Dense Depth Estimation of a Complex Dynamic Scene without Explicit 3D Motion Estimation

Suryansh Kumar[1] Ram Srivatsav Ghorakavi[3] Yuchao Dai[4] Hongdong Li[3], Luc Van Gool[1,2]
[1]ETH Zurich, [2]KU Leuven, [3]Australian National University, [4]Northwestern Polytechnical University
suryansh.kumar@vision.ee.ethz.ch

Abstract

This work proposes an uncommon way to perceive and estimate dense depth maps of a complex dynamic scene from images. Recent geometric methods to solve dense depth map of a complex dynamic scene from images is greatly dependent on the reliable estimates of 3D motion parameters. To estimate and validate the accuracy of these relative motion parameters ***precisely*** *from image feature correspondences, specifically for a dynamic scene, is a challenging task. In this work, we propose an alternative approach that bypasses the 3D motion estimation step and still provides compelling depth results. Given per-pixel optical flow correspondences between two consecutive frames, and the sparse depth prior of the reference frame, we show that, we can effectively recover the dense depth map for the successive frames without solving for 3D motion parameters. Our method assumes a piece-wise planar model of a dynamic scene, which undergoes rigid transformation* ***locally****, and as-rigid-as-possible transformation* ***globally*** *between two successive frames. Under our assumptions, we can avoid the explicit estimation of 3D rotation and translation to estimate scene depth. In essence, our unconventional formulation provides a distinct framework to estimate the dense depth map of a dynamic scene which is incremental and free from 3D relative motion computation. Our proposed method does not perform any object-level motion segmentation or any other high-level prior assumptions about the dynamic scene, as a result, it is applicable to a wide range of scenarios. Experimental results on the benchmark datasets show the competence of our approach for multiple frames.*

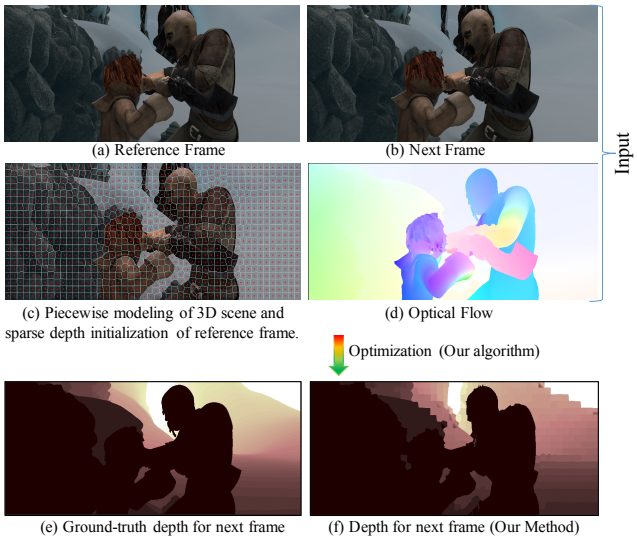


Figure 1. Given consecutive monocular perspective frame (a), (b) of a complex dynamic scene and the dense optical flow correspondences between them (d). Also, assume an approximate sparse depth prior for the reference frame is provided as input (c), then, our algorithm under the piecewise planar approximation of a dynamic scene gives per-pixel depth estimate for the next frame (f) without solving for any motion parameters. (e) ground-truth depth.

1. Introduction

Estimating the dense depth map of a complex dynamic scene from monocular images is an important and well-studied problem in computer vision[35]. Recent developments to solve this problem have gained great attention from several industries involved in augmented reality, autonomous driving, movies, robotics [37, 34, 9, 29, 25, 26] etc. Although recent research to solve this problem has provided some promising theory and results, its success depends on the accurate estimates of 3D motion parameters [23, 32, 38, 10].

To our knowledge, almost all the existing *geometric* solutions to this problem have tried to fit the well-established theory of rigid reconstruction in some way to solve the per-pixel depth of a general *dynamic* scene [30, 23, 32]. Hence, these extensions are intricate to execute and largely depends on per-object or per-superpixel *reliable* motion estimates [30, 23, 32, 1]. The main issue with the available geometric frameworks is that, even if the depth for the first frame or reference frame is known, we must solve for per-

superpixel or per-object 3D motion to obtain the depth for the next frame. Consequently, the composition of their objective function fails to utilize the prior depth knowledge and therefore, does not cascade such prior knowledge well in their framework. In this work, we argue that in a dynamic scene, if the depth for the reference frame is known then it's not obligatory to estimate 3D motion parameters to obtain the depth for the next frame. Hence, the rationale behind relative motion estimation as an essential paradigm for obtaining the depth of a complex dynamic scene seems rather optional under the prior knowledge about the depth of the reference frame and dense optical flow between frames. To endorse our claim, we propose an **alternative approach** which is easy to implement and allow us to get rid of the intricacies related to the optimization on $\mathbb{SE}(3)$ manifold.

We posit that the recent geometric methods to solve this task have been bounded by their inherent dependence on the 3D motion parameters. Consequently, we present a different method to solve the dense depth estimation problem of a dynamic scene. Inspired by the recent work [23], we model the dynamic scene as a set of locally planar surfaces and constrain the change in the scene to be as-rigid-as-possible (ARAP). Recent work by Kumar *et al*. [23] uses local rigidity graph structure to constrain the movement of each local planar structure based on the homography [28] and its inter-frame relative 3D motion. In contrast, we propose that the global ARAP assumption of a dynamic scene may not need explicit 3D motion parameters, and its definition just based on the 3D Euclidean distance metric is a sufficient regularization to supply the depth for the next frame. To this point, one may ask "*Why ARAP assumption for a dynamic scene?*"

Consider a general real-world dynamic scene, the change we observe in the scene between consecutive time frame is not arbitrary, rather it is regular. Hence, if we observe a local transformation closely, it changes rigidly, but the overall transformation that the scene undergoes is non-rigid. Therefore, to assume that the dynamic scene deforms as rigid as possible seems quite convincing and practically works well for most real-world dynamic scenes.

