Predicting Visual Overlap of Images Through Interpretable Non-Metric Box Embeddings

Anita Rau[1⋆], Guillermo Garcia-Hernando[2], Danail Stoyanov[1], Gabriel J. Brostow[1,2], and Daniyar Turmukhambetov[2]

[1]University College London [2]Niantic

www.github.com/nianticlabs/image-box-overlap

Abstract. To what extent are two images picturing the same 3D surfaces? Even when this is a known scene, the answer typically requires an expensive search across scale space, with matching and geometric verification of large sets of local features. This expense is further multiplied when a query image is evaluated against a gallery, *e.g.* in visual relocalization. While we don't obviate the need for geometric verification, we propose an interpretable image-embedding that cuts the search in scale space to essentially a lookup.

Our approach measures the **asymmetric** relation between two images. The model then learns a scene-specific measure of similarity, from training examples with known 3D visible-surface overlaps. The result is that we can quickly identify, for example, which test image is a close-up version of another, and by what scale factor. Subsequently, local features need only be detected at that scale. We validate our scene-specific model by showing how this embedding yields competitive image-matching results, while being simpler, faster, and also interpretable by humans.

Keywords: Image embedding, representation learning, image localization, interpretable representation

1 Introduction

Given two images of the same scene, which one is the close-up? This question is relevant for many tasks, such as image-based rendering and navigation [52], because the close-up has higher resolution details [30] for texturing a generated 3D model, while the other image has the context view of the scene. Related tasks include 3D scene reconstruction [48], robot navigation [5] and relocalization [3, 40, 42, 44], which all need to reason in 3D about visually overlapping images.

For these applications, exhaustive searches are extremely expensive, so efficiency is needed in two related sub-tasks. First, it is attractive to cheaply identify which image(s) from a corpus have substantial overlap with a query image. Most images are irrelevant [53], so progress on whole-image descriptors like [50] narrows that search. Second, for every two images that are expected to have matchable content, the geometric verification of relative pose involves two-view feature

[⋆] Work done during an internship at Niantic.

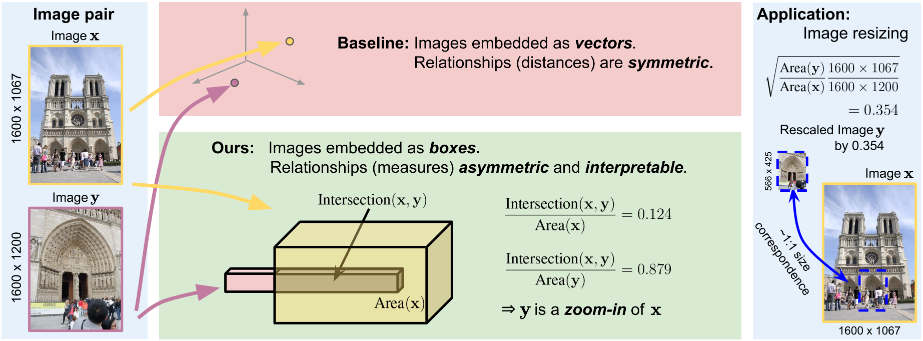


Image pair
Image x
1600 x 1067
Image y
1600 x 1200
Baseline: Images embedded as vectors.
Relationships (distances) are symmetric.
Ours: Images embedded as boxes.
Relationships (measures) asymmetric and interpretable.
Intersection(x, y)
Area(x)
$\frac{\text{Intersection}(\mathbf{x},\mathbf{y})}{\text{Area}(\mathbf{x})} = 0.124$
$\frac{\text{Intersection}(\mathbf{x},\mathbf{y})}{\text{Area}(\mathbf{y})} = 0.879$
⇒ y is a zoom-in of x
Application:
Image resizing
$\sqrt{\frac{\text{Area}(\mathbf{y})}{\text{Area}(\mathbf{x})}\frac{1600 \times 1067}{1600 \times 1200}}$
$= 0.354$
Rescaled Image y
by 0.354
566 x 425
Image x
~1:1 size
correspondence
1600 x 1067

Fig. 1. Given two images from the test set, can we reason about their relationship? How much of the scene is visible in both images? Is one image a zoomed-in version of another? We propose a CNN model to predict box embeddings that approximate the visible surface overlap measures between two images. The surface overlap measures give an interpretable relationship between two images and their poses. Here, we extract the relative scale difference between two test images, without expensive geometric analysis.

matching and pose estimation [17], which can range from being moderately to very expensive [12], depending on design choices. RANSAC-type routines are more accurate and faster if the two images have matching scales [10,64,66], and if the detector and descriptor work well with that scene to yield low outlier-ratios.

