

Dimensions of Motion: Monocular Prediction through Flow Subspaces

Richard Strong Bowen^{*1,2} Richard Tucker^{*1} Ramin Zabih^{1,2} Noah Snavely^{1,2}
 rsb@cs.cornell.edu {richardt, raminz, snavely}@google.com
¹ Google Research ² Cornell Tech

Abstract

We introduce a way to learn to estimate a scene representation from a single image by predicting a low-dimensional subspace of optical flow for each training example, which encompasses the variety of possible camera and object movement. Supervision is provided by a novel loss which measures the distance between this predicted flow subspace and an observed optical flow. This provides a new approach to learning scene representation tasks, such as monocular depth prediction or instance segmentation, in an unsupervised fashion using in-the-wild input videos without requiring camera poses, intrinsics, or an explicit multi-view stereo step. We evaluate our method in multiple settings, including an indoor depth prediction task where it achieves comparable performance to recent methods trained with more supervision. Our project page is at <https://dimensions-of-motion.github.io/>.

1. Introduction

Monocular video is widely used as training data in self-supervised learning of depth prediction from single images. Many such methods (including those that predict additional scene properties such as moving object masks) operate by reconstructing one view from another and are supervised using a photometric loss. To perform this reconstruction, either ground truth camera poses and intrinsics are computed in a pre-processing step like structure from motion or SLAM, or else additional networks are trained to predict camera parameters. Either way, an explicit representation of camera pose and intrinsics is part of the training setup.

We investigate an alternative approach that uses optical flow—the apparent movement of pixels between two images or frames of video—as supervision, without requiring the poses of those frames. Estimating optical flow is still a challenging task, but recent deep learning approaches can produce quite high-quality two-frame optical flow, and achieve good generalization across datasets. How can we use such

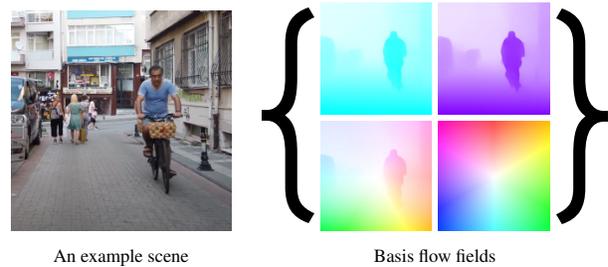


Figure 1. This single frame from a video shows a cyclist in a street scene. From this image alone, can we predict what the optical flow would be between it and the next frame of the video? This single-view flow prediction problem is inherently ill-posed—there could be many possible flow fields, depending on whether the camera is moving left, up, forward, etc, and depending on the motion of the objects in the scene. However, the *space* of possible flow fields should be low-dimensional—that is, it should be spanned by a small number of basis flow fields, some of which (three translations and a rotation) are shown to the right. Moreover, these basis flows will be related to scene properties like depth and instance segmentation, which is why for instance some of these flow fields resemble disparity maps. In this paper, we show how to use these low-dimensional flow subspaces to learn to predict properties like depth from single images, supervised by optical flow computed from Internet videos. *Video imagery used under Creative Commons license from YouTube channels POPtravel (Figs. 1, 2, 5, 6) and SonaVisual (Fig. 4).*

optical flow from pairs of video frames to help supervise *single-image* tasks like monocular depth prediction? If the camera is moving between the pair of frames, then the induced flow will be related to the scene depth, and so we might imagine that the problem of *single-image flow prediction* would be a good proxy for other scene prediction tasks. However, the task of predicting optical flow from a single image is inherently ill-posed, because an infinite family of possible flows could result from different combinations of camera and scene motion. Our approach, then, is not to predict a particular optical flow, nor even a distribution over optical flows, but to predict a low-dimensional flow subspace (a subspace of the much larger space of all theoretically possible optical flows) that contains all realizable instantaneous

* Authors contributed equally.

optical flows (i.e., realizable pixel velocities under small camera or scene motion) given an input image. This overall idea is illustrated in Fig. 1.

In fully unconstrained videos the possible optical flows in a scene are numerous and varied, but prior work has shown that under assumptions of instantaneous flow and a rigid scene, the possible flows form a low-dimensional linear subspace, parameterized by depth or disparity. In settings with potentially moving objects, flow resides within a larger but still low-dimensional subspace which we show can be elegantly parameterized by depth and an object embedding.

We use the novel task of predicting a flow subspace as a proxy to learn to predict depth and objects without using any ground truth labels for them, and without requiring camera poses or estimating them via another network. We predict a subspace which encompasses the possible optical flows from any camera movement and focal length, and employ a simple but novel loss that measures the distance between this subspace and the actual optical flow to a nearby frame (computed using a state-of-the-art method such as RAFT [43]). This allows us to establish a training setup in which the only required input is video frames.

We review in Section 3 the families of flow that arise from camera movement, and extend this analysis to consider moving objects using an *object instance embedding* (Section 3.3). We show how this analysis can be applied—in tandem with a linear solver—to train a deep network to predict scene properties from a single image using pairs of frames from Internet videos as supervision, and conduct experiments on depth prediction and object embedding with the RealEstate10K [57] dataset and with a varied dataset of videos of people walking around cities. Fig. 2 shows an overview of this training setup. On RealEstate10K we obtain comparable performance to other methods on the same dataset, even without using pose or sparse depth supervision (Section 4.1).

2. Related work

2.1. Optical flow

Our method relies on optical flow as a source of supervision; modern two-frame optical flow methods [41, 44] are robust and generalize fairly well across datasets.

A number of flow estimation techniques exploit the relationship between optical flow, scene geometry (i.e. depth) and motion, first analyzed in the context of the human eye [28]. Irani constrains the task of flow estimation between two images using a subspace formulation for instantaneous flow due to camera movement in a rigid 3D scene [20]. The rigidity constraint can be relaxed by treating the subspace as a per-pixel basis for flow and applying regularization to the basis coefficients [35], or by using non-rigid models that allow objects to deform [13]. Flow rank constraint techniques can also be used for tracking and reconstruction [4, 45, 46].

The relationship between optical flow and object or camera movement has also been applied to compute an ‘ideal’ flow for the purpose of comparing and evaluating different flow estimation methods [29], or to recover the underlying camera and scene parameters from flow [17]. More recently, deep learning methods have used this relationship to estimate optical flow simultaneously with object or motion segmentation [51] and camera movement [38]. Other uses of flow subspaces include building a higher-dimensional flow subspace (dimension 500) by applying PCA to a collection of films to facilitate more efficient flow estimation [50], and expressing local phenomena such as affine motion and motion edges using a basis of ‘steerable’ flow fields, with the aim of using the decomposition for motion recognition [7].

