FrameNet: Learning Local Canonical Frames of 3D Surfaces from a Single RGB Image

Jingwei Huang[1] Yichao Zhou[2] Thomas Funkhouser[3,4] Leonidas Guibas[1,5]

[1]Stanford University [2]University of California, Berkeley [3]Princeton University [4]Google [5] Facebook AI Research

Abstract

In this work, we introduce the novel problem of identifying dense canonical 3D coordinate frames from a single RGB image. We observe that each pixel in an image is the projection of a small surface region in the underlying 3D geometry, where a canonical frame can be identified as represented by three orthogonal axes, one along its normal direction and two in its tangent plane. We propose an algorithm to predict these axes from RGB data. Our first insight is that canonical frames computed automatically with recently introduced direction field synthesis methods can provide training data for the task. Our second insight is that networks designed for surface normal prediction provide better results when trained jointly to predict canonical frames, and even better when trained to also predict 2D projections of canonical frames. We conjecture this is because projections of canonical tangent directions often align with local gradients in images, and because those directions are tightly linked to 3D canonical frames through projective geometry and orthogonality constraints. In our experiments, we find that our method predicts 3D canonical frames that can be used in applications ranging from surface normal estimation, feature matching, and augmented reality.

1. Introduction

In recent years, learning to predict 3D properties from a single RGB image has made great progress. For example, monocular depth estimation [34, 25, 47, 44, 12] and surface normal prediction [10, 45, 3, 28] have improved dramatically. There are many applications for these tasks in scene understanding and robot interaction.

The main challenge in this domain is choosing an appropriate representation of 3D geometry to predict. Zhang *et al.* [52] predict dense surface normals and then use geometric constraints to solve for depth with a global optimization. GeoNet [28] predicts both surface normals and depth and then passes them to a refinement network for further optimization. These methods are clever in their use of geometric constraints to regularize dense predictions. However,

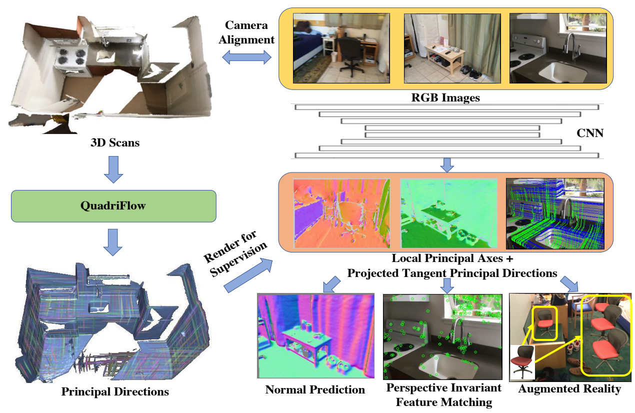

Figure 1. We propose the novel task of predicting dense 3D canonical frames from a single RGB image. We compute the frames from reconstructed meshes using QuadriFlow and render them to images to supervise the task. We train a network that predicts all directions of the frames jointly. We find that predicted tangents provides better surface normals, and are useful for applications like feature matching and augmented reality.

they infer only 2 of the 3 degrees of freedom in a 3D coordinate frame – the rotation in the tangent plane around the surface normal is left unknown. As such, they are missing 3D information critical to many applications. For example, they cannot assist an AR system in placing a picture frame on a wall or a laptop on a table because they don't know the full 3D coordinate frame (including tangent directions) of the wall or table surfaces.

In this work, we propose a novel image-to-3D task: dense 3D canonical frames estimation from a single image (figure 1). This task requires predicting a full 3D coordinate frame defined by the surface normal *and two principal tangent directions* of the surface observed at every pixel in a RGB image. We investigate this task for three reasons. First, we expect that predicting principal tangent directions is easier than predicting normals because they are often aligned with observable patterns in surface textures (e.g., wood grains, fabric weaves, tile seams, etc.) and surface boundaries, which are directly observable in images (figure 2). Second, we expect that joint surface normal and tangent prediction is more robust than normal prediction

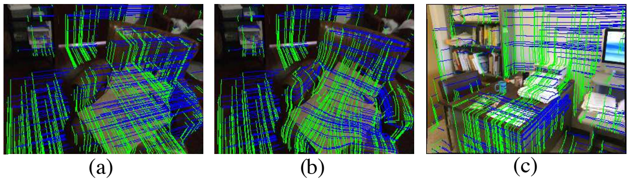

Figure 2. We visualize the directions by picking random seed points and tracing along projected directions in the image. (a) and (b) show the Manhattan and projected principal directions in the same scene, respectively. (c) shows that projected principal directions usually follow the texture directions or object boundaries.

alone due to the regularization provided by orthogonality constraints. Third, we expect that predicting a full canonical 3D coordinate frame at every pixel is useful for many applications, such as augmented reality.

We have implemented an algorithm for this task in a supervised setting. To acquire "ground truth" canonical frames, we leverage data from RGB-D scanning datasets, like ScanNet [9], which provide large sets of images posed within reconstructed 3D meshes. We compute canonical frames on the meshes and render them to the RGB images to produce training data. There are multiple choices for how to define the frames. A simple approach would be to use Manhattan frames; however they reflect only the global scene orientation (figure 2(a)). Instead, we compute locally consistent 4-RoSy canonical frames that follow principal curvatures using the Quadriflow algorithm [21] (figure 2(b)). We find that the surface tangent directions computed this way are consistent with image features and can be learned by a network from 2D data.

The canonical frames are fundamental 3D properties of a scene, as they imply the canonical transformation that maps the 3D surface to the image plane. They provide not only the surface normal, but also canonical tangent directions and their projections onto the image plane. We show that predicting all these directions jointly can improve surface normal estimation, local patch description using SIFT features [26], and allow the insertion of novel objects with correct orientation in augmented reality applications.