For scenes where training data is available, we efficiently address both subtasks, relevant image-finding and scale estimation, with our new embedding. Our proposed model projects whole images to our custom asymmetric feature space. There, we can compare the non-Euclidean measure between image encoding $\mathbf{x}$ to image encoding $\mathbf{y}$, which is different from comparing $\mathbf{y}$ to $\mathbf{x}$ – see Figure 1. Ours is distinct from previous methods which proposed a) learning image similarity with metric learning, *e.g.* [1,3,50], and b) estimation of relative scale or image overlap using geometric validation of matched image features, *e.g.* [29,30]. Overall, we

- advocate that normalized surface overlap (NSO) serves as an interpretable real-world measure of how the same geometry is pictured in two images,
- propose a new box embedding for images, that approximates the surface overlap measure while preserving interpretability, and
- show that the predicted surface overlap allows us to pre-scale images for same-or-better feature matching and pose estimation in a localization task.

The new representation borrows from box embeddings [26,56,60], that were designed to represent hierarchical relations in words. We are the first to adapt them for computer vision tasks, and hereby propose *image* box embeddings. We qualitatively validate their ability to yield *interpretable* image relationships that don't impede localization, and quantitatively demonstrate that they help with scale estimation.

2 Related Work

Ability to quickly determine if two images observe the same scene is useful in multiple computer vision pipelines, especially for SLAM, Structure-from-Motion (SfM), image-based localization, landmark recognition, *etc*.

For example, Bag-of-Words (BoW) models [13] are used in many SLAM systems, *e.g*. ORB-SLAM2 [33], to detect when the camera has revisited a mapped place, *i.e*. a loop closure detection, to re-optimize the map of the scene. The BoW model allows to quickly search images of a mapped area and find a match to the current frame for geometric verification. A similar use case for BoW model can be found in SfM pipelines [12, 20, 48], where thousands or even millions of images are downloaded from the Internet and two-view matching is impractical to do between all pairs of images. Once a 3D model of a scene is built, a user might want to browse the collection of images [53]. Finding images that are zoom-ins to parts of the query and provide detailed, high-resolution images of interesting parts of the scene was addressed in [29, 30], where they modified a BoW model to also store local feature geometry information [36] that can be exploited for Document at a Time (DAAT) scoring [55] and further geometric verification. However, that approach relies on iterative query expansion to find zoomed-in matches, and re-ranking of retrieved images still requires geometric verification, even when choosing images for query expansion. Similarly the loss of detail in SfM reconstructions was identified to be a side-effect of retrieval approaches in [47] and solution based on modification of query expansion and DAAT scoring was proposed, that favors retrieval of images with scale and viewing directions changes. Again, each image query retrieves a subset of images from the image collection in a certain ranking order and geometric verification is performed to establish the pairwise relation of the views. Weyand and Leibe [62] build a hierarchy of iconic views of a scene by iteratively propagating and verifying homography between pairs of images and generating clusters of views. Later, they extended [63] it to find details and scene structure with hierarchical clustering over scale space. Again, the relation between views is estimated with geometric verification.

Schönberger *et al*. [46] propose to learn to identify image pairs with scene overlap without geometric verification for large-scale SfM reconstruction applications to decrease the total computation time. However, their random forest classifier is trained with binary labels, a pair of images either overlap or not, and no additional output about the relative pose relation is available without further geometric verification. Shen *et al*. [50] proposed to learn an embedding which can be used for quick assessment of image overlap for SfM reconstructions. They trained a neural network with triplet loss with ground-truth overlap triplets, however their embedding models minimal overlap between two images, which can be used to rank images according to predicted mutual overlap, but does not provide additional information about relative scale and pose of the image pairs.

Neural networks can be used to directly regress a homography between a pair of images [9, 11, 25, 34], however a homography only explains a single planar surface co-visible between views.

Finally, image retrieval is a common technique in localization [3,43,44]: first, a database of geo-tagged images is searched to find images that are similar to the query, followed by image matching and pose estimation to estimate the pose of the query. Again, the image retrieval reduces the search space for local feature matching, and it is done using a compact representation such as BoW [38], VLAD [2] or Fisher vectors [37]. Recently proposed methods use neural networks to compute the image representation and learn the image similarity using geometric information available for the training images, e.g. GPS coordinates for panoramic images [1], camera poses in world coordinate system [3], camera pose and 3D model of the environment [50]. Networks trained to directly regress the camera pose [22,24] were also shown to be similar to image retrieval methods [45]. The image embeddings are learned to implicitly encapsulate the geometry of a scene so that similar views of a scene are close in the embedding space. However, these representations are typically learned with metric learning, and so they only allow to rank the database images with respect to the query, and no additional camera pose information is encoded in the representations. We should also mention that localization can also be tackled by directly finding correspondences from 2D features in the query to 3D points in the 3D model without explicit image retrieval, *e.g.* [6,40–42].