To realize our intuition, we first decompose the dynamic scene as a collection of moving planes. We consider K-nearest neighbors per superpixel [1] (which is an approximation of a surfel in the projective space) to define our ARAP model. For each superpixel, we choose three points *i.e*., an anchor point (center of the plane), and two other non-collinear points. Since the depth for the reference frame is assumed to be known (for at least 3 non-collinear points per superpixel), we can estimate per plane normal for the reference frame, but to estimate per plane normal for the next frame, we need depth for at least 3 non-collinear points per plane (see Figure 2) §3. If per-pixel depth for the reference frame is known, then the ARAP model can be extended to pixel level without any loss of generality. The only reason for such discrete planar approximation is the computational complexity.

Our ARAP model defined over planes does not take into account the depth continuity along the boundaries of the planes. We address it in the subsequent step by solving a depth continuity constraint optimization problem using the TRW-S algorithm [17] (see Fig. 1 for a sample result). In this work, we make the following contributions:

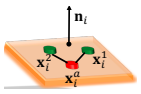


Figure 2. Notation of points on a plane. Three non-collinear points per plane.

- We propose an approach to estimate the dense depth map of a complex dynamic scene that circumvents explicit parameterization of the inter-frame 3D motion. We specify as rigid as possible constraint for the depth estimation by expressing length consistency constraint directly on locally neighboring 3D points.
- Our algorithm under piece-wise planar and as rigid as possible assumption appropriately encapsulates the behavior of a dynamic scene to estimate per-pixel depth.
- Although our algorithm takes two consecutive frames into account, its incremental in nature and therefore, it generalizes to multiple frames without any 3D motion parameters estimation. Experimental results for *two consecutive frames* and *multiple frames* show the validity of our claim §4.

2. Related Work and Our Motivation

Based on our findings, Li.H [27] introduced the first practical method to directly estimate the 3D structure of a scene without explicitly estimating motion. However, this approach solves 3D structure of a **rigid** scene and the formulation can handle few **sparse** points. Very recently, Ji *et al*. [15] extended the Li.H [27] "motion-free" framework to solve **sparse** 3D structure of a **single** non-rigidly moving object using multiple frames (M view, N point) [27]. In contrast, we propose a 3D motion free formulation that provides a **dense** depth map of the **entire dynamic scene** over frames by relying on global as-rigid-as-possible assumption. Recently, numerous papers have been published for the dense depth estimation of a dynamic scene from images. However, for brevity, in this paper, we limit our discussion to the recent papers that are motivated *geometrically* to solve this problem, leading to the easy discourse of our contributions.

In the recent past, two major class of work has been proposed for estimating dense depth of a dynamic scene from two consecutive monocular images [30, 23, 32]. However, all of these methods depend on explicit 3D motion estimation. These methods can broadly be classified as:

(a) Object-level motion segmentation approach: Ranftl *et al*. [32] proposed a three-staged approach to solve dense monocular depth estimation of a dynamic scene. Given the dense optical flow field, the method first performs an object-level motion segmentation using epipolar geometry [12]. Per-object motion segmentation is then used to perform object-level 3D reconstruction using triangulation [12]. To obtain a scene consistent depth map, ordering constraint and smoothness constraints were employed over Quick-shift superpixel [36] graph to deliver the final result.

(b) Object-level motion segmentation free approach: Kumar *et al*. [23] argued that "in a general dynamic scene-setting, the task of densely segmenting rigidly moving object or parts are not trivial". They proposed an over-parametrized algorithm to solve this task without using object-specific motion segmentation. The method dubbed as "Superpixel Soup" showed that under two mild assumptions about the dynamic scene *i.e*., (a) the deformation of the scene is locally rigid and globally as rigid as possible and (b) the scene can be approximated by piece-wise planar model, a scale consistent 3D reconstruction of a dynamic scene can be obtained for both the frames with higher accuracy. Inspired by locally rigid assumption, recently, Noraky *et al*. [30] proposed a method that uses the optical flow and depth priors to estimate pose and 3D reconstruction of a deformable object.

Challenges with such geometric approaches: Although these methods does provide a plausible direction to solve this problem, its usage to real-world applications is rather limited [32, 23, 30]. The major challenge with these approaches is to **correctly** estimate all conceivable 3D motion parameters from image correspondences. The method proposed by Ranftl *et al*. [32] estimates per-object relative rigid motion which is not a sensible choice if the object themselves are deforming. On the other hand, methods such as [23, 30, 24] estimates per superpixel/region relative rigid motion which is sensitive to the size of the superpixels and distance of the surfel from the camera.

The point we are trying to make is, given the depth for the reference frame of a dynamic scene, *can we correctly estimate the depth for the next frame using the aforementioned approaches?*. Maybe yes, but then, we have to again estimate relative rigid motion for each object or superpixel and so on and so forth. Inspired by the "as-rigid-as-possible" (ARAP) intuition [23], in this work, we show that if we know the depth for the reference frame and dense optical flow correspondences between the consecutive frames, then estimating relative 3D motion can be avoided. We can successfully estimate the depth for the next frame by exploiting as-rigid-as-possible global constraints. These depth estimates obtained using ARAP can further be refined using boundary depth continuity constraint.

