Upgrading Optical Flow to 3D Scene Flow through Optical Expansion

Gengshan Yang\textsuperscript{1,*}, Deva Ramanan\textsuperscript{1,2}
\textsuperscript{1}Carnegie Mellon University, \textsuperscript{2}Argo AI
\{gengshay, deva\}@cs.cmu.edu

Abstract

We describe an approach for upgrading 2D optical flow to 3D scene flow. Our key insight is that dense optical expansion – which can be reliably inferred from monocular frame pairs – reveals changes in depth of scene elements, e.g., things moving closer will get bigger. When integrated with camera intrinsics, optical expansion can be converted into a normalized 3D scene flow vectors that provide meaningful directions of 3D movement, but not their magnitude (due to an underlying scale ambiguity). Normalized scene flow can be further “upgraded” to the true 3D scene flow knowing depth in one frame. We show that dense optical expansion between two views can be learned from annotated optical flow maps or unlabeled video sequences, and applied to a variety of dynamic 3D perception tasks including optical scene flow, LiDAR scene flow, time-to-collision estimation and depth estimation, often demonstrating significant improvement over the prior art.

1. Introduction

Estimating 3D motion is crucial for autonomous robots to move safely in a dynamic world. For example, collision avoidance and motion planning in a dynamic requirement hinge on such inferences \cite{9, 34, 35, 41}. Many robotic platforms make use of stereo cameras or time-of-flight sensors for which metric distances are accessible. Here, 3D motion can be determined by searching for correspondence over frames, or registration between 3D point clouds. Such active sensing and fixed-baseline stereo methods struggle to capture far-away objects due to limited baselines and sparse sensor readings. In this work, we analyze the problem of dynamic 3D perception with monocular cameras, which do not suffer from baselines or sparse readings.

Challenges However, estimating 3D motion from monocular cameras is fundamentally ill-posed without making assumptions about the scene rigidity: given a particular 2D flow vector, there is an infinite pair of 3D points along two degrees of freedom (obtained by back-projecting two rays for the source and target pixel - see Fig. 3) that project to the same 2D flow. Intuitively, a close-by object that moves slowly will generate the same 2D flow as a far-away object that moves fast.

Prior work Nevertheless, there have been numerous attempts at monocular dynamic scene reconstruction using multi-body SfM and non-rigid SfM \cite{27, 67}. A recent approach \cite{7} attempts to solve the monocular scene flow problem in its generality. Because such tasks are under-constrained, these methods need to rely on strong prior assumptions, either in the form of prior 3D geometry (typically learned from data-driven scenes) or prior 3D motion (typically rigid-body priors) that are difficult to apply to “in-the-wild” footage. Instead, we derive a simple but direct geometric relationship between 3D motion and 2D correspondence that allows us to extract up-to-scale 3D motion.

Why optical expansion? Human perception informs us that changes in the perceived size of an object are an important cue to determine its motion in depth \cite{52, 55}. Indeed, optical expansion is also a well-known cue for biological navigation, time-to-contact prediction, and looming estimation \cite{16}. Inspired by these observations, we propose to augment 2D optical flow measurements with 2D optical expansion measurements: for each pixel in a reference frame, we estimate both a 2D offset and a relative scale change \((u, v, s)\), as shown in Fig. 2. We show that such measurements can be robustly extracted from an image pair, and importantly, resolve half of the fundamental ambiguity in 3D motion estimation. Because optical expansion is a local
pixelwise measurement, we demonstrate that it can be easily incorporated into existing approaches for self-learning of optical flow, increasing the accuracy.

**3D motion from expansion** Specifically, under a scaled orthographic camera projection model, optical expansion is directly equivalent to the motion-in-depth of the non-rotating scene element that projects to the corresponding source and target pixel. This eliminates one degree-of-freedom. When combined with camera intrinsics and optical flow, optical expansion reveals the true direction of the 3D motion, but not its magnitude. Fig. 3 demonstrates that we now know if an object is moving closer toward or away from the camera, but there is still an overall scale ambiguity, which can also be resolved by specifying the depth of one of the point pairs along its back-projected ray.

**Method** To estimate per-pixel optical expansion, we propose an architecture based on local affine transformations. The relative scale ground-truth is obtained from the existing optical flow and 3D scene flow training datasets. We also present a self-supervised approach to learn optical expansion from photometric information of the input images.