Image Matching If a pair of images have a large viewing angle and scale difference, then estimating relative pose using standard correspondence matching becomes challenging. MODS [32] addresses the problem by iteratively generating and matching warped views of the input image pair, until a sufficient confidence in the estimated relative pose is reached. Zhou *et al.* [66] tackles the problem of very large scale difference between the two views by exploiting the consistent scale ratio between matching feature points. Thus, their two-stage algorithm first does exhaustive scale level matching to estimate the relative scale difference between the two views and then does feature matching only in corresponding scale levels. The first stage is done exhaustively, as there is no prior information about the scale difference, which our embeddings provide an estimate for.

Metric Learning Many computer vision applications require learning an embedding, such that the relative distances between inputs can be predicted. For example, face recognition [49], descriptor learning [4, 19, 31, 39, 57], image retrieval [14], person re-identification [21], *etc.* The common approach is metric learning: models are learned to encode the input data (*e.g.* image, patch, *etc.*) as a vector, such that a desirable distance between input data points is approximated by distances in corresponding vector representations. Typically, the setup uses siamese neural networks, and contrastive loss [16], triplet loss [49], average concentration loss [19] and ranking loss [8] is used to train the network. The distances in the embedding space can be computed as Euclidean distance between two vectors, an inner product of two vectors, a cosine of the angle between two vectors, *etc.* However, learned embeddings can be used to estimate the order of data points, but not other types relations, due to symmetric distance function used for similarity measure. Other relations, for example, hierarchies

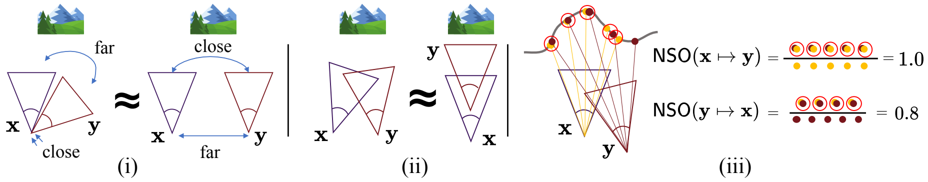


Fig. 2. Illustrations of world-space measures and their properties. (i) Left and right camera configurations result in the same value for weighted sum of rotation translation errors. (ii) Left and right camera configurations result in the same frustum overlap value. (iii) Illustration of our surface overlap. Images $\mathbf{x}$ and $\mathbf{y}$ have resolution of 5 pixels. All 5 pixels of $\mathbf{x}$ are visible in $\mathbf{y}$, meaning the corresponding 3D points of $\mathbf{x}$ are sufficiently close to 3D points backprojected from $\mathbf{y}$. Hence $\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y}) = 1.0$. However, only 80% of pixels of $\mathbf{y}$ are visible by $\mathbf{x}$, thus, $\mathsf{NSO}(\mathbf{y} \mapsto \mathbf{x}) = 0.8$.

require asymmetric similarity measures, which is often encountered in word embeddings [26, 35, 56, 59–61].

Some methods learn representations that are disentangled, so that they correspond to meaningful variations of the data. For example [23, 54, 65] model geometric transformations of the data in the representation space. However these representations are typically learned for a single class of objects, and it is not straight-forward how to use these embeddings for retrieval.

3 Method

Our aim is to i) interpret images according to a geometric world-space measure, and ii) to devise an embedding that, once trained for a specific reconstructed 3D scene, lends itself to easily compute those interpretations for new images. We now explain each, in turn.

3.1 World-space Measures

We want to have interpretable world-space relationship between a pair of images $\mathbf{x}$ and $\mathbf{y}$, and their corresponding camera poses (orientations and positions).

A straightforward world-space measure is the distance between camera centers [1]. This world-space measure is useful for localizing omni-directional or panoramic cameras that observe the scene in 360°. However, most cameras are not omni-directional, so one needs to incorporate the orientation of the cameras. But how can we combine the relative rotation of the cameras and their relative translation into a world-space measure? One could use a weighted sum of the rotation and translation differences [7, 22, 52, 53], but there are many camera configurations where this measure is not satisfactory, *e.g.* Figure 2(i).

