Unsupervised Monocular Depth Reconstruction of Non-Rigid Scenes

Conference Paper

Author(s):
Takmaz, Ayça; Paudel, Danda Pani; Probst, Thomas; Chhatkuli, Ajad; Oswald, Martin R.; Van Gool, Luc

Publication date:
2021

Permanent link:
https://doi.org/10.3929/ethz-b-000523758

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
https://doi.org/10.1109/3DV53792.2021.00091

Funding acknowledgement:
820434 - ENergy aware BIM Cloud Platform in a COst-effective Building REnovation Context (EC)
Unsupervised Monocular Depth Reconstruction of Non-Rigid Scenes

Ayça Takmaz\textsuperscript{2}, Danda Pani Paudel\textsuperscript{1}, Thomas Probst\textsuperscript{1}, Ajad Chhatkuli\textsuperscript{1}, Martin R. Oswald\textsuperscript{2,3}, Luc Van Gool\textsuperscript{1,4}

\textsuperscript{1}Computer Vision Lab, ETH Zurich \quad \textsuperscript{2}Department of Computer Science, ETH Zurich
\textsuperscript{3}University of Amsterdam, Netherlands \quad \textsuperscript{4}VISICS, ESAT/PSI, KU Leuven, Belgium

\texttt{takmaza@ethz.ch}, \{\texttt{paudel,probstt,ajad.chhatkuli,vangool}\}@vision.ee.ethz.ch, \texttt{moswald@inf.ethz.ch}

Abstract

Monocular depth reconstruction of complex and dynamic scenes is a highly challenging problem. While for rigid scenes learning-based methods have been offering promising results even in unsupervised cases, there exists little to no literature addressing the same for dynamic and deformable scenes. In this work, we present an unsupervised monocular framework for dense depth estimation of dynamic scenes, which jointly reconstructs rigid and non-rigid parts without explicitly modelling the camera motion. Using dense correspondences, we derive a training objective that aims to opportunistically preserve pairwise distances between reconstructed 3D points. In this process, the dense depth map is learned implicitly using the as-rigid-as-possible hypothesis. Our method provides promising results, demonstrating its capability of reconstructing 3D from challenging videos of non-rigid scenes. Furthermore, the proposed method also provides unsupervised motion segmentation results as an auxiliary output.

1. Introduction

Understanding the 3D structure of a scene can provide important cues for many tasks such as robot navigation \cite{19}, motion capture \cite{42}, scene understanding \cite{33}, and augmented reality \cite{28}. While humans are exceptionally capable of inferring the non-rigid 3D structures from an image, geometric computer vision techniques either require a large amount of labeled data or are capable of learning only from rigid scenes \cite{84}. However, in the real world, scenes often consist of non-rigid and dynamic elements. Thus, it is quite natural to seek for the ability of inferring the depth from an image of any given scene, regardless of whether it is highly dynamic or not. We refer to Fig. 1 for some examples.

The ability of humans to understand their environment geometrically and semantically is mainly acquired through childhood learning. In an attempt to emulate this ability, learning-based strategies have been applied to the problem of depth from single-view. In fact, many learning-based methods already offer very promising progress in this direction. Among them, supervised methods \cite{40, 12, 37} aim to reconstruct the depth of rigid and non-rigid parts during training, whereas many unsupervised methods \cite{23, 84, 83, 25, 13, 45, 73, 43, 44, 26} mainly employ a training strategy which aims to reconstruct the rigid parts of a scene. Although a recent unsupervised method \cite{39} also reconstructs objects translating on a ground plane, it has limitations in terms of modelling highly non-rigid scenes. Other unsupervised methods for non-rigid reconstruction exploit object-specific priors \cite{59, 77, 80}, or reconstruct sparse \cite{49} or dense \cite{64, 66} points of a single non-rigid object. Such methods, however, do not have the same applicability or motivation as that of single view scene depth estimation. Unlike the monocular setup, the calibrated stereo (or multicamera) methods for learning single view depth \cite{24, 48} do not suffer from ambiguities due to non-rigidity and can handle complete scene depth. However, they have significant limitations, particularly when acquiring training data via calibrated stereo pairs is not feasible.

In this work, we are interested in learning to predict dense depth from a single image using an unsupervised monocular pipeline. More importantly, we would like to reconstruct the depth of all parts of the scene during the training, irrespective of the rigidity/non-rigidity of these parts. Our unsupervised setup assumes that only calibrated monocular videos \textsuperscript{1} with known intrinsics are available during training. Such an assumption is realistic as well as crucial for a wide variety of setups, ranging from consumer to surgical cameras, where depth or stereo acquisition for

\textsuperscript{1}Our method can potentially also be used for multi-view setups.
supervision is often impractical. In this context, learning depth from monocular videos of non-rigid scenes remains an unresolved problem. This is no surprise, given the challenges, e.g., ill-posedness, ambiguities, inconsistent priors.