Some work has addressed the problem of predicting optical flow or motion fields from a single image, often by supervising from video [36], with the aim of producing convincing animations from still images [18], or as an intermediate step in action recognition [10]. Rather than predicting a specific optical flow, Walker *et al.* [49] predict a probability distribution over a quantized coarse flow. These methods are primarily concerned with object motion and suppose a static camera. In addition to methods that predict motion fields, other work directly predicts future frames from a single image [53]. Again, this work often is primarily concerned with object and not camera motion.

We apply the relationship between optical flow and movement, and a subspace characterization like that of Irani [20], in a new context: rather than attempting to estimate optical flow, we use it as a source of supervision to learn to predict scene structure from a single input image.

2.2. Monocular depth

Supervised learning of monocular depth has a long history; supervision may come from active sensing with LIDAR [14] or structured light [32], or from human annotators judging which of a pair of points is closer [9]. As with optical flow, it is difficult and expensive to obtain ground truth to support supervised learning: data may be limited in scale, or spatial density, or both. Instead, depth or disparity may be computed from stereo imagery found online [52], or from 3D movies as in the MiDaS system [37]. By applying multi-view stereo [39], depth can also be obtained from collections of in-the-wild photos [27] or from videos of (artificially) static scenes [26].

When stereo or multi-view imagery with camera poses is available, one can learn depth without explicit depth supervision. Rather, supervision is provided by a photometric loss (reconstructing one view by warping another according to depth) [11, 15, 21]. As view synthesis and depth prediction are related tasks, many single image view synthesis methods naturally produce dense depth as an intermediate step or an auxiliary output. Niklaus *et al.* [34] produce a pan-and-zoom

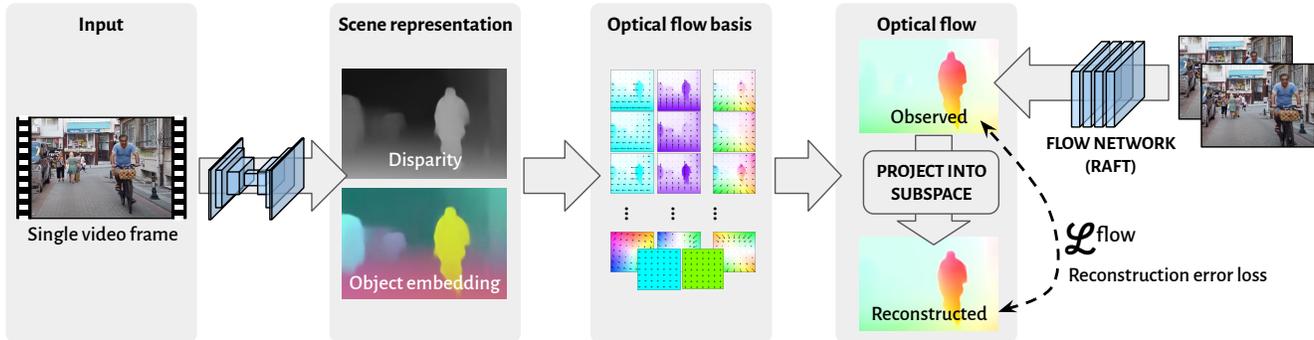


Figure 2. System overview. From a single input frame, our network predicts a scene representation consisting of disparity and (optionally) an embedding of objects into ambient space, from which we generate a basis for a subspace of optical flow. During training, we minimise the distance from this subspace to observed optical flow, computed from the input frame and another frame using a pre-existing network.

effect from a single photo; Tucker and Snavely [47] learn to generate a multiplane image from which depth may be extracted; Li *et al.*'s [24] MINE combines the properties of this representation with those of Neural Radiance fields. These methods may be supervised by a combination of view synthesis and (when available) sparse depth.

The multi-view processing required to compute accurate camera poses can be computationally expensive and require significant manual tuning. Another approach is to train a second neural network to predict relative pose between two or more input images at training time, in tandem with a first network that predicts monocular depth. To handle video sequences featuring non-static scenes, these methods may also predict an ‘explainability’ map [56], an explicit object motion map [23], a self-discovered object motion map [3], optical flow for moving regions [38], or rigid-body object transformations [48]. Zhao *et al.* [55] learn flow and monocular disparity jointly by sampling from dense correspondences to find a pose, with a photometric loss as well as a scale-invariant depth loss. In contrast, our method uses optical flow rather than image reconstruction for supervision, and does not require or predict explicit camera or object poses.

2.3. Linear subspaces in computer vision

Linear subspaces underlie a range of other vision problems in addition to optical flow. They apply also to the *appearance manifold* [31]: given an image, other images from slightly different camera viewpoints will lie in a low-dimensional subspace (in particular, 6D, corresponding to the six degrees of freedom of camera motion). Samples of the local appearance manifold can therefore be used for 6DoF camera tracking [54]. In 3D processing, functional maps in the span of a small basis can be used for non-rigid point cloud registration [19]. There are also classic results in the dimensionality of the space of images of a specific scene under any possible illumination [2, 12, 40], and rank constraints have been studied in the context of motion segmentation when considering multiple rigidly moving objects [25].

2.4. Object instance embeddings

To handle moving objects, we produce a per-pixel *object instance embedding* (detailed in Sec. 3.3), where pixels in the same object should map to the same vector, while pixels in different objects as well as the background should map to different vectors. This embedding is related to the task of instance segmentation [16]. While most instance segmentation approaches employ multi-stage pipelines that include mask proposals, clustering, or other complex techniques, we take an approach similar to Fathi *et al.* [6], who learn an end-to-end, per-pixel 64-dimensional embedding using ground-truth labels (whereas our formulation is self-supervised). A discretization method such as clustering run on such embeddings may produce reasonable instance segmentations. Newell *et al.* [33] produce, in a supervised way, a per-class heatmap and a per-class, one-dimensional index (or ‘tag’) at each pixel to separate instances.

3. Methods

Our approach has two main parts. First, in Section 3.1, we consider the concept of a basis for optical flow, and show a way to train a system that produces such a basis by using observed flow as supervision and learning to minimize a *flow reconstruction error*.

Then, in Sections 3.2–3.3, we identify subspaces of optical flow corresponding to certain assumptions about the scene. In each case we give a basis for optical flow, identify its dimension, and show how it is parameterized by an appropriate scene representation (disparity, object embedding) which could be predicted by a neural network.

