πMatch: Monocular vSLAM and Piecewise Planar Reconstruction Using Fast Plane Correspondences

Carolina Raposo(✉) and João P. Barreto

Institute of Systems and Robotics, University of Coimbra, Coimbra, Portugal
{carolinaraposo,jpbar}@isr.uc.pt

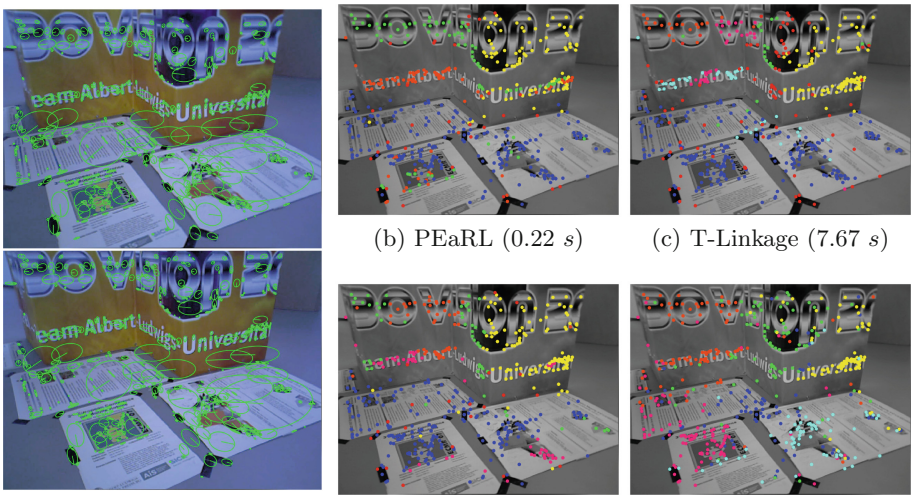
Abstract. This paper proposes πMatch, a monocular SLAM pipeline that, in contrast to current state-of-the-art feature-based methods, provides a dense Piecewise Planar Reconstruction (PPR) of the scene. It builds on recent advances in planar segmentation from affine correspondences (ACs) for generating motion hypotheses that are fed to a PEaRL framework which merges close motions and decides about multiple motion situations. Among the selected motions, the camera motion is identified and refined, allowing the subsequent refinement of the initial plane estimates. The high accuracy of this two-view approach allows a good scale estimation and a small drift in scale is observed, when compared to prior monocular methods. The final discrete optimization step provides an improved PPR of the scene. Experiments on the KITTI dataset show the accuracy of πMatch and that it robustly handles situations of multiple motions and pure rotation of the camera. A Matlab implementation of the pipeline runs in about 0.7 s per frame.

Keywords: Monocular visual SLAM · Piecewise planar reconstruction

1 Introduction

Monocular Visual Simultaneous Localization and Mapping (vSLAM) is the process of estimating the camera position and orientation while building 3D maps of the environment, from a single camera. Although there has been intensive research on this topic, current methods still face several challenges and difficulties, including (i) presence of outliers, (ii) dynamic foregrounds and pure rotation of the camera, (iii) large baselines, (iv) scale drift, (v) density of 3D reconstruction, and (vi) computational efficiency. Nowadays, existing methods for monocular vSLAM follow two distinct approaches: feature extraction and direct image alignment. Each paradigm is effective in solving some of these challenges but, to the best of our knowledge, there is no monocular vSLAM algorithm that is able to tackle all these issues. While feature-based methods work on top of extracted features and are usually robust to outliers by applying RANSAC-based schemes [11], direct methods perform whole image alignment

B. Leibe et al. (Eds.): ECCV 2016, Part VIII, LNCS 9912, pp. 380–395, 2016.
DOI: 10.1007/978-3-319-46484-8_23



(a) Img. pair w/ matched ACs (b) PEaRL (0.22 s) (c) T-Linkage (7.67 s) (d) PEaRL (4.36 s) (e) Affinity Prop. (0.20 s)

Fig. 1. Plane segmentation problem solved using different methods: (b) PEaRL [14] with 300 homography hypotheses, (c) T-linkage [18] with 300 hypotheses,(d) PEaRL with 5000 hypotheses, and (e) affinity propagation [7]. The computational times of PEaRL and T-linkage hamper real-time performance. On the contrary, affinity propagation is fast and is able to detect all the planes present in the image. Red points correspond to outliers. (Color figure online)

and cannot handle outliers [2,4,19]. Moreover, the former work with wide baselines and provide sparse reconstructions, as opposed to the latter that require small baselines, which is typically accomplished by high frame rates that tend to limit image resolution, and provide dense scene models. All feature-based methods [3,15] perform tracking and mapping as separate tasks. This greatly reduces the complexity of the problem, allowing them to work in real-time. On the other hand, direct methods such as [19,20] compute dense depth maps using variational approaches which are computationally expensive and require powerful GPUs to achieve real-time performance. Only recently, direct methods that estimate semi-dense depth maps have been proposed [2,4,5], allowing real-time operation on a CPU. Most feature-based monocular methods assume there is significant camera translation and that the scene is mainly rigid for applying epipolar geometry. However, there might be situations where this does not hold and a scheme to robustly estimate the camera motion is desirable. Both direct and non-direct methods perform poorly in the presence multiple motions and tend to drift in scale. While there is no explicit solution for the first problem in the state-of-the-art, the last issue is typically solved using prior information such as the height of the camera [11,23] or the existence of loop closures for performing global optimization [4].

The advantages of using planes as opposed to point features has been demonstrated by recent work on Structure-from-Motion (SfM) with a stereo camera [21]. Performing PPR in monocular sequences has never been much explored due to the difficulties in detecting planes without knowing the camera motion. One possibility would be to use an hypothesize-and-test framework, such as RANSAC, to fit homographies, but this lacks robustness and is time consuming [14]. Other greedy methods such as J-linkage [24] or its continuous relaxation T-linkage [18] could also be used but they still suffer from low computational efficiency (Fig. 1c). An alternative would be to use discrete optimization to replace greedy methods by a global scheme such as PEaRL [14] but, although there are improvements, the results are still not satisfactory (Figs. 1b and d).

