in I_u^l.

2 Related work

The idea of using one or more images has been previously explored to compute joint optical flow and disparity in [11, 18, 21], where the reference image is segmented and each segment is assumed to be a moving plane. All these algorithms work using four similar steps: 1) Compute an initial estimate of disparity and optical flow, 2) do plane fitting to generate plane hypotheses per image segment, 3) estimate transformation per segment, and 4) do per segment plane inference. This type of approach is known as scene flow estimation. Using disparity planes to estimate sub-pixel stereo disparity has been previously used in [9, 20, 4, 3, 13, 7, 15, 14, 21]. These algorithms can be classified in two categories: fixed plane inference (FPI) and dynamic plane inference (DPI). FPI algorithms usually work by making an initial disparity estimation and then extracting a set of plane hypotheses, which are then used to compute the 3D plane labeling. DPI algorithms use one or more plane hypotheses per pixel (either from a random initialization or pre-computed solution), and then propagate the planes with the "best" scores (depending on the cost function) to neighboring pixels/regions assuming that neighbors/regions may have the same plane. The initial plane labeling is refined in a separate stage, i.e. planes are dynamically updated. DPI algorithms have become the state of the art (e.g. [4, 3, 7, 15]). In this paper we follow the DPI approach.

In order to estimate the transformation from I_l/I_r to

I_u the most common approach is to compute keypoints and recover the camera position as in [6], and then do bundle adjustment to refine the obtained solution (e.g. [1, 17]). These algorithms are designed to work with multiview uncalibrated stereo, and the bundle adjustment process estimates both optimized camera positions and 3D points. Another option is to compute the trifocal tensor using either matching points [6] or lines [22, 16] to recover the missing camera position like in the configuration described in fig.1, or use the trifocal tensor to do point/line transfer, which has the inconvenience of being unreliable at points that are close to the epipolar plane. Using either the multi-view approach or trifocal tensor is an overkill especially when there are two calibrated cameras and only one extra camera's position needs to be computed.

2.1 Contributions

As noted previous approaches to recover the transformation P either relies on optical flow and disparity estimates, or using camera estimation algorithms that do not take into account the particular case of a calibrated stereo rig displacing in space. Our type of scenario is commonly found in vehicles moving either forward or backwards (e.g. [11]). In this paper we present a camera recovery algorithm that exploits existing calibration to constrain and estimate a transformation $P = K[R|T_{lu}]$ (assuming C_l is the origin in fig.1) that maps a 3D point X_i recovered from images I_l/I_r to a point X_i' consistent with the projected point x_i^u in I_u, and in doing so we also develop a pixel cost similarity that seamlessly uses a third image to reduce some of the ambiguities inherent to the standard binocular stereo matching pixel cost. Our contributions are:

- Algorithm to recover a rigid transformation $[R|T_{lu}]$ constrained by the baseline T_r of the calibrated stereo rig.
- Pixel similarity function that integrates three views in a DPI algorithm (we use [8]).

3 Baseline recovery

The case of a calibrated stereo rig moving as in fig.1 has the characteristic that the camera center C_{lu} is projected as the epipoles e'_{lu}/e'_{ru} in I_l/I_r, and the distance between e'_{lu} and e'_{ru} is related to the baseline size. Furthermore, if two fundamental matrices F_{lu} and F_{ru} are available then $[R|T_{lu}]$ and $[R|T_{ru}]$ are extracted, and it is trivial to compute the baseline $T'_r = T_{lu} - T_{ru}$, but most importantly it is possible to measure the error of the recovered baseline. Consider the following case:

$$K \cdot [I|C_l] \cdot C_{lu} - K \cdot [I|C_r] \cdot C_{lu} = e'_{lu} - e'_{ru} \tag{3}$$

Eq.3 assumes that both cameras in the stereo rig are are fronto parallel with the same intrinsic parameters, and to further simplify the situation let the camera C_l be at the origin in world coordinates and thus $C_r = T_r$, $C_{lu} = T_{lu}$. Eq.3 then simplifies to:

$$\begin{aligned} K(\cancelto{\mathbf{0}}{T_{lu} - T_{lu}} - T_r) &= e'_{lu} - e'_{ru} \\ -K \cdot T_r &= e'_{lu} - e'_{ru} \end{aligned} \tag{4}$$