The success of the unsupervised depth learning methods for rigid scenes can be primarily attributed to the advancements in deep learning and the rigid reconstruction constraints used in such methods. This motivates us to explore the non-rigid 3D reconstruction literature employing various assumptions. Our goal is to keep in mind an overview of the literature and to use the gained insights to build our unsupervised monocular pipeline for depth reconstruction in non-rigid scenes. Our main contributions are threefold:

- We reformulate the Non-Rigid Structure-from-Motion (NRSfM) priors in a novel unified framework using the Euclidean distance matrix measures across views. This contribution is summarized in Table 1.
- The utility of the proposed framework is also demonstrated for unsupervised non-rigid monocular depth, by exploiting the as-rigid-as possible (ARAP) prior during CNN training. For the implementation of the ARAP prior, we define and employ a concept of motion embeddings. This contribution is illustrated in Fig. 3.
- Through experiments, we provide interesting new insights towards learning non-rigid scene depths in an unsupervised manner, detailed in our discussion section.

2. Non-Rigid Reconstruction Revisited

Since generic non-rigid 3D reconstruction from a monocular camera is an ill-posed problem, methods in the literature rely on some priors or assumptions about the scene. The most common scene priors can be broadly divided into the following four categories.

**Low-Rank (LR).** The low-rank prior assumes that non-rigid 3D structures can be expressed as a linear combination of finite basis shapes. The landmark work of *Costeira & Kanade* [16] developed for orthographic cameras (or slight variations) uses the LR prior for multi-body 3D scene reconstruction. Since, it has been widely used in various cases for the 3D reconstruction of single [10, 9, 8, 50, 17, 21, 32, 70, 2, 49, 35] and multiple [79, 3] non-rigid objects.

**Scene Motion (SM).** Several notable works of *Shashua et al.* [62, 7, 76, 72] have shown that the known planar/linear motion prior can be exploited to reconstruct the 3D structure of dynamic scenes². Work of *Ozden et al.* [51] also tackled dynamic scene 3D reconstruction using the principle of non-accidentalness. An insightful work of *Hartley and Vidal* [27] reveals that the method of [76] can be indeed extended to the generic non-rigid case, using the low-rank structure and without an explicit motion prior, with only one severe ambiguity, which stems from the fact that the recovered camera motion is relative to the moving points. This in turn implies the unfavorable news that the low-rank prior alone is not sufficient to recover scale-consistent non-rigid 3D structures from monocular projective cameras, when the objects are independently moving in the multi-body setting.

**Isometric Deformation (ID).** To avoid the algebraic prior of LR, *Salzmann et al.* [61] and *Perriollat et al.* [57] introduced the geometric ‘ID’ prior of an object deforming isometrically, which makes non-rigid surface reconstruction possible when used with a known template. Since, the ID prior of the object has been used in many other works [63, 71, 14, 53, 15, 74, 58, 38]. *Taylor et al.* [68] introduced local-rigidity for non-rigid reconstruction, which is a form of the ID prior. Other related geometric priors have also been exploited in the literature [46, 60, 1]. When objects undergo severe deformations, relying on a geometric prior for the objects and avoiding any explicit camera motion estimation have demonstrated state-of-the-art results for non-rigid object reconstruction [55, 52, 58, 22, 47]. The benefit of not estimating the camera motion explicitly can be understood quite intuitively, as the estimated rigid camera motion is merely a motion with respect to the scene. Interested readers may recall the relevant work in [27]. In this regard, the works of *Li* [38], *Ji et al.* [31] and *Chhatkuli et al.* [15] stand out. While reconstructing inextensible structures, [15] formulates an explicit-motion-free problem, which turns out to be equivalent to the rigid case formulation of [38] in the absence of deformations. In fact, the formulation of [15] is shown to be effective for both rigid and non-rigid scenes. All three works [15, 31, 38] are based on the pairwise distance equality in 3D, while [38] uses the same pairwise sampling as [31].

