Dimensions of Motion: 
Monocular Prediction through Flow Subspaces

Richard Strong Bowen*1,2 Richard Tucker*1 Ramin Zabih1,2 Noah Snavely1,2
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 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
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 [42]). 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 [36] 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 [40, 43] 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, 44, 45].

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 [50] and camera movement [37]. 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 [49], 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, 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. [48] 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 [52]. 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 [51], or from 3D movies as in the MiDaS system [36]. By applying multi-view stereo [38], 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 effect from a single photo; Tucker and Snavely [46] 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 [55], an explicit object motion map [23], a self-discovered object motion map [3], optical flow for moving regions [37], or rigid-body object transformations [47]. Zhao et al. [54] 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 [53]. 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, 39], 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 $S$ of this space. $S$ consists of linear combinations of a set of flows $\Delta_i$ that form a flow-basis $\mathcal{B}$:

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

$$S = \text{span}(\mathcal{B})$$  \hspace{1cm} (2)

The individual fields $\Delta_i$, and hence the subspace $S$, are specific to $I$ 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 $S$, we quantifiy how well the space $S$ explains an observed optical flow $\Delta$, by finding $\hat{\Delta} \in S$ with minimum distance from $\Delta$ via projection of $\Delta$ into $S$. We first find an orthonormal basis for $S$ via a (differentiable) singular-value decomposition on the matrix of basis vectors $[\Delta_0|\Delta_1|\cdots|\Delta_{n-1}]$, where each $\Delta_i$ is here viewed as a column vector with $2HW$ elements. The left singular vectors form an orthonormal basis for $S$, from which we can compute $\hat{\Delta}$. For more details, see the supplemental material.

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

$$L_\text{flow} = \|\Delta - \hat{\Delta}\|_2$$  \hspace{1cm} (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 [42] to produce the observed flow $\Delta$ 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 $(T_x, T_y, T_z)$, the basis vectors are:

$$\Delta_{T_x} = \left[ \begin{array}{c} d \cdot f_x \\ 0 \end{array} \right], \quad \Delta_{T_y} = \left[ \begin{array}{c} 0 \\ d \cdot f_y \end{array} \right], \quad \Delta_{T_z} = \left[ \begin{array}{c} d \cdot (c_x - u) \\ d \cdot (c_y - v) \end{array} \right]$$  \hspace{1cm} (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:

$$\Delta_{R_x} = \left[ \begin{array}{c} f_x (u - c_x)(v - c_y) \\ f_y (u - c_x)^2 + f_z (v - c_y)^2 \end{array} \right]$$  \hspace{1cm} (5)

$$\Delta_{R_y} = \left[ \begin{array}{c} f_x (u - c_x)^2 + f_z (v - c_y)^2 \\ f_y (u - c_x)(v - c_y) \end{array} \right]$$  \hspace{1cm} (6)

$$\Delta_{R_z} = \left[ \begin{array}{c} f_y (v - c_y) \\ f_z (c_x - u) \end{array} \right]$$  \hspace{1cm} (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:

$$\mathcal{B}_\text{camera} = \{\Delta_{T_x}, \Delta_{T_y}, \Delta_{T_z}, \Delta_{R_x}, \Delta_{R_y}, \Delta_{R_z}\}.$$  \hspace{1cm} (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.
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):

$$\Delta_{R_x} = f_y \Delta_{R^1_x} + \frac{1}{f_y} \Delta_{R^2_x}, \quad \Delta_{R_y} = f_x \Delta_{R^1_y} + \frac{1}{f_x} \Delta_{R^2_y}$$

where

$$\Delta_{R^1_x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \Delta_{R^2_x} = \begin{bmatrix} (u - c_x)(v - c_y) \\ (v - c_y)^2 \end{bmatrix}, \quad \Delta_{R^1_y} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \Delta_{R^2_y} = \begin{bmatrix} (u - c_x)^2 \\ (u - c_x)(v - c_y) \end{bmatrix}.$$  \hspace{1cm} (10)  