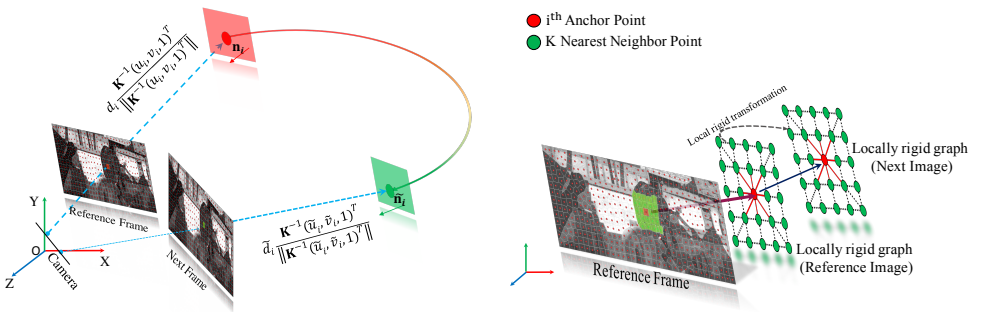
The next concern could be *why we are trying to abort the 3D motion data to solve this problem?* Firstly, as alluded to above, such formulation can help avoid involved optimization on $\mathbb{SE}(3)$ manifold. Secondly, it simplifies the underlying objective function which is relatively neat and easy to solve. Thirdly, it provides a distinct view to think about the behavior of a dynamic scene which generally pivots around the confusion of structure motion and camera motion and its relative inference from image data. Lastly, it provides the flexibility to solve for depth at a pixel level rather than at an object level or superpixel level which is hard to realize using rotation and translation based approaches [30, 23, 32]. Nevertheless, to reduce the overall computation, we stick to optimize our objective function at the superpixel level.

3. Piecewise Planar Scene Model

Inspired by the recent work on dense depth estimation of a general dynamic scene [23], our model parameterizes the scene as a collection of a piece-wise planar surface, where each local plane is assumed to be moving over frames. The global deformation of the entire scene is assumed to be as rigid as possible. Moreover, we assign the center of each plane (anchor point) to act as a representative for the entire points within that plane (see Fig.3). In addition to the anchor point of each plane, we take two more points from the same plane in such a way that these three points are non-collinear (see Fig.4). This strategy is used to define our as rigid as possible constraint between the reference frame and next frame without using any 3D motion parameters. As the depth for the reference frame and the optical flow between the two successive frames is assumed to be known a priori, each local planar region is described using only four parameters —normal and depth, instead of nine [23].

Our model first assigns each pixel of the reference frame to a superpixel using SLIC algorithm [1] and each of these superpixels then acts as a representative for its 3D plane geometry. Since the global change of the dynamic scene is assumed to be ARAP, the transformation that each plane undergoes from the first frame to the next frame should be as minimum as possible. The solution to global ARAP constraint supply depth for three points per plane in the next frame, which is used to estimate the normal and depth of the plane. The estimated depth and normal of each plane are then used to calculate per-pixel depth in the next frame.

Although our algorithm is described for the classical two-frame case, it is easy to extend to the multi-frame case. The energy function we define below is solved in two steps: First, we solve for the depth of each superpixel in the next frame using as rigid as possible constraint. Due to the piecewise planar approximation of the scene, the overall solution to the depth introduces discontinuity along the boundaries. To remove the blocky artifacts —due to the discretization of the scene, we smooth the obtained depth along the boundaries of all the estimated 3D plane in the second step us-



(a) (b)

Figure 3. (a) Piece-wise planar approximation of a dynamic scene. Each superpixel is assumed to be an approximation of a 3D plane in the projective space. The center of the plane is shown with a filled circle (anchor point). (b) Decomposition of the scene into a local graph structure. Locally rigid graph model with its k-nearest neighbor is shown for the reference frame and the next frame.

ing TRWS [17]. If the ARAP cost function is extended to pixel-level then the boundary continuity constraint can be avoided [13]. Nevertheless, over-segmentation of the scene provides a good enough approximation of a dynamic scene and is computationally easy to handle.

3.1. Model Overview

Notation: We refer two consecutive perspective image $\mathbf{I}, \mathbf{I}'$ as the reference frame and next frame respectively. Vectors are represented by bold lowercase letters, for *e.g.* '$\mathbf{x}$' and the matrices are represented by bold uppercase letters, for *e.g.* '$\mathbf{X}$'. The 1-norm, 2-norm of a vector is denoted as $|.|_1$ and $\|.\|_2$ respectively.

3.2. As-Rigid-As-Possible (ARAP)

The idea of ARAP constraint is well known in practice and has been widely used for shape modeling and shape manipulation [14]. Recently Kumar *et al.* [23] exploited this idea to estimate scale consistent dense 3D structure of a dynamic scene. Our idea to use ARAP constraint in this work is inspired by [23, 24].

Let $(d_i,\ d_j)$ and $(\tilde{d}_i,\ \tilde{d}_j)$ be the distance of two neighboring 3D points i, j from the reference coordinate in the consecutive frames. Let $(u_i, v_i, 1)^T$, $(u_j, v_j, 1)^T$ be its 2D image coordinate in the reference frame and $(\tilde{u}_i, \tilde{v}_i, 1)^T$, $(\tilde{u}_j, \tilde{v}_j, 1)^T$ be its image coordinate in the next frame respectively. If '$\mathbf{K}$' denotes the intrinsic camera calibration matrix then, $e_i = \mathbf{K}^{-1}(u_i, v_i, 1)^T / \|\mathbf{K}^{-1}(u_i, v_i, 1)^T\|_2$, $e_j = \mathbf{K}^{-1}(u_j, v_j, 1)^T / \|\mathbf{K}^{-1}(u_j, v_j, 1)^T\|_2$ is the unit vector in the direction of the i^{th}, j^{th} 3D point respectively for the reference frame. Similarly, the corresponding unit vectors in the next frame is denoted with $\tilde{e_i}, \tilde{e_j}$ (see Fig. 3(a)). Using these notations, we define the ARAP constraint as:

$$\Phi^{\text{arap}} = \sum_{i=1}^{3N} \sum_{j \in \mathcal{N}_i^k} w_{ij}^{(1)} \Big| \underbrace{\|d_i e_i - d_j e_j\|_2}_{\text{reference frame}} - \underbrace{\|\tilde{d}_i \tilde{e_i} - \tilde{d}_j \tilde{e_j}\|_2}_{\text{next frame}} \Big|_1 \tag{1}$$

