# ViewSynth: Learning Local Features from Depth Using View Synthesis

Jisan Mahmud[†1]
jisan@cs.unc.edu
Rajat Vikram Singh[2]
singh.rajat@siemens.com
Peri Akiva[†3]
peri.akiva@rutgers.edu
Spondon Kundu[2]
spondon.kundu@siemens.com
Kuan-Chuan Peng[†4]
kpeng@merl.com
Jan-Michael Frahm[1]
jmf@cs.unc.edu

[1] University of North Carolina at Chapel Hill, Chapel Hill, NC
[2] Siemens Corporate Technology, Princeton, NJ
[3] Rutgers University, New Brunswick, NJ
[4] Mitsubishi Electric Research Laboratories, Cambridge, MA

**Abstract**

The rapid development of inexpensive commodity depth sensors has made keypoint detection and matching in the depth image modality an important problem in computer vision. Despite great improvements in recent RGB local feature learning methods, adapting them directly in the depth modality leads to unsatisfactory performance. Most of these methods do not explicitly reason beyond the visible pixels in the images. To address the limitations of these methods, we propose a framework ViewSynth, to jointly learn: (1) viewpoint invariant keypoint-descriptor from depth images using a proposed *Contrastive Matching Loss*, and (2) view synthesis of depth images from different viewpoints using the proposed *View Synthesis Module* and *View Synthesis Loss*. By learning view synthesis, we explicitly encourage the feature extractor to encode information about not only the visible, but also the occluded parts of the scene. We demonstrate that in the depth modality, ViewSynth outperforms the state-of-the-art depth and RGB local feature extraction techniques in the 3D keypoint matching and camera localization tasks on the RGB-D dataset 7-Scenes, TUM RGBD and CoRBS in most scenarios. We also show the generalizability of ViewSynth in 3D keypoint matching across different datasets.

## 1 Introduction

Accurate local feature correspondence matching is a crucial step in many computer vision applications like structure-from-motion and multi-view stereo [1, 15, 36, 37], image retrieval [2, 29, 50], geo-localization [19, 20, 55], camera localization [10, 35, 45], and object pose estimation [13]. Most of these applications utilize RGB based local features. Unlike RGB



[†]This work was done when Jisan Mahmud and Peri Akiva were interns and Kuan-Chuan Peng was a staff scientist at Siemens Corporate Technology.

images, depth images contain 3D information [27], and they are invariant to color, texture, and illumination changes [24, 49]. These properties have encouraged recent studies that use depth images in 3D correspondence matching [12, 25, 27], object pose estimation[13], human pose estimation [28, 48], etc. Depth images are especially suitable for establishing local feature correspondences [12, 25, 27] when high color, texture, or illumination variation is expected, which motivates us to propose a framework for learning keypoints and descriptors from depth images towards the tasks of 3D keypoint matching and camera localization.

Most of the handcrafted feature based [25, 26, 27, 34], and deep-learning based [30, 54] keypoint-descriptor extraction methods take a *detect-then-describe* (*DtD*) approach, where keypoints and descriptors are estimated separately. D2Net [10] shows that a *detect-and-describe* (*DaD*) approach - where detection and description are jointly estimated, and leads to better performance. However, despite the state-of-the-art (SOTA) performance in the RGB modality, [10] is not directly applicable in the challenging depth image modality due to model collapse [52], which we explain in Sec 3.1. Moreover, most keypoint-descriptor learning methods are designed to primarily learn from the commonly visible parts of RGB [10, 30, 33] or depth [12, 13] image pairs. They do not explicitly enforce encoding information about the occluded parts of the scene. In contrast, we propose to explicitly enforce the learned features (which are used to generate keypoints and descriptors) to be able to synthesize the depth from different viewpoints, including the areas occluded in the original viewpoint. Therefore, the learned features are trained to include the information beyond the visible pixels. Sitzmann *et al*. [40] showed that learning 3D-structure-aware scene representation can improve various tasks like few-shot reconstruction, shape and appearance interpolation, novel view synthesis, etc. Inspired by [40], we hypothesize that learning view synthesis of depth images will encourage encoding such 3D-structure-aware information, which would be useful for extracting keypoints and descriptors that can be correctly matched.

To this end, we propose a local feature learning framework **ViewSynth** (Figure 1) which (1) jointly estimates keypoints and descriptors with a *DaD* approach, and (2) explicitly enforces encoding information beyond the visible pixels by learning view synthesis. First, we propose a *Contrastive Matching Loss*, $L_{cm}$, which uses a contrastive loss coupled with the hardest negative sampling [11] to learn viewpoint invariant keypoint-descriptor, while circumventing the model collapse [52] problem. We also propose the *View Synthesis Module* (**VSM**), which takes in a depth image and a relative pose, and synthesizes the depth image from that relative pose, and *View Synthesis Loss*, $L_v$ to train VSM. VSM consists of: the *Grid Transformation Encoder* (**GTE**), which encodes the transformation-related parameters between the images, and the *Depthmap Synthesis Network* (**DSN**), which uses the output of GTE and dense features of one depth image to synthesize the other depth image.

In summary, we make the following contributions: (1) we propose the **ViewSynth** framework, which learns view synthesis using the *View Synthesis Module* (VSM) (composed of the *Grid Transformation Encoder* (GTE) and the *Depthmap Synthesis Network* (DSN)) towards improving keypoint matching, (2) the *Contrastive Matching Loss*, $L_{cm}$, for learning keypoints and descriptors in depth images, (3) the *View Synthesis Loss*, $L_v$, to train VSM. ViewSynth outperforms the SOTA depth [12] and SOTA RGB [10, 33] local feature methods in the 3D keypoint matching and camera localization tasks on RGB-D dataset 7-Scenes [38], TUM [42], and CoRBS [51] datasets, while showing better generalizability across datasets.