**Contributions** We summarize our contribution as follows. (1) We theoretically derive the effectiveness of optical expansion to reduce the ambiguities inherent to monocular scene flow. (2) We propose a neural architecture for normalized scene flow estimation that encodes strong geometry knowledge and leads to better interpretability and generalization. (3) We demonstrate the effectiveness of optical expansion across a variety of benchmark tasks, establishing new SOTA results for optical scene flow, LiDAR scene flow, time-to-collision estimation - while being significantly faster than prior methods, and improving results for self-supervised optical flow. (4) We apply dense optical expansion to two-frame depth estimation and show improvements over triangulation-based methods in numerically-unstable regions near the epipole.

2. Related Work

**Visual correspondence** Visual correspondence dates back to the early work in human visual perception and 3D reconstruction, where it was found point correspondence needs to be solved to perceive depth and 3D structures can be recovered from their projected points [36, 57]. Affine correspondence defines a 2×2 affine transform between the neighborhood of point correspondences, to encode higher-order information about the scene geometry [3, 4, 47]. Similar to affine correspondence, we extract local information of point correspondences to encode rich geometric information about motion-in-depth, but not rotation or shear.

**Scale estimation** The concept of scale changes of visual features is well studied in the context of feature descriptor and matching [5, 32, 43] as well as dense optical flow [45, 60, 62]. In these approaches, scale is often treated as a discrete auxiliary variable for producing better descriptors and feature matches, but not estimated as a continuous quantity at a fine scale. Some other approaches either directly estimate the intrinsic scales by Laplacian filtering [42] or compute the scale changes from the divergence of optic flow fields [10, 61], but give sub-accurate results. Instead, our method produces continuous dense optical expansion reliably in a data-driven fashion. Moreover, the relationship between relative scale and depth changes has been explored for 3D reconstruction [13, 44, 50, 65] as well as collision avoidance in robotics [21, 34, 41]. However, prior methods often focus on object-level scale changes and sparse interest points. Our work extends the concept of relative scale and depth changes to the dense, low-level correspondence tasks of 3D scene flow estimation.

**Monocular dynamic reconstruction** Prior work on monocular 3D motion estimation casts the task as a sub-problem of monocular scene reconstruction, attempting to jointly recover both motion and depth [27, 46, 49, 58]. Because of the ill-posed nature of this problem, they either rely on strong motion priors such as multi-rigid body [46, 58] and as rigid as possible [27], or strong shape priors such as low rank and union-of-subspaces [18, 67]. Those handcrafted priors hallucinate good reconstructions when their
assumptions are met, but in other cases not applicable. On the other hand, when scene elements are piece-wise rigid, we can reconstruct up-to-scale local motions with planar homographies. However, homography estimation is sensitive to noise in 2D correspondences, requiring the use of strong priors to regularize the problem [27]. In this work, we propose a simpler representation of local motion, e.g. optical expansion, which can be reliably estimated from real-world imagery because fewer degrees of freedom are needed to be inferred.

3. Approach

In this section, we first establish the relationship between optical expansion and motion-in-depth under scaled orthographic projection model. Then we derive a direct relationship between motion-in-depth, normalized 3D motion, and scene flow. Finally we propose a neural architecture of learning optical expansion and normalized 3D flow.

3.1. Optical expansion

Here we explicitly derive the relationship between optical expansion, which describes the change of the perceived size of objects, and motion-in-depth. We begin with a simple pinhole camera model that projects a 3D point \( P = (X, Y, Z) \) into an image position \( (x, y) \):

\[
p = (x, y) = \frac{f}{Z} (X, Y),
\]

where \( f \) is the focal length. Under a scaled orthographic camera model, the projection of all points on an object can be computed with an orthographic projection onto a fronto-parallel plane followed by a perspective projection of the plane [20]. Such an approximation is reasonable if the variation in depth of an object is small compared to its distance from the camera. With the physical length of an object being \( L \) and its orientation \( \sigma \) defined as the angle between the surface normal and the camera z-axis, the projected length of the object is then given by

\[
l = \frac{f L}{Z} = \frac{f L \cos \sigma}{Z},
\]

where \( L = L \cos \sigma \) accounts for the foreshortening. Assume the scene is locally rigid and one of the rigid pieces changes its depth across two frames from \( Z \) to \( Z' \), while keeping its physical size \( L \) and orientation \( \sigma \) unchanged. Define the optical expansion \( s \) to be the ratio of its projected lengths \( l' / l \), and define the motion-in-depth \( \tau \) to be the ratio of depths \( Z' / Z \). We can now derive that \( s = 1 / \tau \) assuming 1) a scaled orthographic camera model and 2) the scene elements are not rotating relative to the camera (Fig. 4):

\[
l = \frac{f L}{Z}, l' = \frac{f L'}{Z'} \quad \Rightarrow \quad s = \frac{l'}{l} = \frac{Z}{Z'} = \frac{1}{\tau} \tag{1}
\]

3.2. Normalized scene flow

In the last section, we showed that motion-in-depth \( \tau \) can be computed from optical expansion \( s \) for a scaled orthographic camera model. In this section, we show that motion-in-depth \( \tau \) can be combined with camera intrinsics \( K \) to compute a normalized 3D scene flow vector.