Another example of a world-space measure is frustum overlap [3]. Indeed, if we extend the viewing frustum of each camera up to a cutoff distance D, we can assume that the amount of frustum overlap correlates with the positions and orientation of the cameras. However, the two cameras can be placed multiple ways and have the same frustum overlap – see Figure 2(ii). So, the normalized frustum overlap value does not provide interpretable information about the two camera poses.

We propose to use normalized surface overlap for the world-space measure. See Figure 2(iii) for an illustration. Formally, normalized surface overlap is defined as

$$\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y}) = \mathsf{overlap}(\mathbf{x} \mapsto \mathbf{y})/N_{\mathbf{x}}, \text{ and} \tag{1}$$

$$\mathsf{NSO}(\mathbf{y} \mapsto \mathbf{x}) = \mathsf{overlap}(\mathbf{y} \mapsto \mathbf{x})/N_{\mathbf{y}}, \tag{2}$$

where $\mathsf{overlap}(\mathbf{x} \mapsto \mathbf{y})$ is the number of pixels in image $\mathbf{x}$ that are visible in image $\mathbf{y}$, and $\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y})$ is $\mathsf{overlap}(\mathbf{x} \mapsto \mathbf{y})$ normalized by the number of pixels in $\mathbf{x}$ (denoted by $N_{\mathbf{x}}$), hence it is a number between 0 and 1 and it can be represented as a percentage. To compute it, we need to know camera poses and the depths of pixels for both image $\mathbf{x}$ and image $\mathbf{y}$. The pixels in both images are backprojected into 3D. The $\mathsf{overlap}(\mathbf{x} \mapsto \mathbf{y})$ are those 3D points in $\mathbf{x}$ that have a neighbor in the point cloud of image $\mathbf{y}$ within a certain radius.

The normalized surface overlap is not symmetric, $\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y}) \neq \mathsf{NSO}(\mathbf{y} \mapsto \mathbf{x})$, because only a few pixels in image $\mathbf{x}$ could be viewed in image $\mathbf{y}$, but all the pixels in $\mathbf{y}$ could be viewed in image $\mathbf{x}$. This asymmetric measure can have an interpretation that image $\mathbf{y}$ is a *close-up view of a part the scene* observed in image $\mathbf{x}$. So, the normalized surface overlap provides an interpretable relation between cameras; please see Figure 4 for different cases.

In addition to the visibility of pixels in $\mathbf{x}$ by image $\mathbf{y}$, one could also consider the angle at which the overlapping surfaces are observed. Thus, we weight each point in $\mathsf{overlap}(\mathbf{x} \mapsto \mathbf{y})$ with $\cos(\mathbf{n_i}, \mathbf{n_j})$, $\mathbf{n_i}$ denotes the normal of a pixel $\mathbf{i}$ and $\mathbf{n_j}$ is the normal of the nearest 3D point of $\mathbf{i}$ in image $\mathbf{y}$. This will reduce the surface overlap between images observing the same scene from very different angles. Two images are difficult to match if there is substantial perspective distortion due to a difference in viewing angle. Incorporating the angle difference into the world-space measure captures this information.

3.2 Embeddings

We aim to learn a representation and embedding of RGB images. The values computed on those representations should approximate the normalized surface overlap measures, but without access to pose and depth data. Such a representation and embedding should provide estimates of the surface overlap measure cheaply and generalize to test images for downstream tasks such as localization.

Vector Embeddings A common approach in computer vision to learn embeddings is to use metric learning. Images are encoded as vectors using a CNN, and the network is trained to approximate the relation in world-space measure by distances in vector space.

However, there is a fundamental limitation of vector representations and metric learning: the distance between vectors corresponding to images $\mathbf{x}$ and $\mathbf{y}$ can only be symmetric. In our case, the normalized surface overlaps are not symmetric. Hence they cannot be represented with vector embeddings. We could compromise and compute a symmetric value, for example the average of normalized surface overlaps computed in both directions,

$$\mathsf{NSO}^{sym}(\mathbf{x},\mathbf{y}) = \mathsf{NSO}^{sym}(\mathbf{y},\mathbf{x}) = \frac{1}{2}(\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y}) + \mathsf{NSO}(\mathbf{y} \mapsto \mathbf{x})). \quad (3)$$

Or, only consider the least amount of overlap in one of the directions [50], so $\mathsf{NSO}^{min}(\mathbf{x},\mathbf{y}) = \mathsf{NSO}^{min}(\mathbf{y},\mathbf{x}) = \min(\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y}), \mathsf{NSO}(\mathbf{y} \mapsto \mathbf{x}))$. As shown in Figure 3 any such compromise would result in the loss of interpretability.