**As-Rigid-As-Possible (ARAP).** Non-rigid shape modelling using the ARAP assumption is very prevalent in computer graphics [6, 29, 65]. The ARAP assumption maximizes rigidity while penalising stretching, shearing, and compression. ARAP concept has also been used in many works [54, 36] for monocular non-rigid 3D reconstruction. In particular, two works of *Parashar et al.* [54] and *Kumar et al.* [36] are noteworthy. Using the shape template of an object, [54] reconstructs the deformed volumetric 3D under the ARAP assumption. On the other hand, [36] demonstrates the practicality of ARAP in depth densification and refinement for non-rigid scenes, using multi-view setups. More interestingly to us, the ARAP prior is sufficient to resolve the scale ambiguity between freely moving parts, i.e. the issue previously discussed by *Hartley & Vidal* in [27].

In summary, the computation of camera motion becomes ambiguous in complex non-rigid scenes for a monocular camera. Nevertheless, the structure of both rigid scenes and non-rigid objects³ can still be recovered without any explicit

---

²The recent work [39] can be seen as an adaptation of [7].

³Given that the assumed object’s prior is sufficient.
Let us define a weight matrix \( W \in \mathbb{R}^{n \times n} \), whose entries are \( w_{ij} = 1 \) if \( j \in \mathcal{N}_r(i) \) for some radius \( r \), and \( w_{ij} = 0 \) otherwise. Then the relaxed isometric prior of \( [61, 15, 58] \) aims to reconstruct the depth by solving,

\[
\text{find } \{ \Lambda^k \}, \\
\text{s.t. } W \odot (E(\Lambda^k) - E(\Lambda^l)) = 0, \forall k, l, \\
\Lambda^k > 0, \forall k, 
\]

where \( \odot \) represents the Hadamard-product. Intuitively, (2) aims to preserve the local Euclidean distances across views. The constraints \( \Lambda^k > 0 \) ensure positive depths in all views. It is important to note that for sufficiently large radii \( r \) the problem of (2) is equivalent to that of the Example 3.2.

### 4. This Work

One may use the formulation in Eq. (2) to learn to reconstruct both rigid scenes or non-rigid objects, for known \( W \). We however, are interested in reconstructing complex scenes consisting of both, which requires additional priors (recall Section 2). Moreover, it is unclear how to obtain the weight matrix \( W \) in general cases. We use the ARAP assumption to address the scale ambiguity problem. In this process, the weight matrix is also learned along with the depth. If one aims to exploit other non-rigid priors, later in Section 4.2 we provide a thorough analysis in that direction.

### 4.1. Problem Formulation

We aim to estimate the depth \( \Lambda \) for each view, and the weight matrix \( W \) for given view-pairs. Our weights \( w_{ij} \in [0, 1] \) can be interpreted as rigidity scores, between points \( X_i \) and \( X_j \). Since, the concept of rigidity is meaningful only for two (or more) views, our rigidity scores are computed accordingly. We consider two points to be rigidly connected, if their distance does not change across views. If one seeks for pair-wise rigidity on generic graphs, several interpretations can be derived based on the graph connectivity [75, 20, 38]. In the context of this paper, we formulate the ARAP assumption as follows:

\[
\text{find } \{ \Lambda^k \}, \\
\text{s.t. } W \odot (E(\Lambda^k) - E(\Lambda^l)) = 0, \forall k, l, \\
\Lambda^k > 0, \forall k, 
\]

Notations. We denote matrices with uppercase and their elements with double-indexed lowercase letters: \( A = (a_{ij}) \). Similarly, we write vectors and index them as: \( a = (a_i) \). The inequality \( A > 0 \) refers to \( (a_{ij}) > 0 \), unless mentioned otherwise. We use special uppercase Latin or Greek letters for sets and graphs, such as \( \mathcal{S} \) and \( \mathcal{G} \). The lowercase Latin letters, as in \( a \), are used for scalars. The set of neighbors of \( i \) within a radius \( r \) is given by \( \mathcal{N}_r(i) \).

### Non-rigid Structure-from-Motion (NRSfM).

Our problem formulation is camera motion independent, similar to [38, 61, 15]. We pose the NRSfM problem as finding point-wise depth in each view. We use superscript \( k \) to denote the \( k \)-th image (among \( n \)) and subscript \( i \) to denote the \( i \)-th point (among \( n \)). As illustrated in Fig. 2, we represent the unknown depth as \( \lambda^k_i \) and the known homogeneous image coordinates as \( u^k_i \). The set of points in the \( k \)-th view is given as \( X^k = \{ X^k_i \}_{i = 1}^{n} \). The Euclidean distance between point \( X^k_i \) and point \( X^k_j \) is denoted as \( d^k_{ij} \). In our formulation, we use the Euclidean distance matrix:

**Definition 3.1 (Euclidean distance matrix)** Euclidean Distance Matrix (EDM), say \( E \in \mathbb{R}^{n \times n} \), is a matrix representing the spacing of \( n \) points in 3-dimensional Euclidean space, say \( X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{3 \times n} \), and the entries of \( E \) are given by, \( e_{ij} = d^k_{ij} = \| x_i - x_j \|^2 \).