Overall, the core contributions of the paper are:

- Identifying an important new 3D vision problem: local canonical frame estimation from RGB images.
- Using projected tangent principal directions to improve canonical frames estimation, outperforming existing works on surface normal estimation.
- Exploiting tangent projected principal directions to compute perspective invariant feature descriptors.
- Inserting new elements in the scene in a manner aware of perspective distortions, for augmented reality.

2. Related Work

3D from Single Image. Estimating 2.5D geometry properties from a single image has become popular in recent years. Traditional methods aim at understanding low-level image information and geometry constraints. For example, Torralba *et al.* [41] exploits the scene structure to estimate the absolute depth values. Saxena *et al.* [33] uses handcrafted features to predict the depth based on Markov random fields. Hoiem *et al.* [18] recovers scene layout guided by the vanishing points and lines. Shi *et al.* [35] estimates the defocus blur and uses it to assist depth estimation.

With the availability of large-scale dataset and the success of deep learning, many methods have been proposed for depth or/and surface normal estimation. For depth estimation, Eigen *et al.* [11] uses CNN to predict indoor depth maps on the NYUv2 dataset. With the powerful backbone network architecture like VGG [37] or ResNet [16], depth estimation can be further improved [13, 46]. DORN [12] proposes a novel ordinary loss and achieves the state-of-the-art in KITTI [14]. For surface normal estimation, Wang *et al.* [45] incorporate vanishing point and layout information in the network architecture. Eigen and Fergus [10] trained a coarse-to-fine CNN to refine the details of the normals. The skip-connected architecture [3] is proposed to fuse hidden layers for surface normal estimation.

Since surface normal and depth are related to each other, another set of methods aimed at jointly predicting both to improve the performance. Wang *et al.* [43] exploits the consistency between normal and depth in planar regions. GeoNet [28] proposes a refinement network to enhance the depth and normal estimation from each other. Zhang *et al.* [52] predict the normal and solve a global optimization problem to complete the depth. We take a further step by jointly estimating all axes of a 3D canonical frame at each pixel, which helps both regularize the prediction through constraints and is useful in applications (see Sec. 5).

Local Canonical Frames Computing local canonical frames on surfaces is a fundamental step for many problems. 3DLite [19] builds canonical frames in fitted 3D planes for color optimizations. GCNN [27] defines local frames with spherical coordinates and applies discrete patch operators on tangent planes. ACNN [6] introduces the anisotropic heat kernels derived from principal curvatures so that it can apply convolutions in canonical frames defined by principal axes. Such canonical frame is also used in Xu *et al.* [48] for nonrigid segmentation, by Tatarchenko *et al.* [39, 20] for semantic segmentation of the 3D scenes. We aim at recognizing such frames from 2D images, and compute them from 3D surfaces to supervise the learning.

TextureNet [20] highlights the challenges of computing robust local canonical frames at planar surface regions, where the principal curvatures are undetermined or highly influenced by noise or uneven sampling. Therefore, it proposes to compute a 4-RoSy orientation field to represent the principal directions. The 4-RoSy orientation field is an important concept in the geometry processing commu-

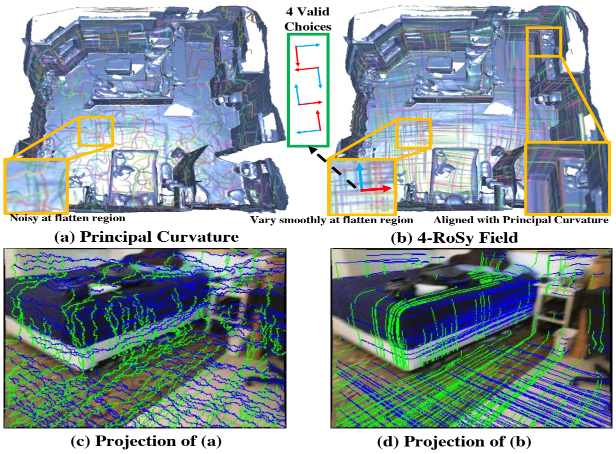

Figure 3. (a) computes the direction field from estimated principal curvatures. Noise exists in both the geometry and the projections in images, as shown in (c). (b) computes the 4-RoSy field using QuadriFlow [21] and produces robust tangent principal directions, as shown in (d) as the projection in the image plane.

nity [30, 23]. The target directions are aligned with the principal curvatures [8, 7], but regularized by additional energy to vary smoothly. This can be achieved by optimizing a nonlinear energy by periodic functions [17, 29] or a mixed-integer representation [30, 5]. In our work, we use QuadriFlow [21] to optimize the 4-RoSy field so that it aligns with the principal curvatures at the curved surface and ensures smoothness in flat regions (where principal directions are ill-defined), as well as robustness to noise.

3. Approach

In this section, we develop our approach for learning local canonical frames from RGB images. First, we discuss the ground truth labeling of canonical frames from 2D images in section 3.1. Then, we discuss the concept of projected tangent principal directions in section 3.2. Finally in section 3.3, we propose several energy terms that encourage the neural network to predict consistent local canonical frames assisted by the projected tangent principal directions. Since we focus on the behavior of the local canonical frames rather than the neural network architecture, we can adopt any neural network that predicts per-pixel features (see experiments in Sec. 4 and 5).

3.1. Local Canonical Frames Generation

To label the canonical frames, we need a dataset with 3D meshes aligned with RGB images to compute frames from geometry and render them to images as ground truth. We choose ScanNet [9] for our experiments.

We compute canonical frames as surface normals and tangent principal directions with the scene geometry. It is straightforward to compute surface normals, but tangent principal directions at flat regions are hard to compute especially in the presence of noise. As visualized in figure 3(a,c), the tangent principal directions can be pretty noisy. To solve this problem, we adopt the 4-RoSy field using QuadriFlow [21] as proposed by TextureNet [20], as shown in figure 3(b,d): This field generates consistent directions which vary smoothly at flatter regions and are aligned with the principal curvatures at curved surfaces. The cross-field is 4-RoSy since there are four valid choices for the tangent principal directions at each vertex. Considering this, we pick any pair of orthogonal tangent vectors in the cross field to represent the principal directions, but we also view the other three alternatives as valid ground truth.