Recent work using affine correspondences (ACs) [22] has shown that it is possible to establish necessary conditions for two ACs to belong to the same plane. The authors define an error metric that allows to quickly segment ACs in planes, without the need to generate homography hypotheses as in hypothesize-and-test approaches. We build on this recent advance and propose a complete vSLAM pipeline that relies on plane features, named πMatch. ACs are extracted and quickly clustered into coplanar regions using affinity propagation [7] (Fig. 1e) based on the new metric. For each plane cluster, a fast, robust scheme estimates the corresponding homography, which is decomposed into two solutions for rotation R and translation $\mathbf{t}$ (Sect. 2.1). The obtained motion hypotheses are used as input to a PEaRL formulation that merges close motions and decides about multiple motion situations (e.g. dynamic foreground, pure rotation of camera, etc.) (Sect. 2.2). Given the refined camera motion, the initial plane hypotheses are also merged and refined in a PEaRL framework, and, as an option, used as input to a standard Markov Random Field (MRF) formulation [1,8] for dense pixel labeling and subsequent PPR (Sect. 2.3). This two-view pipeline is applied to each image pair, providing camera motion estimations up to scale. As a final step, we use a fast scheme for scale estimation based on the minimization of the reprojection error that benefits from the high accuracy in the estimation of R and $\mathbf{t}$. This is followed by a discrete optimization step for improving the final PPR of the scene (Sect. 4). πMatch makes considerable advances in handling the aforementioned difficulties, being advantageous with respect to the state-of-the-art methods (Table 1).

2 Two-View SfM and PPR Using πMatch

We propose πMatch, a Structure-from-Motion framework that is able to automatically recover the camera motion and a PPR of the scene from a monocular sequence. For each image pair, ACs are extracted and used for computing the error metric of compatibility between two ACs and an homography proposed in [22]. These measures of similarity between pairs of ACs are used for segmenting planes by affinity propagation (AP) [7]. A robust MSAC scheme [25] is then applied to each cluster for filtering out outliers. This step provides a plane segmentation and a set of motion hypotheses, from which the ones present in the

Table 1. Advantages of the proposed method πMatch over existing feature-based and direct methods.

	Feature-based	Direct	πMatch
Robust to outliers	+	−	+
Dynamic foreground/pure rotation	−	−	+
Wide baselines	+	−	+
Scale drift problem	*Camera height*	*Loop closure*	*No priors*
Model density	−	+	+
Computational efficiency	*Real-time*	*Parallelizable*	*Near real-time*

image pair are selected in a PEaRL [14] framework. The dominant one, which is assumed to be the camera motion, is identified and refined. Another PEaRL step is applied for plane merging and refinement, and a final standard MRF [1,8] can be used for dense pixel-labeling. Figure 2 shows the sequence of steps of the proposed pipeline. The next subsections detail each building block using the image pair of Fig. 1 as an illustrative example.

2.1 Generation of Motion Hypotheses

An AC consists in a point correspondence $(\mathbf{x}, \mathbf{y})$ across views and a non-singular 2×2 matrix A that maps image points surrounding $\mathbf{x}$ into image points in the neighbourhood of $\mathbf{y}$, with

$$\mathbf{x} = \begin{bmatrix} x_1 \; x_2 \end{bmatrix}^\mathsf{T}, \mathbf{y} = \begin{bmatrix} y_1 \; y_2 \end{bmatrix}^\mathsf{T}, \mathsf{A} = \begin{bmatrix} a_1 \; a_3 \\ a_2 \; a_4 \end{bmatrix}. \tag{1}$$

Recent research on ACs [22] has shown that 2 ACs, $(\mathbf{x}, \mathbf{y}, \mathsf{A})$ and $(\mathbf{z}, \mathbf{w}, \mathsf{B})$, must satisfy 4 conditions in order to be compatible with the same homography:

$$\begin{aligned} (\mathbf{w}-\mathbf{y})^\mathsf{T}\mathsf{PA}(\mathbf{z}-\mathbf{x}) &= 0 \\ (\mathbf{w}-\mathbf{y})^\mathsf{T}\mathsf{PB}(\mathbf{z}-\mathbf{x}) &= 0 \\ \begin{bmatrix} s + a_2b_3 - a_3b_2 & -(a_1b_3 - a_3b_1) \\ a_2b_4 - a_4b_2 & s - (a_1b_4 - a_4b_1) \end{bmatrix} (\mathbf{w}-\mathbf{y}) &= \mathbf{0}, \text{ with.} \\ s = \frac{[-a_2+b_2 \;\; a_1-b_1](\mathbf{w}-\mathbf{y}) - (a_1b_2 - a_2b_1)(x_1 - z_1)}{(x_2 - z_2)} \quad \text{and } \mathsf{P} &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \end{aligned} \tag{2}$$

The authors devised an error metric from this result, which was validated in a plane segmentation experiment. Following this idea, for each pair of ACs, we compute an error metric by taking the average of the errors obtained for each condition, which are the values of the expression on the left-hand side of each equation in the system of Eq. 2. For C ACs, this results in a $C \times C$ matrix of similarities between pairs of ACs, which is fed to an AP method [7] for clustering the ACs into scene planes. Since all data points are assigned to a cluster, the obtained segmentation contains outliers. Moreover, there are cases in which AP

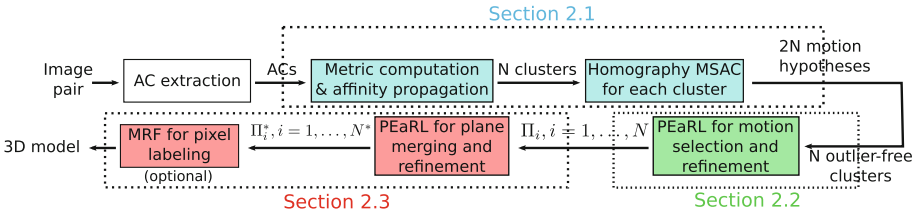


Fig. 2. Pipeline of the proposed method πMatch for two-view SfM and PPR. ACs are used for segmenting the scene into planes and providing motion hypotheses. The existing motions are selected in a PEaRL framework, and the dominant one is identified and refined. Plane hypotheses are generated from the clustering result and the refined motion, and PEaRL is again used for plane segmentation and refinement. A standard MRF scheme can be used for dense pixel-labeling.

tends to oversegment, providing several clusters that correspond to the same scene plane. This is shown in Fig. 3a, where the ground plane is segmented into 3 different clusters and there are data points incorrectly labeled.