All 3D points projected in the image using the intrinsic matrix K are equal up to a scale factor [6] and thus from eq.4 the following relation is derived:

$$\|T_r\| = S\|K^{-1}(e'_{lu} - e'_{ru})\| \tag{5}$$

When calibration is available and T_r is known there are only three unknowns: S, e'_{lu} and e'_{ru}. The epipoles e'_{lu} and e'_{ru} are extracted from fundamental matrices F_{lu} and F_{ru}, which can be computed from keypoints that are consistent across three views (I_l,I_r and I_u^l). Therefore it is trivial to compute the scale factor S and estimate a baseline T'_r (fig.1) using eq.4, and R can be extracted from F_{lu}. Note T'_r and T_r should be same, however due to noise in the points used to estimate F_{lu}, F_{ru}, T'_r is an approximation of T_r.

3.1 Finding consistent transformations

The process of estimating the baseline T'_r can be stated as finding updated $[\hat{R}|\hat{T}_{lu}]$ and $[\hat{R}|\hat{T}_{ru}]$ such that they can be used to: approximate the baseline T_r, obtain fundamental matrices $\hat{F}_{lu}$ and $\hat{F}_{ru}$ that produce a minimal Sampson error when evaluated, with both T_{lu}, T_{ru} translating points to the same depth, because we have fronto parallel cameras, and minimum reprojection error of a point X_i to the third view. This is expressed as the following optimization problem:

$$\begin{aligned} \arg \min_{\hat{R},\hat{T}_{lu},\hat{T}_{ru}} \sum_{i=0}^{n} [ds(x_{li}, x_{li}^u, F_{lu}) + ds(x_{ri}, x_{li}^u, F_{ru}) \\ + \|x_{li}^u - \phi(PX_i))\|] + \|T_r - T'_r\| \end{aligned} \tag{6}$$

where X_i^t is a 3D point at time t, x_{li}, x_{ri}, x_{li}^u are the projections of X_i in I_l, I_r, I_u^l with $P = K[R|T_{lu}]$. $\hat{F}_{lu} = K^{-1'}[\hat{T}_{lu}]_\times \hat{R} K^{-1}$ and $\hat{F}_{ru} = K^{-1'}[\hat{T}_{ru}]_\times \hat{R} K^{-1}$ are fundamental matrices consistent with the recovered baseline, $ds(x, x', F)$ is the Sampson error. Eq.6 is parametrized such that $\hat{R} = R_{\Delta\theta_x \Delta\theta_y \Delta\theta_z} R$, $\hat{T}_{lu} = T_{lu} + (\Delta T_{lu}^x, \Delta T_{lu}^y, \Delta T_u^z)$ and $\hat{T}_{ru} = T_{ru} + (\Delta T_{ru}^x, \Delta T_{ru}^y, \Delta T_u^z)$ where R , $T_{lu} = \beta K^{-1} e'_{lu}$, and $T_{ru} = \beta K^{-1} e'_{ru}$ are the initial estimates with $\beta = K_{11}\|T_r\|/\|e'_{lu} - e'_{ru}\|$ assuming a single focal length. Both $\hat{T}_{lu}, \hat{T}_{ru}$ share ΔT_u^z, which gives a total of 8 parameters to optimize, three for rotation and five for translation. To ensure that initial (T_{lu} and T_{ru}) move points to the same depth their z component is set to the same initial value, selecting either of T_{lu}^z, T_{ru}^z. Eq.6 is minimized using the Levenberg-Marquardt algorithm to estimate $R_{\Delta\theta_x \Delta\theta_y \Delta\theta_z}$, $(\Delta T_{lu}^x, \Delta T_{lu}^y)$, $(\Delta T_{ru}^x, \Delta T_{ru}^y)$, and ΔT_u^z to update transformations and make them consistent with the three views and the stereo rig baseline T_r, i.e. recover the baseline. Finally, a second solution $\hat{R}', \hat{T}'_{lu}, \hat{T}'_{ru}$ is computed by minimizing again eq.6 using the previously estimated $R_{\Delta\theta_x \Delta\theta_y \Delta\theta_z}$ with $(\Delta T_{lu}^x = 0, \Delta T_{lu}^y = 0)$, $(\Delta T_{ru}^x = 0, \Delta T_{ru}^y = 0)$, $\Delta T_u^z = 0$ as initial estimates, and keeping the best solution. This compensates for noisy initial estimates of T_{lu} and T_{ru}.