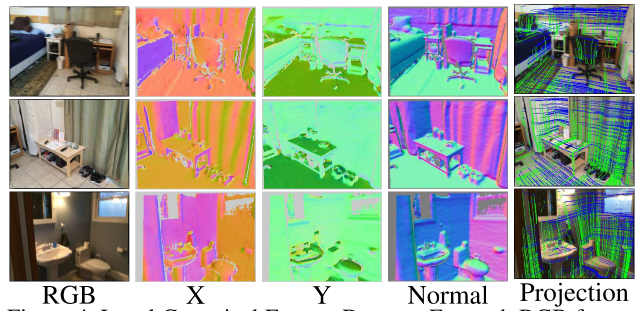

Figure 4. Local Canonical Frames Dataset. For each RGB frame, we render the corresponding tangent principal directions (X and Y) for each pixel. The surface normal can be computed as the cross product of the principal directions.

We store the computed local canonical frames on top of mesh vertices and render them to images after transforming them to the camera space. For each triangle to be rendered, we enumerate the $90^\circ N (N \in \mathbb{Z})$ degree rotations to the tangent principal directions of the last two vertices, so as to align them with the first vertex before the standard rasterization stage. This is to deal with the 4-way rotational ambiguities in the cross field. For each RGB image, we render and save the tangent principal directions as two images, as shown in figure 4 as X and Y. The ground truth normal can be directly computed as the cross product of them.

3.2. Projected Principal Directions

Since we aim to predict 3D principal tangent directions from their appearances into RGB images, we first derive the projective geometry that relates them.

For a pixel $\mathbf{p} = (p_x, p_y)$ in the canonical camera coordinate system, its 3D position of the pixel can be represented as $\mathbf{P} = (p_x d, p_y d, d)$ where d is the depth value. Suppose the pixel has two tangent principal directions $\mathbf{i}$ and $\mathbf{j}$, and we want to analyze their projections. For $\mathbf{i} = (i_x, i_y, i_z)$, we can project a line segment $l(\mathbf{P}, \delta, \mathbf{i})$ that connects endpoints $\mathbf{P}$ and $\mathbf{P} + \delta \cdot \mathbf{i}$ into the image as $l_p(\mathbf{P}, \delta, \mathbf{i})$, which is the offset from $\mathbf{p}$ to the projection of $\mathbf{P} + \delta\mathbf{i}$:

$$l_p(\mathbf{P}, \delta, \mathbf{i}) = \frac{\mathbf{P} + \delta\mathbf{i}}{(\mathbf{P} + \delta\mathbf{i})_z} - \mathbf{p} = (i_x - p_x i_z, i_y - p_y i_z)\frac{\delta}{d + \delta i_z}. \tag{1}$$

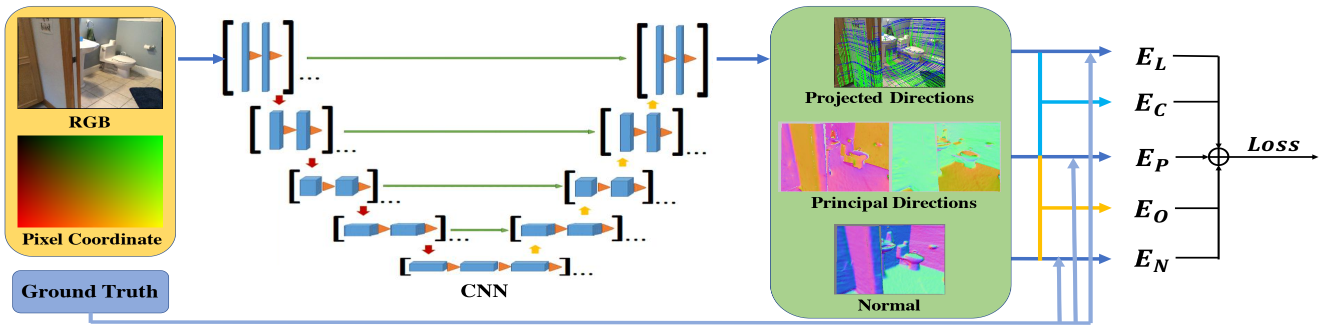

Figure 5. To estimate the local canonical frames, we feed the RGB image and the canonical pixel coordinate map to the network. The output is a 13-dimensional vector for each pixel including two projected tangent principal directions, two 3D tangent principal directions, and one normal vector. We propose a new loss that utilizes the projected directions to improve the estimation of the canonical frames.

We find several ways to translate the projected line segment as a property of the pixel, as shown in equation 2,3,4. The most straightforward idea is to define the property as the projection of the unit 3D line segment from the pixel through the principal directions, represented as

$$l_p^1(\mathbf{P}, \mathbf{i}) := l_p(\mathbf{P}, 1, \mathbf{i}). \tag{2}$$

This simple definition, however, requires a complex mathematical form including the depth value as a hidden information. Thus it could be hard to learn. Another property is the normalized projected principal direction, or

$$l_p^u(\mathbf{P}, \mathbf{i}) := \frac{l_p(\mathbf{P}, \delta, \mathbf{i})}{||l_p(\mathbf{P}, \delta, \mathbf{i})||_2} = \frac{(i_x - p_x i_z, i_y - p_y i_z)}{||(i_x - p_x i_z, i_y - p_y i_z)||_2}. \tag{3}$$