Different strategies to train vector embeddings are valid. One could consider training a symmetric version using NSO^{sym} or an asymmetric one using NSO; we hypothesize both are equivalent for large numbers of pairs. We empirically found the best performance with the loss function $\mathcal{L}_{\text{vector}} = ||\mathsf{NSO}^{sym}(\mathbf{x},\mathbf{y}) - 1 - ||f(\mathbf{x}) - f(\mathbf{y})||_2||_2$, where f is a network that predicts a vector embedding.

Box Embeddings We propose to use box representations to embed images with non-metric world-space measures. Our method is adapted from geometrically-inspired word embeddings for natural language processing [26, 56, 60]. The box representation of image $\mathbf{x}$ is a D-dimensional orthotope (hyperrectangle) parameterized as a $2D$-dimensional array $\mathbf{b_x}$. The values of this array are split into $m_{\mathbf{b_x}}$ and $M_{\mathbf{b_x}}$, which are the lower and upper bounds of the box in D-dimensional space. Crucially, the box representation allows us to compute the *intersection* of two boxes, $\mathbf{b_x} \wedge \mathbf{b_y}$, as

$$\mathbf{b_x} \wedge \mathbf{b_y} = \prod_d^D \sigma\left(\min(M^d_{\mathbf{b_x}}, M^d_{\mathbf{b_y}}) - \max(m^d_{\mathbf{b_x}}, m^d_{\mathbf{b_y}})\right), \quad (4)$$

and the *volume* of a box,

$$A(\mathbf{b_x}) = \prod_d^D \sigma(M^d_{\mathbf{b_x}} - m^d_{\mathbf{b_x}}), \quad (5)$$

where $\sigma(v) = \max(0, v)$. This definition of $\sigma()$ has a problem of zero gradient to v for non-overlapping boxes that should be overlapping. As suggested by Li *et al*. [26], we train with a smoothed box intersection: $\sigma_{\texttt{smooth}}(v) = \rho \ln(1 + \exp(v/\rho))$, which is equivalent to $\sigma()$ as ρ approaches 0. Hence, we can approximate world-space surface overlap values with normalized box overlaps as

$$\mathsf{NBO}(\mathbf{b_x} \mapsto \mathbf{b_y}) = \frac{\mathbf{b_x} \wedge \mathbf{b_y}}{A(\mathbf{b_x})} \approx \mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y}) \text{ and} \quad (6)$$

$$\mathsf{NBO}(\mathbf{b_y} \mapsto \mathbf{b_x}) = \frac{\mathbf{b_y} \wedge \mathbf{b_x}}{A(\mathbf{b_y})} \approx \mathsf{NSO}(\mathbf{y} \mapsto \mathbf{x}). \quad (7)$$

For training our embeddings, we can minimize the squared error between the ground truth surface overlap and the predicted box overlap of a random image pair $(\mathbf{x}, \mathbf{y})$ from the training set, so the loss is

$$\mathcal{L}_{\text{box}} = ||\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y}) - \mathsf{NBO}(\mathbf{b_x} \mapsto \mathbf{b_y})||_2^2. \quad (8)$$

It is important to note that computing the volume of a box is computationally very efficient. Given two boxes, intersection can be computed with just min, max and multiplication operations. Once images are embedded, one could build an efficient search system using R-trees [15] or similar data structures.

3.3 Implementation Details

We use a pretrained ResNet50 [18] as a backbone. The output of the 5-th layer is average pooled across spatial dimensions to produce a 2048-dimensional vector, followed by two densely connected layers with feature dimensions 512 and $2D$ respectively, where D denotes the dimensionality of the boxes. The first D values do not have a non-linearity and represent the position of the center of the box, and the other D values have softplus activation, and represent the size of the box in each dimension. These values are trivially manipulated to represent $m_{\mathbf{b_x}}$ and $M_{\mathbf{b_x}}$. We found $D = 32$ to be optimal (ablation in supp. material). Input images are anisotropically rescaled to 256×456 resolution. We fix $\rho = 5$ when computing smoothed box intersection and volume. We train with batch size $b = 32$ for $20 - 60k$ steps, depending on the scene. The ground-truth overlap both for training and evaluation is computed over the original resolution of images, however the 3D point cloud is randomly subsampled to 5000 points for efficiency. Surface normals are estimated by fitting a plane to a 3×3 neighborhood of depths. We used 0.1 as the distance threshold for two 3D points to overlap.