3.2 Computing initial estimates

The initial transformations (R, T_{lu}, and T_{ru}) and 3D points (X_i) are estimated by performing the following steps:

(1) Compute matching ASIFT [12] key points $(x_{li}, x_{ri}, x_{li}^u)$ for views I_l, I_r and I_u^l.

(2) Compute the 3D points X_i^t from key points x_{li}, x_{ri}.
(3) Compute F_{lu} from x_{li}, x_{li}^u, and F_{ru} from x_{ri}, x_{li}^u using the normalized 8-point algorithm [6].
(4) Compute R from F_{lu} using the algorithm described in [6], and (T_{lu}, T_{ru}) as in sec.3.1.

4 Stereo matching

The pixel similarity function (from eq.1) used in this paper is made up from three terms $\hat{C}_p(D_p) = C_p(D_p) + U(D_p) + O(D_p)$. $C_p(D_p)$ the aggregation function from [8] applied to the raw pixel similarity cost $c_p(D_p)$ (eq.9), $U(D_p)$ is the uniqueness term, and $O(D_p)$ the out of range term.

$$U(D_p) = \begin{cases} \tau_{unique} & : L(Dp) \\ 0 & : \text{otherwise} \end{cases} \tag{7}$$

$$O(D_p) = \begin{cases} 1 - exp(-|D_p - minD|/\sigma_d) & : D_p < minD \\ 1 - exp(-|D_p - maxD|/\sigma_d) & : D_p > maxD \\ 0 & : \text{otherwise} \end{cases} \tag{8}$$

The local uniqueness term $U(D_p)$ from eq.7 penalizes pixels with multiple matches, with $L(Dp)$ true when a pixel p is mapped to a pixel $p + D_p(p)$ which has more than one match. Fig.2 shows an example of uniqueness constraint violation: two pixels (red arrows) in left image scanline map to a single pixel in right image (red pixel). The out of range term $O(D_p)$

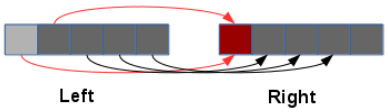


Figure 2. Uniqueness constraint violation

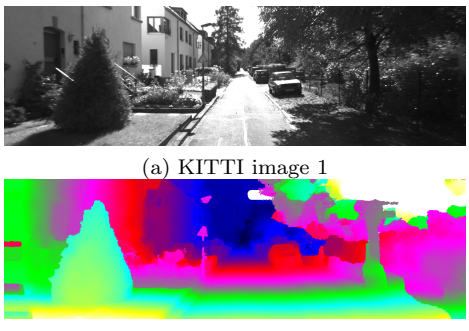
(eq.8) penalizes disparity values that lie outside a defined search range, where $minD$ and $maxD$ are the minimum and maximum of the disparity search range, while σ_d is the maximum deviation allowed for values outside the search range. The non-aggregated pixel similarity function is given by:

$$c_p(D_p) = \alpha c_p^1(D_p(p)) + c_p^2(D_p(p)) \tag{9}$$

$$\begin{aligned} c_p^1(D_p) = {} & \alpha_t \cdot min(|\nabla I_l(p) - \nabla I_r(p + D_p(p))|, \tau_{grad}^b) \\ & + (1 - \alpha_t) \cdot min(|\nabla I_l(p) - \nabla I_u^l(\phi(PX))|, \tau_{grad}^t) \end{aligned} \tag{10}$$