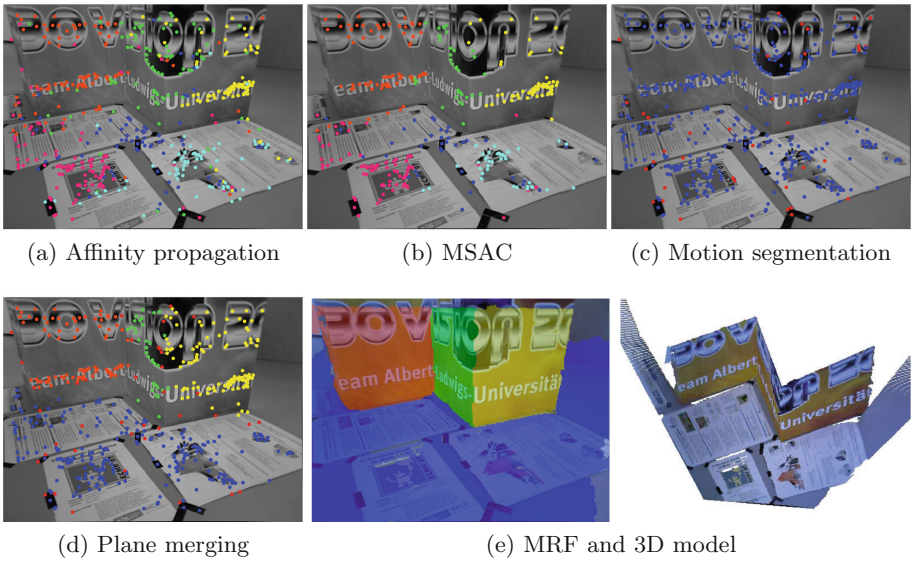
In order to filter out outliers, each cluster is used as input to a MSAC framework for homography estimation. We consider the minimal set of 2 ACs for generating homography hypotheses as proposed in [22], which provides a speed-up of approximately 3× when compared using a 4-point minimal set of point correspondences. Also, since each MSAC is performed for each cluster independently, they can run in parallel, significantly speeding up the process.

The output of this MSAC step is a set of homographies and corresponding outlier-free clusters (Fig. 3b). Decomposing each homography yields two solutions for the camera rotation R, translation $\mathbf{t}$, and plane $\mathbf{n}$, up to scale [17].

2.2 PEaRL for Motion Selection

Cases in which the camera motion is a pure rotation must be correctly identified since nor the scene planes neither the scale of translation can be recovered, and schemes must be devised to overcome this problem (Sect. 4). The previous step of the pipeline outputs N outlier-free clusters and $2N$ motion hypotheses. Firstly, the motions that correspond to pure rotations are identified. This is done by considering the corresponding homography H and computing the distance between the identity matrix I and the matrix $\mathsf{H}\mathsf{H}^\mathsf{T}$. We opted to use metric Φ_4 proposed in [13] for computing this distance as it is the most computationally efficient. Homographies for which this distance lies below a pre-defined threshold are decomposed and only the rotation component is considered by setting $\mathbf{t} = \mathbf{0}$.

There may be more than one motion present in the image due to moving objects in the foreground. In case these objects are planar, they will be identified by the plane segmentation step of the pipeline. Thus, a scheme to decide which planes correspond to rigid structures is required. We propose to solve this problem by selecting the motions present in the image in a PEaRL framework, and afterwards identifying the camera motion. The motion selection task can be



(a) Affinity propagation (b) MSAC (c) Motion segmentation

(d) Plane merging (e) MRF and 3D model

Fig. 3. Results for the image pair in Fig. 1a after each step of the pipeline. (a) AP tends to oversegment and does not identify outliers. (b) A robust scheme is required for filtering each cluster. (c) The best motion hypothesis is selected and (d) plane segmentation is performed, where the original plane hypotheses are merged. (e) πMatch provides an accurate 3D model of the scene from only two views. Colors across images identify planes. Outliers are shown in red. (Color figure online)

cast as a labeling problem where the nodes of the graph are the point correspondences $\mathbf{p}$, to which a label $l_{\mathbf{p}}$ must be assigned. The label set $\mathcal{L} = \{\{\mathcal{R}_0, \mathcal{T}_0\}, l_\emptyset\}$ consists of the set of motion hypotheses $\{\mathcal{R}_0, \mathcal{T}_0\}$ and the discard label $l_\emptyset$. This labeling problem is solved by minimizing an energy function E defined by

$$E(\mathbf{l}) = \underbrace{\sum_{\mathbf{p}} D_{\mathbf{p}}(l_{\mathbf{p}})}_{\text{Data term}} + \underbrace{\lambda_S \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{N}} w_{\mathbf{pq}}\delta(l_{\mathbf{p}} \neq l_{\mathbf{q}})}_{\text{Smoothness term}} + \underbrace{\lambda_L|\mathcal{L}_{\mathbf{l}}|}_{\text{Label term}}, \tag{3}$$

where λ_S and λ_L are weighting constants, $\mathbf{l}$ is the labeling being analysed, $\mathcal{N}$ is the neighbourhood of $\mathbf{p}$, weights $w_{\mathbf{pq}}$ set penalties for each pair of neighbouring data points $\mathbf{p}, \mathbf{q}$, and δ is 1 whenever the condition inside the parentheses is satisfied and 0 otherwise. The data term $D_{\mathbf{p}}$ is defined as the symmetric transfer error (STE) [12] if the label corresponds to a pure rotation and the Sampson distance [12] otherwise. Two nodes $\mathbf{p}$ and $\mathbf{q}$ are neighbours if they belong to the same cluster from the set of clusters provided by the MSAC step (Sect. 2.1). We set $w_{\mathbf{pq}} = 1$, meaning that an equal penalty is set to all neighbours. This definition of neighbourhood forces points belonging to the same scene plane to be assigned the same motion label. Finally, the label term forces the algorithm to use as few motion hypotheses as possible. Due to the small size of the label set