3.1. Learning a subspace of optical flow

For an input image I of size $H \times W$, the space of all possible optical flow fields is $\mathbb{R}^{H \times W \times 2}$, since flow consists of a separate 2D motion vector for each pixel. But only a tiny fraction of these theoretical optical flows are actually realizable given a specific scene. We represent such possible

flows as a low-dimensional subspace \mathcal{S} of this space. \mathcal{S} consists of linear combinations of a set of flows Δ_i that form a flow-basis \mathcal{B} :

$$\mathcal{B} = \{\Delta_0, \Delta_1, \dots, \Delta_{n-1}\}, \Delta_i \in \mathbb{R}^{H \times W \times 2} \quad (1)$$

$$\mathcal{S} = \text{SPAN}(\mathcal{B}) \quad (2)$$

The individual fields Δ_i , and hence the subspace \mathcal{S} , are specific to \mathbf{I} and not global across all images. While in general the set of plausible flows is not a linear subspace, in the *instantaneous flow* limit the space is closed under linear combination; as long as our time interval is such that rotation and forward motion are small, the instantaneous model is a good approximation of flow [17, 20, 28].

To learn to predict \mathcal{B} , we quantify how well the space \mathcal{S} explains an observed optical flow Δ , by finding $\hat{\Delta} \in \mathcal{S}$ with minimum distance from Δ via projection of Δ into \mathcal{S} . We first find an orthonormal basis for \mathcal{S} via a (differentiable) singular-value decomposition on the matrix of basis vectors $[\Delta_0 | \Delta_1 | \dots | \Delta_{n-1}]$, where each Δ_i is here viewed as a column vector with $2HW$ elements. The left singular vectors form an orthonormal basis for \mathcal{S} , from which we can compute $\hat{\Delta}$. For more details, see the supplemental material.

The distance from Δ to \mathcal{S} , or equivalently the error in our reconstructed flow $\hat{\Delta}$, is the *flow reconstruction loss*:

$$\mathcal{L}^{\text{flow}} = \|\Delta - \hat{\Delta}\|_2 \quad (3)$$

Because the SVD routine we use is differentiable, gradient can flow back from this loss through to the basis vectors. In lieu of images with ground truth optical flow we sample pairs of nearby frames from video sequences, using one image from each pair to generate the basis \mathcal{B} and running a state-of-the-art flow network [43] to produce the observed flow Δ from the pair of images.

In practice, the basis \mathcal{B} is not the direct output of our network: instead we output a representation of the scene from which \mathcal{B} can be directly computed. In the next sections, we therefore consider flow bases corresponding to specific types of motion.

3.2. Instantaneous flow from camera motion

Optical flow arises from the motion of the camera and of objects in the scene. If the scene is stationary, then all flow comes from camera motion, and we can characterize it explicitly.

We will consider the *instantaneous optical flow* at a point in time. Suppose a world point (x, y, z) projects onto the sensor at pixel (u, v) at time $t = 0$. The instantaneous flow is the apparent velocity at this pixel: $\Delta = (u', v')$. The instantaneous flow is well-studied within computer vision [17, 20] and in other fields [28]. For a given disparity map and camera intrinsics, the instantaneous flow depends *linearly* on the six parameters of translational and rotational velocity, i.e. the

six degrees of freedom in the camera pose. Consequently, the set of possible instantaneous flows forms a linear space with six basis vectors, which we now enumerate. (For more details and derivations, see for example Heeger and Jepsen §3 [17] or Irani, Appendix A [20].)

Camera translation. For translation along each axis (\mathbf{T}_x , \mathbf{T}_y , \mathbf{T}_z), the basis vectors are:

$$\Delta_{\mathbf{T}_x} = \begin{bmatrix} d \cdot f_x \\ 0 \end{bmatrix}, \Delta_{\mathbf{T}_y} = \begin{bmatrix} 0 \\ d \cdot f_y \end{bmatrix}, \Delta_{\mathbf{T}_z} = \begin{bmatrix} d \cdot (c_x - u) \\ d \cdot (c_y - v) \end{bmatrix} \quad (4)$$

Here f_x and f_y are the x and y focal lengths of the camera, (c_x, c_y) is the principal point, and d (a function of (u, v)) is the disparity or inverse depth at (u, v) . The translational flow fields are horizontal and vertical for translation in x and y , and radial (centered on the principal point) for translation in z , and in all three cases the flow is proportional to disparity d since the further away objects are, the less they appear to move when the camera translates. Note that there is the usual scale ambiguity between translation velocity and disparity.

Camera rotation. The basis vectors for rotation about the x , y and z axes are:

$$\Delta_{\mathbf{R}_x} = \begin{bmatrix} \frac{1}{f_y}(u - c_x)(v - c_y) \\ f_y + \frac{1}{f_y}(v - c_y)^2 \end{bmatrix} \quad (5)$$

$$\Delta_{\mathbf{R}_y} = \begin{bmatrix} f_x + \frac{1}{f_x}(u - c_x)^2 \\ \frac{1}{f_x}(u - c_x)(v - c_y) \end{bmatrix} \quad (6)$$

$$\Delta_{\mathbf{R}_z} = \begin{bmatrix} \frac{f_x}{f_y}(v - c_y) \\ \frac{f_y}{f_x}(c_x - u) \end{bmatrix} \quad (7)$$

As expected, flow from rotation does not depend on disparity, since motion induced by pure camera rotation is independent of depth. At large focal lengths, flow from rotation about the x (or y) axis is almost vertical (or horizontal) and uniform; at smaller focal lengths the effects of curvature are more apparent, especially at the corners. For rotation around the z -axis, flow is circular (or elliptical if $f_x \neq f_y$) around the optical center.

Bases. Combining translation and rotation, we have a basis for the six-dimensional space of flow due to camera motion:

$$\mathcal{B}_{\text{camera}} = \{\Delta_{\mathbf{T}_x}, \Delta_{\mathbf{T}_y}, \Delta_{\mathbf{T}_z}, \Delta_{\mathbf{R}_x}, \Delta_{\mathbf{R}_y}, \Delta_{\mathbf{R}_z}\}. \quad (8)$$

Fig. 3 depicts this flow basis for an example scene.

A common case when dealing with real-world imagery is that f_x and f_y are unknown but equal, and c_x and c_y are known (or assumed to be at the center of the image). Can we produce a basis for flow due to camera motion in this case? Since basis vectors may be freely scaled up or down, only the flows from rotation about the x - and y -axes are problematic.