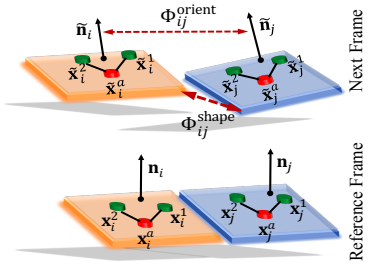


Figure 4. Intuition on orientation and shape regularization. Anchor point and two non-collinear points are shown in red and green respectively. Dark red line show the change in the next frame.

Here, N is the total number of planes used to approximate the scene and $\mathcal{N}_i^k$ is the 'k' neighboring planes local to i^{th} superpixel (see Fig. 3(b)). $w_{ij}^{(1)}$ is the exponential weight fall off based on the image distance of the points. $w_{ij}^{(1)}$ parameter slowly breaks the rigidity constraint if the points are far apart in the image space. This constraint encapsulates our idea that the change in the distance of i^{th} point relative to its local neighbors in the next frame should be as minimum as possible. Note that the summation goes over $3N$ rather than N due the reason discussed in Sec. §1

3.3. Orientation and Shape Regularization

Solving the ARAP constraint gives us the distance for three non-collinear points per-plane in the next frame. We use these distances and corresponding unit vectors to solve for per plane normals in the next frame. Let these three 3D points corresponding to the i^{th} superpixel in the next frame be denoted as $\tilde{\mathbf{x}}_i^a = \tilde{d}_i^a \tilde{e}_i^a$, $\tilde{\mathbf{x}}_i^1 = \tilde{d}_i^1 \tilde{e}_i^1$ and $\tilde{\mathbf{x}}_i^2 = \tilde{d}_i^2 \tilde{e}_i^2$ respectively. We estimate the normals in the next frame as:

$$\tilde{\mathbf{n}}_i = \frac{(\tilde{d}_i^a \tilde{e}_i^a - \tilde{d}_i^1 \tilde{e}_i^1) \times (\tilde{d}_i^a \tilde{e}_i^a - \tilde{d}_i^2 \tilde{e}_i^2)}{\|(\tilde{d}_i^a \tilde{e}_i^a - \tilde{d}_i^1 \tilde{e}_i^1) \times (\tilde{d}_i^a \tilde{e}_i^a - \tilde{d}_i^2 \tilde{e}_i^2)\|_2}. \tag{2}$$

where superscript 'a' is used intentionally to denote the anchor point, which is assumed to be at the center of each

plane (see Fig. 4).

(a) Orientation smoothness constraint: Once we compute the normal for each plane and 3D coordinates of the anchor point (which lies on the plane), we estimate the depth of the plane as follows:

$$\tilde{\mathbf{n}}_i^T \tilde{\mathbf{x}}_i^a + \tilde{d}_i^{\pi_a} = 0 \tag{3}$$

The computed depth of the plane is then used to solve for per-pixel depth in the next frame —assuming that the camera intrinsic matrix is known [23, 24, 12]. To encourage smoothness in the change of angles between each adjacent planes (see Fig. 4), we define orientation regularization as

$$\Phi_{ij}^{\text{orient}} = \lambda_1 \rho_1 \Big(1 - \frac{|\tilde{\mathbf{n}}_i^T \tilde{\mathbf{n}}_j|}{\|\tilde{\mathbf{n}}_i\| \|\tilde{\mathbf{n}}_j\|}\Big), \tag{4}$$

where, λ_1 is an empirical constant and $\rho_1(x) = \min(|x|, \sigma_1)$ is the truncated l_1 function with σ_1 as a scalar parameter.

(b) Shape smoothness constraint: In our representation, the dynamic scene model is approximated by the collection of piecewise planar regions. Hence, the solution to per-pixel depth obtained using Eq. (1) to Eq. (3) may provide discontinuity along the boundaries of the planes in 3D (see Fig. 4). To allow smoothness in the 3D coordinates for each adjacent planes along their region of separation, we define the shape smoothness constraint as

$$\Phi^{\text{shape}} = \sum_{(i,j)\in N_b} w_{ij}^{(2)} \rho_2 (\underbrace{\|d_i e_i - d_j e_j\|_2^2}_{\text{reference frame}} + \underbrace{\|\tilde{d}_i \tilde{e}_i - \tilde{d}_j \tilde{e}_j\|_2^2}_{\text{next frame}}). \tag{5}$$

The symbol 'N_b' denotes the set of boundary pixels of i^{th} superpixel that are shared with the boundary pixel of other superpixels. The weight $w_{ij}^{(2)} = \exp(-\beta \|\mathbf{I}_i - \mathbf{I}_j\|_2)$ takes into account the color consistency of the plane along the boundary points —weak continuity constraint [4]. Since all the pixels within the same plane are assumed to share the same model, smoothness for the pixels within the plane is not essentially required. Similar to orientation regularization, $\rho_2(x) = \min(|x|, \sigma_2)$ is the truncated l_1 penalty function with σ_2 as a scalar parameter. The overall optimization steps of our method is provided in **Algorithm (1)**.