Fig. 4. Image pairs with the extracted ACs for 4 different scenarios: 1 - normal motion, 2 - dominant dynamic foreground, 3 - static camera/pure rotation, and 4 - four motions besides the camera motion. Examples 1 to 3 are from the KITTI dataset [9,10] and Example 4 is from the Hopkins dataset [26]. πMatch is applied to each scenario and the results are shown in Figs. 5 and 6.

(typically 8–14 motion hypotheses), this discrete optimization step is very fast. Figure 3c shows that the algorithm selected only one motion and some points were assigned the discard label $l_\emptyset$. If more than one motion is chosen, the one to which more clusters are associated is selected as the camera motion. In case this is satisfied by more than one hypothesis, the one to which more points were assigned is considered. The camera motion is finally refined with the selected inliers in a standard bundle adjustment with point correspondences.

2.3 Plane Refinement and PPR

Having the refined camera motion, the final step of the two-view pipeline is to merge and refine the initial plane hypotheses obtained from the AP step. This can only be done if the camera motion is not a pure rotation. Otherwise, the algorithm stops. For each cluster associated to the camera motion, a plane hypothesis is generated by reconstructing its points and finding the 3D plane that best fits the point cloud by linear least squares. From the set of plane hypotheses $\mathcal{P}_0$, the objective is to find the minimum number of planes that best describes the scene. Similarly to Sect. 2.2, this task can be cast as a labeling problem where the goal is to assign each point $\mathbf{p}$ to a label from the label set $\mathcal{P} = \{\mathcal{P}_0, l_\emptyset\}$. Again, this is solved by minimizing an energy function E defined as in Eq. 3, where the data cost is the STE obtained for the homographies computed using the refined camera motion and the plane hypotheses. In this case, our set of neighbours $(\mathbf{p}, \mathbf{q}) \in \mathcal{N}$ is determined by a Delaunay triangulation of points to account for possible small errors in the initial plane segmentation. The weights $w_{\mathbf{pq}}$ are defined as the inverse distance between points $\mathbf{p}$ and $\mathbf{q}$ because closer points are more likely to belong to the same plane. Figure 3d shows that the three initial plane hypotheses belonging to the ground plane were correctly merged into one plane, allowing its proper estimation. Also, some incorrectly labeled points in the

MSAC stage (Fig. 3b) were now assigned the correct label. Each selected plane is then refined in an optimization scheme that minimizes the STE. Since each plane is refined independently, this procedure can be performed in parallel, providing a significant speed-up. As a final step, a dense pixel labeling can be obtained using a standard MRF formulation [1,8]. Figure 3e shows that the proposed method is able to provide an accurate and visually pleasing dense PPR from only two views.

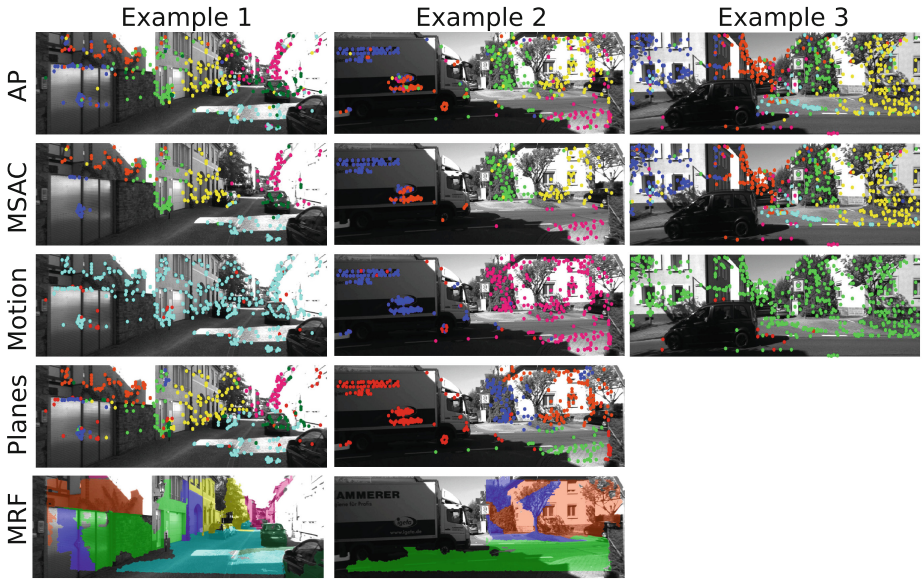
3 Two-View Experimental Results

In this section, we apply the proposed two-view pipeline to 4 different example scenarios (Fig. 4) and show the obtained results after each step. The first three examples were selected from the KITTI dataset [9,10] and illustrate cases of normal motion, dominant dynamic foreground caused by a large vehicle moving, and static camera. The last example is the situation of a moving camera observing multiple planar motions, and was selected from the Hopkins dataset [26].