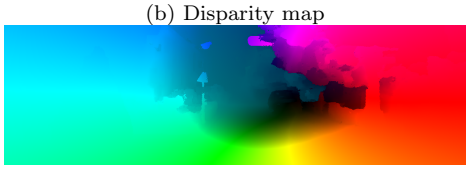
$$\begin{aligned} c_p^2(D_p) = {} & \alpha_t \cdot min(\chi(I_l, I_r, p, D_p), \tau_{cen}^b) \\ & + (1 - \alpha_t) \cdot min(\chi(I_l, I_u^l, p, D_p), \tau_{cen}^t) \end{aligned} \tag{11}$$

I_l is the reference image, I_r and I_u^l are the target images. $c_p^1(D_p)$ is the truncated absolute differences of gradients using using $(\tau_{grad}^b, \tau_{grad}^t)$. $c_p^2(D_p)$ is the truncated Hamming distance of the census transform using $(\tau_{cen}^b, \tau_{cen}^t)$, χ computes the census transform and Hamming distance at pixel p with disparity plane D_p , α balances the pixel-wise cost influence. In this way the I_u^l is included to improve the binocular match cost. The trinocular cost influence is balanced with α_t to prevent points in the image I_u^l from having too much influence in case they have changed position, i.e. reduce outliers. Plane hypothesis generation and inference is done using the DPI algorithm from [8].



(a) KITTI image 1

(b) Disparity map



(c) Optical flow

Figure 3. Result for KITTI 2012 test image 1.

5 Experimental results

The baseline recovery algorithm is evaluated using the KITTI 2012 data set, where groundtruth depth maps are projected to a third image displaced in time and then optical flow is computed using the recovered motion from our algorithm and compared with the groudtruth (tab.1). The proposed stereo matching approach was evaluated using the KITTI 2012 stereo disparity and optical flow benchmarks, and compared with a binocular algorithm (tab.2). Our algorithm is also compared to the state of the art competitors (the best performing convolutional neural network algorithms and the best performing algorithm not using convolutional neural networks). The algorithms compared appear in both data set evaluation tables. Our approach is among the top performers on the KITTI 2012[1] optical flow (tab.3) and stereo (tab.4) benchmarks (submitted as TBR). Fig.3 shows an example of the resulting disparity map and its mapping to optical flow using the recovered motion (images are displayed using false color). Tab.1 compares our approach to recover the camera motion with the 6 points algorithm $6PT$ to recover 3 cameras [6]. The evaluation uses 40 images from KITTI 2012 and measures the average pixel displacement error of all pixels in the optical flow evaluated computed using our approach (using every 5th image from KITTI 2012). We report the error of the initial camera motion estimate on all images (*avg. init.*) and error after refinement (*avg. ref.*) using our approach. Our algorithm has lower error on initialization and it is further reduced after refinement, whereas $6PT$ has a large error on initialization (even after using RANSAC) also it is slower as it optimizes 24 vs. 8 parameters using our approach. The $6PT$ algorithm was refined using our approach. Tab.2 show that using the baseline recovery and the proposed trinocular cost gives better results in non-occluded areas, and also shows that the baseline recovery algorithm works as intended in images with no moving objects. In the KITTI benchmark, our algorithm ranks 11^{th} (out of 85), and 15^{th} (out of 89) for KITTI 2012 optical flow and stereo respectively. The evaluation on KITTI 2012 proved challenging due colored intensity images that

[1] see supporting material for parameter settings.

are not properly aligned to the ground truth shape image, causing problems for the aggregation algorithm. The top performing competitors achieve high performance by: using scene specific content to eliminate ambiguities (e.g. cars in Disp.v2), training specifically for the data sets (e.g. MCNCC, SDF), or using 2-3 image pairs to estimate disparity (e.g. PRSM, OSF). By contrast the proposed algorithm achieves top performing results in multiple data set by: using only the left, right and $t+1$ left images, using baseline recovery, not using scene specific features (e.g. cars), and not computing optical flow directly but instead mapping disparities using the recovered motion.

Table 1. Baseline recovery accuracy.

Algorithm	avg. init.	avg. ref.	time secs.
Our	6.47	0.52	0.23
$6PT$	9.16	0.53	152.91

Table 2. Trinocular vs. Binocular evaluation.