4. Experimental Evaluation

We performed the experimental evaluation of our approach on two benchmark datasets, namely MPI Sintel [5] and KITTI [9]. These two datasets provides complex and realistic environment to test and compare dense depth estimation algorithms. We compared the accuracy of our approach against two recent state-of-the-art methods [23, 32] that use geometric approach to solve dynamic scene dense depth estimation from monocular images. These comparisons are performed using three different dense optical flow estimation algorithms, namely PWC-Net [33], FlowFields [3] and Full Flow [6]. All the depth estimation accuracies are reported using mean relative error (MRE) metric. Let $\tilde{d}$ be the estimated depth and $\tilde{d}^{gt}$ be the ground-truth depth, then MRE is defined as MRE $= \frac{1}{P}\sum_{i=1}^{P} \frac{|\tilde{d}_i - \tilde{d}_i^{gt}|}{\tilde{d}_i^{gt}}$, where '$P$' denotes the total number of points. The statistical results for DMDE [32] and Superpixel Soup [23, 24] are taken from their published work for comparison.

Implementation Details: We over-segment the reference frame into 1000-1200 superpixels using SLIC algorithm[1] to approximate the scene. We use a fixed value of $d_{i\sigma}$ = 1 and $\mathcal{N}_i^k$ = 20-25 for all the experiments. For computing the dense optical flow correspondences between images we used PWC-Net[33], FlowFields[3] and Full Flow[6] algorithm. The depth for the reference image is initialized using Mono-Depth[11] model on the KITTI dataset and using S.Soup algorithm [23] on the MPI-Sintel dataset. The proposed optimization is solved in two stages, firstly Eq.(6) is optimized using SQP[31] algorithm and Eq.(7) is optimized

---

Algorithm 1 : Dense Depth Estimation without using 3D motion

Input: $(\mathbf{I}, \mathbf{I}')$, optical_flow$(\mathbf{I}, \mathbf{I}')$, $\mathbf{K}$, depth for reference frame.
Output: Dense depth map for the next frame.
1: Over-segment the reference frame into N superpixels [1].
2: Assign anchor point for each superpixel and two other points in the same plane such that these three points are non-collinear (see Fig. 4).
3: Use K-NN algorithm over superpixels to get the K-nearest neighbor index set.
4: Solve for per-superpixel depth in the next frame §3.2

$$\Phi^{\texttt{arap}} \rightarrow \underset{\tilde{d}_i}{\texttt{minimize}}$$
$$\texttt{subject to:} \quad \tilde{d}_i > 0, \quad |\tilde{d}_i - d_i| < d_{i\sigma}(\textit{optional}) \tag{6}$$

where, $d_{i\sigma}$ is the variance in the depth.

Note: The second constraint provides a trust region for the fast and proper convergence of a non-convex problem (Fig.10). Can be thought of as max/min restriction to the scene deformation.
5: Estimate the normal for each plane in the next frame Eq. (2).
6: Estimate the depth for each plane Eq. (3).
7: Solve per pixel depth for the next frame using per plane depth ($\tilde{d}_i^{\pi_a}$), $\mathbf{K}$, normal of the plane and its image coordinate. For plane boundaries use 4-nearest neighbor pixel depth average.
8: Refine the depth of the next frame by minimizing Eq. (4), Eq. (5) with respect to depth and normal [17] §3.3.

$$(\Phi^{\texttt{orient}} + \Phi^{\texttt{shape}}) \rightarrow \underset{\tilde{d}_i, \tilde{\mathbf{n}}_i}{\texttt{minimize}}$$
$$\texttt{subject to:} \quad \tilde{d}_i > 0, \|\tilde{\mathbf{n}}_i\| = 1. \tag{7}$$

9: (**For multi-frame**) For generalizing the idea to multi-frame, repeat the above steps by making the next frame as the reference frame and new frame as the next frame.

Figure 5. Depth results on the MPI Sintel dataset[5] for the next frame under two frame experimental setting. $\mathbf{2^{nd}}$ and $\mathbf{3^{rd}}$ row show our depth results and ground-truth depth results respectively.

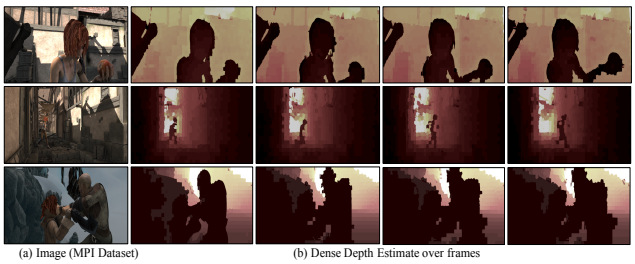


Figure 6. Results on MPI Sintel dataset [5] under multi-frame experimental setting. (a) Image frame for which the depth is initialized. (b) Depth estimation results using our method over frames.

OF↓ / Methods	DMDE[32]	S.Soup[24]	Ours
PWC Net [33]	-	-	0.1848
Flow Fields [3]	0.2970	0.1643	0.1943
Full Flow [6]	-	0.1933	0.2144

Table 1. Comparison of dense depth estimation methods under *two consecutive frame setting* against the state-of-the-art approaches on the **MPI Sintel dataset** [5]. For consistency, the evaluations are performed using mean relative error metric (MRE).

using TRW-S[17] algorithm. The choice of the optimizer is purely empirical and the user may choose other optimization algorithms to solve our cost function. We implemented our algorithm on a commodity desktop computer using C++ /MATLAB which takes 10-12 minutes of processing time.

The implementation is performed under two different experimental settings. In the first setting, given the sparse (*i.e.* for three non-collinear points per superpixel) depth estimate of a dynamic scene for the reference frame, we estimate the per-pixel depth for the next frame. In the second experimental setting, we generalize this idea of two frame depth estimation to multiple frames by computing the depth estimates over frames. For the ease of understanding, MATLAB codes are provided in the appendix to show the working of ARAP idea on synthetic dynamic scene examples.