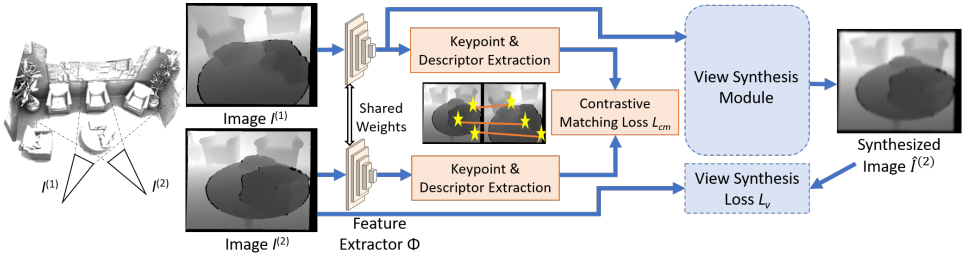


Figure 1: The ViewSynth framework. Dense features, keypoints and descriptors are extracted from depth images $I^{(1)}, I^{(2)}$. *Contrastive Matching Loss* supervises keypoint and descriptor learning. Simultaneously, *View Synthesis Module* trained with *View Synthesis Loss* synthesizes depth image from $I^{(2)}$'s view from $I^{(1)}$'s features.

## 2 Related work

**Sparse local feature learning:** While traditional methods like SIFT [26], SURF [3] and ORB [34] use handcrafted features for extracting keypoint-descriptor, deep-learning based methods [10, 30, 33, 46] outperform the traditional ones by learning features from the images. Most of these methods take a *DtD* approach [3, 6, 8, 26, 30, 34, 54]. In contrast, D2Net [10] and R2D2 [33] propose *DaD* approaches that share all [10], or most [33] parameters between keypoint detection and description, and achieve the SOTA keypoint matching performance in RGB modality. Our experiments show that these methods are either not trainable, or underperform when directly adapted in the depth modality. Moreover, none of these methods explicitly seek to encode information beyond the visible pixels in the images, towards keypoint-descriptor extraction. In contrast, ViewSynth explicitly learns to encode this information by learning view synthesis, and shows that this is beneficial for keypoint matching. Suwajanakorn *et al*. [43] proposed 3D keypoint learning from RGB images via geometric reasoning to obtain a fixed number of keypoints for pose estimation. Their method only detects keypoints without providing their descriptors, but ViewSynth can provide both.

**Learning from depth data:** The reliance on depth data has recently seen a surge in applications like 3D object detection [9, 32], facial emotion recognition [44], people counting [9, 16], activity recognition [18, 24] and human pose estimation [28, 48]. Recently, the authors of [13] learn modality-invariant keypoints between RGB and rendered depth images for object pose estimation. Georgakis *et al*. [12] learn keypoints and descriptors from depth images towards 3D correspondence matching. Most depth image keypoint-descriptor methods either take a *DtD* approach [25, 27, 56], or do not share all parameters between detection and description [12, 13]. D2Net [10] showed that a *DaD* approach that shares all parameters between detection and description outperforms the *DtD* methods, or methods that do not share all parameters. In addition, these methods also do not explicitly seek to encode information about the occluded parts of the images, which ViewSynth is designed to overcome.

**Synthesizing novel views:** The use of view synthesis has been largely focused on generating information missing in current views with known applications in point cloud reconstruction by depth image synthesis [22], depth image super-resolution [41], layered 3D scene inference [47], image inpainting [51, 53] and image-to-image translation [7, 17]. Zhou *et al*. [57] predicts an appearance flow to synthesize novel views from an image. Sitzmann *et al*. [40] learns to implicitly represent a scene in a 3D-structure-aware manner, towards novel view synthesis, shape and appearance interpolation, and few-shot reconstruction. These methods typically do not generate keypoints or descriptors for matching. In

contrast, ViewSynth learns view synthesis in conjunction with learning keypoint-descriptor to improve keypoint matching performance.

# 3 Methodology

Here we describe our proposed depth image local feature learning framework **ViewSynth**. We use a *DaD* technique for learning keypoints and descriptors; while also learning 3D-structure-aware depth image representation using view synthesis.

## 3.1 Learning keypoints and descriptors

Given a depth image $I$, we first use VGG-16 [39] up to the `conv_4_3` layer as a deep feature extractor $\Phi$ to extract features $F = \Phi(I)$. Here, $F \in \mathbb{R}^{h \times w \times f}$ is 8 times downsampled compared to $I$. Here $h$, $w$ and $f$ refer to the height, width and channels of $F$ respectively. Features along location $(i, j)$ are indicated by Keypoints and their descriptors are extracted from $F$ using the joint keypoint detection-description technique described by [10]. Applying $L_2$ normalization to $F$ produces the descriptor at each spatial position, $D_{i,j} = F_{i,j}/||F_{i,j}||_2$. Potential keypoints are detected from $D$ with their respective soft detection scores $S \in \mathbb{R}^{h \times w}$ [10] during training. $S_{i,j}$ indicates how confident the local feature extraction method is in being able to correctly match the keypoint at $(i, j)$ in other images. The hard feature detection technique [10] is used during inference.