4 Experiments

We conduct four main experiments. First, we validate how our box embeddings of images capture the surface overlap measure. Do these embeddings learn the asymmetric surface overlap and preserve interpretability? Do predicted relations of images match camera pose relations?

Second, we demonstrate that the proposed world measure is superior to alternatives like frustum overlap for the task of image localization. We evaluate localization quality on a small-scale indoor dataset and a medium-scale outdoor dataset. We also show that images retrieved with our embeddings can be ranked with metric distances, and are "backwards-compatible" with existing localization pipelines, despite also being interpretable.

Third, we demonstrate how the interpretability of our embeddings can be exploited for the task of localization, specifically for images that have large scale differences to the retrieved gallery images. Finally, we show how our embeddings give scale estimates that help with feature extraction.

Datasets Since our proposed world-space measure requires depth information during training, we conduct our experiments on MegaDepth [27] and 7Scenes [51] datasets.

MegaDepth is a large scale dataset including 196 scenes, each of which consists of images with camera poses and camera intrinsics. However, only a subset of images have a corresponding semi-dense depth map. The dataset was originally proposed for the task of SfM reconstruction and to learn single-image depth estimation methods. The scenes have varying numbers of images, and large scenes have camera pose variations with significant zoom-in and zoom-out relationships. Most images do not have corresponding semi-dense depth maps, and may have only two-pixel ordinal labels or no labels at all. The low number of images with suitable depth also necessitates the generation of our own train and test splits, replacing the provided sets that are sampled across images with and without depths. As depth is needed for the computation of the ground-truth surface overlap, we only consider four scenes that provide enough data for training, validation and testing. So, we leave 100 images for validation and 100 images for testing, and the remaining images with suitable depth maps are used for training: 1336, 2130, 2155 and 1471 images for Big Ben, Notre-Dame, Venice, and Florence scenes, respectively.

7Scenes is an established small-scale indoor localization benchmark consisting of 3 to 10 sequences of a thousand images (per scene) with associated depth maps Kinect-acquired ground-truth poses. To evaluate our localization pipeline we follow the training and test splits from [51] and compare to methods such as RelocNet [3], which uses frustum overlap as a world-space measure.

4.1 Learning Surface Overlap

Figure 5 shows qualitative results of ground-truth surface overlaps and the predicted box overlaps between a random test image as query and training images as gallery. See supplementary materials for more examples.

In Table 4.2 we compare vector embeddings against box embeddings to experimentally validate that asymmetric surface overlap cannot be learned with a symmetric vector embedding. We evaluate the predictions against ground truth surface overlap on $1,000$ random pairs of test images. For each pair of images, we can compute ground truth $\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y})$ and $\mathsf{NSO}(\mathbf{y} \mapsto \mathbf{x})$. We report results on three metrics: L_1-Norm[1], the root mean square error (RMSE)[2] and the prediction accuracy. The prediction accuracy is defined as the percentage of individual overlaps that is predicted with an absolute error of less than 10%. Note, that the ground-truth depth information for the images may be incomplete, so there is inherent noise in the training signal and ground-truth measurements which makes smaller percentage thresholds not meaningful.

These results confirm that box embeddings are better than vectors at capturing the asymmetric world-space overlap relationships between images.

4.2 Interpreting Predicted Relations

For these experiments, we learn box embeddings for images in different scenes of the Megadepth dataset [27]. A trained network can be used to predict box

[1] L_1-Norm: $\frac{1}{N}\sum|\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y}) - \mathsf{NBO}(\mathbf{b_x} \mapsto \mathbf{b_y})| + |\mathsf{NSO}(\mathbf{y} \mapsto \mathbf{x}) - \mathsf{NBO}(\mathbf{b_y} \mapsto \mathbf{b_x})|$.

[2] RMSE: $\sqrt{\frac{1}{N}\sum[(\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y}) - \mathsf{NBO}(\mathbf{b_x} \mapsto \mathbf{b_y}))^2 + (\mathsf{NSO}(\mathbf{y} \mapsto \mathbf{x}) - \mathsf{NBO}(\mathbf{b_y} \mapsto \mathbf{b_x}))^2]}$.

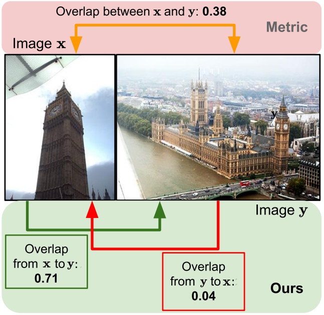

Overlap between x and y: 0.38
Metric
Image x
Image y
Overlap from x to y: 0.71
Overlap from y to x: 0.04
Ours

Fig. 3. Top: Metric learning can only represent a single distance between two images of Big Ben. Bottom: While 4% of image **y** is visible in image **x**, 71% of image **x** is visible vice versa. The average of overlaps is 38%.