Given camera intrinsics \( K \), for a 3D point changing its position from \( P \) to \( P' \), we have

\[
P = \lambda K^{-1} \tilde{p}, \quad P' = \lambda' K^{-1} \tilde{p}',
\]

where \( \tilde{p} \) and \( \tilde{p}' \) are homogeneous 2D coordinates with the last coordinate being 1, and \( \lambda, \lambda' \) are scale factors. Because the last row of the intrinsic matrix \( K \) is \((0,0,1)\), scale factors are directly equal to the depth of each point: \( \lambda = Z \) and \( \lambda' = Z' \).

Following prior work [39], we model scene flow as 3D motion vectors relative to the camera, which factorizes out the camera motion. The scene flow \( t \) is then computed as:

\[
t = P' - P = K^{-1} (Z' \tilde{p}' - Z \tilde{p}) = ZK^{-1} \left[ \tau (\tilde{u} + \tilde{p}) - \tilde{p} \right] \quad \text{where} \quad \tilde{u} = \tilde{p}' - \tilde{p} = ZK^{-1} \left[ (\tau - 1) \tilde{p} + \tau \tilde{u} \right] = \bar{Z} \tilde{t} \quad \text{where} \quad \tilde{t} = K^{-1} \left[ (\tau - 1) \tilde{p} + \tau \tilde{u} \right] \tag{2}
\]

We denote \( \tilde{t} \) as the “normalized scene flow”, which is a vector pointing in the direction of the true 3D scene flow. It can be “upgraded” from 2D flow \( \tilde{u} \) knowing motion-in-depth \( \tau \) and camera intrinsics \( K \). When augmented with the true depth of the point in either frame \( Z \) or \( Z' \) (following an analogous derivation to the above), normalized scene flow can be further “upgraded” to the true 3D scene flow.

3.3. Learning normalized scene flow

In this section we introduce a network architecture for optical expansion and normalized scene flow estimation, and describe ways of learning optical expansion, either in a supervised fashion, or with self-supervised learning.
Figure 5. Network architecture for estimating normalized scene flow. 1) Given two consecutive images, we first predict dense optical flow fields using an existing flow network. 2) Then we estimate the initial optical expansion with a local affine transform layer, which is refined by a U-Net architecture taking affine fitting error and image appearance features as guidance [25]. 3) To correct for errors from the scaled-orthographic projection and rotation assumptions, we predict the difference between optical expansion and motion-in-depth with another U-Net. Finally, a dense normalized scene flow field is computed using Eq. 2 by combining \((u, v, \tau)\) with camera intrinsics \(K\).

**Network** We separate the task of estimating normalized scene flow into three sequential steps: (1) optical flow estimation, where the \((u, v)\) component is predicted from an image pair, (2) optical expansion estimation, where the optical expansion component \(s\) is estimated conditioned on the optical flow, and (3) motion-in-depth estimation \(\tau\), where the optical expansion is refined to produce correct outputs for a full perspective camera model. Finally, normalized scene flow can be computed given camera intrinsics. We design an end-to-end-trainable architecture for the above steps, as shown in Fig. 5. An ablation study in Sec. 5 discusses different design choices that affect the performance.

**Local affine layer** To extract dense optical expansion over two frames, we propose a local affine layer that directly computes the expansion of local 3x3 patches over two frames, as described in the three following steps:

1) **Fit local affine motion models.** Given a dense optical flow field \(u\) over a reference frame and a target frame, we fit a local affine transformation \(A \in \mathbb{R}^{2x2}\) [3] for each pixel \(x_c = (x_c, y_c)\) over its 3x3 neighborhood \(N(x_c)\) in the reference image by solving the following linear system:

\[
(x' - x_c') = A(x - x_c), \quad x \in N(x_c),
\]

where \(x' = x + u(x)\) is the correspondence of \(x\).

2) **Extract expansion.** We compute optical expansion of a pixel as the ratio of the areas between the deformed vs original 3x3 grid: \(s = \sqrt{\text{det}(A)}\).

3) **Compute fitting errors.** We compute the residual \(L_2\) error of the least-squares fit from Eq. 3 (indicating the confidence of the affine fit) and pass this in as an additional channel to the optical refinement network.