We learn local features by training the network $\Phi$ with pairs of depth images with some overlap, and by encouraging it to learn correct keypoint correspondences between depth images. Given a pair of depth images $(I^{(1)}, I^{(2)})$ normalized to $[0, 1]$, their corresponding dense features $F^{(1)}$, $F^{(2)}$, soft keypoint scores $S^{(1)}, S^{(2)}$ and descriptors $D^{(1)}, D^{(2)}$ are first extracted as described earlier. We define $\pi(F^{(j)})$ as the grid of spatial positions of $F^{(j)}$; and $\mathbf{C}_{gt}$ as ground truth correspondences between the points visible in both images based on their 3D world coordinates. For each correspondence $(c_1, c_2) \in \mathbf{C}_{gt}$ such that $c_1 \in \pi(F^{(1)}), c_2 \in \pi(F^{(2)})$, we minimize the *positive descriptor distance*, $p(c_1, c_2) = ||D^{(1)}_{c_1} - D^{(2)}_{c_2}||_2$ to encourage descriptor similarity between correct correspondences. We also maximize *negative descriptor distance*, the descriptor distance between the most confounding incorrect correspondences. For $c_1$, we compute the descriptor distance of the most confounding incorrect correspondence in $I^{(2)}$: $n(c_1, c_2) = min_k ||D^{(1)}_{c_1} - D^{(2)}_{k}||_2$, where $k \in \pi(F^{(2)})$, and $||k - c_2||_2 > \tau$. $\tau$ defines a boundary around each correctly matched keypoint, within which we do not consider any point as a negative match. Following [10], we use $\tau = 4$ pixels. Similarly, we compute $n(c_2, c_1)$, the most confounding incorrect correspondence distance for $c_2$.

D2Net uses a triplet loss to encourage $p(c_1, c_2)$ to be smaller than $\min(n(c_1, c_2), n(c_2, c_1))$ upto some margin. Interestingly, we observe that this loss often led to a model collapse [52], where all descriptors collapsed into a singular representation in earlier phases of training. We presume that the inherent difficulty associated with learning pose invariant representation from often noisy depth data, coupled with a high learning rate, and the hard negative sampling of D2Net led to this phenomenon. We propose to use a contrastive loss [14] to avoid this problem. Unlike the triplet loss, the contrastive loss encourages the network to learn the exact same descriptor for a keypoint across depth images from any viewpoint. This is desirable in ViewSynth, since we want the densely extracted features to encode information in a viewpoint invariant fashion. The contrastive loss for $D^{(1)}_{c_1}$ is:

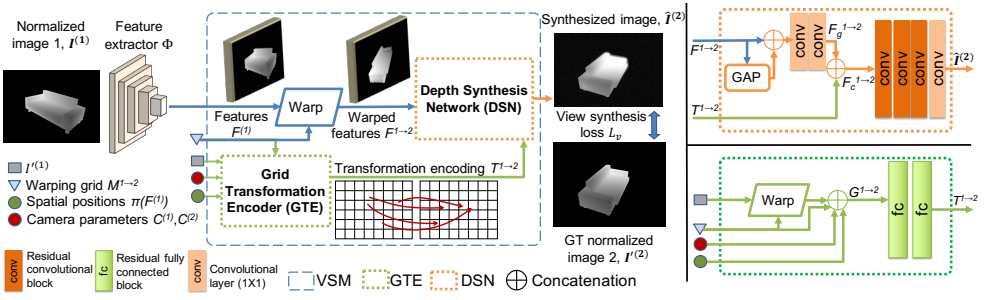


Figure 2: VSM's goal is to synthesize a depth image from the features of another depth image and the relative transformation between the two depth images. Here, VSM takes in the dense representation $F^{(1)}$ of the depth image $I^{(1)}$ and the parameters related to pixel-wise transformation from $I^{(1)}$ to $I^{(2)}$, and synthesizes the normalized representation $\hat{I}^{(2)}$ from the view of $I^{(2)}$. See Sec. 3.2 for details.

$L_c(c_1,c_2) = 0.5p(c_1,c_2)^2 + 0.5\max(0, m - n(c_1,c_2))^2$. Margin $m$ is empirically set to 1.5 for all of our experiments. Similarly, for $D_{c_2}^{(2)}$ we compute $L_c(c_2,c_1)$. Finally, our proposed *Contrastive Matching Loss*, $L_{cm}$ is defined as:

$$L_{cm} = \frac{\sum_{c_1,c_2 \in \mathbf{C}_{gt}} S_{c_1}^{(1)} S_{c_2}^{(2)} \left(L_c(c_1,c_2) + L_c(c_2,c_1)\right)}{\sum_{c_1,c_2 \in \mathbf{C}_{gt}} S_{c_1}^{(1)} S_{c_2}^{(2)}}. \tag{1}$$

$L_{cm}$ is a weighted average of the contrastive loss terms based on the keypoint scores. Minimizing $L_{cm}$ drives the relative scores of the correspondences with lower contrastive loss to get larger, and vice versa. Simultaneously, $L_{cm}$ also drives all contrastive losses to be smaller. Optimization of $L_{cm}$ is done wrt the parameters of feature extraction network Φ.

Although Georgakis *et al.* [12] use contrastive loss for learning descriptors - they learn keypoints and descriptors separately, while our method learns them jointly. Another key difference is: for each keypoint, [12] only considers a single match on the other image, and penalizes depending on whether the match was correct or not. In contrast, our method considers *all* possible matches on the other image, and decides the correct match and the most confounding incorrect match, which better lets our method have dissimilar descriptors for different keypoints.