Results on MPI Sintel Dataset: This dataset gives an ideal setting to evaluate depth estimation algorithms for complex dynamic scenes. It contains image sequences with intricate motions and severe illumination change. Moreover, the large number of non-planar scenes and non-rigid deformations makes it a suitable choice to test the piecewise planar assumption. We selected seven set of scenes namely alley_1, alley_2, ambush_5, bandage_1, bandage_2, market_2 and temple_2 from the clean category of this dataset to test our method.

(a) Two-frame results: While testing our algorithm for the two-frame case, the reference frame depth is initialized using recently proposed superpixel-soup algorithm [23]. The optical flow between the frames is obtained using methods such as PWC-Net [33], Flow Fields [3] and Full Flow [6]. Table (1) shows the statistical performance comparison of our method against other geometric approaches. The statistics clearly show that our alternative way performs equally well without using any 3D rotation or translation. Qualitative results within this setting are shown in Fig. 5.

(b) Multi-frame results: In the multi-frame setting, only the depth for the first frame is initialized. The result obtained for the next frame is then used for the upcoming frames to estimate its dense depth map. Since we are dealing with the dynamic scene, the environment changes slowly and therefore, the error starts to accumulate over frames. Fig. 9(a) reflects this propagation of error over frames. Qualitative results over multiple frames are shown in Fig. 6.

Results on KITTI Dataset: The KITTI dataset has emerged as a standard benchmark dataset to evaluate the performance of dense depth estimation algorithms. It contains images of outdoor driving scenes with different lighting conditions and large camera motion. We tested our algorithm on both KITTI raw data and KITTI 2015 benchmark. For KITTI dataset, we used Monodepth method [11] to initialize the reference frame depth. Dense optical flow correspondences are obtained using the same aforementioned methods. For consistency, the depth estimation error measurement on KITTI dataset follows the same order of 50 meters as presented in [11] work.

(a) Two-frame results: KITTI 2015 scene flow dataset provides two consecutive frame pair of a dynamic scene to test algorithms. Table (2) provides the depth estimation results of our algorithm in comparison to other competing methods. Here, our results are a bit better using PWC-Net [33] optical flow and Monodepth [11] depth initialization. Fig. 7 shows the qualitative results using our approach in comparison to the Monodepth [11] for the next frame.

(b) Multi-frame results: To test the performance of our algorithm on multi-frame KITTI dataset, we used KITTI raw dataset specifically from the city, residential and road category. The depth for only the first frame is initialized using Monodepth [11] and then we estimate the depth for the upcoming frames. Due to very large displacement in the scene per frame (>150) pixels, the rate of change of error accumulation curve for KITTI dataset (Fig. 9(b)) is a bit steeper than MPI Sintel. Fig. 8 and Fig. 9(b) show the qualitative

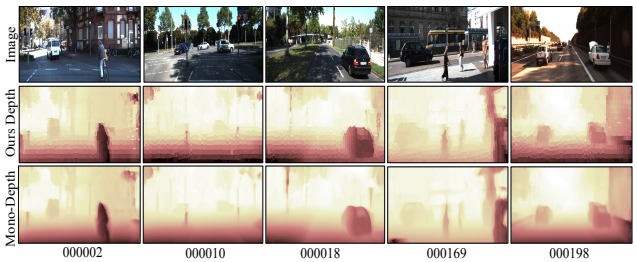


Figure 7. Results on KITTI 2015 benchmark dataset under two frame experimental setting. **3^{rd} row:** Monodepth [11] results on the same sequence for the next frame for qualitative comparison.

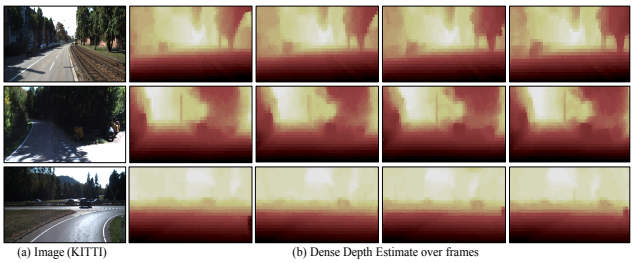


Figure 8. Results on KITTI raw dataset under multi-frame experimental setup. (a) Reference image for which the depth is initialized using Monodepth [11] (b) Dense depth results over frames using our algorithm.

OF↓ / Methods	DMDE[32]	S.Soup[23]	Ours
PWC Net [33]	-	-	0.1182
Flow Fields [3]	0.1480	0.1254	0.1372
Full Flow [6]	-	0.1437	0.1665

Table 2. Comparison of dense depth estimation under *two consecutive frame setting* against the state-of-the-art approaches on **KITTI dataset** [5]. For consistency, the evaluations are performed using mean relative error metric (MRE). The results are better due to monodepth initialization for the reference frame.

Algo.	DT	GLRT	BMM	PTA	DMDE	S.Soup	Ours
Data	(SF)	(MF)	(MF)	(MF)	(TF)	(TF)	(TF)
MPI-S	0.483	0.410	0.312	0.317	0.297	0.164	0.194
KITTI	0.270	0.411	0.390	0.409	0.148	0.125	0.137

Table 3. Statistical comparison of our approach with two-frame and multi-frame approaches on different benchmark dataset. For S.Soup and DMDE [23, 32, 24] we used their previously reported results as their implementations are not publicly available. SF, MF, TF refers to single frame, multi-frame and two-frame based approach respectively. The reference are DT[16], GLRT[8], BMM[7], PTA[2], DMDE [32], S.Soup [23, 24]. For consistency, these comparisons are done using Flow-Fields optical flow [3]

results and depth error accumulation over frames on KITTI raw dataset respectively.