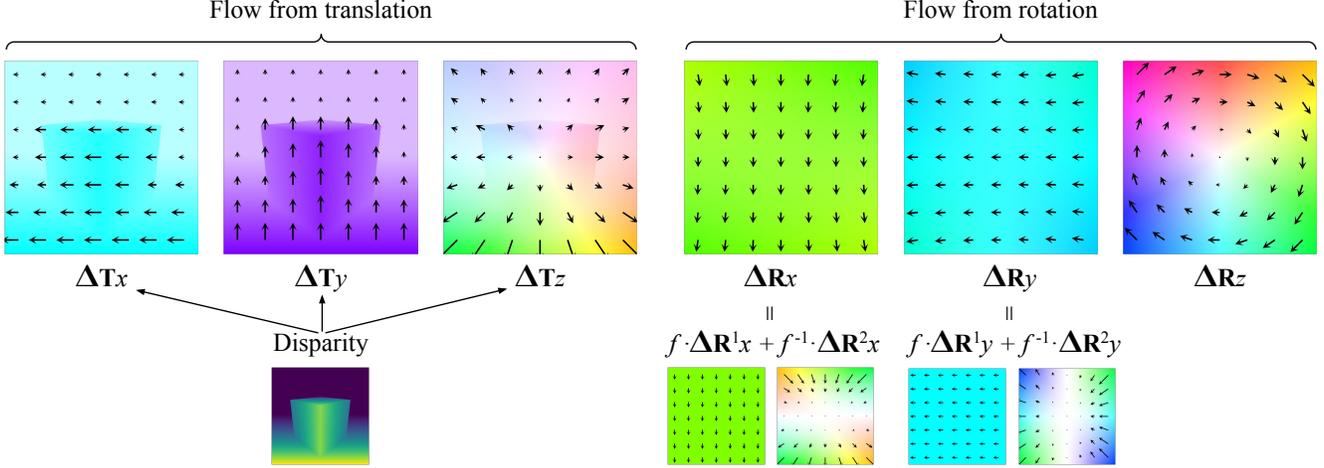


Figure 3. Flow basis for camera motion with six degrees of freedom, shown on a simple cube scene (best viewed in color). The translation flows are derived from the depicted scene disparity. See Section 3.2.

We can separate out the terms in f and $1/f$, replacing each of these two flow fields by a pair (as also shown in Fig. 3):

$$\Delta_{\mathbf{R}x} = f_y \Delta_{\mathbf{R}^1x} + \frac{1}{f_y} \Delta_{\mathbf{R}^2x}, \Delta_{\mathbf{R}y} = f_x \Delta_{\mathbf{R}^1y} + \frac{1}{f_x} \Delta_{\mathbf{R}^2y} \quad (9)$$

where

$$\Delta_{\mathbf{R}^1x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \Delta_{\mathbf{R}^2x} = \begin{bmatrix} (u - c_x)(v - c_y) \\ (v - c_y)^2 \end{bmatrix}, \quad (10)$$

$$\Delta_{\mathbf{R}^1y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \Delta_{\mathbf{R}^2y} = \begin{bmatrix} (u - c_x)^2 \\ (u - c_x)(v - c_y) \end{bmatrix}. \quad (11)$$

We thus end up with a basis of eight flow fields, parameterized by disparity d , to cover the space of camera movement with unknown focal length. Note that this basis actually covers a slightly larger space, since although we assume $f_x = f_y$ we do not have a way to enforce this (nonlinear) constraint in our decomposition of the rotation flows.

3.3. Instantaneous flow from object motion

Suppose now that the camera is stationary but that a rigid object in the scene is moving. What does the resulting space of possible optical flow fields look like? For all points (u, v) outside the moving object the flow will be zero. For points *within* the object, we observe that for any rigid object motion (rotation or translation) there is an equivalent camera motion, and thus the space of flow from rigid object motion is exactly the same as the space of flow from camera motion restricted to points in the object. That is, given a binary object mask m which is 1 within the object and 0 elsewhere, a basis for rigid movement of the object is given by

$$\mathcal{B}_m = \{m\Delta \mid \Delta \in \mathcal{B}_{\text{camera}}\}. \quad (12)$$

Alternatively we may consider a flow basis for *object translation* only, which is just three dimensions per object:

$$\mathcal{B}_m^{\text{translation}} = \{m\Delta_{\mathbf{T}x}, m\Delta_{\mathbf{T}y}, m\Delta_{\mathbf{T}z}\}. \quad (13)$$

Hence, one way to produce a basis for flow due to object motion would be to predict disparity d and a set of object masks. But we can instead model potential movers in the scene *without* explicit masks, by introducing an *object instance embedding* $\phi(u, v) \in \mathbb{R}^A$. This embedding, like much higher-dimensional embeddings used in instance segmentation [6], gives for each pixel a unit vector in an embedding space of dimension A . (In our experiments, $A = 6$.) The idea is that pixels in the same object should map to the same point in this space, but that different objects, and the background, should map to different and linearly-independent points.

With up to A objects (including the background) at linearly-independent positions in this space, a matrix $\mathbf{M} \in \mathbb{R}^{3 \times A}$ is sufficient to describe a mapping from embedding space to movement in the x -, y - and z -axes, allowing each object to move independently. The movement at each pixel is then $\mathbf{M}\phi$, and the flow $\Delta_{\mathbf{M}\phi}$ due to object and camera translation is:

$$\begin{aligned} \Delta_{\mathbf{M}\phi} &= (\mathbf{M}\phi) \cdot \begin{bmatrix} \Delta_{\mathbf{T}x} \\ \Delta_{\mathbf{T}y} \\ \Delta_{\mathbf{T}z} \end{bmatrix} \\ &= \sum_{i=0}^{A-1} \mathbf{M}_{0i} \phi_i \Delta_{\mathbf{T}x} + \mathbf{M}_{1i} \phi_i \Delta_{\mathbf{T}y} + \mathbf{M}_{2i} \phi_i \Delta_{\mathbf{T}z}. \end{aligned} \quad (14)$$

Thus a basis for the space of possible flows $\Delta_{\mathbf{M}\phi}$ is given by

$$\mathcal{B}_\phi^{\text{translation}} = \{\phi_i \Delta_{\mathbf{T}x}, \phi_i \Delta_{\mathbf{T}y}, \phi_i \Delta_{\mathbf{T}z} \mid 0 \leq i < A\}. \quad (15)$$