## 3.2 Learning from view synthesis

Inspired by [40, 43], we hypothesize that learning 3D-structure-aware depth image representation using view synthesis can assist in learning local features more suitable for correct matching. Intuitively, learning to synthesize views from unseen viewpoints can assist the keypoint detection and description process by encoding 3D structure information of the scene. Consider depth images $I^{(1)}$ and $I'^{(2)}$ in Figure 2. $I'^{(j)}$ is the 8 times downsampled version to $I^{(j)}$. Some surfaces of the sofa observed in $I^{(1)}$ are occluded in $I'^{(2)}$. We propose the *View Synthesis Module* (VSM) (Figure 2) to learn to synthesize $I'^{(2)}$, from $F^{(1)}$, and $T^{1\to2}$: the relative transformations from $I^{(1)}$ to $I^{(2)}$. We hypothesize that learning view synthesis will encourage $F^{(1)}$ to encode information beyond the visible pixels of the depth image, which can improve keypoint matching accuracy. We summarize a high-level overview of VSM -

- VSM first warps the features of $I^{(1)}$ onto $I^{(2)}$'s perspective.
- GTE encodes the $I^{(1)}$ to $I^{(2)}$ pixel-level transformation-related parameters along each spatial position.
- DSN then uses the warped features of $I^{(1)}$ and the transformation features from GTE to predict the estimated depth intensity along each pixel of $I^{(2)}$.
- ViewSynth is end-to-end differentiable, and learning view synthesis encourages the feature extractor $\Phi$ to learn to encode information about the visible and occluded parts of $I^{(1)}$.

During training, we use depth image pairs $(I^{(1)}, I^{(2)})$, with camera parameters $C^{(1)}$, $C^{(2)}$ respectively to learn view synthesis. $C^{(j)}$ refers to the intrinsic and extrinsic parameters of camera $j$. We compute the mapping grid $M^{1\to2}$, which indicates where each spatial location of $F^{(2)}$ is located in $F^{(1)}$. $M^{1\to2}$ is mathematically defined on $C^{(1)}$, $C^{(2)}$ and computed using $I'^{(2)}$. VSM uses $I'^{(2)}$ only to compute the mapping grid $M^{1\to2}$, and not directly as an input to any part of the neural network of VSM. Instead, VSM takes $F^{(1)}$, $\pi(F^{(1)})$, $I'^{(1)}$, $C^{(1)}$, $C^{(2)}$ as input, and utilizes $M^{1\to2}$ to synthesize the normalized depth image from $C^{(2)}$'s pose: $\hat{I}^{(2)}$.

VSM first uses the mapping grid $M^{1\to2}$ to warp the dense feature representation $F^{(1)}$ onto the image space of $F^{(2)}$ to obtain the warped representation, $F^{1\to2} = \text{Warp}(F^{(1)}; M^{1\to2})$. The *Grid Transformation Encoder* (GTE) then computes $T^{1\to2}$. Finally, the *Depthmap Synthesis Network* (DSN) uses $F^{1\to2}$ and $T^{1\to2}$ to synthesize $\hat{I}^{(2)}$. VSM is needed only during training to learn view synthesis, and not required for keypoint-descriptor generation during inference.

**Grid Transformation Encoder (GTE):** $L_{cm}$ seeks to learn a viewpoint invariant depth image representation. Hence, to synthesize $\hat{I}^{(2)}$, it is essential to use $T^{1\to2}$ to incorporate viewpoint specific information. GTE is designed to encode $T^{1\to2}$. First, we compose a grid of transformation related parameters $G^{1\to2} \in \mathbb{R}^{h\times w\times f'_t}$. Along spatial position $(i,j)$, $G^{1\to2}_{i,j}$ is created by concatenating: 1) $\text{Warp}(I'^{(1)}; M^{1\to2})(i,j)$, 2) $\pi(F^{(1)})(i,j)$, 3) $M^{1\to2}_{i,j}$ and 4) the relative pose between $C^{(1)}$ and $C^{(2)}$. The detail about how $G^{1\to2}$ is composed is in the supplementary material. Then, similar to [57] we encode $G^{1\to2}$ using two fully-connected residual blocks, and obtain $T^{1\to2} \in \mathbb{R}^{h\times w\times f_t}$. The architecture of the fully-connected residual blocks is in the supplementary material. We empirically set $f_t = 96$.

**Depthmap Synthesis Network (DSN):** DSN (Figure 2) takes as input $F^{1\to2}$ and $T^{1\to2}$; and synthesizes $\hat{I}^{(2)}$. First, we apply global average pooling (GAP) [23] on $F^{1\to2}$. GAP features are then concatenated across every spatial position of $F^{1\to2}$ and passed through two $1\times1$ convolutions to obtain $F^{1\to2}_g$. $F^{1\to2}_g$ captures spatial position specific local information, and also the global context captured by GAP. Then we concatenate $F^{1\to2}_g$ with $T^{1\to2}$ along the channel dimension to obtain $F^{1\to2}_c$. Finally, three residual convolutional blocks and a $1\times1$ convolutional layer follow to synthesize $\hat{I}^{(2)}$. We use convolutional blocks to reason about the spatial neighborhoods of each pixel in the final phase of DSN. The architecture of the convolutional residual block is in the supplementary material.