This representation removes the influence of depth as the challenging hidden property. Since the projection usually aligns with the image gradients, it can be as easy as the task of predicting the normalized gradient for the neural network. However, though this is an easy task, the unit projected direction cannot determine the original 3D direction. As shown in figure 6(a), a 2D direction in an image is corresponding to a plane in the 3D world, in which any 3D direction could be a valid solution. Fortunately, we can simplify the definition as

$$l_p^*(\mathbf{P}, \mathbf{i}) := (i_x - p_x i_z, i_y - p_y i_z). \tag{4}$$

This excludes the influence of the depth and gives enough supervision to the directions in 3D space. Mathematically, given the prediction of $\mathbf{l}_p^*(\mathbf{P}, \mathbf{i}) = (l_x^{\mathbf{i}}, l_y^{\mathbf{i}})$, we can compute direction $\mathbf{i} = (i_x, i_y, i_z)$ by solving the system 5:

$$\begin{cases} i_x - p_x i_z = l_x^{\mathbf{i}} \\ i_y - p_y i_z = l_y^{\mathbf{i}} \\ i_x^2 + i_y^2 + i_z^2 = 1\,. \end{cases} \tag{5}$$

3.3. Joint Estimation

We could train a network to estimate the projected principal directions $\mathbf{i}_p = l_p^*(\mathbf{P}, \mathbf{i})$ and $\mathbf{j}_p = l_p^*(\mathbf{P}, \mathbf{j})$, and directly infer $\mathbf{i}$ and $\mathbf{j}$ according to equation 5 for canonical frames estimation. However, we find that this approach does not lead to a robust canonical frames. Therefore, we propose to jointly estimate the canonical frames as well as the projected tangent principal directions, and enforce their orthogonality and projection consistency with additional soft energy constraints. We expect that the extra constraints will provide a regularization that can help the network learn.

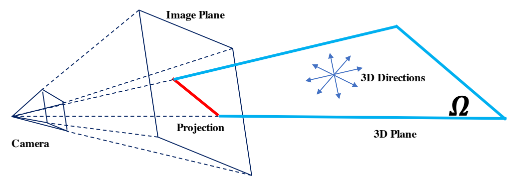

Figure 6. Each projected direction in the image plane (shown in red) corresponds to a 3D plane Ω in the scene. Any 3D direction inside the plane is a valid candidate for this direction.

Our proposed solution is illustrated in figure 5. The neural network can be viewed as a black box function that predicts per-pixel features for the RGB image. Since the projected tangent principal directions relate to the pixel coordinate in the canonical camera, we feed the canonical pixel coordinate together with its RGB values into the network as the input. The network outputs a 13-dimensional vector includes two tangent principal directions $\mathbf{i}$ and $\mathbf{j}$, their 2D projections $\mathbf{i}_p$ and $\mathbf{j}_p$, and the surface normal $\mathbf{n}$.

We propose a set of energies so that projected tangent principal directions can assist the local principal axes estimation. The loss energy E is a linear combination of five energy terms as shown in equation 6,

$$\begin{aligned} E &= \lambda_L E_L + \lambda_P E_P + \lambda_N E_N + \lambda_C E_C + \lambda_O E_O \\ E_L &= \min_{0 \le k \le 4} ||[\mathbf{i}_p, \mathbf{j}_p] - R_k([\mathbf{i}_p^{gt}, \mathbf{j}_p^{gt}])||_2^2 \\ E_P &= \min_{0 \le k \le 4} ||[\mathbf{i}, \mathbf{j}] - R_k([\mathbf{i}^{gt}, \mathbf{j}^{gt}])||_2^2 \\ E_N &= ||N - N^{gt}||_2^2 \\ E_C &= ||l_p^*(\mathbf{i}) - \mathbf{i}_p||_2^2 + ||l_p^*(\mathbf{j}) - \mathbf{j}_p||_2^2 \\ E_O &= ||N - \mathbf{i} \times \mathbf{j}||_2^2\,, \end{aligned} \tag{6}$$

where $R_1([\mathbf{a},\mathbf{b}]) = [-\mathbf{b},\mathbf{a}]$ and $R_k = R_1 \circ R_{k-1}(k>1)$.

Specifically, E_L measures the distance between the predicted tangent principal directions and the ground truth in the 2D projected space. R_k represents the $90°k$ degree rotation around the normal axis. E_L removes the rotational ambiguity by enumerating the possible $90°k$ rotations and measure the minimum L2 loss among them. Similarly, E_P measures the minimum L2 loss of tangent principal directions in the 3D space, and E_N measures the L2 loss of the surface normal estimation. In order to connect the tangent principal directions to their projections, we design E_C to measure the consistency between the projected predicted directions ($l_p^*(\mathbf{i})$,$l_p^*(\mathbf{j})$) and the predicted one ($\mathbf{i}_p$,$\mathbf{j}_p$) by the network. Finally, we also hope the influence can be propagated to the surface normal, so we add an orthogonality constraint E_O to enforce that the surface normal is orthogonal to the tangent principal directions.

Since all the distances are roughly on the same scale, we set $\lambda_L = \lambda_P = \lambda_N = 1$ to balance the penalty for errors for different vectors. To enforce the system to predict orthogonal canonical frames with consistent 2D projection, we set $\lambda_C = \lambda_O = 5$ in our experiments to provide slightly stronger constraints between network predictions.

4. Evaluation

In this section, we describe a series of experiments to evaluate our method for local canonical frames estimation and do ablation studies using the ScanNet dataset [9]. Unless otherwise specified, we used the DORN architecture [12] as the backbone for the architecture in fig. 5, and we used equation 4 for the projected tangent principal directions, since they gave the best results (see below). The main conclusion of these tests is that jointly predicting the projected tangent directions and enforcing the consistency loss are major contributors to the success of local principal axes and surface normal estimation.

How well can canonical frames be estimated from RGB? Our first experiment simply investigates how well our algorithm can predict the canonical frames. Since this is a new task, there is no suitable comparison to prior work. However, we can still gain insight into the problem by comparing errors in predicted normals, principal tangent principal directions, and projected tangent principal directions. The results in table 1 show that prediction of projected tangent principal directions have least error, surface normals have most error, and tangent principal directions are in the middle. This suggests that predicting tangent directions is less error prone than normals, which should be expected since they largely align with textures and gradients in the input image (figure 7).