Comparison: We compared the performance of our algorithm against some of the recent dynamic scene 3D reconstruction and depth estimation methods. Our comparisons are done against the methods that are, in general, motivated geometrically to solve this problem. Table (3) shows an overall comparison of our method with recent approaches. We observed that our method works reasonably well under piece-wise rigid and planar model without solving for 3D motion parameters (see Table 3). Note that all the other methods perform object-level or region level motion estimation to report the result. In contrast, our method can handle dynamic scene with independent moving objects over multiple frames without explicitly estimating 3D motion. Although S.Soup [23, 24] performs better than our approach, however, such framework struggles in multiple frame 3D reconstruction. Also, estimating per superpixel motion is challenging for such framework. Hence, our approach has some clear advantage of estimating depth when 3D motion estimates are not correct for a dynamic scene.

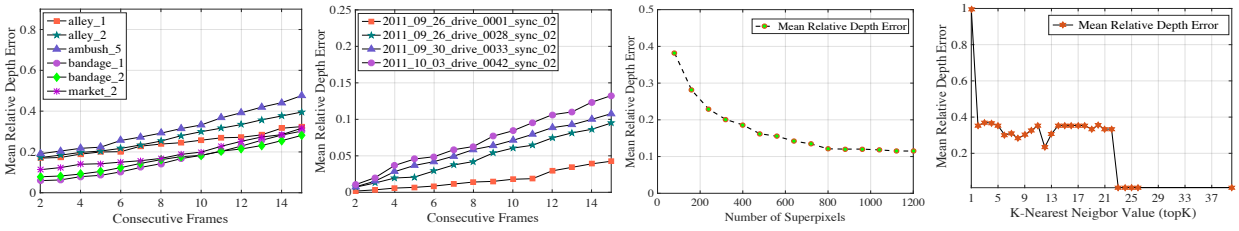
Statistical Analysis: Besides standard experiments under the aforementioned variable initialization, we conducted other experiments to better understand the behavior of the proposed idea. We used synthetic examples shown in Fig. 11 to provide better interpretation of the algorithmic behaviour. Matlab codes with synthetic examples are also provided in the supplementary material for reference.

(a) Effect of the variable N: The number of superpixels to approximate the dynamic scene can affect the performance of our method. A small number of superpixel can provide poor depth result, whereas a very large number of superpixel can increase the computation time. Fig. 9(c) shows the change in the accuracy of depth estimation with respect to the change in the number of superpixels. The curve suggests that for KITTI and MPI Sintel 1000-1200 superpixel provides a reasonable approximation to the dynamic scenes.

(b) Effect of the variable $\mathcal{N}_i^k$: The number of K-nearest neighbors to define the local rigidity graph can have a direct effect on the performance of the algorithm. Although $\mathcal{N}_i^k$=$20 - 25$ works well for the tested benchmarks, it is purely an empirical parameter and can be different for a distinct dynamic scene. Fig. 9(d) demonstrates the performance of the algorithm with the change in the values of $\mathcal{N}_i^k$.

(c) Performance of the algorithm under noisy initialization: This experiment is conducted to study the sensitivity of the method to noisy depth initialization. Fig. 10(a) shows the change in the 3D reconstruction accuracy with the variation in the level of noise from 1% to 9%. We introduced the Gaussian noise using randn() MATLAB function and the results are documented for the example shown in Fig. 11 after repeating the experiment for 10 times and taking its average values. We observe that our algorithm can provide arguable results when the noise level is high.

(d) Performance of the algorithm under restricted isometry constraint with Φ^{arap} objective function: While minimizing the ARAP objective function under the $|\tilde{d}_i - d_i| < d_{i\sigma}$ constraint, we restrict the convergence trust region of the optimization. This constraint makes the algorithm works extremely well —both in timing and accuracy, if an ap-



(a) (b) (c) (d)

Figure 9. (a)-(b) Accumulation of error over frames for MPI and KITTI dataset respectively. (c) Change in the depth estimation accuracy w.r.t number of superpixel. (d) Variation in the depth accuracy as a function of k-nearest neighbor ($\mathcal{N}_i^k$)

(a) (b) (c)

Figure 10. (a) Depth results for the next frame with different levels of Gaussian noise in the reference frame coordinate initialization. (b) Variation in the performance with the change in the $d_{i\sigma}$ values for synthetic example. (c) Convergence curve of the ARAP objective function (light green). Quick convergence with comparable accuracy on the same example can be achieved by using restricted isometric constraint in just 60-70 iteration (dark-green).

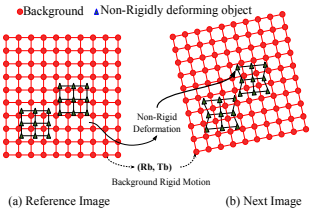


Figure 11. Synthetic example to conduct in-depth analysis of our ARAP cost function. Two objects are deforming independently over a rigid background motion. The objects are at a finite separation from the background. For numerical details on this example kindly go through suppl. material.

proximate knowledge about the deformation that the scene may undergo is known a priori. Fig. 10(b) show the 3D reconstruction accuracy as a function of $d_{i\sigma}$ for the example shown in Fig. 11. Clearly, if we anticipate the scene transformation a priori, we can get high accuracy in less time. See Fig. 10(c) which show the quick convergence by using this constraint under a suitable range of $d_{i\sigma}$. On contrary, if we relax it too much the error can increase (see Fig. 11 for $d_{i\sigma} = 4$). The possible reason for this that, by relaxing too much our rigidity assumption may not hold anymore.

(e) Nature of convergence of our ARAP optimization:
1) Without restricted isometry constraint: As rigid as possible minimization Φ^{arap} under the constraint $\tilde{d}_i > 0$ is alone a good enough constraint to provide acceptable results. However, it may take a considerable number of iterations to do so. Fig. 10(c) shows the convergence curve (light-green).