Let \( \Lambda^k = [\lambda^k_1, \lambda^k_2, \ldots, \lambda^k_n] \in \mathbb{R}^n \), be the sought depth of the \( k \)-th view. We represent the 3D structure in the form, \( X^k(\Lambda^k) = [X^k_1, \lambda^k_1 u^k_1, \lambda^k_2 u^k_2, \ldots, \lambda^k_n u^k_n] \in \mathbb{R}^{3 \times n} \). Let us define the Gram matrix \( G(\Lambda^k) = (X^k(\Lambda^k))^T X^k(\Lambda^k) \in \mathbb{R}^{n \times n} \). The EDM for the \( k \)-th view is then given by:

\[
E(\Lambda^k) = \text{diag}(G(\Lambda^k))1^T - 2G(\Lambda^k) + \text{diag}(G(\Lambda^k))1^T. 
\]

NRSfM aims to estimate \( \{ \Lambda^k \} \) by imposing priors on \( E(\Lambda^k) \). For clarity, we present an example of the rigid case.

**Example 3.2 (The rigid structure prior)** For noise- and outlier-free rigid scenes, the estimated \( \{ \Lambda^k, \Lambda^l \} \) must satisfy, \( E(\Lambda^k) - E(\Lambda^l) = 0 \), for all \( k, l = 1, \ldots, n \). Intuitively, the distance between any pair of points must be preserved across views. Method of [38] is a variant of this example.

The prior of the above example is also referred as the global rigidity constraint [75, 20]. Weaker than the global rigidity is local rigidity. Interested readers can find a thorough study of local/global rigidity in [38]. Here, we are interested in the non-rigid isometric deformation prior [57, 68, 61, 15].

**4.1. Problem Formulation**

We aim to estimate the depth \( \Lambda \) for each view, and the weight matrix \( W \) for given view-pairs. Our weights \( w_{ij} \in [0, 1] \) can be interpreted as rigidity scores, between points \( X_i \) and \( X_j \). Since, the concept of rigidity is meaningful only for two (or more) views, our rigidity scores are computed accordingly. We consider two points to be rigidly connected, if their distance does not change across views. If one seeks for pair-wise rigidity on generic graphs, several interpretations can be derived based on the graph connectivity [75, 20, 38]. In the context of this paper, we formulate the ARAP assumption as follows:

\[
\text{find } \{ \Lambda^k \}, \\
\text{s.t. } W \odot (E(\Lambda^k) - E(\Lambda^l)) = 0, \forall k, l, \\
\Lambda^k > 0, \forall k, 
\]

where \( \odot \) represents the Hadamard-product. Intuitively, (2) aims to preserve the local Euclidean distances across views. The constraints \( \Lambda^k > 0 \) ensure positive depths in all views. It is important to note that for sufficiently large radii \( r \) the problem of (2) is equivalent to that of the Example 3.2.

**References**

[38] 61, 15, 58].

**Figure 2:** **Pairwise distance formulation.** The pair of 3D Euclidean distances in \( k \)-th and \( l \)-th views are given by \( d^k_{ij} = \| x_i^k - x_j^k \| \) and \( d^l_{ij} = \| x_i^l - x_j^l \| \), respectively. Every 3D point, say \( x_i^k = \lambda^k_i u_i^k \), is expressed using the unknown depth and known homogeneous image coordinates, i.e. \( \lambda^k_i \) and \( u_i^k \).

Camera motion estimation. On the other hand, the scale consistent reconstruction intrinsically requires some additional prior, such as the ARAP assumption used in this work.

**3. Preliminaries**

**Notations.** We denote matrices with uppercase and their elements with double-indexed lowercase letters: \( A = (a_{ij}) \). Similarly, we write vectors and index them as: \( a = (a_i) \). The inequality \( A > 0 \) refers to \( (a_{ij}) > 0 \), unless mentioned otherwise. We use special uppercase Latin or Greek letters for sets and graphs, such as \( \mathcal{S} \) and \( \mathcal{G} \). The lowercase Latin letters, as in \( a \), are used for scalars. The set of neighbors of \( i \) within a radius \( r \) is given by \( \mathcal{N}_r(i) \).
Definition 4.1 (As-rigid-as-possible) For a point set under deformation, the as-rigid-as-possible model assumes that every pair-wise distance of the fully connected graph between points, respects at least some degree of rigidity.