3D Frame	mean	median	rmse	11.25°	22.5°	30°
Normal	15.28	8.14	23.36	60.6	78.6	84.7
Principal	12.26	7.88	16.85	63.7	84.3	90.8
Projection	7.55	4.46	11.36	79.8	93.0	96.3

Table 1. Testing mean average error of local principal axes estimation on ScanNet [9]. We evaluate surface normals, tangent principal directions their projections predicted by our network.

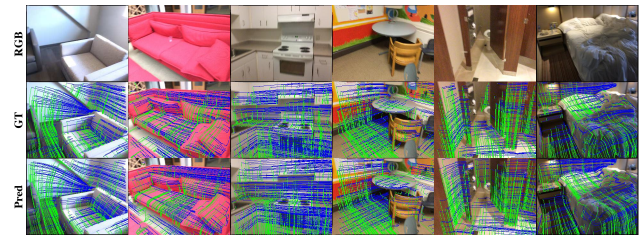

Figure 7. Visualization of the projected principal directions. Our estimation is similar to the ground truth at curved surfaces or texture smooth regions. The predicted directions align with textures and gradients in the input image.

Method	UNet	SkipNet	GeoNet	DORN
Normal	21.08	20.84	20.37	16.42
Normal-YZ	17.49	17.17	16.71	12.51
Normal-XZ	18.05	17.16	17.68	13.00
Normal-XY	29.05	29.71	29.08	22.57
Principal	17.55	15.78	15.41	12.53
Principal-YZ	21.15	21.96	20.61	16.19
Principal-XZ	22.67	21.87	21.57	16.65
Principal-XY	11.47	9.96	9.53	7.55

Table 2. Mean angle errors of normals and tangent principal directions and their projections to three orthogonal planes on ScanNet.

Which frame directions are easiest to predict? To further investigate the relative challenge of predicting different components of the local canonical frames, we perform experiments in which we separately train normals and tangent principal directions in 3D space with L2 losses and evaluate them with mean angle errors of their projections to three planes in camera space, as illustrated in figure 8. The prediction errors and their projected components, listed in table 2, suggest that the errors of the tangent principal directions are less than those of normals, and the projected errors on the image plane are smaller than those on the other two planes for tangent principal directions. This again suggests that the network can predict tangent principal directions better than surface normals, especially for the components projected into the image plane. Interestingly, the projected errors for the normal in the image plane is the largest, which might be because the network learns tangent principal directions in the latent space and propagates the errors from XZ and YZ planes to the image plane by the cross product.

How does each loss contributes to the estimation? We next study how our proposed consistency losses influence

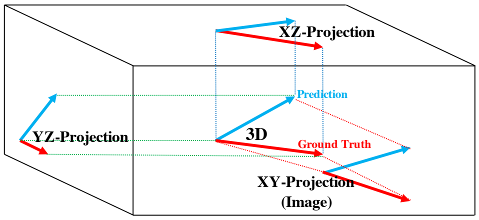

Figure 8. By projecting the directions into XY, YZ, XZ planes in the camera space, we can measure the projected angle error.

Method	UNet	SkipNet	GeoNet	DORN
E_N	21.08	20.36	19.77	16.42
E_N,E_P	21.04	20.45	19.64	16.29
E_N,E_P,E_L	20.62	19.47	19.26	15.45
E_N,E_P,E_L,E_O	20.58	19.43	19.18	15.41
E_N,E_P,E_L,E_C	19.79	19.44	19.02	15.31
All Losses	**19.68**	**19.39**	**18.96**	**15.28**

Table 3. We test mean average angle errors for surface normal predictions with different combination of loss terms on ScanNet. E_L and E_C has major contributions to the improvement, suggesting the importance of the projected principal directions.

the learning process. In table 3, we present the testing mean average angle for surface normals w/o. certain parts of losses during training on ScanNet. We note that by directly predicting all E_N and E_P together, there is already an improvement. The reason could be that the correlation between predicted principal directions and the 3D frames are automatically learned from the data distribution. However, the improvement is minor without predicting the projected principal directions with E_L. With orthogonal or consistency constraints, the performance can be further improved and achieve maximum with both.

Does the method generalize to different networks? To study the generality of our approach, we tested it with different network architectures. Table 3 shows that our joint losses improve performance for all the tested networks including UNet[31], SkipNet[3], GeoNet[28] and DORN[12].

Which definition of projected directions is best? In equation 2 3 4, we propose three choices for projected tangent principal directions. We use UNet [31] to separately train and test them on ScanNet [9] as shown in table 4. The mean angle error for equation 2 is the highest as a complex function related to the depth. The error for equation 4 is only slightly higher than that in equation 3, but equation 4 can explicitly guide the 3D directions with the consistency loss E_C. Therefore, we select equation 4 together with the canonical frames for joint estimation.

ScanNet	mean	median	rmse	11.25°	22.5°	30°
$l_p^1(\mathbf{P},\mathbf{i})$	11.13	7.63	15.00	65.1	86.2	92.5
$l_p^u(\mathbf{P},\mathbf{i})$	**7.35**	**4.38**	**10.94**	**81.2**	**93.6**	**96.7**
$l_p^*(\mathbf{P},\mathbf{i})$	7.56	4.46	11.36	79.8	93.0	96.3

Table 4. Testing mean average error of different choices for projected tangent principal directions on ScanNet dataset.

ScanNet	mean	median	rmse	11.25°	22.5°	30°
UNet	21.08	14.21	28.55	40.8	66.9	76.3
UNet-Ours	19.68	12.43	27.58	46.1	70.6	78.8
SkipNet	20.36	13.74	28.63	45.4	68.2	77.4
SkipNet-Ours	19.39	10.85	27.52	53.2	72.7	79.3
GeoNet	19.77	11.34	28.51	49.7	70.4	77.7
GeoNet-Ours	18.96	9.84	27.29	54.6	73.5	80.1
DORN	16.42	8.64	24.94	58.7	76.7	82.9
DORN-Ours	**15.28**	**8.14**	**23.36**	**60.6**	**78.6**	**84.7**

Table 5. Evaluation on Surface Normal Predictions. We train and test our algorithm with different network architectures on the ScanNet [9] dataset. Assisted by our joint loss, the performances of all networks are improved.