Algorithm	%bad noc	%bad occ	avg. noc	avg. occ
Our	3.07	4.13	0.69	0.86
bino	3.22	4.12	0.72	0.86

Table 3. Optical flow evaluation. Non-anonymous entries are used for comparison: PRSM[18], OSF[11], SDF[2].

Algorithm	%bad noc	%bad occ	avg. noc	avg. occ
Our^{11th}	4.24	7.50	0.9	1.5
PRSM^{1st}	2.46	4.23	0.7	1.0
OSF^{5th}	3.47	6.34	1.0	1.5
SDF^{9th}	3.80	7.69	1.0	2.3

Table 4. Disparity evaluation. Non-anonymous entries are used for comparison: Disp. v2[5], MCNCC[19].

Algorithm	%bad noc	%bad occ	avg. noc	avg. occ
Our^{15th}	3.09	4.29	0.70	0.90
Disp. v2^{4th}	2.37	3.09	0.70	0.80
MCNCC^{5th}	2.43	3.63	0.70	0.90
PRSM^{10th}	2.78	3.00	0.70	0.70
OSF^{19th}	3.28	4.07	0.80	0.90

6 Conclusions

The baseline recovery is to the best of our knowledge a novel technique to recover camera motion that integrated easily in a DPI dense trinocular algorithm. The proposed algorithm successfully exploits the temporal displacement of a third image to accurately recover camera motion and also delivers high performing optical flow and disparity estimation results even though only the general motion is computed, no pre-computed optical flow is used, and no convolutional neural network (e.g. [19, 5, 2]) or prior 3D models (e.g. cars) are used.

Acknowledgments: This research was done with the support of the Mexican CONACYT programme and the European Commission.

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S.M. Seitz, and R. Szeliski. Building rome in a day. *Commun. ACM*, 54(10):105–112, 2011.

[2] M. Bai, W. Luo, K. Kundu, and R. Urtasun. Exploiting semantic information and deep matching for optical flow. *ECCV*, pages 154–170, 2016.

[3] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. Pmbp: Patchmatch belief propagation for correspondence field estimation. *IJCV*, 110(1):2–13, 2012.

[4] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. *BMVC*, 11:1–11, 2011.

[5] F. Guney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. *CVPR*, 2015.

[6] R.I. Hartley and A. Zisserman. Multiple view geometry in computer vision. 2004.

[7] P. Heise, S. Klose, B. Jensen, and A. Knoll. Patchmatch with huber regularization for stereo matching. *ICCV*, pages 2360–2367, 2013.

[8] L. Horna and R.B. Fisher. 3d plane labeling stereo matching with content aware adaptive windows. *VISAPP*, 2017.

[9] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. *ICPR*, 3:15–18, 2006.

[10] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *TPAMI*, 28(10):1568–1583, 2007.

[11] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. *CVPR*, 2015.

[12] J.M. Morel and G.Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[13] C. Olsson, J. Ulén, and Y. Boykov. In defense of 3d-label stereo. *CVPR*, pages 1730–1737, 2013.

[14] S. Sinha, D. Scharstein, and R. Szeliski. Efficient high-resolution stereo matching using local plane sweeps. *CVPR*, pages 1582–1589, 2014.

[15] T. Taniai, Y. Matsushita, and T. Naemura. Graph cut based continuous stereo matching using locally shared labels. *CVPR*, pages 1613–1620, 2014.

[16] M. Trager, J. Ponce, and M. Hebert. Trinocular geometry revisited. *IJCV*, 120(2):134–152, 2016.

[17] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment — a modern synthesis. *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms*, pages 298–372, 2000.

[18] C. Vogel, K. Schindler, Konrad, and S. Roth. 3d scene flow estimation with a piecewise rigid scene model. *IJCV*, pages 1–28, 2015.

[19] J. Žbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *JMLR*, 2015.

[20] O. J. Woodford, I. D. Reid, P. H. S. Torr, and A. W. Fitzgibbon. On new view synthesis using multiview stereo. *BMVC*, pages 1–10, 2007.

[21] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. *ECCV*, pages 756–771, 2014.

[22] M. Zhao and R. Chung. Critical configurations of lines to geometry determination of three cameras. *ICPR*, pages 1–5, 2008.