Crucially, we implement the above steps as dense, pixel-wise, and differential computations as Pytorch layers that efficiently run on a GPU with negligible compute overhead.

**Learning expansion (supervised)** To train the optical expansion network that predicts \(s\), one challenge is to construct the optical expansion ground-truth. The common solution of searching over a multi-scale image pyramid is infeasible because it gives sparse and inaccurate results. Instead, we extract expansion from the local patches of optical flow fields [10, 61]. Specifically, for each pixel with optical flow ground-truth, we fit an affine transform over its 7x7 neighborhood and extract the scale component, similar to the local affine layer. Pixels with a high fitting error are discarded. In practice, we found optical expansion ground-truth can be reliably computed for training, given the high-quality optical flow datasets available [2, 8, 12, 26, 39, 37].

**Learning motion-in-depth** To train the motion-in-depth network that predicts \(\tau\), we use existing 3D scene flow datasets, from which the ground-truth motion-in-depth can be computed as the ratio between the depth of corresponding points over two frames,

\[
\tau^*(x) = \frac{Z''(x + u^*(x))}{Z'(x)}
\]

where \(Z'\) and \(Z''\) are the ground-truth depth in the reference
and target frame respectively, and \( u^* \) is the ground-truth optical flow.

**Losses** We empirically find that supervised learning of optical expansion yields better performance than self-supervised learning (245 vs 336 in log-\( L_1 \) error, as in Tab. 5 and Tab. 6), and therefore supervised learning is used throughout experiments in Sec. 4.1-4.3. Here, multi-task \( L_1 \) losses are used to train the optical expansion (\( s \)) and motion-in-depth (\( \tau \)) networks jointly:

\[
L = \sum_x |\sigma_s(x) - \log s^*(x)| + |\sigma_\tau(x) - \log \tau^*(x)|,
\]

where \( \sigma_s \) and \( \sigma_\tau \) are the predicted log-scale expansion and motion-in-depth, and \( s^* \) and \( \tau^* \) are ground-truth labels. The loss is summed over pixels with valid labels. We experimented with stagewise training, but found joint end-to-end training simpler while performant.

## 4. Experiments

We first evaluate our method on 3D scene perception tasks including optical scene flow, LiDAR scene flow and time-to-collision estimation. We then show results of self-supervised training of optical expansion models. Finally, we conclude with qualitative results that apply optical expansion for rigid depth estimation during forward or backward translational camera movements that are difficult for traditional structure-from-motion.

**Setup** We freeze the pre-trained optical flow network \([64]\) and train our normalized scene flow model on Driving, Monkaa, and KITTI-15 \([37, 39]\). For KITTI, we split the original 200 images with ground-truth into train and validation sets. Specifically, for optical scene flow, we select every 5 images for validation, and add the rest 160 images for training; while for LiDAR scene flow, we follow the split of MeteorNet \([30]\) and use the first 100 out of 142 images with LiDAR point clouds for training, and the rest for validation. We follow a two-stage training protocol \([22]\), and empirically choose a larger learning rate of 0.01. Pre-training on Driving and Monkaa takes 60k iterations, and fine-tuning on KITTI takes 30k iterations.

### 4.1. Optical scene flow

We first compare to baselines on KITTI-15 validation set, where the standard metrics of scene flow and error of log motion-in-depth (MiD) are used \([39]\).

Table 1. Scene flow estimation on KITTI-15 validation set. D1, D2, Fl, and SF measure the percentage error of disparity, optical flow and overall scene flow prediction respectively. MiD measures the L1 error of log motion-in-depth \( \log(D_2/D_1) \) scaled by 10,000X. Monocular methods are listed at the top, while stereo-based methods are listed below. Methods with \( ^\dagger \) use validation data to train. Our method outperforms the monocular baselines by a large margin, and beats the stereo baselines in terms of MiD.

<table>
<thead>
<tr>
<th>Method</th>
<th>D1</th>
<th>Fl</th>
<th>D2</th>
<th>SF</th>
<th>MiD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delta ([23])</td>
<td>14.51</td>
<td>6.00</td>
<td>78.87</td>
<td>83.26</td>
<td>2237</td>
</tr>
<tr>
<td>Warp+copy ([53])</td>
<td>14.51</td>
<td>6.00</td>
<td>27.73</td>
<td>31.16</td>
<td>623</td>
</tr>
<tr>
<td>Ours</td>
<td>14.51</td>
<td>6.00</td>
<td>16.71</td>
<td>19.65</td>
<td>75</td>
</tr>
<tr>
<td>FlowNet3 ([23])</td>
<td>6.95</td>
<td>32.41</td>
<td>20.89</td>
<td>37.09</td>
<td>537</td>
</tr>
<tr>
<td>^FlowNet3-ft ([23])</td>
<td>1.51</td>
<td>7.36</td>
<td>4.70</td>
<td>9.60</td>
<td>217</td>
</tr>
<tr>
<td>PRSM ([59])</td>
<td>4.05</td>
<td>8.32</td>
<td>7.52</td>
<td>10.36</td>
<td>124</td>
</tr>
<tr>
<td>OSF ([40])</td>
<td>3.98</td>
<td>8.95</td>
<td>7.69</td>
<td>10.16</td>
<td>115</td>
</tr>
</tbody>
</table>