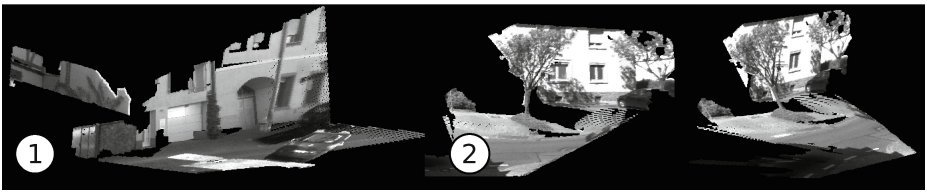
In all experiments, affine covariant features were extracted with the Difference of Gaussians operator using the VLFeat library [27]. We limit the number of extracted ACs to approximately 500 for computational efficiency. We used the publicly available implementations of AP [6] and graph cut optimization [28] for the PEaRL steps. The estimations of the relative rotation R and translation $\mathbf{t}$ up to scale are compared with the ground truth R_{GT} and $\mathbf{t}_{GT}$. The error in rotation (e_{R}) is quantified by the angular magnitude of the residual rotation $\mathsf{R}^{\mathsf{T}}\mathsf{R}_{GT}$ and the error in translation $e_{\mathbf{t}}$ is defined as the angle between vectors $\mathbf{t}$ and $\mathbf{t}_{GT}$. In all experiments, red points correspond to outliers.

Figure 5 shows the outcome of each step of the pipeline for the KITTI image pairs. The first example corresponds to the most common scenario of a moving camera and static scene. AP initially segmented the scene into 7 clusters which were then merged into 6 clusters corresponding to different scene planes. Not only the larger planes corresponding to the ground and building façade were recovered, but also the smaller orange plane was accurately estimated, as shown in the 3D model of Fig. 5b. Moreover, the camera motion was accurately estimated: $e_{\mathsf{R}} = 10e-3^{\circ}, e_{\mathbf{t}} = 1.2^{\circ}$.

Example 2 illustrates the case of dominant dynamic foreground, where AP detects 5 different clusters, 2 of which correspond to the moving vehicle. The PEaRL step described in Sect. 2.2 correctly detects two motions in the image, where the one to which more clusters are associated is selected as the camera motion. After refinement with the inliers (magenta points in the third row) the rotation and translation errors are $e_{\mathsf{R}} = 14e-3^{\circ}$ and $e_{\mathbf{t}} = 0.98^{\circ}$, respectively. 3 planes are then segmented in the image, with the remaining points being labeled as outliers. A final 3D model of the scene is shown in Fig. 5b, where it can be seen that even the faraway plane corresponding to building façade is accurately estimated. VISO2-Mono uses a scheme for detecting if the camera motion is too small, providing the identity matrix as the result for the camera motion in those cases. For this image pair, although the true translation has a norm of



(a) Results for each step of the pipeline. Red points correspond to outliers.



(b) Final 3D model

Fig. 5. (a) Results obtained after each step of the proposed pipeline for the first 3 scenarios in Fig. 4. Since scenario 3 corresponds to a static camera, the planes cannot be estimated and thus the last two steps are not performed. (b) PPR obtained for scenarios 1 and 2. (Color figure online)

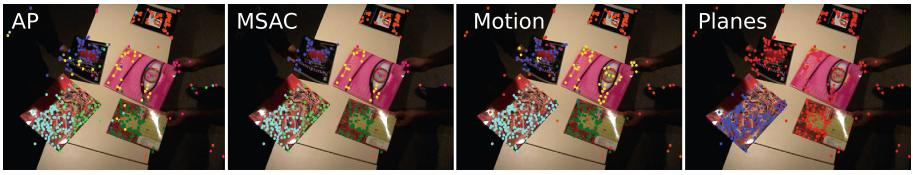


Fig. 6. Results obtained for the scenario of presence of multiple motions in Fig. 4. The dominant motion is selected as the one which has the largest number of associated clusters, which, in this case, does not correspond to the camera motion. This leads to the segmentation of only one plane in the last step of the pipeline, and all remaining correspondences being assigned as outliers (red). (Color figure online)

Table 2. Computational times on a Intel Core i7 3.4 GHz

AC extraction & matching	Metric computation & affinity propagation	Homography MSAC	Motion segmentation	Plane merging	Total
0.21 s	0.20 s	0.08 s	0.07 s	0.09 s	0.65 s

$||\mathbf{t}_{GT}|| = 46.4$ cm, VISO2-Mono identified this case as small motion and did not provide an estimation. By increasing the threshold, we forced VISO2-Mono to estimate the camera motion and observed that it selected many points on the moving vehicle as inliers, providing a poor estimation of the camera motion: $e_{\mathsf{R}} = 1.99°$, $e_{\mathbf{t}} = 77.9°$ and $||\mathbf{t}|| = 11.1$ m.

The third example corresponds to the case of static camera. In fact, there is a residual rotation which allows the scene to be correctly segmented into planes and the camera rotation to be accurately estimated ($e_{\mathsf{R}} = 40e - 4°$). However, since the translation component is negligible, it is not possible to estimate the scene planes. By forcing VISO2-Mono to provide an estimation for the camera motion, poor results were obtained: $e_{\mathsf{R}} = 0.03°$, $e_{\mathbf{t}} = 26.0°$ and $||\mathbf{t}|| = 69.3$ cm.

Figure 6 shows example 4 that consists in a moving camera observing 4 different planar motions. It can be seen that 7 clusters were initially segmented, and 5 different motions were correctly detected by the PEaRL framework described in Sect. 2.2. Since the larger plane was initially segmented into two clusters, its motion is incorrectly identified as the camera motion and only this plane is segmented in the final step. In this case, the rigid structure has little image support, so a more sophisticated scheme for identifying the camera motion is required. A possibility would be to used temporal consistency as proposed in [16].

Table 2 shows the computational times of each step of the proposed pipeline. Except for the C++ implementation of the graph cut optimization [28], the rest of the algorithm is implemented in Matlab. We believe that a C++ implementation of the whole algorithm would allow it to reach a frame rate of 5–10 fps.

4 vSLAM Pipeline

In this section we describe our proposed method πMatch that takes as input a sequence of images and outputs the camera motions and a PPR of the scene. We presented in Sect. 2 a two-view pipeline that takes as input a pair of images and outputs the camera motion, with the translation estimated up to scale, along with the PPR of the scene. In order to be able to work with image sequences, the relative scale of translation between motions must be estimated.