5. Applications

In this section, we investigate whether the estimation of local canonical frames is useful for applications. We first study surface normal estimation, a direct application of our method. In addition, we study how 3D canonical frames can be utilized for perspective invariant feature descriptors and augmented reality.

5.1. Surface Normal Estimation

Test on ScanNet We first compare the performance of our surface normal estimation with state-of-the-art methods on ScanNet [9]. We use our approach to train four networks and evaluate them according to ground truth provided by RGBD. Table 8 shows the results for all networks including UNet [31], SkipNet [3], GeoNet [28] and DORN [12]. With the assistance of the projected tangent principal directions, the normal prediction is better for all architectures.

Figure 13 visualizes the normals predicted using DORN with and without our method. With our approach, the errors are smaller especially at object boundaries, possibly because of the additional supervision given by the projected tangent principal directions.

Test on NYUv2 We test different versions of our network on NYUv2 [11] as a standard evaluation dataset. Since NYUv2 does not provide reconstructed 3D meshes, we cannot get ground truth 3D frames. Therefore, we train the network on ScanNet datasets and directly test on NYUv2, as shown in Table 9. Note that GeoNet-origin [28] is specifically trained and tested on NYUv2 and is the current state-of-the-art method on normal estimation for that dataset. Other rows are networks trained with and without our joint

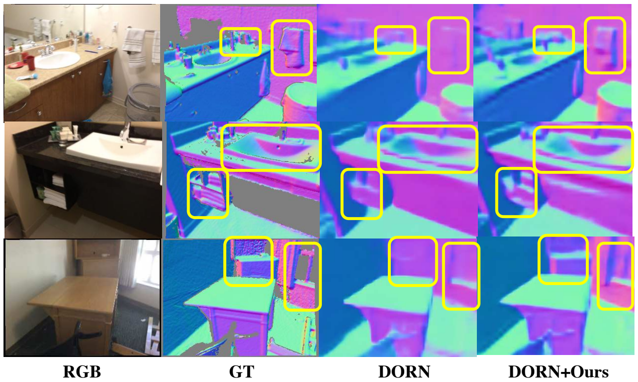

Figure 9. Visual comparison of the results. With our joint loss, the predicted surface normals produce less errors and more details.

NYUv2	mean	median	rmse	11.25°	22.5°	30°
GeoNet-origin	19.0	11.8	26.9	48.4	71.5	**79.5**
ScanNet	mean	median	rmse	11.25°	22.5°	30°
UNet	23.46	17.58	29.90	29.9	60.9	72.7
UNet-Ours	22.09	15.45	29.26	36.9	64.5	74.9
SkipNet	22.27	14.25	30.60	42.0	64.8	73.5
SkipNet-Ours	20.68	13.42	28.33	46.3	67.4	76.0
GeoNet	22.02	14.55	29.79	40.7	64.9	73.9
GeoNet-Ours	20.22	13.23	28.19	47.9	68.0	76.4
DORN	19.12	11.60	27.06	49.0	70.6	78.5
DORN-Ours	**18.63**	**11.16**	**26.61**	**50.2**	**71.6**	**79.5**

Table 6. Normal prediction on NYUv2 [11]. GeoNet-origin trained and tested on NYUv2 [28]. DORN-Ours trained on ScanNet performs best among all.

losses on ScanNet and tested on NYUv2.

Although GeoNet performs worse than GeoNet-origin by training only on ScanNet without fine-tuning, we still achieve better performance with the DORN [12] architecture and our loss (DORN-Ours). Moreover, all networks show better performance with our loss, implying a robust advantage of our joint estimation.

5.2. Keypoint Matching

Predicting local transformations is important for keypoint feature matching [26, 4, 40, 15, 51, 36, 49]. For example, SIFT [26] estimates scale and camera-plane rotations to provide invariance to those transformations. Since our network estimates a full local 3D canonical frames, we can additionally estimate a projective warp. Specifically, predicting the pairs of projected tangent principal directions (in equation 4) for pixel $\mathbf{p}$ as $\mathbf{i}_p$ and $\mathbf{j}_p$ the local patch $\mathbb{P}$ is warped to $\mathbb{P}^*$ as shown in equation 7:

$$\mathbb{P}^*(\mathbf{x}) = \mathbb{P}([\mathbf{i}_p, \mathbf{j}_p]\mathbf{x})\,. \tag{7}$$

To investigate this feature, we performed a simple experiment with SIFT [26]. We augmented the standard SIFT descriptor computation to account for perspective warps implied by our predicted canonical frames. Specifically, we

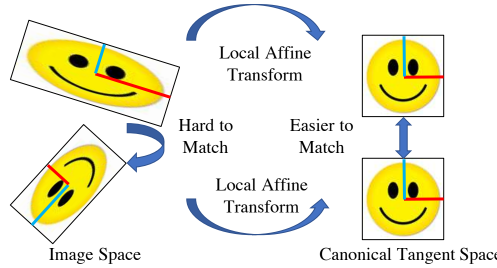

Figure 10. By warping the local patch from the image to the canonical tangent plane of the surface, feature descriptors are invariant to the camera perspectives. Keypoint matching could be improved.