Our solution Since our expansion network only provides motion-in-depth over two frames, to generate the full scene flow vector, we use an off-the-shelf monocular depth estimation network MonoDepth2 \([17]\) to predict \( d_1 \), which is the disparity of frame one. To predict \( d_2 \), the disparity of frame one pixels that have moved to frame two, we simply divide \( d_1 \) by the predicted motion-in-depth.

**Validation performance** To compute \( d_2 \), Schuster et al. \([53]\) warp the second-frame disparity maps to the first frame using forward flow, without dealing with occlusions; we consider a stronger baseline that further copies the disparity of out-of-frame pixels from the first frame, denoted by “Warp+copy”. Following FlowNet3 \([23]\), we also train a refinement network to hallucinate the disparities of the regions occluded in the second frame. As for monocular scene flow, shown in the first group of Tab. 1, our method outperforms baselines by a large margin.

We further consider baselines using stereo cameras to estimate the metric depth at both frames: PRSM \([59]\) and OSF \([40]\) are stereo-based methods that break down the image into rigid pieces and jointly optimize their depth and 3D motion. To evaluate MiD, we simply divide their predicted \( d_2 \) by \( d_1 \). As a result, our method achieves the lowest error in terms of MiD, reducing the error by 10X for monocular baselines, and outperforming the stereo baseline by a large margin (115 v.s. 75). A visual example is shown in Fig. 6. This demonstrates the effectiveness of modeling optical expansion, which models the structure of the scene to infer optical flow and depth. When evaluating on KITTI validation set, we empirically find that supervised learning yields better performance than self-supervised learning (245 vs 336 in log-\( L_1 \) error).

Figure 6. Results for image “000105” in the KITTI val set. Top: Motion-in-depth between two frames where bright indicates points moving towards the camera; bottom: error maps of motion-in-depth. Our method predicts more accurate motion-in-depth than the baselines.
Table 2. Scene flow estimation on KITTI-15 benchmark foreground pixels. All metrics are errors in percentage shown for the foreground pixels. The best among the same group are bolded, and the best among all are underlined. Monocular methods are listed at the top, while stereo-based methods are listed below.

<table>
<thead>
<tr>
<th>Method</th>
<th>D1</th>
<th>D2</th>
<th>Fl</th>
<th>SF</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mono-SF [7]</td>
<td>26.94</td>
<td>32.70</td>
<td>19.64</td>
<td>39.57</td>
<td>41</td>
</tr>
<tr>
<td>Ours-mono</td>
<td>27.90</td>
<td>31.59</td>
<td>8.66</td>
<td>36.67</td>
<td>0.2</td>
</tr>
<tr>
<td>PRSM [59]</td>
<td>10.52</td>
<td>15.11</td>
<td>13.40</td>
<td>20.79</td>
<td>300</td>
</tr>
<tr>
<td>DRISF [33]</td>
<td>4.49</td>
<td>9.73</td>
<td>10.40</td>
<td>15.94</td>
<td>0.75</td>
</tr>
<tr>
<td>Ours-stereo</td>
<td>3.46</td>
<td>8.54</td>
<td>8.66</td>
<td>13.44</td>
<td>2</td>
</tr>
</tbody>
</table>

* The expansion and motion-in-depth networks take 15ms for KITTI-sized images on a TITAN Xp GPU, giving a total run time of 200ms together with flow. Also notice that both PRSM and Mono-SF run on a single-core CPU, and could possibly be parallelized for a better speed.

relative scale change via optical expansion.

**Test performance (fg objects)** We then evaluate our method on scene flow prediction for foreground objects on KITTI-15 benchmark, as shown in Tab. 2. We first compare against Mono-SF, the only monocular scene flow method on the benchmark. It formulates monocular scene flow estimation as an optimization problem, and use probabilistic predictions of a monocular depth network as one energy term. Notice although our disparity error D1 is similar to Mono-SF, we obtain better D2 and SF metrics, which indicates that our prediction of normalized scene flow is more accurate.