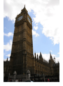
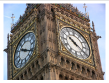
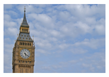
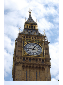
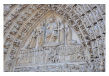
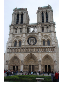
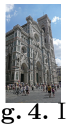
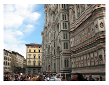
x	**y**	NSO ($\mathbf{x} \mapsto \mathbf{y}$)	NSO ($\mathbf{y} \mapsto \mathbf{x}$)	Interpretation
		15.2% (15.3%)	83.1% (83.5%)	**y** is zoom-in on **x**
		80.8% (87.7%)	88.7% (89.2%)	**x** and **y** are clone-like
		85.3% (86.9%)	5.3% (4.8%)	**y** is zoom-out of **x**
		23.7% (20.1%)	22.3% (28.5%)	**y** is oblique-out or crop-out of **x**

Fig. 4. Interpretation of image relationship based on the normalized surface overlap (NSO) between two images (predicted from just RGB with our approach and ground-truth using privileged 3D data). Four different relationships between image pairs can be observed. When NSO($\mathbf{x} \mapsto \mathbf{y}$) is low but NSO($\mathbf{y} \mapsto \mathbf{x}$) is high, this indicates that most pixels of **y** are visible in **x**. Therefore **y** must be a close-up of **x**.

representations for any image. We re-use our training data (which incidentally also have ground-truth depths) to form the gallery. Each query image **q** comes from the test-split. We compute its box representation $\mathbf{b_q}$ and compute surface overlap approximations using box representations as specified in Eq. (1) and (2).

First, we analyze predicted relations qualitatively. We introduce two terms to discuss the relations between query image **q** and retrieved image **r**: *enclosure* and *concentration*. Enclosure refers to the predicted surface overlap from query to retrieved image, *i.e.* $\mathsf{NBO}(\mathbf{b_q} \mapsto \mathbf{b_r})$. Concentration refers to the predicted surface overlap from retrieved image to query, *i.e.* $\mathsf{NBO}(\mathbf{b_r} \mapsto \mathbf{b_q})$. Thus, if a retrieved image has a large value for enclosure, then it observes a large number of the query's pixels. See Figure 4 for other interpretations.

To demonstrate the interpretability of the predicted relations between the images, we retrieve and show gallery images in different ranges of enclosure and concentration. Figure 5 shows results on two different scenes. On the left we see qualitative and quantitative results for the query image from the test data, and the images retrieved from the training set. As can be seen, the images in different quadrants of enclosure and concentration ranges are interpretable with different amounts of zoom-in, or zoom-out, or looking clone-like, or exhibiting crop-out/oblique-out. On the right, we retrieve images from the larger test set (without depth maps) plotted according to the estimated enclosure and concentration. This qualitative result demonstrates how box embeddings generalize to the test set. Please see supplementary materials for more examples.