For every two consecutive motions $(\mathsf{R}_i, \mathbf{t}_i)$ and $(\mathsf{R}_{i+1}, \mathbf{t}_{i+1})$, where $(\mathsf{R}_i, \mathbf{t}_i)$ is the motion between frames i and $i+1$, the scale of translation s_i^{i+1} is estimated by fixing the norm of $\mathbf{t}_i$ and computing the new translation vector $s_i^{i+1}\mathbf{t}_{i+1}$. We consider point tracks between frames i and $i+2$ and start by reconstructing the 3D points in frames i and $i+1$ using motions $(\mathsf{R}_i, \mathbf{t}_i)$ and $(\mathsf{R}_{i+1}, \mathbf{t}_{i+1})$, respectively. We consider as inliers the 3D points whose reprojection error lies

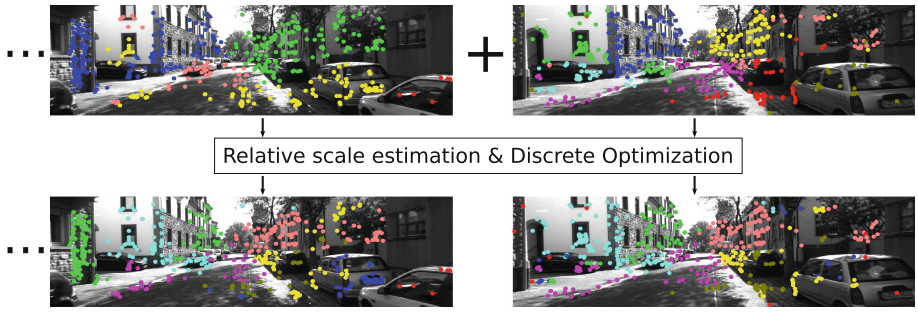


Fig. 7. The two-view pipeline described in Sect. 2 is a applied to each new image pair and its scale is estimated. This allows to select the best planes across multiple views in a PEaRL framework. An important advantage is the backpropagation of planes: the fronto-parallel plane corresponding to the building façade (cyan in the output images) is correctly detected in the incoming image and backpropagated to previous images. Colors identify planes in the output images. Red identifies outlier points. (Color figure online)

below a pre-defined threshold. We observed that the accurate estimation of the rotation and direction of translation allows a good selection of inliers. The two sets of reconstructed 3D points $\mathbf{X}_i$ and $\mathbf{X}_{i+1}$ correspond to the same scene points represented in different reference frames. Thus, using motion $(\mathsf{R}_i, \mathbf{t}_i)$, $\mathbf{X}_i$ can be represented in reference frame $i+1$, and scale s_i^{i+1} is initialized by taking the median of the element-wise ratio $\frac{\mathbf{X}'_i}{\mathbf{X}_{i+1}}$, where $\mathbf{X}'_i = \mathsf{R}_i\mathbf{X}_i + \mathbf{t}_i$. Scale s_i^{i+1} is then refined by minimizing the maximum reprojection error of the 3D points $\mathbf{X}'_i$ in frames $i+1$ and $i+2$, computed using motion $(\mathsf{R}_{i+1}, s_i^{i+1}\mathbf{t}_{i+1})$:

$$s_i^{i+1*} = \min_{s_i^{i+1}} \sum_k \left(\max(d_k^{i+1}, d_k^{i+2})\right)^2, \tag{4}$$

where d_k^i is the reprojection error of point k in frame i. Due to the good selection of inliers, this procedure provides accurate results. Also, since we only optimize one parameter, the computational time of this refinement step is very low (approximately 18 ms in our experiments). For images in which the camera motion is a pure rotation, the scale is not estimated. When the camera resumes the movement, the scale is determined using the new motion and the previous one which was not a pure rotation. This scheme allows the relative scale information to be kept through the whole sequence.

The last step of the pipeline concerns the refinement of the piecewise planar structure by selecting the best planes across multiple frames. This is an adaptation of the discrete optimization step proposed in [21] for stereo sequences, where the authors propose to refine the camera motion and the PPR in a PEaRL framework by considering multiple stereo pairs simultaneously. In this case, for the sake of computational efficiency and since both the camera motion and the planes have already been refined, we propose to include a final discrete optimization

step in a sliding window approach for improving the overall PPR. As explained in [21], optimizing over multiple frames allows the backpropagation of planes, significantly improving the accuracy and visual perception of the 3D model. Figure 7 depicts this advantage, where it can be seen that the fronto-parallel plane of the building façade is detected in the new image and backpropagated to the previous one, providing a much more realistic 3D model.

We formulate this discrete optimization as a labeling problem where the goal is to minimize an objective function E defined as

$$E(\mathbf{l}) = \underbrace{\sum_i \sum_{\mathbf{p}^i} D_{\mathbf{p}^i}(l_{\mathbf{p}^i})}_{\text{Data term}} + \underbrace{\lambda_{S'} \sum_i \sum_{(\mathbf{p}^i,\mathbf{q}^i)\in\mathcal{N}'} w_{\mathbf{p}^i\mathbf{q}^i}\delta(l_{\mathbf{p}^i} \neq l_{\mathbf{q}^i})}_{\text{Smoothness term}} + \underbrace{\lambda_{L'}|\mathcal{L}_\mathbf{l}|}_{\text{Label term}}, \quad (5)$$