Projecting into this basis implicitly finds a matrix \mathbf{M} . To allow for camera rotation too we add the various $(\Delta_{\mathbf{R}})$ to

Method (Dataset)	iBims-1						NYU Depth V2					
	rel↓	log10↓	RMS↓	σ_1 ↑	σ_2 ↑	σ_3 ↑	rel↓	log10↓	RMS↓	σ_1 ↑	σ_2 ↑	σ_3 ↑
<i>Supervised by depth</i>												
DIW (DIW) [5]	0.25	0.10	1.00	0.61	0.86	0.95	0.25	0.1	0.76	0.62	0.88	0.96
MegaDepth (Mega/DIW) [27]	0.20	0.08	0.78	0.70	0.91	0.97	0.21	0.08	0.65	0.68	0.91	0.97
MiDaS v2.1 (MiDaS 10) [37]	0.14	0.06	0.57	0.84	0.97	0.99	0.16	0.06	0.50	0.80	0.95	0.99
3DKenBurns (Mega/NYU/3DKB) [34]	0.10	0.04	0.47	0.90	0.97	0.99	0.08	0.03	0.30	0.94	0.99	1.00
<i>Supervised by view synthesis plus sparse depth, using pose from SfM</i>												
Single-view MPI (RE10K) [47]	0.21	0.08	0.85	0.70	0.91	0.97	0.15	0.06	0.49	0.81	0.96	0.99
MINE (RE10K) [24]	0.11	0.05	0.53	0.87	0.97	0.99	0.11	0.05	0.40	0.88	0.98	0.99
<i>Supervised by flow reconstruction only</i>												
Ours (RE10K)	0.12	0.05	0.55	0.85	0.97	0.99	0.12	0.05	0.43	0.86	0.97	0.99

Table 1. Depth prediction quality measured with the iBims-1 [22] and NYU-V2 [32] benchmarks. Our method, trained on RealEstate10K, achieves comparable performance with the best methods among those without extra, hard-to-scale supervision (such as structured light, as in NYU). Additionally, we remove the need for an explicit posing or a structure from motion step in the pipeline in, e.g., MINE. See Section 4.1.

our basis, giving a basis parameterized by (d, ϕ) with $3A + 3$ dimensions, or $3A + 5$ for unknown focal length. Allowing for *object* rotation is also straightforward (add the $(\phi_i \Delta_{\mathbf{R}})$ to the basis), but we did not find that it improved performance.

4. Experiments

We use the setup described in Section 3.1 to learn monocular depth prediction (Section 4.1) and to learn depth together with object embedding (Section 4.2). In either case, our architecture roughly follows DispNet [30], using an encoder-decoder architecture with skip connections, described more fully in the supplementary material. For depth, the output is a $H \times W \times 1$ tensor of disparities, with sigmoid activation applied, from which we generate an 8-dimensional basis as described in Section 3.2. For instance embedding experiments the output is $H \times W \times (1 + A)$, i.e., 1 dimension for disparity (again with sigmoid activation) and A dimensions for the embedding (normalized to be unit-length at each pixel), from which we generate a $3A + 5$ dimensional basis as described in Section 3.3. Training data consists of frames from monocular video, with observed optical flow estimated using RAFT [43]. No use was made of camera intrinsics, poses, or ground truth depth data in training.

Our implementation is in TensorFlow [1] and uses `tf.linalg.svd` for the singular value decomposition (see Section 3.1). In our experiments with object embedding, we run the solver twice: once using only the 8-dimensional camera-movement flow basis, and again using the full $(3A + 5)$ -dimensional basis, and we train using reconstruction losses from both solves.

4.1. Learning disparity on static scenes

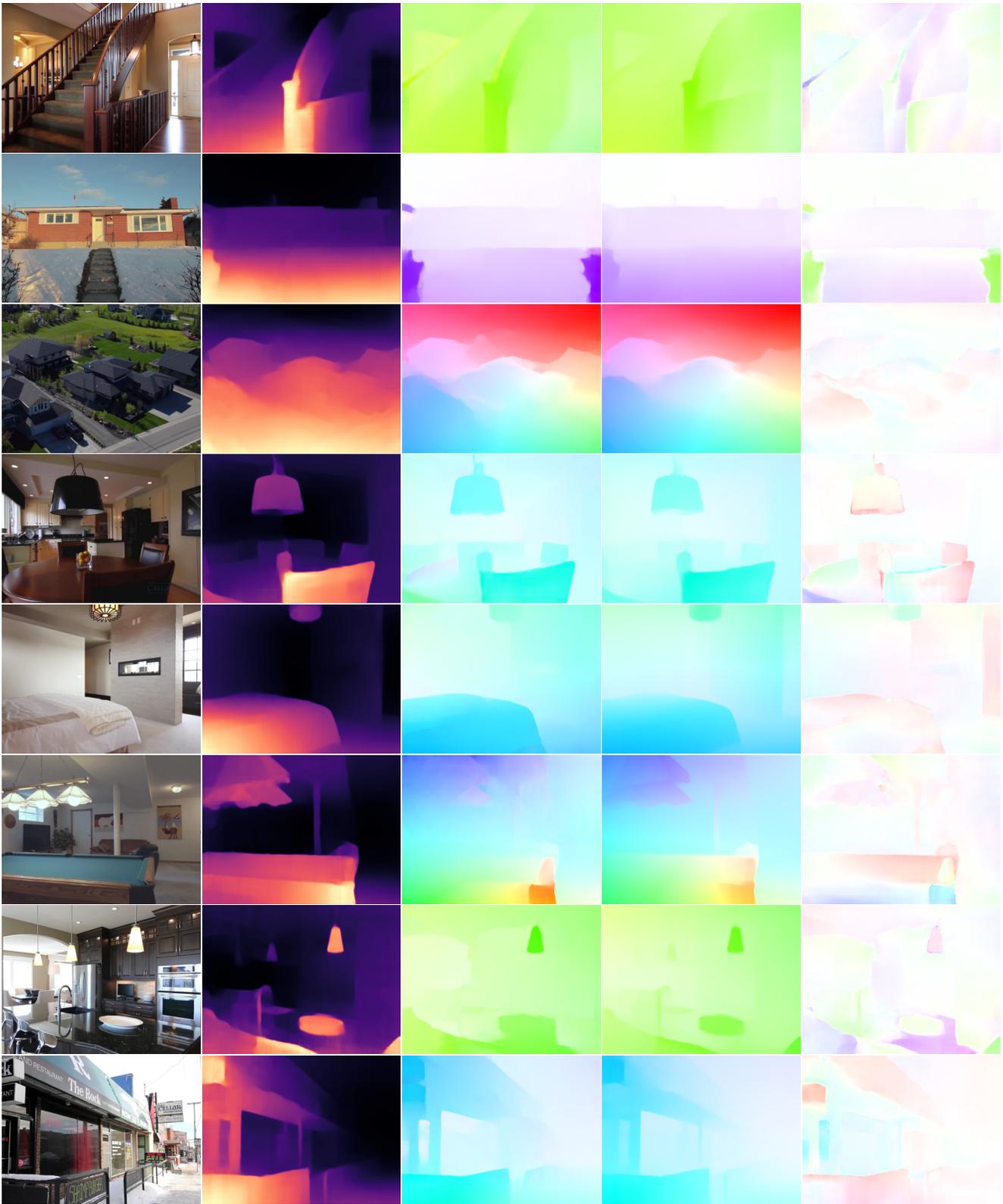
To investigate depth prediction on static scenes, we train on images from the RealEstate10K dataset [57], but without making any use of the included camera intrinsics or