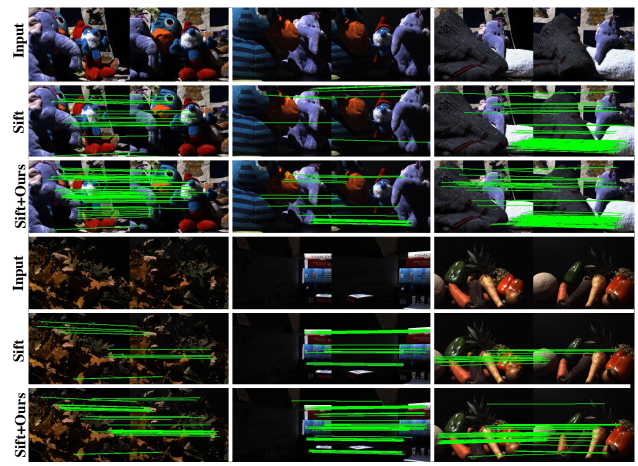

Figure 11. Visualize the matching between SIFT with and without our warping. With our warping, SIFT finds more correct matches.

SURF [4]	ORB [32]	Daisy [40]	BRISK [24]
.224	.127	.262	.193
VGG [22]	MatchNet [15]	DeepDesc [36]	PN-Net [2]
.271	.198	.257	.267
SIFT [26]	ASIFT [50]	LIFT [49]	SIFT+Ours
.272	.265	.317	.335

Table 7. Matching score of descriptors on the DTU dataset.

detect keypoints using SIFT [26], and extract the SIFT descriptors on the warped patch using our estimated local projected tangent principal directions.

To evaluate our modified descriptor, we compare it with other methods on the DTU dataset [1], where scenes are captured with different lighting and viewpoints. We visualize the correct matching produced by SIFT with and without our local image warping in figure 11. As a result, the local image warping reduces the perspective distortions and produce more correct matches. We also test the matching score as "the ratio of ground truth correspondences that can be recovered by the whole pipeline over the number of features proposed by the pipeline in the shared viewpoint region" [49]. As shown in table 7, SIFT [26] outperforms most methods. Since our method additionally reduces the perspective effects using the projected tangent principal directions, we can further improve the SIFT performance. Note that ASIFT [50] also shares the limitation of SIFT [26] to different viewpoints, and extracts keypoints from the image with various affine transforms. Therefore, they usually provide many more correct matching but also more outliers. That is why the matching score produced by ASIFT [50] is slightly lower than SIFT [26]. However, it sometimes shows better robustness assisted by geometric filters in certain applications.

5.3. Augmented Reality

A particularly compelling application of predicting 3D surface frames is augmented reality – i.e., it enables adding new elements to a scene with appropriate 3D orientations.

Decal Attachment As a simple example, we investigate warping virtual decals added to RGB images based on the estimated 3D frame (first two rows of figure 12). In our experiment, we ask the user to select one pixel in an RGB image to indicate the center point for the decal on a surface. If we assume the surface is planar, we can compute the homography transformation required to align the decal with the scene geometry. Suppose the selected pixel is $\mathbf{p}$ with two estimated principal directions $\mathbf{i}$ and $\mathbf{j}$ and depth d. Then, the center of the pattern (x_c, y_c) is located at $K^{-1}\mathbf{p}\cdot d$ where K is the camera intrinsics. We additionally suppose that the target distance of neighboring pixels of the pattern attached to the scene is $\delta \cdot d$. Then, for pixel (x, y) in the pattern, the homogeneous coordinate in the scene is

$$\mathbf{P}(x,y) = K\cdot(K^{-1}\mathbf{p}\cdot d+\mathbf{i}\cdot(x-x_c)\delta d+\mathbf{j}\cdot(y-y_c)\delta d)\,. \quad (8)$$

Therefore, the homography transform can be inferred as

$$H = K[\delta\mathbf{i}, \delta\mathbf{j}, K^{-1}\mathbf{p} - \delta(x_c\mathbf{i} + y_c\mathbf{j})]\,. \quad (9)$$

Here, δ represents the relative scale of the pattern to the depth of the pixel, which can be controlled by the user. Beyond this point, our local frame even enables deformable pattern attachment on curved surfaces. Similarly, the homogeneous coordinate of any pixel $\mathbf{x}_t$ can be computed as

$$\mathbf{P}(\mathbf{x}_t) = \mathbf{p} + \delta K \cdot \int_{\mathbf{x}_c}^{\mathbf{x}_t} [\mathbf{i}(\mathbf{P}(\mathbf{x})), \mathbf{j}(\mathbf{P}(\mathbf{x}))] d\mathbf{x}\,. \quad (10)$$

We use the simple explicit Euler method to evolve $\mathbf{P}(\mathbf{x})$, where the path of the integration starts from the center, and follows the order guided by the breadth first search, where the expansion is from one pixel to those among its four neighbors which are not yet visited. Several examples of deformable attachment is shown in figure 12. The user can control δ to specify the size of the attached patterns.

Object Placement We can also use the local 3D frame defined by predicted principal axes to render 3D objects into

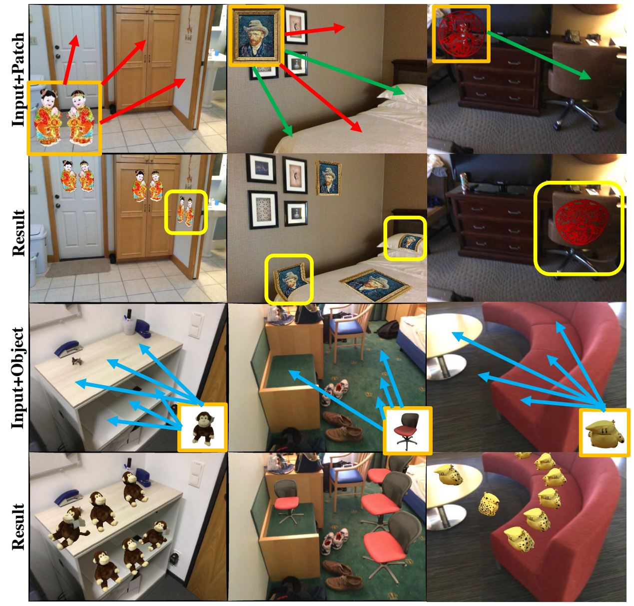

Figure 12. Adding new elements in the scene. We use red arrows to represent rigid attachment, green to represent deformable attachment, and blue to represent object placement.