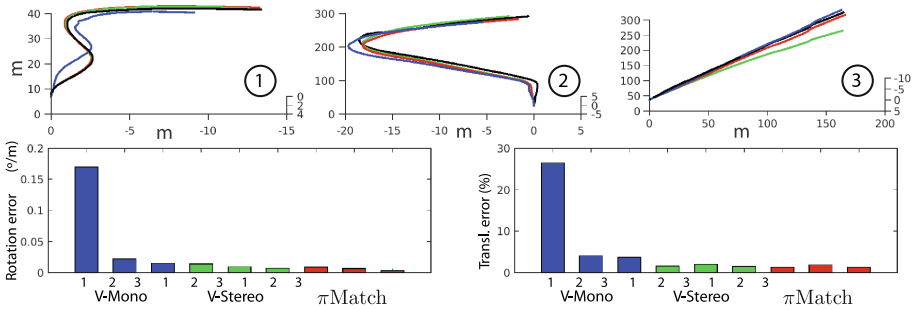
where the label set is the union of the planes detected in each image pair i separately ($\mathcal{L} = \{\bigcup_i \Pi^i, l_\emptyset\}$), the nodes $\mathbf{p}^i$ are the point correspondences in all images i and $\lambda_{S'}$ and $\lambda_{L'}$ are weighting constants. We use the refined motions $\mathsf{R}_i, \mathbf{t}_i, s_i^{i+1}$ to represent the planes in the label set in all reference frames i and compute the STE for defining the data cost $D_{\mathbf{p}^i}$. The neighbourhood $\mathcal{N}'$ is defined by Delaunay triangulation of the points in each image i. We also define as neighbours the points $\mathbf{p}^i$ and $\mathbf{q}^i$ that correspond to the same point track, and set the weight $w_{\mathbf{p}^i\mathbf{q}^i}$ to a large value in this case. This forces points belonging to the same track to be assigned the same label across frames. The remaining weights $w_{\mathbf{p}^i\mathbf{q}^i}$ are the inversely proportional to the distance between $\mathbf{p}^i$ and $\mathbf{q}^i$. In our experiments, for a sliding window of 5 frames (4 camera motions) this optimization took around 50 ms.

5 Large-Scale Experiments

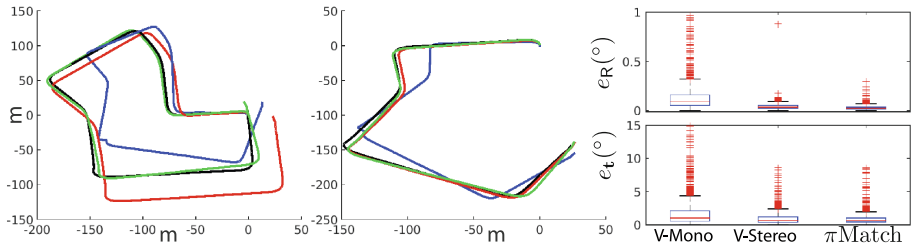
This section reports experiments on 4 sequences of the KITTI dataset [9,10] performed with the monocular method VISO2-Mono [11], the stereo method VISO2-Stereo [11], and our proposed method πMatch. Figure 8 shows the results obtained for the 3 methods, with the errors being quantified using the error metric described in Sect. 3 and the metric proposed in [11].

The first observation is that, when compared to the other monocular method VISO2-Mono, our method is far more superior in the estimation of rotation and translation. Regarding the scale estimation, while VISO2-Mono uses information about the height of the camera, πMatch does not make any prior assumptions and still significantly outperforms this method. Moreover, another important observation is that for the 3 shortest sequences, πMatch also manages to outperform the stereo method VISO2-Stereo, begin particularly more accurate in the estimation of the rotation. This demonstrates the effectiveness of our proposed motion hypotheses generation and selection scheme.

Regarding the 1100-frame sequence, the trajectory makes it evident that VISO2-Stereo outperforms our method. However, from the boxplots showing the individual rotation e_R and translation $e_\mathbf{t}$ errors, it can be seen that πMatch



(a) 1 - 125 frames, 2 - 268 frames, 3 - 395 frames



(b) 1100 frames. V-M: $59e{-}3\ °/m$, 10.8% V-S: $18e{-}3\ °/m$, 2.4% πM: $44e{-}4\ °/m$, 4.4%

Fig. 8. Results obtained on 4 sequences of the KITTI dataset [9,10] using the monocular method VISO2-Mono [11], the stereo method VISO2-Stereo [11], and our proposed monocular method πMatch. The bar plots and caption (b) show the average rotation and translation errors computed using the metric proposed in [10]. The boxplots show the distribution of rotation (e_R) and translation (e_t) errors computed for each image pair.

provides more accurate estimations, leading to the conclusion that the reason for the overall inferior performance of our method is some inaccuracy in the scale estimation. We observed that the estimation of the scale is frequently very accurate, and only fails in few cases. Due to the propagation of error, one poorly estimated scale will influence all subsequent ones, which does not happen in stereo methods. In order to illustrate this fact, we show in Fig. 8b the trajectories for the same sequence after removing the first 300 frames, where it can be seen that the πMatch outperforms VISO2-Stereo. For this sub-trajectory, VISO2-Stereo provided an error of $23e-3°$/m in rotation and 2.55 % in translation while our method was more accurate: $50e-4°$/m in rotation and 1.96 % in translation.

In Fig. 9, the PPR obtained for the 268-frame sequence demonstrates not only the accuracy of our method but also the importance of the last discrete optimization step, where the best planes across multiple frames (5 in this case) are selected. Since we are simply concatenating the individual PPRs for each image pair, the final 3D model would be visually significantly worse if this optimization stage had not been used. This experiment shows that πMatch performs

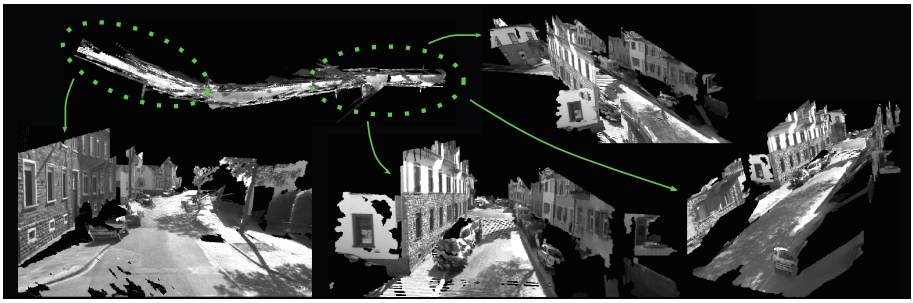


Fig. 9. PPR of the 268-frame sequence in Fig. 8a. A proper alignment of the individual PPRs of each image pair is observed, confirming the good quality of the scale estimation step. Some areas are shown in greater detail.

accurate vSLAM and dense PPR from monocular sequences, significantly outperforming the monocular method VISO2-Mono, and also being superior in the estimation of rotation to the state-of-the-art stereo system VISO2-Stereo.