We can now formalize our problem statement as follows.

\[
\min \{ \lambda^k \}, \{ W^{kl} \} \eta, \quad s.t. \quad | W^{kl} \odot (E(\Lambda^k) - E(\Lambda^l)) | \leq \eta \| W^{kl} \|_{1,1}, \quad (3)
\]

\[ 1 \geq W^{kl} \geq \tau, \lambda^k > 0, \forall k. \]

Here, the positive scalars \( \tau \) and \( \eta \) are the rigidity threshold and the rigidity adjusted maximum allowed distance error, respectively. Very often the point pairs from non-rigid objects respect rigidity. This is when many priors are best justified. In general, local rigidity does not imply global rigidity. For large fully connected graphs though, pair-wise rigidity means global rigidity. We relax this constraint by allowing different edges to have different rigidity scores. We like to draw the reader’s attention on two key (and somewhat related) aspects of our formulation: (1) image pair-wise rigidity scores; and (2) global connectivity.

Our motivation for using individual \( W^{kl} \) stems from the following observation: across all frame pairs, most of the pairs from non-rigid objects stop, or a rigid surgical tool poking an isometric organ surface. A rigid car driving on the rigid road with linear constant velocity maintain highest rigidity with respect to parallel lines/planes\(^4\) along the velocity direction. In fact, the exact rigidity scores to the other points can be measured (or bounded) in some or all image pairs, as is known. This is particularly interesting, if the motion prior and image semantics are known, e.g. a rigid car driving on the rigid road with stops, or a rigid surgical tool poking an isometric organ surface. We represent a set of edges whose rigidity scores can be measures (or bounded) in some or all image pairs, as \( S^{kl} = \{ s_{ij}^{kl} | e \in \{ e_{ij} \} \}. \) These scores then can also be used (if necessary with bounds) in our formulation. Needless to say, the SM prior can be used in conjunction with LR in practice \([76, 27]\). An overview of commonly used priors as discussed in Section 2 in relation to previous works is given in Table 1, and presented in a unified framework. Our method can be used for all priors unified in Table 1. However, different priors are useful depending on the scene.

**Proof** The proof is provided in the suppl. material.

### 4.2. Relation to Previous Works

Heretofore, we have discussed the relationship of our formulation in Eq. (3) to rigidity, ID, and ARAP priors. Now, we further explore the same with respect to LR and SM, for the completeness of our theoretical framework. To do so, let us denote \( D^{kl} = E(\Lambda^k) - E(\Lambda^l) \) and its stacked version \( \mathbf{D} = [(\mathbf{D}^{12})^T, \ldots, (\mathbf{D}^{kl})^T, \ldots, (\mathbf{D}^{(m-1)m})^T]^T \). Our investigation leads to the following relationships between the formulation used in this paper and the prior of the low-rank purely on structures, without explicit camera motion.

**Proposition 4.2 (EDM of Low-rank structures)** For 3D structures that can be represented using \( b \) linear basis,
between the views

trix between points

ity

trix W

problem given in (3), we aim to estimate the weight ma-

k and

prior. For that purpose, we predict depths for the views

H

image of size

Revisiting the initial optimization

rigidity scores

W

the dense correspondences for the view pair

spondences between views

In order to use the ARAP objective (3), we establish corre-

f

from

neural network

f

Objective and overview.

Using a deep convolutional

I

timization objective can be extended to multi-view images

for a given image pair

I

views

k

and

l

from image data, given the dense correspondences between

pairs. A pre-trained flow network used for dense correspondences is included for completeness. The training objective is given in Eq. (6).

Having obtained the motion embedding matrix \( M^{kl} \), we derive the entries of \( W^{kl} \) from the pairwise distances between the motion embedding vectors \( m_i^{kl} \) and \( m_j^{kl} \). It should be noted that we expect a higher weight when the distance between the motion embedding vectors are smaller, and vice versa. With this consideration, we formulate the motion similarity scores for every pair (as illustrated in Fig. 4) as,

\[
    w^{kl}_{ij} = 1 - \tanh(||m_i^{kl} - m_j^{kl}||),
\]

(4)

Edge Sampling. We aim to compute the motion similarity scores, \( w^{kl}_{ij} \), ideally for all possible point pair combinations. However, performing this computation is not tractable considering the number of pixels. Therefore, from the graph for the \( k \)-th view, say \( G^k = (V = \mathcal{X}^k, E = \mathcal{X}^k \times \mathcal{X}^k) \), we uniformly sample a random subset of all edges (point pairs) for the loss computation. The sampled edges are denoted as \( S^k \). Using this subset \( S^k \), we compute a weight matrix \( W^k_{S} \), whose entries are given by

\[
    w^{kl}_{S,ij} = \begin{cases} 
    w^{kl}_{ij} & (X^k_i, X^k_j) \in S^k, \\
    0 & \text{otherwise.}
    \end{cases}
\]