Our method of estimating motion-in-depth and $d_2$ is also applicable to stereo scene flow, where we directly take GANet [66], the SOTA method on D1 metric, to predict the disparity of the first frame $d_1$. To obtain $d_2$, we divide the $d_1$ with estimated motion-in-depth as before. As a result, we obtained the SOTA accuracy on foreground depth change $D2$ and scene flow $SF$, which further demonstrates the effectiveness of our method for upgrading optical flow to 3D scene flow. In comparison, we effectively reason about relative depth change at a low cost (15ms), instead of explicitly computing the disparity at frame two. This gives us improved accuracy, spatial consistency and reduced latency.

**4.2. LiDAR scene flow**

Given two consecutive LiDAR scans of the scene, the LiDAR scene flow task is defined as estimating the 3D motions of the point clouds. Prior work either register two point clouds by optimization [11], or train a network to directly predict the 3D motion [19, 29, 30].

**Our solution** Practically, LiDAR scans are usually paired with monocular cameras. Therefore, we use such monocular images to predict optical flow and expansion and convert them to normalized scene flow by Eq. 2. To obtain 3D scene flow for the point clouds, we project them onto the image plane and use LiDAR depth to “upgrade” the normalized scene flow to full 3D scene flow.

**Evaluation protocol** We compare with prior work on 42 KITTI validation images, using the evaluation protocol from MeteorNet [30]: raw LiDAR points are projected onto the image plane and the ground-truth 3D flow is constructed from disparity and flow annotations. Methods are scored by 3D end-point-error (EPE, L2 distance between vectors).

**Baselines** Among all the point-based methods, FlowNet3D and MeteorNet are finetuned on the same set of KITTI images as ours, and the numbers are taken from their paper. HPLFlowNet is trained on FlythingThings [37] and we modify their code to run on raw point clouds. ICP-global finds a single rigid transformation that best describes the motion of all the scene points, and does not deal with non-rigid elements. We further consider stereo scene flow methods [23, 40, 59], where the projected LiDAR depth and $\frac{d_1}{\tau}$ are used to determine the depth-wise flow displacements.

**Results** As in Tab. 3, our method trained on synthetic dataset already performs better than all the point-based methods as well as FlowNet3. After fine-tuning on KITTI, it out-performs all the stereo-based methods, except for PRSM, which takes 100X more inference time. Compared to point-based methods where exact 3D correspondence may not exist in the sparse scan, our method estimates normalized scene flow on a much denser pixel grid, which leads to higher precision. A visual example is shown in Fig. 7.

**4.3. Time-to-collision estimation**

Modelling time-to-collision (TTC) is important for robots to avoid collisions and plan the trajectory [9, 14, 34, 35, 41]. Indeed, knowing the motion-in-depth directly tells us the time a point takes to collide with the image plane by $T_c = \frac{Z}{Z - Z'}T = \frac{T}{1 - \tau}$, assuming a constant velocity, where $T$ is the sampling interval of the camera and $\tau$ is motion-in-depth [21].
convert the motion-in-depth estimations to time-to-collision and compare our method with the baselines in Tab. 4.

We treat TTC prediction as a binary classification task where we predict whether the TTC is less than \( 1s, 2s, 5s \) for each pixel [34]. The sampling interval is set as 0.1s and only the points with positive TTC ground-truth are evaluated. We compute the accuracy over 40 KITTI validation images as used in optical scene flow evaluation.

We find that OSF and PRSM perform reasonably well on TTC estimation, which is consistent with their high accuracy on motion-in-depth estimation. Our monocular method outperforms all the baselines for all time intervals, indicating it makes better predictions on possible future collisions.

### 4.4. Self-learning of optical expansion

We explore the task of self-supervised learning of optical flow and expansion. Our network is trained on 6800 images from KITTI depth estimation dataset [56] for 20k iterations, where the sequences that appear in KITTI-15 scene flow training set are excluded. Then we evaluate 40 validation KITTI-15 images as used in optical scene flow.