RGB images, as shown in the last two rows of figure 12. For this application, predicting the full 3D orientation of the scene geometry is critical, so that objects can be planes not only in accordance with the surface normal, but also in the appropriate rotation around the normal (e.g., so that the front is facing the right way). For example, the stuffed animals in the bottom left of figure 12 would appear unnatural if they were facing the wall. This could also eases mixed reality data augmentation for vision tasks, where existing methods require the depth image for plane detection [42].

6. Conclusion

We have proposed the novel problem of densely estimating local 3D canonical frames from a single RGB image. We formulated the problem as a joint estimation of surface normals, canonical tangent directions, and projected tangent directions. We find that this approach leads to superior performance as compared to previous work on normal estimation and other tasks, including local projectively-invariant feature extraction and AR novel object insertion in images. Further study is warranted to investigate what other geometric properties can be predicted from RGB using similar methods and how they can be exploited in application settings.

Acknowledgements

This work is supported in part by a Samsung GRO grant, NSF grant DMS-1546206, a Vannevar Bush Faculty Fellowship as well as a IAS/TUM Hans Fischer Faculty Fellowship.

References

[1] Henrik Aanæs, Anders Lindbjerg Dahl, and Kim Steenstrup Pedersen. Interesting interest points. *International Journal of Computer Vision*, 97(1):18–35, 2012.

[2] Vassileios Balntas, Edward Johns, Lilian Tang, and Krystian Mikolajczyk. Pn-net: Conjoined triple deep network for learning local image descriptors. *arXiv preprint arXiv:1601.05030*, 2016.

[3] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5965–5974, 2016.

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[5] David Bommes, Henrik Zimmer, and Leif Kobbelt. Mixed-integer quadrangulation. In *ACM Transactions On Graphics (TOG)*, volume 28, page 77. ACM, 2009.

[6] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197, 2016.

[7] Frédéric Cazals and Marc Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, 22(2):121–146, 2005.

[8] David Cohen-Steiner and Jean-Marie Morvan. Restricted Delaunay triangulations and normal cycle. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, pages 312–321. ACM, 2003.

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas A Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, volume 2, page 10, 2017.

[10] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[11] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[12] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[13] 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*, pages 740–756. Springer, 2016.

[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.

[15] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 517–526. ACM Press/Addison-Wesley Publishing Co., 2000.

[18] Derek Hoiem, Alexei A Efros, and Martial Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

[19] Jingwei Huang, Angela Dai, Leonidas J Guibas, and Matthias Nießner. 3dlite: towards commodity 3d scanning for content creation. *ACM Trans. Graph.*, 36(6):203–1, 2017.

[20] Jingwei Huang, Haotian Zhang, Li Yi, Thomas Funkhouser, Matthias Nießner, and Leonidas Guibas. Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. *arXiv preprint arXiv:1812.00020*, 2018.

[21] Jingwei Huang, Yichao Zhou, Matthias Nießner, Jonathan Richard Shewchuk, and Leonidas J Guibas. Quadriflow: A scalable and robust method for quadrangulation. In *Computer Graphics Forum*, volume 37, pages 147–160. Wiley Online Library, 2018.

[22] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip Torr, and David Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE transactions on visualization and computer graphics*, 21(11):1241–1250, 2015.

[23] Yu-Kun Lai, Miao Jin, Xuexiang Xie, Ying He, Jonathan Palacios, Eugene Zhang, Shi-Min Hu, and Xianfeng Gu. Metric-driven RoSy field design and remeshing. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):95–108, 2010.

[24] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 IEEE international conference on computer vision (ICCV)*, pages 2548–2555. Ieee, 2011.

[25] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3372–3380, 2017.

[26] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[27] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[28] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pages 283–291, 2018.

[29] Nicolas Ray, Bruno Vallet, Laurent Alonso, and Bruno Levy. Geometry-aware direction field processing. *ACM Transactions on Graphics (TOG)*, 29(1):1, 2009.

[30] Nicolas Ray, Bruno Vallet, Wan Chiu Li, and Bruno Lévy. n-symmetry direction field design. *ACM Transactions on Graphics (TOG)*, 27(2):10, 2008.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[32] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. 2011.

[33] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006.

[34] Evan Shelhamer, Jonathan T Barron, and Trevor Darrell. Scene intrinsics and depth from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 37–44, 2015.

[35] Jianping Shi, Xin Tao, Li Xu, and Jiaya Jia. Break ames room illusion: depth from general single images. *ACM Transactions on Graphics (TOG)*, 34(6):225, 2015.

[36] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[38] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[39] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2018.

[40] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):815–830, 2010.

[41] Antonio Torralba and Aude Oliva. Depth estimation from image structure. *IEEE Transactions on pattern analysis and machine intelligence*, 24(9):1226–1238, 2002.

[42] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. *arXiv preprint arXiv:1901.02970*, 2019.

[43] Peng Wang, Xiaohui Shen, Bryan Russell, Scott Cohen, Brian Price, and Alan L Yuille. Surge: Surface regularized geometry estimation from a single image. In *Advances in Neural Information Processing Systems*, pages 172–180, 2016.

[44] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive o-cnn: a patch-based deep representation of 3d shapes. In *SIGGRAPH Asia 2018 Technical Papers*, page 217. ACM, 2018.

[45] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015.

[46] Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *European Conference on Computer Vision*, pages 842–857. Springer, 2016.

[47] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5354–5362, 2017.

[48] Haotian Xu, Ming Dong, and Zichun Zhong. Directionally convolutional networks for 3d shape segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2698–2707, 2017.

[49] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016.

[50] Guoshen Yu and Jean-Michel Morel. Asift: An algorithm for fully affine invariant comparison. *Image Processing On Line*, 1:11–38, 2011.

[51] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4353–4361, 2015.

[52] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single rgb-d image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–185, 2018.