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

$$\mathcal{B}_m = \{m \Delta \mid \Delta \in \mathcal{B}_{\text{camera}}\}.$$  \hspace{1cm} (12)

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

$$\mathcal{B}_{\text{translation}} = \{m \Delta_{T_x}, m \Delta_{T_y}, m \Delta_{T_z}\}.$$  \hspace{1cm} (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 $\phi(u, v) \in \mathbb{R}^4$. 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 $M \in \mathbb{R}^{3 \times A}$ 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 $M\phi$, and the flow $\Delta_{M\phi}$ due to object and camera translation is:

$$\Delta_{M\phi} = (M\phi) \cdot \begin{bmatrix} \Delta_{T_x} \\ \Delta_{T_y} \\ \Delta_{T_z} \end{bmatrix} = \sum_{i=0}^{A-1} M_{0i} \phi_i \Delta_{T_x} + M_{1i} \phi_i \Delta_{T_y} + M_{2i} \phi_i \Delta_{T_z}.$$  \hspace{1cm} (14)

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

$$\mathcal{B}_{\phi} = \{\phi_i \Delta_{T_x}, \phi_i \Delta_{T_y}, \phi_i \Delta_{T_z} \mid 0 \leq i < A\}.$$  \hspace{1cm} (15)

Projecting into this basis implicitly finds a matrix $M$. To allow for camera rotation too we add the various ($\Delta_R$) to...
our basis, giving a basis parameterized by \((d, \phi)\) with \(3A + 3\) dimensions, or \(3A + 5\) for unknown focal length. Allowing for object rotation is also straightforward (add the \((\phi, \Delta R)\) 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 [42]. 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 [56], 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 \(\times\) 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 [56]. 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

| Method (Dataset) | iBims-1 | NYU Depth V2 |
|------------------|---------|--------------|
|                  | rel \(\downarrow\) | log10 \(\downarrow\) | RMS \(\downarrow\) | \(\sigma_1 \uparrow\) | \(\sigma_2 \uparrow\) | \(\sigma_3 \uparrow\) | rel \(\downarrow\) | log10 \(\downarrow\) | RMS \(\downarrow\) | \(\sigma_1 \uparrow\) | \(\sigma_2 \uparrow\) | \(\sigma_3 \uparrow\) |
| **Supervised by depth** |
| 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) [36] | 0.14    | 0.06         | 0.57     | 0.84 | 0.97 | 0.99     | 0.16    | 0.06            | 0.50     | 0.80 | 0.95 | 0.99     |
| 3DKenBums (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     |
| **Supervised by view synthesis plus sparse depth, using pose from SfM** |
| Single-view MPI (RE10K) [46] | 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     |
| **Supervised by flow reconstruction only** |
| 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, e.g., MINE. See Section 4.1.
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 $\phi$. The spatial gradient of $\phi$ (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 $\phi$ 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 $\phi$, 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.

![Input image](image1.png)  ![Predicted disparity](image2.png)  ![Embedding gradient](image3.png)  ![Embedding PCA first 3 dims](image4.png)

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 [41], 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.
of moving people by watching frozen people. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), pages 4521–4530, 2019.

[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.

[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.

[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.

[30] Nikolaus Mayer, Eddy Ilg, Philipp 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.

[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.

[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.

[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.

[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.

[35] Tal Nir, Alfred M Bruckstein, and Ron Kimmel. Over-parameterized variational optical flow. International Journal of Computer Vision, 76(2):205–216, 2008.

[36] René Ranftl, Katrin Larsinger, 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.

[37] 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.

[38] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, (CVPR), June 2016.

[39] Amnon Shashua. Geometry and Photometry in Three-Dimensional Visual Recognition. PhD thesis, Massachusetts Institute of Technology, 1993.
ing the local appearance manifold. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2007. 3

[54] 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

[55] 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

[56] 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