2) With restricted isometry constraint: Employing an approximate bound on the deformation that the scene may undergo in the next time instance can help fast convergence with similar accuracy. Fig. 10(c) shows that comparable accuracy can be achieved in just 60-70 iterations.[1]

5. Conclusion

The problem of estimating per-pixel depth of a dynamic scene, where the complex motions are prevalent is a challenging task to solve. Quite naturally, previous methods rely on standard relative 3D motion estimation techniques to solve this problem, which in fact is a non-trivial task for a non-rigid scene. In contrast, this paper introduces an alternative way to perceive this problem, which essentially trivializes the notion of 3D motion estimation as a compulsory step. Most of the real-world dynamic scenes if observed closely, it can be inferred that it locally transforms rigidly and globally as rigid as possible. Using such acute observation, we propose an algorithm to solve dense depth estimation task using piece-wise planar and locally rigid approximation of a scene without explicitly solving for 3D motion. Results on the benchmark datasets are provided to validate the competence of our idea. We hope that our idea may open up a new direction for 3D vision research.

Note: For details on limitations, comprehensive discussion and Maltab code, kindly go through the supplementary material.

[1]Note: Per iteration cost without isometry constraint is: 3.6s, whereas, with isometry constrain it is 1.72s, when tested on MPI Sintel dataset.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. In *IEEE transactions on Pattern Analysis and Machine Intelligence*, volume 34, pages 2274–2282. IEEE, 2012.

[2] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1442–1456, 2011.

[3] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *IEEE international Conference on Computer Vision*, pages 4015–4023, 2015.

[4] Andrew Blake and Andrew Zisserman. *Visual reconstruction*. MIT press, 1987.

[5] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, 2012.

[6] Qifeng Chen and Vladlen Koltun. Full flow: Optical flow estimation by global optimization over regular grids. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4706–4714, 2016.

[7] Yuchao Dai, Hongdong Li, and Mingyi He. A simple prior-free method for non-rigid structure-from-motion factorization. *International Journal of Computer Vision*, 107(2):101–122, 2014.

[8] Katerina Fragkiadaki, Marta Salas, Pablo Arbeláez, and Jitendra Malik. Grouping-based low-rank trajectory completion and 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 55–63, 2014.

[9] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The KITTI dataset. In *Int. J. Rob. Res.*, volume 32, pages 1231–1237, Sept. 2013.

[10] Clément Godard, Oisin Mac Aodha, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. *arXiv preprint arXiv:1806.01260*, 2018.

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 7, 2017.

[12] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[13] Michael Hornáček, Frederic Besse, Jan Kautz, Andrew Fitzgibbon, and Carsten Rother. Highly overparameterized optical flow using patchmatch belief propagation. In *European Conference on Computer Vision*, pages 220–234. Springer, 2014.

[14] Takeo Igarashi, Tomer Moscovich, and John F Hughes. As-rigid-as-possible shape manipulation. In *ACM transactions on Graphics*, volume 24, pages 1134–1141. ACM, 2005.

[15] Pan Ji, Hongdong Li, Yuchao Dai, and Ian Reid. "maximizing rigidity" revisited: A convex programming approach for generic 3d shape reconstruction from multiple perspective views. In *ICCV*, pages 929–937. IEEE, 2017.

[16] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *T-PAMI*, 36(11):2144–2158, 2014.

[17] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

[18] Suryansh Kumar. Jumping manifolds: Geometry aware dense non-rigid structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5346–5355, 2019.

[19] Suryansh Kumar. Non-rigid structure from motion: Prior-free factorization method revisited. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 51–60, 2020.

[20] Suryansh Kumar, Anoop Cherian, Yuchao Dai, and Hongdong Li. Scalable dense non-rigid structure-from-motion: A grassmannian perspective. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[21] Suryansh Kumar, Yuchao Dai, and H.Li. Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion. In *Pattern Recognition*, volume 71, pages 428–443. Elsevier, May 2017.

[22] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Multi-body non-rigid structure-from-motion. In *International Conference on 3D Vision (3DV)*, pages 148–156. IEEE, 2016.

[23] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames. In *IEEE International Conference on Computer Vision*, pages 4649–4657, Oct 2017.

[24] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Superpixel soup: Monocular dense 3d reconstruction of a complex dynamic scene. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI), 2019*, 2019.

[25] Suryansh Kumar, Ayush Dewan, and K Madhava Krishna. A bayes filter based adaptive floor segmentation with homography and appearance cues. In *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, page 54. ACM, 2012.

[26] Suryansh Kumar, M Siva Karthik, and K Madhava Krishna. Markov random field based small obstacle discovery over images. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 494–500. IEEE, 2014.

[27] Hongdong Li. Multi-view structure computation without explicitly estimating motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2777–2784. IEEE, 2010.

[28] Ezio Malis and Manuel Vargas. *Deeper understanding of the homography decomposition for vision-based control*. PhD thesis, INRIA, 2007.

[29] Moustafa Meshry, Dan B. Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[30] James Noraky and Vivienne Sze. Depth estimation of non-rigid objects for time-of-flight imaging. In *IEEE Interna-*

tional Conference on Image Processing, pages 2925–2929. IEEE, 2018.

[31] Michael JD Powell. A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical analysis*, pages 144–157. Springer, 1978.

[32] René Ranftl, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun. Dense monocular depth estimation in complex dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4058–4066, 2016.

[33] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[34] Lech Świrski, Christian Richardt, and Neil A. Dodgson. Layered photo pop-up. In *SIGGRAPH Posters*, 2011.

[35] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[36] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *European Conference on Computer Vision*, pages 705–718. Springer, 2008.

[37] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. *arXiv preprint arXiv:1912.08804*, 2019.

[38] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 7, 2017.