As for baselines, “Brightness” compares the difference between the intensity values of the reference and target pixel, and “Census” compares between intensity values of a \( K \times K \) patch around the reference pixel and their correspondences. Both methods do not provide supervision signals for optical expansion. Our scale-aware loss provides supervision for optical expansion, and combined with census loss, gives the best performance, as shown in Tab. 5.

\[
Z = \frac{x - \text{FoE}_x}{u} t_{cz} \quad \text{and} \quad Z = \frac{y - \text{FoE}_y}{v} t_{cz},
\]

where \( \text{FoE} = (\frac{t_{cx}}{t_{cz}}, \frac{t_{cy}}{t_{cz}}) \) and \((u, v)\) is the displacement of reference point \((x, y)\) [31]. Notice when only lateral movement exists, the above is equivalent to \( Z = t_{cx}/u \). Motion-in-depth \( \tau \) also tells us the depth via time-to-contact,

\[
Z = \frac{1}{1 - \tau} t_{cz}.
\]

Assuming the errors according to triangulation and time-to-contact are \( \epsilon_{Z_1} \) and \( \epsilon_{Z_2} \) respectively, we have

\[
\epsilon_{Z_1} \sim \frac{1}{|u|^2}, \quad \epsilon_{Z_2} \sim \frac{1}{(1 - \tau)^2}.
\]
which indicates large error occurs when flow is smaller for the triangulation solution, and large error occurs when optical expansion is close to 1 for the time-to-contact solution. Interestingly, it is always the case that for points near FoE where displacements are small, the optical expansion is either greater than one (moving forward) or smaller than one (moving backward) [41], giving robust signals for reconstructing points near the FoE as shown in Fig. 8.

5. Ablation

Setup To demonstrate the advantage of our method over alternatives for estimating optical expansion, we perform an extensive set of diagnostics. For all experiments, we train the network for 20k iterations on Driving and Monkaa with a batch size of 8, and test on the 40 KITTI validation images used in optical scene flow experiments. We also test on the sintel training set, which compared to KITTI, has a batch size of 8, and test on the 40 KITTI validation images used in optical scene flow experiments. We also test on the sintel training set, which compared to KITTI, has extensive set of diagnostics. For all experiments, we train the network for 20k iterations on Driving and Monkaa with a batch size of 8, and test on the 40 KITTI validation images used in optical scene flow experiments.

Comparison to expansion-based options We first remove the residual prediction structure and directly learn to regress the optical expansion from the initial prediction and find the performance drops slightly. Then we investigate the effectiveness of input features. Replacing initial expansion to flow predictions as inputs increases the error by 50.2% on KITTI and 39.8 on Sintel, which shows the initial scale extracted from local affine transform is crucial for estimating the optical expansion. We then replace initial expansion with the reference and warped target image features (by flow) as inputs, and find the error rises by 76.5% on KITTI and 109.6% on Sintel, which indicates it is difficult to learn optical expansion directly from image features. To demonstrate the improvement from the optical expansion network, we evaluate the raw scale component extracted from the local affine transforms, which increases the error by 42.4% on KITTI and 57.8% on Sintel.

Matching over scales We consider a scale matching network baseline that searches for scale over a pyramid of images [45, 60, 62]. At the quarter feature resolution, we discretize the \( s \in [0.5, 2] \) into \( S = 9 \) intervals in log space, and construct a pyramid by scaling the reference image features. Then a 3D cost volume of size \( H/4, W/4, S \) is constructed by taking the dot product between reference feature and target feature pyramid warped by the optical flow prediction. The cost volume is further processed by 3D convolutions and soft-argmin regression following prior work on stereo matching [24, 63]. However, this approach faces a hard time predicting the optical expansion correctly. We posit the signals in raw images feature is not strong enough for the matching network to directly reason about expansions.

6. Discussion

We explore problems of 3D perception using monocular cameras and propose to estimate optical expansion, which provides rich information about relative depth change. We design a neural architecture for optical expansion and normalized scene flow, associated with a set of supervised or self-supervised learning strategies. As a result, significant improvements over prior art on multiple 3D perception tasks are achieved, including LiDAR scene flow, optical scene flow, and time-to-collision estimation. For future work, we think dense optical expansion is a valuable low-level cue for motion segmentation and robot collision avoidance. Moreover, the geometric relationship between optical expansion and normalized scene flow is currently established assuming a scaled orthographic camera model and non-rotating scene elements. Extending it to a perspective camera model with rotating scene elements would be interesting. Finally, background rigidity is a powerful prior for depth and motion estimation, and incorporating it with our local estimates would further improve the performance.

Acknowledgements: This work was supported by the CMU Argo AI Center for Autonomous Vehicle Research. Thanks to Chaoyang Wang and Peiyun Hu for insightful discussions, and friends at CMU for valuable suggestions.