**View Synthesis Loss ($L_v$):** We train VSM using the *View Synthesis Loss* $L_v$, which encourages the similarity between $\hat{I}^{(2)}$ and $I'^{(2)}$. We define $L_v$ as:

$$L_v = \left( \sum_{i \in P^{1\to2}} ||\hat{I}^{(2)}_i - I'^{(2)}_i||_1 \right) \Big/ |P^{1\to2}| \,. \tag{2}$$

| Dataset \ Property | # of Scenes | # of Sequences | Sensor Type | # Training/Testing Images |
|---|---|---|---|---|
| MSR-7 [38] | 7 | 18 | Kinect | 26K/17K |
| TUM [42] | 11 | 55 | Kinect | 18K/4K |
| CoRBS [51] | 4 | 20 | Kinect v2 | 26K/6K |

Table 1: The datasets and their properties used in our experiments.

Here, $P^{1\to 2}$ is the set of pixels in $I'^{(2)}$ that correspond to the 3D points contained within the camera view frustum of $I^{(1)}$, but possibly are occluded in $I^{(1)}$. The overall loss $L$ for the ViewSynth is $L = L_{cm} + \alpha L_v$, where we empirically set $\alpha = 10$ throughout all experiments.

# 4 Experimental evaluation

We evaluate the target tasks on three datasets: RGB-D dataset 7-Scenes (denoted as MSR-7) [38], TUM RGBD-SLAM (3D object reconstruction subset) [42], and CoRBS [51]. Summarized in Table 1, each dataset is a compilation of tracked sequences of real RGB-D camera frames of naturally occurring indoor scenes.

**Experimental Protocol** We follow the same experimental setup as [12] to evaluate local feature learning from depth images on the 3D correspondence matching and camera localization tasks. Pairs of depth images captured 10 or 30 frames apart are used during training. For evaluation, a *reference 3D keypoint-descriptor repository*, $\mathcal{R}$ is created from the training-set images. This is done by extracting the 50 highest scoring keypoints and their descriptors from each image, and putting these keypoints' 3D world coordinates and their descriptors into $\mathcal{R}$. Next, we extract the 50 highest scoring keypoints from each test-set image and match them against the keypoints in $\mathcal{R}$ based on the closest $L_2$ descriptor distance. A match is considered correct if the 3D world coordinates of the matched keypoints are within a 3D distance threshold ($0.1m$, $0.25m$ or $0.5m$). To evaluate camera localization, we use an experimental protocol similar to [35]. Same as before, we match the keypoints of each test-set depth image to $\mathcal{R}$, and then estimate the camera pose using the RANSAC based EPnP solver [6, 21]. The camera localization accuracy is measured in different *position error* and *orientation error* thresholds. We evaluate on (0.5m, 2°), (1m, 5°) and (5m, 10°) thresholds.

**Baseline** We use the SOTA depth local feature extractor Georgakis' method [12], and SOTA RGB local feature extractors R2D2 [33] and D2Net [10] adapted for depth modality as our baselines. As the original D2Net led to the model collapse [52] in all experimental setups, we add a modified D2Net baseline (mD2Net) which uses all negative sampling instead of the hardest negative sampling for descriptor learning. For keypoint-descriptor extraction, we used the 3-scale detection setting [10] for all D2Net baselines and ViewSynth. We also add an additional R2D2 baseline $\text{R2D2}_{s_3}$, which uses 3 scales (instead of 5) for keypoint aggregation similar to ViewSynth and the D2Net baselines. While training the R2D2 baselines, the image pairs were resized to $256 \times 192$ to fit them into the GPU memory. During evaluation, full resolution images were used for keypoint-descriptor extraction. Finally, we add another baseline $\text{D2Net}_{L_{cm}}$ which uses the D2Net architecture and $L_{cm}$.

**Results** Figure 3 shows the qualitative results of keypoint matching, where ViewSynth obtains higher number of correct matching pairs than mD2Net and $\text{D2Net}_{L_{cm}}$. Table 2 shows the mean matching accuracy (MMA) for each method in the 3D keypoint matching task, for $0.1m$, $0.25m$ and $0.5m$ 3D distance thresholds. D2Net could not be trained as it faced the model collapse [52] in all experiments. ViewSynth outperforms all the listed baselines in

| MMA Threshold | 0.1m | | 0.25m | | 0.1m | | 0.25m | | 0.1m | | 0.25m | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # of Frames Apart | 10 | 30 | 10 | 30 | 10 | 30 | 10 | 30 | 10 | 30 | 10 | 30 |
| Dataset | TUM | | | | CoRBS | | | | MSR-7 | | | |
| D2Net [10] | Collapsed | | | | Collapsed | | | | Collapsed | | | |
| mD2Net | 8.72 | 3.62 | 20.48 | 12.60 | 17.10 | 13.93 | 29.83 | 28.13 | 45.69 | 45.02 | 61.31 | 59.55 |
| R2D2 [33] | 20.84 | - | 37.34 | - | 42.08 | - | 51.26 | - | 61.55 | - | 66.30 | - |
| R2D2$_{s_3}$ [33] | 16.74 | - | 33.24 | - | 34.70 | - | 43.15 | - | 50.07 | - | 55.60 | - |
| D2Net$_{L_{cm}}$ | 33.38 | 23.93 | 53.19 | 45.82 | 56.73 | 51.53 | 71.24 | 66.65 | 79.87 | 80.35 | **89.84** | 90.30 |
| ViewSynth (ours) | **34.75** | **35.63** | **59.45** | **57.39** | **67.30** | **52.69** | **72.43** | **69.25** | **80.10** | **80.56** | 89.70 | **90.72** |

Table 2: Comparison of MMA on TUM, CoRBS, and MSR-7 datasets, trained on 10/30-frames-apart setting. Acronyms: mD2Net: modified D2Net; D2Net$_{L_{cm}}$: D2Net with loss $L_{cm}$; ViewSynth: D2Net$_{L_{cm}}$+ $L_v$, our proposed method.