(5)

5. Learning Depth Using ARAP

We first use problem (3) to learn depth reconstruction from image data, given the dense correspondences between views \( k \) and \( l \). We will then formalize our learning objective for a given image pair \( I^k \) and \( I^l \) of views \( k \) and \( l \). Our optimization objective can be extended to multi-view images \( I = \{I^1, I^2, \ldots, I^m\} \) of the same scene. An overview of our learning pipeline is shown in Fig. 3.

Objective and overview. Using a deep convolutional neural network \( \phi_{\theta_\phi}(I) : \mathbb{R}^{H \times B \times 3} \rightarrow [0, \mathbb{R}^+]^{H \times B} \) parameterized by \( \theta_\phi \), we wish to estimate the depth for a given RGB image of size \( H \times B \times 3 \), as the output of the network.

In order to train the depth network, we use the ARAP prior. For that purpose, we predict depths for the views \( k \) and \( l \) as: \( \Lambda^k = \phi_{\theta_\phi}(I^k) \) and \( \Lambda^l = \phi_{\theta_\phi}(I^l) \), respectively. In order to use the ARAP objective (3), we establish correspondences between views \( k \) and \( l \) with a pre-trained optical flow network \( \phi_{\theta_f}(I \times I) \), whose weights are frozen. We use the dense correspondences for the view pair \( (k, l) \) obtained from \( \phi_{\theta_f}(I^k, I^l) \) to compute the difference of EDMs given by \( E(\Lambda^k) - E(\Lambda^l) \). Next we describe how we obtain the rigidity scores \( W^{kl} \) for the view pair \( (k, l) \).

Motion embeddings. Revisiting the initial optimization problem given in (3), we aim to estimate the weight matrix \( W^{kl} \), whose entries \( w^{kl}_{ij} \in [0, 1] \) represent the rigidity between points \( (X^k_i, X^l_j) \), across the transformation between the views \( (k \leftrightarrow l) \). To derive this weight matrix \( W^{kl} \), we propose to learn per-pixel motion embed-

Fig. 3: Unsupervised learning of depth reconstruction and motion embeddings. Our model consists of a depth and a per-pixel motion embedding networks, which jointly learn depth map, and motion embeddings whose similarity expresses the rigidity between scene point pairs. A pre-trained flow network used for dense correspondences is included for completeness. The training objective is given in Eq. (6). After training, only a single-pass through the depth network is required for the inference of depth from an image.
we only iterate through the indices with non-zero entries by exploiting the sparsity of (3)). Computation of (6) can be efficiently performed using (4), followed by a $\tau$-offset and clamping of the weight values to $[0, 1]$ in progression (to enforce the constraints of (3)). Computation of (6) can be efficiently performed by exploiting the sparsity of $W_{S}^{kl}$. During the calculation, we only iterate through the indices with non-zero entries of $W_{S}^{kl}$. We additionally incorporated a weight-norm regularization term in our training objective, i.e., $\beta \| W_{S}^{kl} \|_{1,1}$, to control the maximization of the weights. Further implementation details can be found in the supplementary. We summarize our loss computation process in Algorithm 1.

**Network structure.** Our depth network $f_{\theta_d}(I)$ consists of a ResNet-18 based encoder and a decoder. The input to our depth network is a single RGB image, and the output from the network is a single depth map of the same size. Our per-pixel motion embedding network $f_{\theta_m}(I \times I)$ also consists of a multi-input ResNet-18 based encoder and a decoder. The input to the motion embedding network is a pair of images, whose channels are concatenated to create a single input tensor before being passed through the network. The output from the motion embedding network is a motion-embedding map, which, in our case, has 3 channels.