<table>
<thead>
<tr>
<th>Method</th>
<th>KITTI log-L</th>
<th>Sintel log-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>245</td>
<td>78</td>
</tr>
<tr>
<td>w/o residual</td>
<td>255</td>
<td>83</td>
</tr>
<tr>
<td>affine → flow</td>
<td>383</td>
<td>116</td>
</tr>
<tr>
<td>affine → warp</td>
<td>450</td>
<td>174</td>
</tr>
<tr>
<td>Raw affine transform</td>
<td>363</td>
<td>131</td>
</tr>
<tr>
<td>Matching over scales</td>
<td>541</td>
<td>145</td>
</tr>
</tbody>
</table>

Figure 8. Rigid depth estimation with optical flow vs motion-in-depth. (a): overlaid input frames, where the pixel motion is relative small for the marked region near the focus of expansion. (b): distance to the focus of expansion from image coordinates, given by \( ||p - \text{FoE}|| \). (c): flow correspondences visualized by the Middlebury color wheel. (d): depth estimation by triangulation of flow correspondences, where the estimation for the marked region near the focus of expansion is corrupted due to small displacements. (e): motion-in-depth estimation. (f): depth reconstruction by time-to-contact, where the depth estimation near the focus of expansion is more robust than the triangulation method.
7. Supplementary Material

7.1. Two-view SfM with normalized scene flow

**Problem setup** In Sec. 4.5, we discussed using motion-in-depth $\tau$ to estimate depth $Z$ for a static scene when the camera pose is given. Here, we solve for the camera motion $(R_c, t_c)$ and the 3D shape $P = (X, Y, Z)$ jointly given normalized scene flow $\hat{t}$ from two views of a static scene. This can be formulated as an optimization problem, and we want to minimize the 3D re-projection error

$$L(R_c, t_c, Z) = \sum_k w_k d(R_c P_k + t_c, P_k + t_k)^2$$

$$= \sum_k w_k ||R_c P_k + t_c - (P_k + t_k)||^2$$

$$= \sum_k w_k ||R_c (Z_k \tilde{P}_k) + t_c - (Z_k \tilde{P}_k + Z_k \hat{t}_k)||^2$$

$$= \sum_k w_k ||Z_k (R_c \tilde{P}_k - \tilde{P}_k - \hat{t}_k) + t_c||^2,$$

where $k$ is the index of each point, $w$ is the weight assigned to each point, and $\tilde{P} = K^{-1}\tilde{p}$ are normalized 3D coordinates.

**Solution** Empirically, we find coordinate descent gives a robust solution to the above optimization problem. We alternate between pose $(R_c, t_c)$, and scales $\{Z_1, \ldots, Z_K\}$ as follows. Fixing the scale $Z_k$, the problem becomes finding a rigid transformation between two registered point sets. The solution can be described as: 1) align the center of two point clouds, 2) solve for rotation using SVD, and 3) solve for translation [54]. Fixing $(R_c, t_c)$, scales $\{Z_1, \ldots, Z_K\}$ can be obtained by solving a least square problem.

**Results** We initialize the depth to one for all 3D points $Z_k = 1, k \in \{1 \ldots, N\}$, and perform steepest descent for each sub-problem for five iterations. Qualitative results on KITTI and Blackbird [1] are shown in Fig. 9 and Fig. 11.

Figure 9. Results on KITTI sequence “10-03-0034” (not in the training sequences) frame 840, where the scene motion can be described by a rigid transform. From top to bottom: reference image, depth prediction from MonoDepth2 [17], our depth prediction and fitting error (uncertainty). Our method jointly estimates camera pose together with scene geometry, and predicts sharper boundaries.

Figure 10. Normalized scene flow v.s. optical flow. (a)-(b): overlaid images of two consecutive frames. (c)-(d): visualization of normalized scene flow using negative and positive color hemispheres. Notice that normalized scene flow provides information about depth change, where in (c) the front car moving left-inwards is colored green and in (d) the car moving left-outwards is colored blue. (e)-(f): visualization of optical flow using the Middlebury color wheel [2]. In comparison, optical flow is not sensitive to change of depth, where left-moving cars are all bluish, no matter moving towards or away from the camera.
Figure 11. Results on Blackbird dataset [1], which is a unmanned aerial vehicle dataset for aggressive indoor flight. Prior method of estimating the scene geometry either uses sparse point correspondences, for example COLMAP [51], or rely on strong data prior, for example MonoDepth2 [17]. Our method produces dense and reliable depth without strong priors by making use of dense optical expansion.
References


[21] Berthold KP Horn, Yajun Fang, and Ichiro Masaki. Time to contact relative to a planar surface. In IEEE Intelligent Vehicles Symposium. IEEE, 2007. 2, 6


[23] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In ECCV, 2018. 5, 6, 7


[27] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames. In ICCV, 2017. 1, 2, 3