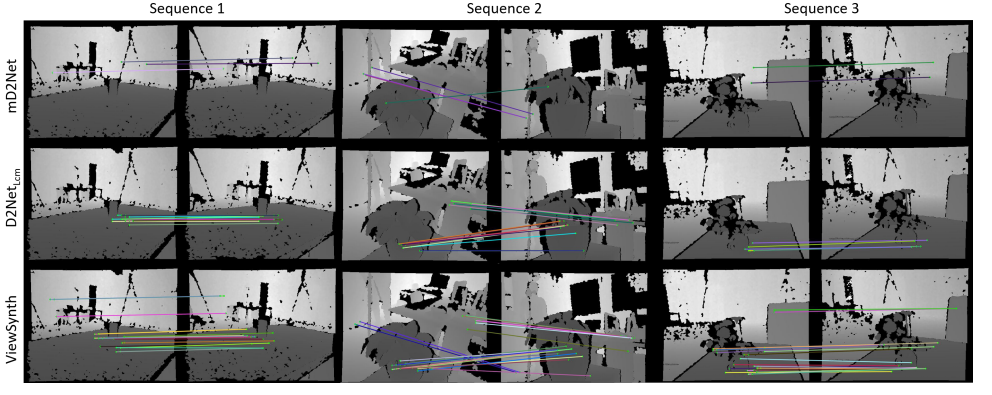


Figure 3: ViewSynth shows that learning view synthesis allows more correct keypoints matches between image pairs.

most settings. R2D2 had a convergence issue for 30-frames-apart training setting, for which it could not produce any keypoint. Possibly, R2D2's sensitive approximated average precision failed to produce meaningful gradients for noisy depth images with large viewpoint variation. ViewSynth beats Georgakis' method [12] by 80.65 vs. 41.20 MMA in the MSR-7 dataset for 10-frames-apart training setting, which is the only experimental setting conducted by [12] on this dataset. ViewSynth outperforms the listed baselines in most settings of the camera localization task (Table 3, 4). Figure 4 demonstrates that ViewSynth is able to synthesize occluded surfaces, which it was trained to learn. More quantitative evaluations of 3D keypoint matching in the depth modality, additional qualitative results, and training details can be found in the supplementary material.

**Ablation** We study the effect of $L_{cm}$ and $L_v$ on both tasks. Figure 3 and Table 2, 4 show the efficacy of $L_{cm}$, where D2Net$_{L_{cm}}$ outperforms mD2Net in every case, while original D2Net fails to learn. ViewSynth in addition uses VSM and $L_v$ to learn view synthesis, and outperforms D2Net$_{L_{cm}}$ in most settings in Table 2, beating all the listed baselines. VSM is more effective in the 30-frames-apart training setting compared to the 10-frames-apart training setting; as it learns view synthesis from larger viewpoint variation with more occlusion (Table 2). This supports that - learning view synthesis encodes information beneficial for improving keypoint matching accuracy. We also see in the camera localization task (Table

| Threshold | 0.5m, 2° | | 1m, 5° | | 5m, 10° | | 0.5m, 2° | | 1m, 5° | | 5m, 10° | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # of Frames Apart | 10 | 30 | 10 | 30 | 10 | 30 | 10 | 30 | 10 | 30 | 10 | 30 |
| Method \ Dataset | TUM | | | | | | CoRBS | | | | | |
| D2Net [10] | Collapsed | | Collapsed | | Collapsed | | Collapsed | | Collapsed | | Collapsed | |
| mD2Net | 1.18 | 1.77 | 4.81 | 6.51 | 9.67 | 12.49 | 1.90 | 4.18 | 6.85 | 11.56 | 13.40 | 18.51 |
| R2D2 [33] | 7.44 | - | 18.70 | - | 26.40 | - | **26.03** | - | **41.91** | - | **48.72** | - |
| R2D2$_{s_3}$ [33] | 6.76 | - | 17.10 | - | 23.68 | - | 25.28 | - | 38.93 | - | 44.48 | - |
| ViewSynth (ours) | **7.70** | **7.58** | **23.02** | **16.60** | **35.49** | **27.18** | 8.19 | **8.57** | 23.36 | **30.29** | 47.78 | **50.52** |

Table 3: Camera localization accuracy (%) on TUM and CoRBS datasets, with 10/30-frames-apart training setting. For most localization correctness thresholds, our proposed method outperforms the SOTA.

| Threshold | 0.5m, 2° | | 1m, 5° | | 5m, 10° | |
|---|---|---|---|---|---|---|
| # of Frames Apart | 10 | 30 | 10 | 30 | 10 | 30 |
| Method \ Dataset | MSR-7 | | | | | |
| D2Net [10] | Collapsed | | Collapsed | | Collapsed | |
| mD2Net | 15.46 | 14.11 | 37.68 | 34.66 | 53.98 | 50.24 |
| R2D2 [33] | **47.20** | - | 68.13 | - | 74.49 | - |
| R2D2$_{s_3}$ [33] | 43.94 | - | 61.61 | - | 67.78 | - |
| D2Net$_{L_{cm}}$ | 31.52 | 21.92 | 66.33 | **58.25** | 85.24 | **82.61** |
| ViewSynth (ours) | 34.60 | **23.83** | **70.09** | 57.04 | **86.67** | 80.34 |

Table 4: Camera localization accuracy (%) on MSR-7 dataset, with 10/30-frames-apart training setting. ViewSynth outperforms baselines for most localization correctness thresholds.