**Loss Formulation.** Using (4) and (5), we reformulate the problem of (3) as a loss function to train our networks, parameterized by $\theta = \{ \theta_d, \theta_m \}$, as follows:

$$L_\theta(A^k, L^i, W_{S}^{kl}) = \frac{\|W_{S}^{kl} \odot (E(A^k) - E(L^i))\|_{1,1}}{\alpha \| W_{S}^{kl} \|_{1,1}}. \quad (6)$$

The normalization factor $\alpha = \|E(A^k) + E(L^i)\|_{1,1}$ is introduced for numerical stability and to avoid reconstructions with near-zero depth values. The constraints of (3) are imposed in the network output: Depths $\Lambda^k$ are ensured to be positive by using a sigmoid on inverse depth output from the network. Similarly, the weights $W_{S}^{kl}$ are bounded by us-

![Figure 4: Motion similarity scores. For a given point pair $(X^k_i, X^k_j)$ from the scene graph $G^k$, we retrieve the associated motion embeddings $m^k_i$ and $m^k_j$, from which we derive motion similarity scores. These scores express how much we expect the 3D distance to be preserved for each point pair.](image)

**Algorithm 1** \[L_\theta^{kl} = \text{computeLossARAP}(l^k, l^i)\]

1: Sample a set of edges $S$ with their vertices $V$.
2: Estimate the motion embedding $M^{kl} = f_{\theta_m}(l^k, l^i)$.
3: Compute $W_{S}^{kl}$ using $M^{kl}$ and (4) for $S$.
4: Establish $(i, j)$ between $(k, l)$ using $F^{kl} = f_{\theta_f}(l^k, l^i)$.
5: Start loop $s = k, l$
6: Estimate the depth $\Lambda^s = f_{\theta_d}(l^s)$
7: Reconstruct 3D $X^s(\Lambda^s) = \sum_{u \in \Lambda^s} u_i^s$ for $V^s$.
8: Compute the EDM $E(\Lambda^s)$ using $X^s(\Lambda^s)$.
9: End loop
10: Compute loss $L_\theta(A^k, L^i, W_{S}^{kl})$ using (6).
11: Return $L_\theta(A^k, L^i, W_{S}^{kl})$.

**Datasets.** In our experiments, we use MPI Sintel [11], VolumeDeform [30], and Hamlyn Laparoscopic Video Dataset [82]. From the MPI Sintel training subset, we select a set of 14 final-pass sequences with varying levels of motions. We also use the VolumeDeform dataset consisting of 8 sequences for further experiments on deforming scenes. Lastly, we evaluate our method on the Hamlyn Centre Laparoscopic Video Dataset, which consists of rectified stereo image pairs collected from a partial nephrectomy. Ground-truth depth maps for VolumeDeform are obtained from the provided depth recordings; and OpenSFM [5, 4] was used to obtain ground-truth depth for Hamlyn using the calibrated stereo pairs in the dataset. For Sintel, both ground-truth depth maps as well as optical flow maps are provided, which is not the case for the other two datasets. A pre-trained supervised optical flow network, RAFT [69], performs reasonably well on VolumeDeform for estimating dense correspondences. On Hamlyn, due to a large domain gap, we fine-tune an unsupervised model, DDFlow [41].

**Training Details.** We follow two different sets of experiments: test-time training or inference. As the small volumes of MPI Sintel and VolumeDeform datasets do not yet enable us to train a depth network that can generalize, we perform test-time training for each image sequence, i.e., we train separate models for each sequence, for our methods and the deep baselines. In this pipeline, for all of the methods, the training is rather used as an optimization procedure for solving for the depth in all frames, given an image sequence. For the Hamlyn dataset where the amount of data enables us to train models that could generalize, we present inference results. We use the original training and test split from the Hamlyn dataset, and further create train/val/test splits. We implemented our model in PyTorch [56]. In all experiments, the ResNet18-based depth encoder and motion-embedding encoder were initialized with weights from ImageNet [18] pre-training. Adam [34] opti-
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Abs Rel</td>
<td>Sq Rel</td>
</tr>
<tr>
<td>Mostly Rigid</td>
<td>0.691</td>
<td>2.879</td>
</tr>
<tr>
<td>Li et al. [39]</td>
<td>0.871</td>
<td>5.267</td>
</tr>
<tr>
<td>Seq.</td>
<td>Ours w/ motion</td>
<td>0.562</td>
</tr>
<tr>
<td>Ours w/o motion</td>
<td>0.549</td>
<td>2.441</td>
</tr>
<tr>
<td>Mostly Non-Rigid</td>
<td>0.774</td>
<td>3.595</td>
</tr>
<tr>
<td>Li et al. [39]</td>
<td>1.217</td>
<td>9.254</td>
</tr>
<tr>
<td>Seq.</td>
<td>Ours w/ motion</td>
<td>0.562</td>
</tr>
<tr>
<td>Ours w/o motion</td>
<td>0.519</td>
<td>1.363</td>
</tr>
<tr>
<td>All Parts</td>
<td>PackNet [28]</td>
<td>0.673</td>
</tr>
<tr>
<td>Li et al. [39]</td>
<td>0.992</td>
<td>6.855</td>
</tr>
<tr>
<td>Seq.</td>
<td>Ours w/ motion</td>
<td>0.578</td>
</tr>
<tr>
<td>Ours w/o motion</td>
<td>0.540</td>
<td>2.301</td>
</tr>
</tbody>
</table>