poses. This dataset consists of frames from internet real estate videos, including both indoor and outdoor settings. Scenes are mostly static and feature a variety of different camera movements. We train on pairs of images 3–10 frames apart, at a resolution of 240×320 . We evaluate on the iBims-1 [22] and NYU-V2 Depth [32] test sets, which contain ground truth depth maps, following the protocol of Niklaus *et al.* [34]. As shown in Table 1, our method achieves comparable performance to other methods trained on this dataset that *do* use camera information or a potentially expensive multi-view stereo or structure-from-motion step in the pipeline. Example outputs are shown in Fig. 4.

4.2. Learning on dynamic scenes

Walking-Tours dataset. To demonstrate the object embedding technique on scenes with moving camera and moving objects, we collect and train on a dataset of internet videos. These videos are chosen by searching for “walking around (city)” for each of the largest 50 cities in the world by population, and collected and processed following the same approach as the RealEstate10K dataset [57]. In total, we consider about 250 videos and about 1.2M frames. The videos, which are mostly tours shot from hand-held or vehicle-mounted cameras, are dynamic and feature both camera and object motion. They vary in geographic location, image quality, camera hardware, and resolution.

Qualitative analysis. Qualitative results are shown in Fig. 5. We find that the network simultaneously learns disparity and a (soft) instance segmentation. Since the instance embedding vector is a unit vector in \mathbb{R}^6 , we visualize it in three ways. In Fig. 5, we reduce the dimension using principal component analysis [8] and show the top three principal dimensions in RGB; we also show the (spatial) gradient magnitude of the embedding, which highlights strong edges. For example, the first two rows of Fig. 5 show that the network has assigned



Input image

Predicted disparity

Flow

Reconstructed flow

Flow error

Figure 4. Experiment on RealEstate10K. Image, disparity predicted by our network, flow, reconstructed flow, and flow error are shown here on test data. (Flow is between source and a target image; the latter is not shown here). See Section 4.1.

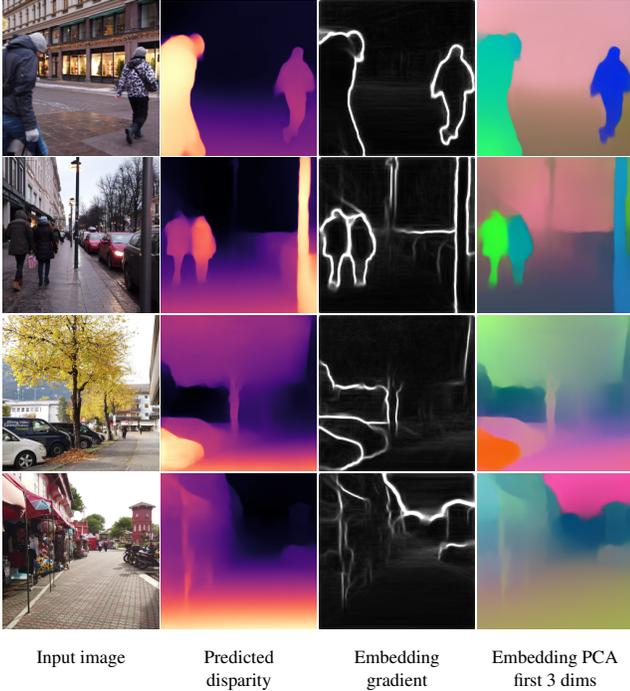


Figure 5. Qualitative results on Walking-Tours with visualizations of the object embedding ϕ . The spatial gradient of ϕ (third column) separates cars and people, while also oversegmenting somewhat by drawing edges at strong depth discontinuities. The fourth column shows the first three dimensions of ϕ after PCA. Instances are separated, though there is some undesirable smooth variation within individual objects and the background. See Section 4.2.

different instances of people different vectors. In Fig. 6, we show a proof of concept of using the embedding for instance segmentation: we (manually) choose a few seed points on each of several objects. Then, every other pixel is assigned to the closest seed in bilateral space, i.e., where the distance is the the (weighted) sum of distances in euclidean and embedding space.

Oversegmentation. One issue is that, beyond the limits imposed by the dimensionality of ϕ , the network has no incentive not to oversegment. In particular, it places embedding edges between regions where the difference between flows is hard to predict from a single image. These include object boundaries, as desired, but also includes large depth discontinuities, such as between a foreground and midground objects. Thus the network will sometimes place object-embedding edges at, e.g., the horizon, as seen in Fig. 5.

5. Discussion

We show that our subspace model can be applied to the tasks of disparity estimation and object instance embedding from in-the-wild internet videos without the use of camera intrinsics/pose or multiview stereo.