6 Conclusions

We describe the first feature-based pipeline for vSLAM and dense PPR from a monocular sequence. It works by extracting ACs and employing a recently proposed error metric [22] for detecting scene planes. These planes are used for generating motion hypotheses that allow not only the accurate estimation of the camera motion, but also of other motions present in the image, in a PEaRL framework. The refined camera motion and initial plane hypotheses are used in another PEaRL scheme, yielding good PPRs of the scene from two views. The extension to longer sequences is done by estimating the scale between every two consecutive image pairs, and a final discrete optimization step allows the exchange of planes between frames, providing improved PPR results. The final experiment shows that scale drift may occur due to a few poor estimations of the scale of translation. As future work, we intend to devise a method to overcome this problem by making use of the detected planes. The idea is that since planes are more constant over time than points, using plane correspondences across frames could significantly reduce the scale drift. The total execution time of πMatch mainly implemented in Matlab is approximately 0.72 s. We will implement a C++ version of the pipeline, which we expect to run in about 5–10 fps.

Acknowledgments. Carolina Raposo acknowledges the Portuguese Science Foundation (FCT) for funding her PhD under grant SFRH/BD/88446/2012. The authors also thank FCT and COMPETE2020 program for generous funding through project VisArthro with reference PTDC/EEI-AUT/3024/2014.

References

1. Antunes, M., Barreto, J.P., Nunes, U.: Piecewise-planar reconstruction using two views. Image Vis. Comput. **46**, 47–63 (2016). http://www.sciencedirect.com/science/article/pii/S0262885615001390
2. Concha, A., Civera, J.: DPPTAM: dense piecewise planar tracking and mapping from a monocular sequence. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5686–5693, September 2015
3. Davison, A., Reid, I., Molton, N., Stasse, O.: MonoSLAM: real-time single camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)
4. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 834–849. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10605-2_54
5. Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 1449–1456, December 2013
6. Frey, B.J.: Affinity propagation. http://www.psi.toronto.edu/index.php?q=affinity%20propagation
7. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315**, 972–976 (2007). www.psi.toronto.edu/affinitypropagation
8. Gallup, D., Frahm, J.M., Pollefeys, M.: Piecewise planar and non-planar stereo for urban scene reconstruction. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1418–1425, June 2010
9. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the kitti dataset. Int. J. Robot. Res. (IJRR) **32**, 389–395 (2013)
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
11. Geiger, A., Ziegler, J., Stiller, C.: StereoScan: dense 3D reconstruction in real-time. In: Intelligent Vehicles Symposium (IV) (2011)
12. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004). ISBN: 0521540518
13. Huynh, D.Q.: Metrics for 3D rotations: comparison and analysis. J. Math. Imaging Vis. **35**(2), 155–164 (2009). http://dx.doi.org/10.1007/s10851-009-0161-2
14. Isack, H., Boykov, Y.: Energy-based geometric multi-model fitting. Int. J. Comput. Vis. **97**(2), 123–147 (2012). http://dx.doi.org/10.1007/s11263-011-0474-7
15. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proceedings of Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2007), Nara, Japan, November 2007
16. Lourenço, M., Stoyanov, D., Barreto, J.P.: Visual odometry in stereo endoscopy by using PEaRL to handle partial scene deformation. In: Linte, C.A. (ed.) AE-CAI 2014. LNCS, vol. 8678, pp. 33–40. Springer, Heidelberg (2014). http://dx.doi.org/10.1007/978-3-319-10437-9_4
17. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S.: An Invitation to 3-D Vision: From Images to Geometric Models. Springer, New York (2003)
18. Magri, L., Fusiello, A.: T-linkage: a continuous relaxation of J-linkage for multi-model fitting. In: IEEE Conference on Computer Vision and Pattern Recognition (2014), pp. 3954–3961 (2014)

19. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: dense tracking and mapping in real-time. In: Proceedings of the 2011 International Conference on Computer Vision, ICCV 2011, pp. 2320–2327. IEEE Computer Society, Washington, DC (2011). http://dx.doi.org/10.1109/ICCV.2011.6126513
20. Pizzoli, M., Forster, C., Scaramuzza, D.: REMODE: probabilistic, monocular dense reconstruction in real time. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 2609–2616, May 2014
21. Raposo, C., Antunes, M., Barreto, J.P.: Piecewise-planar StereoScan:Structure and motion from plane primitives. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part II. LNCS, vol. 8690, pp. 48–63. Springer, Heidelberg (2014). http://dx.doi.org/10.1007/978-3-319-10605-2_4
22. Raposo, C., Barreto, J.P.: Theory and pratice of structure-from-motion using affine correspondences. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016). http://arthronav.isr.uc.pt/~carolina/files/CVPRsubm.pdf
23. Song, S., Chandraker, M.: Robust scale estimation in real-time monocular SFM for autonomous driving. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1566–1573, June 2014
24. Toldo, R., Fusiello, A.: Robust multiple structures estimation with J-linkage. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 537–547. Springer, Heidelberg (2008). http://dx.doi.org/10.1007/978-3-540-88682-2_41
25. Torr, P.H.S., Zisserman, A.: Mlesac: A new robust estimator with application to estimating image geometry. Comput. Vis. Image Underst. **78**, 138–156 (2000)
26. Tron, R., Vidal, R.: A benchmark for the comparison of 3-D motion segmentation algorithms. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007, CVPR 2007, 1–8 June 2007
27. Vedaldi, A., Fulkerson, B.: VLFeat: an open and portable library of computer vision algorithms (2008). http://www.vlfeat.org/
28. Veksler, O., Delong, A.: Multi-label optimization. http://vision.csd.uwo.ca/code/