Plane labeling trinocular stereo matching with baseline recovery
Luis Horna
The University of Edinburgh
s1145121@sms.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 $X_l$ to $x_l^P$ in $I_l$. Finding the optimal 3D disparity plane labeling $D$ is modeled as an optimization problem where the objective is to minimize eq.1.

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

$E(D)$ 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 $q$. $N_{numP}$ 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 $n_p = (n_p^x, n_p^y, n_p^z)$ and disparity $d_p$. The disparity of pixel $q = (x_q, y_q)$ using $D_p$ is given by:

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

where $a = -\hat{n}_p^x/n_p^z$, $b = -\hat{n}_p^y/n_p^z$ and $c = (\hat{n}_p^x \times x_q + \hat{n}_p^y \times y_q + \hat{n}_p^z \times d_p)/\hat{n}_p^z$ as in [4]. $C_p$ is a function that measures the similarity/dissimilarity of three pixels, e.g. $I_l(p)$ is compared to $I_l(p + D_p(p))$ and $I_u^*(\phi(P \cdot X))$ 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 $I_u^*$, 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 $T_r$.

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 $I_u^*$.

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
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 okaykill 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 rely 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 $P = K[R[T]]$ (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 $P = K[R[T]]$ 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 $C_l$ is projected as the epipoles $e_r^u/e_u^u$ in $I_l/I_r$, and the distance between $e_r^u$ and $e_u^u$ is related to the baseline size. Furthermore, if two fundamental matrices $F_{Iu}$ and $F_{ru}$ are available then $[R[T]]$ and $[R[T]]$ 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:

\[
K \cdot [I[C_l] \cdot C_{Iu} - K \cdot [I[C_r] \cdot C_{Iu}] = e_{lu} - e_{ru} \tag{3}
\]

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

\[
(K[T_{lu} - T_{ru}]^0 - T_r) = e_{lu} - e_{ru} - K \cdot T_r = e_{lu} - e_{ru} \tag{4}
\]

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 $e_{lu}^r$ and $e_{ru}^r$ are extracted from fundamental matrices $F_{Iu}$ and $F_{ru}$, which can be computed from keypoints that are consistent across three views ($I_l$, $I_r$, and $I_u$). 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 $F_{Iu}$. Note $T_r'$ and $T_r$ should be the same, however due to noise in the points used to estimate $F_{Iu}$, $F_{ru}$, $T_r'$ 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 $[R[T_{lu}]]$ and $[R[T_{ru}]]$ such that they can be used to: approximate the baseline $T_r$, obtain fundamental matrices $F_{lu}$ and $F_{ru}$ 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:

\[
\arg \min_{R,T_{lu},T_{ru}} \sum_{i=0}^{n} ||ds(x_i, x_i^u, F_{Iu}) + ds(x_i, x_i^u, F_{ru}) + ||x_i^u - \phi(PX_i)|| ||T_r - T_{lu}|| \tag{6}
\]

where $X_i^u$ is a 3D point at time $t$, $x_i$, $x_i^u$, $x_i^r$ are the projections of $X_i$ in $I_l/I_r/I_u$ with $P = K[R[T]]$. $F_{lu} = K^{-1}[T_l] \times RK^{-1}$ and $F_{ru} = K^{-1}[T_u] \times RK^{-1}$ are fundamental matrices consistent with the recovered baseline, $ds(x, x', F)$ is the Sampson error. Eq.6 is parametrized such that $R = R_{\Delta \theta} \Delta \rho \Delta \phi \Delta \theta$, $T_{lu} = T_{lu} + (\Delta T_{lu}^x, \Delta T_{lu}^y, \Delta T_{lu}^z)$ and $T_{ru} = T_{ru} + (\Delta T_{ru}^x, \Delta T_{ru}^y, \Delta T_{ru}^z)$ where $R$, $T_{lu}$, $T_{ru}$ are the initial estimates with $\beta = K_{11}/||T_{lu}||/||e_{lu} - e_{ru}||$ assuming a single focal length. Both $T_{lu}$, $T_{ru}$ share $\Delta T_{r}^z$, which gives a total of 8 parameters to optimize, for three rotation and five for translation. To ensure that initial ($T_{lu}$ and $T_{ru}$) move the same to the depth their z component is set to the same initial value, selecting either of $T_{lu}$, $T_{ru}$. Eq.6 is minimized using the Levenberg-Marquardt algorithm to estimate $R_{\Delta \theta} \Delta \rho \Delta \phi \Delta \theta$, $(\Delta T_{lu}^x, \Delta T_{lu}^y, \Delta T_{lu}^z)$, $(\Delta T_{ru}^x, \Delta T_{ru}^y, \Delta T_{ru}^z)$, and $\Delta T_{r}^z$ 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 $R$, $T_{lu}^r$, $T_{ru}^r$ is computed by minimizing again eq.6 using the previously estimated $R_{\Delta \theta} \Delta \rho \Delta \phi \Delta \theta$, with $\Delta T_{lu}^z = 0$, $\Delta T_{ru}^z = 0$, $(\Delta T_{lu}^x = 0, \Delta T_{lu}^y = 0, \Delta T_{ru}^x = 0, \Delta T_{ru}^y = 0)$, $\Delta T_{r}^z = 0$ 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 $(x_i, x_i^u, x_i^r)$ for views $I_l$, $I_r$, and $I_u$. 