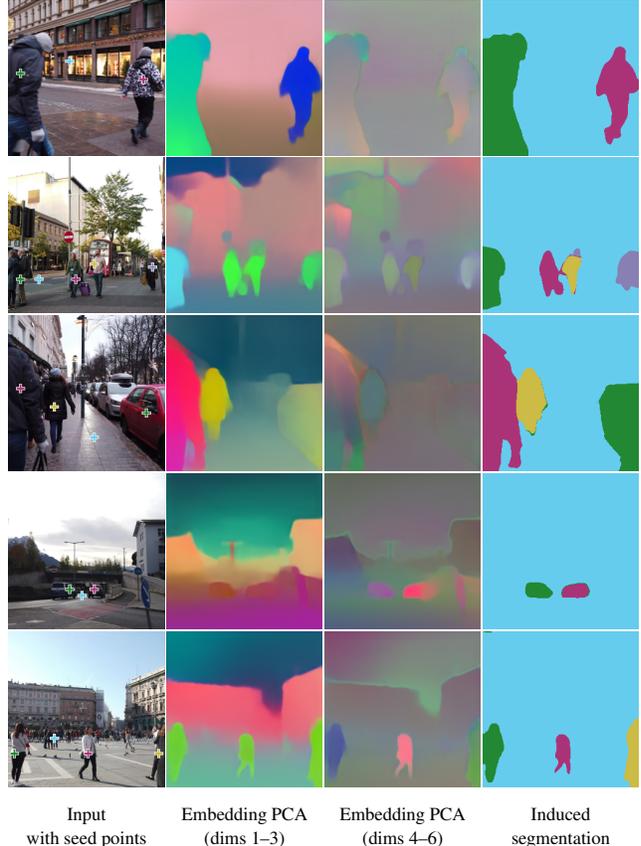


Figure 6. Example segmentation from our instance embedding. We manually choose seed points for objects and background, and induce a segmentation by coloring each pixel according to which of the seed points is closest to it in bilateral embedding space. Using only self-supervision, the network has learned to separate person and car instances. See Section 4.2.

Our approach has some limitations. One is that we rely on the instantaneous-flow assumption, and so our method is suitable for use only on video datasets in which the motion is not too fast. It is unlikely to be suitable for image-collection datasets such as Megadepth [27]. Relatedly, our method depends on flow, which could be degraded by large motions, occlusion, specularities, and so on.

Our method is also limited by an independence assumption for dynamic scene content. For example, for a given image pair an object that has the same motion as the camera will have the same flow as an object at infinite depth. This situation is common in driving datasets; we find that when trained on datasets such as KITTI [14] or Waymo [42], the network tends to do well on static portions of the scene but assigns a very large depth to cars moving in the same direction as the capturing car. Our method is more suitable to in-the-wild internet-collected datasets such as RealEstate10K or Walking-Tours, which tend to feature more general camera motions.

References

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. [6](#)
- [2] P.N. Belhumeur and D.J. Kriegman. What is the set of images of an object under all possible lighting conditions? In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 270–277, 1996. [3](#)
- [3] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in Neural Information Processing Systems*, 32, 2019. [3](#)
- [4] W Brand. Morphable 3d models from video. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, volume 2, pages II–II. IEEE, 2001. [2](#)
- [5] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, 2016. [6](#)
- [6] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P Murphy. Semantic instance segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*, 2017. [3](#), [5](#)
- [7] David J. Fleet, Michael J. Black, Yaser Yacoob, and Allan D. Jepson. Design and use of linear models for image motion analysis. *Int. J. Comput. Vis.*, 36(3):171–193, 2000. [2](#)
- [8] Karl Pearson F.R.S. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. [6](#)
- [9] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 2002–2011, 2018. [2](#)
- [10] Ruohan Gao, Bo Xiong, and Kristen Grauman. Im2Flow: Motion hallucination from static images for action recognition. In *CVPR*, 2018. [2](#)
- [11] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision, (ECCV)*, pages 740–756. Springer, 2016. [2](#)
- [12] Rahul Garg, Hao Du, Steven M. Seitz, and Noah Snavely. The dimensionality of scene appearance. In *IEEE International Conference on Computer Vision, (ICCV)*, pages 1917–1924, 2009. [3](#)
- [13] Ravi Garg, Luis Pizarro, Daniel Rueckert, and Lourdes Agapito. Dense multi-frame optic flow for non-rigid objects using subspace constraints. In *Asian Conference on Computer Vision*, pages 460–473. Springer, 2010. [2](#)
- [14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. [2](#), [8](#)
- [15] Clément Godard, Oisín Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 270–279, 2017. [2](#)
- [16] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, pages 1–19, 2020. [3](#)
- [17] David J Heeger and Allan D Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, 1992. [2](#), [4](#)
- [18] Aleksander Holynski, Brian L Curless, Steven M Seitz, and Richard Szeliski. Animating pictures with Eulerian motion fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 5810–5819, 2021. [2](#)
- [19] Jiahui Huang, Tolga Birdal, Zan Gojcic, Leonidas J Guibas, and Shi-Min Hu. Multiway non-rigid point cloud registration via learned functional map synchronization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. [3](#)
- [20] Michal Irani. Multi-frame correspondence estimation using subspace constraints. *International Journal of Computer Vision*, 48(3):173–194, 2002. [2](#), [4](#)
- [21] Ue-Hwan Kim and Jong-Hwan Kim. Revisiting self-supervised monocular depth estimation. *ArXiv*, abs/2103.12496, 2021. [2](#)
- [22] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Körner. Evaluation of CNN-based single-image depth estimation methods. In Laura Leal-Taixé and Stefan Roth, editors, *European Conference on Computer Vision Workshop (ECCVWS)*, pages 331–348. Springer International Publishing, 2018. [6](#)
- [23] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. In Jens Kober, Fabio Ramos, and Claire Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 1908–1917. PMLR, 16–18 Nov 2021. [3](#)
- [24] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. MINE: Towards continuous depth MPI with NeRF for novel view synthesis. In *IEEE/CVF International Conference on Computer Vision, (ICCV)*, pages 12578–12588, October 2021. [3](#), [6](#)
- [25] Ting Li, Vinutha Kallem, Dheeraj Singaraju, and Rene Vidal. Projective factorization of multiple rigid-body motions. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2007. [3](#)
- [26] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths

- of moving people by watching frozen people. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 4521–4530, 2019. 2
- [27] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 2041–2050, 2018. 2, 6, 8
- [28] Hugh Christopher Longuet-Higgins and Kvetoslav Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 208(1173):385–397, 1980. 2, 4
- [29] Marco Mammarella, Giampiero Campa, Mario L Fravolini, and Marcello R Napolitano. Comparing optical flow algorithms using 6-DOF motion of real-world rigid objects. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1752–1762, 2012. 2
- [30] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 4040–4048, 2016. 6
- [31] H. Murase and S.K. Nayar. Visual Learning and Recognition of 3D Objects from Appearance. *Int. J. Comput. Vis.*, 14(1):5–24, Jan 1995. 3
- [32] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision, (ECCV)*, 2012. 2, 6
- [33] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*, 2017. 3
- [34] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3D Ken Burns effect from a single image. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019. 2, 6
- [35] Tal Nir, Alfred M Bruckstein, and Ron Kimmel. Over-parameterized variational optical flow. *International Journal of Computer Vision*, 76(2):205–216, 2008. 2
- [36] Silvia L. Pinteá, Jan C. van Gemert, and Arnold W. M. Smeulders. Déjà vu: Motion prediction in static images. In *European Conference on Computer Vision, (ECCV)*, 2014. 2
- [37] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 2, 6
- [38] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 12240–12249, 2019. 2, 3
- [39] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, June 2016. 2
- [40] Amnon Shashua. *Geometry and Photometry in Three-Dimensional Visual Recognition*. PhD thesis, Massachusetts Institute of Technology, 1993. 3
- [41] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 8934–8943, 2018. 2
- [42] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 2446–2454, 2020. 8
- [43] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision, (ECCV)*, pages 402–419. Springer, 2020. 2, 4, 6
- [44] Zachary Teed and Jia Deng. RAFT-3D: scene flow using rigid-motion embeddings. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 8375–8384. Computer Vision Foundation / IEEE, 2021. 2
- [45] Lorenzo Torresani and Christoph Bregler. Space-time tracking. In *European Conference on Computer Vision*, pages 801–812. Springer, 2002. 2
- [46] Lorenzo Torresani, Danny B Yang, Eugene J Alexander, and Christoph Bregler. Tracking and modeling non-rigid objects with rank constraints. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, volume 1, pages I–I. IEEE, 2001. 2
- [47] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 551–560, 2020. 3, 6
- [48] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. SfM-Net: Learning of structure and motion from video, 2017. 3
- [49] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *IEEE International Conference on Computer Vision, (ICCV)*, pages 2443–2451, 2015. 2
- [50] Jonas Wulff and Michael J Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 120–130, 2015. 2
- [51] Jonas Wulff, Laura Sevilla-Lara, and Michael J Black. Optical flow in mostly rigid scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 4671–4680, 2017. 2
- [52] Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, Yang Xiao, Ruiho Li, and Zhenbo Luo. Monocular relative depth perception with web stereo data supervision. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 311–320, 2018. 2
- [53] Tianfan Xue, Jiajun Wu, Katherine L Bouman, and William T Freeman. Visual dynamics: Stochastic future generation via layered cross convolutional networks. *Trans. Pattern Analysis and Machine Intelligence*, 41(9):2236–2250, 2019. 2

- [54] Hua Yang, Marc Pollefeys, Greg Welch, Jan-Michael Frahm, and Adrian Ilie. Differential camera tracking through linearizing the local appearance manifold. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2007. [3](#)
- [55] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 9151–9161, 2020. [3](#)
- [56] 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, (CVPR)*, pages 1851–1858, 2017. [3](#)
- [57] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph.*, 37(4), July 2018. [2](#), [6](#)

A. Network architecture

Our network (modeled on those of [27] and [46]), is detailed in the following table:

Input	k_1	c_1	k_2	c_2	Output
Norm(I)	7	32	7	32	conv ₁
↓(conv ₁)	5	64	5	64	conv ₂
↓(conv ₂)	3	128	3	128	conv ₃
↓(conv ₃)	3	256	3	256	conv ₄
↓(conv ₄)	3	512	3	512	conv ₅
↓(conv ₅)	3	512	3	512	conv ₆
↓(conv ₆)	3	512	3	512	conv ₇
↓(conv ₇)	3	512	3	512	conv ₈
↑(conv ₈) + conv ₇	3	512	3	512	conv ₉
↑(conv ₉) + conv ₆	3	512	3	512	conv ₁₀
↑(conv ₁₀) + conv ₅	3	512	3	512	conv ₁₁
↑(conv ₁₁) + conv ₄	3	256	3	256	conv ₁₂
↑(conv ₁₂) + conv ₃	3	128	3	128	conv ₁₃
↑(conv ₁₃) + conv ₂	3	64	3	64	conv ₁₄
↑(conv ₁₄) + conv ₁	3	64	3	64	conv ₁₅
conv ₁₅	3	32	3	32	conv ₁₆
conv ₁₆	3	C	-	-	output

Each row above (except the last) describes a pair of convolutional layers in sequence with kernel sizes k_1, k_2 and number of output channels c_1, c_2 . **Input** shows the input to the first layer, where Norm denotes ImageNet-style normalization, ↓ denotes maxpooling with a pool size of 2 (thus halving the size), ↑ denotes nearest-neighbour upscaling by a factor of 2, and + is concatenation. Each layer is followed by ReLU activation.

The final row shows a single convolutional layer which outputs C channels. In our disparity experiments, $C = 1$ and is followed by sigmoid activation. In our disparity plus embedding experiments, $C = 7$: one channel for disparity (with sigmoid activation) and six for embedding (normalized to be unit-length at each pixel).

B. SVD details

To compute $\hat{\Delta}$, as described in Section 3, we assemble the matrix whose column space is \mathcal{S} :

$$M = [\Delta_0 | \Delta_1 | \dots | \Delta_{n-1}], \quad (16)$$

the dimensions of which are $2HW \times n$. Before assembling M , we normalize rotational basis vectors to have norm 1 and translational basis vectors to have norm 2 (prior to pointwise multiplication by disparity). We compute the singular-value decomposition of M :

$$M = U\Sigma V^T. \quad (17)$$

We choose the columns of U corresponding to singular values (entries in Σ) greater than a threshold ε (in our experiments, $\varepsilon = 1 \times 10^{-5}$); calling this submatrix U_s we compute $\hat{\Delta}$ via

$$\hat{\Delta} = U_s U_s^T \Delta. \quad (18)$$

C. Training details

We use the ADAM optimizer with a learning rate of 5×10^{-5} and an L2 regularization on network weights of 1×10^{-6} , and train asynchronously using ten workers with a batch size of 4 per worker. In those experiments which learn an object embedding, we project the ground truth flow twice and compute two losses: one using the basis only resulting from the learned disparity (loss weight 0.5), and one using the full projection described in Section 3.4 (loss weight 1.0). In experiments without object embedding, flow reconstruction loss has a weight of 1.0.

We found in training that the network sometimes produces very large values for disparity or instance embedding; we apply a regularization loss on disparity before sigmoid activation of $L(z) = \max(0, z - 5)$; and a loss on instance-embedding before normalization, $L(z) = \max(0, (\sum_i z_i^2) - 1)$. Each of these are averaged over the image and applied with a weight of 1×10^{-6} . We train for about 5M steps. We choose the best model and checkpoint from five runs based on flow reprojection loss on a held-out validation set.