Table 2: Depth reconstruction evaluation for VolumeDeform [30] and MPI Sintel [11] datasets. Sequences are classified into two categories: mostly rigid or non-rigid. For MPI Sintel, scenes with less than 10% dynamic pixels are labeled as mostly-rigid. VolumeDeform split is based on our qualitative analysis. Our method with motion embeddings results in superior performance in highly non-rigid scenes in both datasets. Best results are in bold, second best are underlined. Both of our methods generally perform better than our baselines, and for the non-rigid sequences, we can see the benefits of using our method with motion embeddings.

**Figure 5:** Our qualitative results with depth and motion networks for MPI Sintel dataset. Top to bottom: RGB, predicted depth, and predicted embeddings. Embedding colors encoding information about motion clusters are not related across sequences.

mizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ was used, in combination with a learning rate decay by 0.1 every 10 epochs. We follow a two-stage training. In the first stage, we jointly train the motion-embedding network and the depth network. In the second stage, we freeze the weights of the motion-embedding network, and re-initialize the depth network training. In the latter stage, we apply a $\tau$-offset to the weights, to impose the constraint from (3). For the selection of the edges, we uniformly sample pairs of points with correspondences in the consecutive frame. We empirically find 100K pairs (edges) to provide a suitable trade-off between increased memory requirements and slow convergence. Further details are in the supplementary material.

**Depth reconstruction results.** Our qualitative results are demonstrated in Fig. 1, 5 and 6. We report our results by performing per-image median ground-truth scaling for each method, as introduced in [84]. For the MPI Sintel dataset, we evaluate the performance only where the GT depth is smaller than 50 m. For the other datasets, we evaluate the depth for every point where the GT is available. As shown in Table 2, 3 and 4, we report results from “Ours w/o motion”, and “Ours w/o motion”. “Ours w/ motion” refers to the setting we described in Section 5, where we learn the motion similarity scores serving the ARAP prior. “Ours w/o motion” refers to the same unsupervised pipeline, except that all rigidity scores are explicitly set to 1. With this modification, piecewise-rigidity is implied during training. In Table 3, we further compare the performance of our methods, by separately evaluating the performance in dynamic and static parts of the images. In Table 4, the results from the non-rigid reconstruction models DLH [17], MDH [58] and MaxRig [31] were obtained using the complete test sequence to perform reconstruction. The evaluations for DLH and MaxRig were performed only at the reconstructed points as these methods provide sparse reconstructions.

**Motion segmentation results.** We also evaluate the performance of our per-pixel motion embedding. As the ground-truth motion-embeddings are not available, we per-
**Table 5:** Average overall pixel accuracy on MPI Sintel [11].

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean ACC</th>
<th>Mean IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tanii et al. [67]</td>
<td>0.872</td>
<td>0.728</td>
</tr>
<tr>
<td>Yang et al. [81]</td>
<td>0.755</td>
<td>0.519</td>
</tr>
<tr>
<td>Ours w/ motion</td>
<td>0.912</td>
<td>0.731</td>
</tr>
</tbody>
</table>

**7. Conclusion**

We demonstrated the possibility for unsupervised learning of depth using monocular videos of non-rigid scenes in order to infer depth from a single image. Unsupervised learning is enabled by established priors used in the NRSfM literature. Building upon existing works, we reformulated commonly used NRSfM priors within a unified framework suitable for neural network training. We further investigated the utility of the ARAP prior for dense depth reconstruction from a single view. Our experiments demonstrate improvements over the alternative methods on numerous datasets. The improvements can be attributed to our geometric error-based loss function (6) computed directly in the 3D reconstruction space. We assumed only the minimum necessary prior in this work, as we aimed to learn without any supervision (including additional scene priors). Depending on scene properties, different priors can be integrated into our pipeline in a similar way, boosting the performance further.

**Acknowledgments.** This research was partly supported by Innosuisse funding (Grant No. 34475.1 IP-ICT), by FIFA, and by the ETH Zurich project with Specta and EU H2020 ENCORE (Grant No. 820434).
References


[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei Fei Li. Imagenet: a large-scale hierarchical image database. In IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 06 2009. 6


[56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming


Walter Whiteley. R rigidity and scene analysis, 2004. 3