4), that ViewSynth beats other methods in most settings. Table 5 demonstrates the generalizability of ViewSynth, where it beats D2Net$_{L_{cm}}$ in the keypoint matching task across different scenes of the same dataset, and across different datasets using the same or different depth sensors. ViewSynth also beats mD2Net in all cases (in the supplementary).

**Discussion** The quantitative (Table 2, 3, 4, 5) and qualitative (Figure 3) results of the keypoint matching and camera localization tasks on different datasets show that ViewSynth outperforms the SOTA methods in most experimental settings. R2D2 and R2D2$_{s_3}$ perform reasonably in some settings, but they fail to learn when training image pairs contain large viewpoint variations. In all cases, the original D2Net was not trainable due to the model collapse [52]. While mD2Net circumvents this using the all negative sampling for learning descriptors, it still leads to a poor performance. D2Net$_{L_{cm}}$ demonstrates the efficacy of the proposed loss $L_{cm}$ by beating mD2Net in all cases. ViewSynth in addition utilizes VSM and $L_v$ to learn view synthesis, and outperforms D2Net$_{L_{cm}}$ and all the other baselines for both tasks in most settings. These results assert the effectiveness of learning view synthesis for keypoint-descriptor extraction from depth images - towards the 3D keypoint matching and camera localization tasks.

## 5 Conclusion

We show that the SOTA RGB keypoint detection-description methods either are not trainable (D2Net [10]), or do not perform well (R2D2 [33]) in the depth image modality. Towards improving keypoint matching in the depth modality, we propose a framework *ViewSynth* to learn view synthesis in conjunction with learning keypoint-descriptor from depth images in a joint fashion. We propose the *Contrastive Matching Loss*, $L_{cm}$, to learn keypoints and descriptors jointly. We show that, learning view synthesis of depth images from different

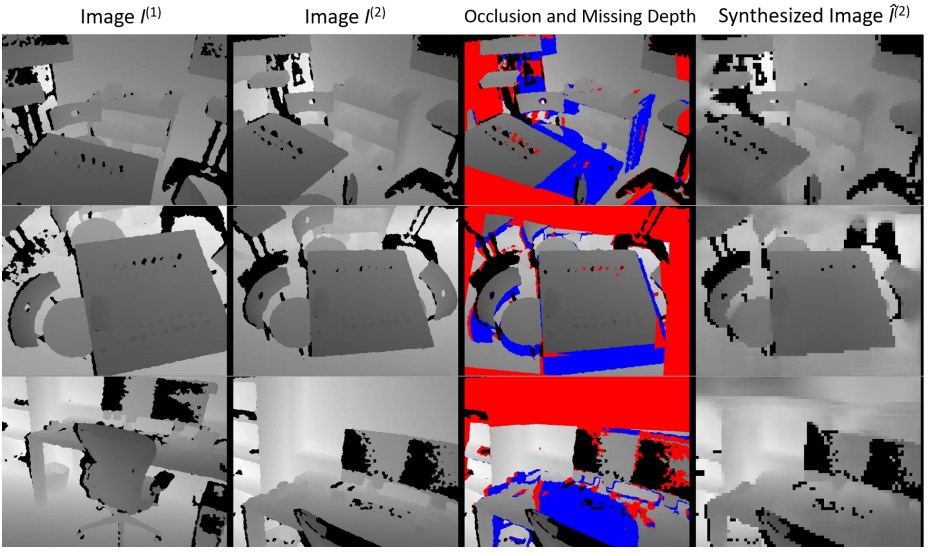


Figure 4: View synthesis outputs on MSR-7 scenes. **Blue** highlighted area indicates the parts of $I^{(2)}$ that are occluded in $I^{(1)}$. **Red** highlight indicates the change in pose between $I^{(1)}$, $I^{(2)}$ and the missing information in $I^{(1)}$. $\hat{I}^{(2)}$ shows that VSM can synthesize the depth views in the blue occluded regions.

| MMA Threshold | | $0.1m$ | | $0.25m$ | | $0.5m$ | |
|---|---|---|---|---|---|---|---|
| Trained on | Tested on | D2Net$_{L_{cm}}$ | ViewSynth | D2Net$_{L_{cm}}$ | ViewSynth | D2Net$_{L_{cm}}$ | ViewSynth |
| MSR-7 (w/o fire scene) | MSR-7 fire scene | 83.32 | **84.85** | 91.36 | **92.94** | 93.03 | **94.80** |
| MSR-7 | TUM | 27.93 | **29.05** | 51.89 | **53.88** | 67.36 | **69.08** |
| MSR-7 | CoRBS | 44.96 | **46.47** | 61.65 | **63.32** | 73.62 | **75.00** |

Table 5: Generalizability of ViewSynth framework on the 3D keypoint matching task using MMA metric. ViewSynth generalizes better than D2Net$_{L_{cm}}$ - to a new scene (row 1), new dataset with the same sensor (row 2), and across a dataset with different depth sensor (row 3) (see Table 1 for the dataset details). All experiments are in the 30-frames-apart setting.

viewpoints using our proposed *View Synthesis Module* (VSM) and the *View Synthesis Loss*, $L_v$, encourages the network to encode information that improves the performance in keypoint matching. ViewSynth outperforms the SOTA in the 3D keypoint matching and camera localization task across the MSR-7, TUM and CoRBS datasets in most cases. We also demonstrate the generalizability of ViewSynth in 3D keypoint matching across different datasets. These evaluations attest the efficacy of ViewSynth in learning keypoints and descriptors for 3D keypoint matching and camera localization.