(2) Compute the 3D points $X^t_i$ from key points $x_{li}, x_{ri}$.
(3) Compute $F_{lu}$ from $x_{li}, x_{lu}$, and $F_{ru}$ from $x_{ri}, x_{ru}$ 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 $C_p(D_p) = C_p(D_p) + U(D_p) + O(D_p)$. $C_p(D_p)$ 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.

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

$$O(D_p) = \begin{cases} 1 - \exp(-|D_p - \text{min} D|/\sigma_d) : D_p < \text{min} D \\ 1 - \exp(-|D_p - \text{max} D|/\sigma_d) : D_p > \text{max} D \\ 0 : \text{otherwise} \end{cases}$$

The local uniqueness term $U(D_p)$ from eq.7 penalizes pixels with multiple matches, with $L(D_p)$ 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 $\text{min} D$ and $\text{max} D$ are the minimum and maximum of the disparity search range, while $\sigma_d$ is the maximum deviation allowed for values outside the search range. The non-aggregated pixel similarity function is given by:

$$c_p(D_p) = \alpha^t_1(D_p(p)) + c_p^2(D_p(p))$$

$$c_p^1(D_p) = \alpha_1 \cdot \min(|\nabla I_1(p) - \nabla I_1(p + D_p(p))|, \tau^p_{\text{grad}})$$

$$+ (1 - \alpha_1) \cdot \min(|\nabla I_1(p) - \nabla I_1(p + \phi(PX))|, \tau^p_{\text{cen}})$$

$$c_p^2(D_p) = \alpha_2 \cdot \min(\chi(I_1, I_1, p, D_p), \tau^p_{\text{cen}})$$

$$+ (1 - \alpha_2) \cdot \min(\chi(I_1, I_1, p, D_p), \tau^p_{\text{cen}})$$

$I_1$ is the reference image, $I_1$ and $I_2$ are the target images. $c_p^2(D_p)$ is the truncated absolute differences of gradients using $\tau^p_{\text{grad}}$. $c_p^2(D_p)$ is the truncated Hamming distance of the census transform using $\tau^p_{\text{cen}}$. $\chi$ computes the census transform and Hamming distance at pixel $p$ with disparity plane $D_p$, $\alpha$ balances the pixel-wise cost influence. In this way the $I_2$ is included to improve the binocular match cost. The trinocular cost influence is balanced with $\alpha$ to prevent points in the image $I_2$ 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].

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 groundtruth (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 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 11th (out of 85), and 15th (out of 89) for KITTI 2012 optical flow and stereo respectively. The evaluation on KITTI 2012 proved challenging due colored intensity images that...
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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>avg. init.</th>
<th>avg. rel.</th>
<th>time secs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our</td>
<td>6.47</td>
<td>0.52</td>
<td>0.23</td>
</tr>
<tr>
<td>6PT</td>
<td>9.16</td>
<td>0.53</td>
<td>152.91</td>
</tr>
</tbody>
</table>

Table 2. Trinocular vs. Binocular evaluation.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>%bad</th>
<th>%bad</th>
<th>avg.</th>
<th>avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our</td>
<td>3.07</td>
<td>4.13</td>
<td>0.69</td>
<td>0.86</td>
</tr>
<tr>
<td>bino</td>
<td>3.22</td>
<td>4.12</td>
<td>0.72</td>
<td>0.86</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>%bad</th>
<th>%bad</th>
<th>avg.</th>
<th>avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our 11th</td>
<td>4.24</td>
<td>7.50</td>
<td>0.9</td>
<td>1.5</td>
</tr>
<tr>
<td>PRSM 1st</td>
<td>2.46</td>
<td>4.23</td>
<td>0.7</td>
<td>1.0</td>
</tr>
<tr>
<td>OSF 5th</td>
<td>3.47</td>
<td>6.34</td>
<td>1.0</td>
<td>1.5</td>
</tr>
<tr>
<td>SDF 9th</td>
<td>3.80</td>
<td>7.69</td>
<td>1.0</td>
<td>2.3</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>%bad</th>
<th>%bad</th>
<th>avg.</th>
<th>avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our 16th</td>
<td>3.09</td>
<td>4.29</td>
<td>0.70</td>
<td>0.90</td>
</tr>
<tr>
<td>Disp. v24th</td>
<td>2.37</td>
<td>3.09</td>
<td>0.70</td>
<td>0.80</td>
</tr>
<tr>
<td>MCNCC 5th</td>
<td>2.43</td>
<td>3.63</td>
<td>0.70</td>
<td>0.90</td>
</tr>
<tr>
<td>PRSM 9th</td>
<td>2.78</td>
<td>3.00</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>OSF 9th</td>
<td>3.28</td>
<td>4.07</td>
<td>0.80</td>
<td>0.90</td>
</tr>
</tbody>
</table>

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