**Acknowledgement** Part of the efforts from Jisan Mahmud and Jan-Michael Frahm is supported by NSF grant No. IIS-1816148.

# References


[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. *arXiv preprint arXiv:1510.07493*, 2015.

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, June 2008. ISSN 1077-3142. doi: 10.1016/j.cviu.2007.09.014.

[4] Enrico Bondi, Lorenzo Seidenari, Andrew D Bagdanov, and Alberto Del Bimbo. Real-time people counting from depth imagery of crowded environments. In *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 337–342. IEEE, 2014.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[7] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7781–7790, 2019.

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.

[9] Bertram Drost and Slobodan Ilic. 3d object detection and localization using multimodal point pair features. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 9–16. IEEE, 2012.

[10] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019.

[11] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[12] Georgios Georgakis, Srikrishna Karanam, Ziyan Wu, Jan Ernst, and Jana Košecká. End-to-end learning of keypoint detector and descriptor for pose invariant 3d matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1965–1973, 2018.

[13] Georgios Georgakis, Srikrishna Karanam, Ziyan Wu, and Jana Kosecka. Learning local rgb-to-cad correspondences for object pose estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[14] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings - 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006*, volume 2, pages 1735–1742, 2006. ISBN 0769525970. doi: 10.1109/CVPR.2006.100.

[15] Jared Heinly, Johannes L Schönberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3287–3295, 2015.

[16] Rabah Iguernaissi, Djamal Merad, and Pierre Drap. People counting based on kinect depth data. In *ICPRAM*, pages 364–370, 2018.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[18] Ahmad Jalal, Yeon-Ho Kim, Yong-Joong Kim, Shaharyar Kamal, and Daijin Kim. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern recognition*, 61:295–308, 2017.

[19] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Predicting good features for image geo-localization using per-bundle vlad. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1170–1178, 2015.

[20] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3251–3260. IEEE, 2017.

[21] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009.

[22] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[23] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[24] Mengyuan Liu and Hong Liu. Depth context: A new descriptor for human activity recognition by using sole depth sequences. *Neurocomputing*, 175:747–758, 2016.

[25] Yanli Liu, Heng Zhang, Hanlei Guo, and Neal N Xiong. A fast-brisk feature detector with depth information. *Sensors*, 18(11):3908, 2018.

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

[27] Karol Matusiak, Piotr Skulimowski, and Pawel Strumillo. Depth-based descriptor for matching keypoints in 3d scenes. *International Journal of Electronics and Telecommunications*, 64(3):299–306, 2018.

[28] Derek McColl, Zhe Zhang, and Goldie Nejat. Human body pose interpretation and classification for social human-robot interaction. *International Journal of Social Robotics*, 3(3):313, 2011.

[29] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3465, 2017.

[30] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: Learning local features from images, 2018.

[31] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[32] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[33] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: Repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019.

[34] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer, 2011.

[35] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018.

[36] Johannes L Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[37] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[38] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, June 2013. doi: 10.1109/CVPR.2013.377.

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

[40] Vincent Sitzmann, Michael Zollhoefer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 1119–1130. Curran Associates, Inc., 2019.

[41] Xibin Song, Yuchao Dai, and Xueying Qin. Deeply supervised depth map super-resolution as novel view synthesis. *IEEE Transactions on circuits and systems for video technology*, 2018.

[42] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[43] Supasorn Suwajanakorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Advances in neural information processing systems*, pages 2059–2070, 2018.

[44] Mariusz Szwoch and Pawel Pieniazek. Facial emotion recognition using depth data. *2015 8th International Conference on Human System Interaction (HSI)*, pages 271–277, 2015.

[45] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7199–7209, 2018.

[46] Yurun Tian, Bin Fan, and Fuchao Wu. L2-net: Deep learning of discriminative patch descriptor in euclidean space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 661–669, 2017.

[47] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 302–317, 2018.

[48] Keze Wang, Shengfu Zhai, Hui Cheng, Xiaodan Liang, and Liang Lin. Human pose estimation from depth images via inference embedded multi-task learning. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1227–1236, 2016.

[49] Keze Wang, Liang Lin, Chuangjie Ren, Wei Zhang, and Wenxiu Sun. Convolutional memory blocks for depth data representation learning. In *IJCAI*, pages 2790–2797, 2018.

[50] Xiang-Yang Wang, Jun-Feng Wu, and Hong-Ying Yang. Robust image retrieval based on color histogram of local feature regions. *Multimedia Tools and Applications*, 49(2): 323–345, 2010.

[51] Oliver Wasenmüller, Marcel Meyer, and Didier Stricker. Corbs: Comprehensive rgb-d benchmark for slam using kinect v2. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–7. IEEE, 2016.

[52] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017.

[53] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6721–6729, 2017.

[54] 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.

[55] A. R. Zamir and M. Shah. Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1546–1558, Aug 2014. doi: 10.1109/TPAMI.2014.2299799.

[56] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1802–1811, 2017.

[57] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016.