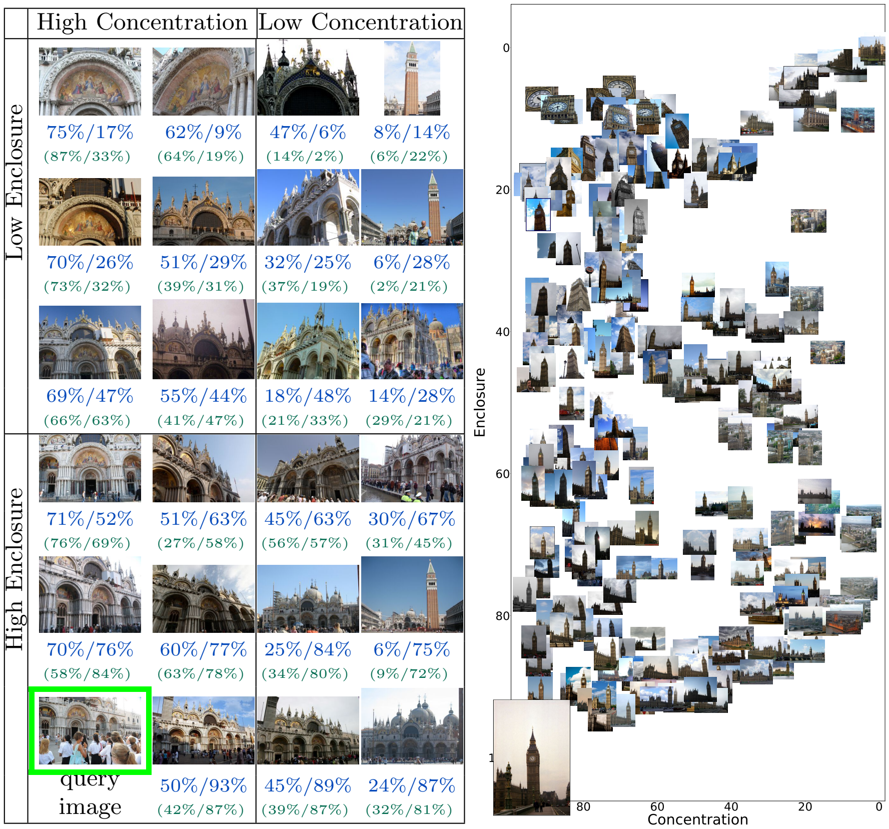


Fig. 5. Left: Results of predicted and ground-truth enclosure and concentration (defined in Sec 4.2) relative to the query image indicated by a green frame. The numbers below each image indicate the predicted and ground-truth concentration/enclosure. Right: Results of predicting enclosure and concentration between a query image from the test set and test images (including test images without depth maps). Images are plotted at the coordinates of predicted (enclosure, concentration). Note in both plots how results in the upper right quadrant are oblique views, and bottom-right show zoomed-out views. The upper left quadrants show zoomed-in views, depending on the range of (low) enclosure selected.

4.3 Querying Box Embeddings for Localization

We now compare the differences between surface overlap and frustum overlap measures for retrieving images in a localization task. The task is to find the camera pose of a query image $\mathbf{q}$ with respect to the images in the training set, where the latter have known camera poses and semi-dense depths. The image embeddings are used to retrieve the top k-th image ($k = 1$) from the training data that are closest according to each embedding measure. After retrieval, 2D−3D correspondences are found between the query image and k-th retrieved image's

Table 1. Evaluation of box *vs.* vector embeddings, trained on normalized surface overlap. We measure the discrepancy between the predicted and ground-truth overlaps on the test set. Across all measures and scenes, the non-metric embeddings with box representations outperform metric learning with vector representations

	$\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y})$ with boxes			$\mathsf{NSO}^{sym}(\mathbf{x}, \mathbf{y})$ with vectors		
	L_1-Norm	RMSE	Acc.$<$ 0.1	L1-Norm	RMSE	Acc.$<$ 0.1
Notre-Dame	0.070	0.092	93.3%	0.244	0.249	60.9%
Big Ben	0.126	0.138	82.6%	0.429	0.350	24.8%
Venice	0.066	0.112	89.9%	0.164	0.193	75.1%
Florence	0.063	0.094	89.7%	0.145	0.162	76.5%

Table 2. Results on 7Scenes dataset [51]. "Repr." indicates the embedding representation as box (B) or vector (V). Q specifies if the world space measure is symmetric (S) or asymmetric (A). Reported numbers show translation and rotation errors in meters and degrees respectively. The results indicate that symmetric surface overlap is superior to frustum overlap when represented with vectors. Asymmetric surface overlap box embeddings are similar to symmetric surface overlap vector embeddings, except for the Stairs scene. The last two rows show the generalization ability of our embeddings: the two embeddings were trained on Kitchen and used for retrieval on other scenes

Method	Repr.	Q	Chess	Fire	Heads	Office	Pumpkin	Kitchen	Stairs
DenseVLAD [58]	V	S	0.03, 1.40°	0.04, 1.62°	0.03, 1.21°	0.05, 1.37°	0.07, 2.02°	0.05, 1.63°	0.16, 3.85°
NetVLAD [1]	V	S	0.04, 1.29°	0.04, 1.85°	0.03, 1.14°	0.05, 1.45°	0.08, 2.16°	0.05, 1.77°	0.16, 4.00°
PoseNet [22]			0.32, 8.12°	0.47, 14.4°	0.29, 12.0°	0.48, 7.68°	0.47, 8.42°	0.59, 8.64°	0.47, 13.6°
RelocNet [3]			0.12, 4.14°	0.26, 10.4°	0.14, 10.5°	0.18, 5.32°	0.26, 4.17°	0.23, 5.08°	0.28, 7.53°
Active Search [42]			0.04, 1.96°	0.03, 1.53°	0.02, 1.45°	0.09, 3.61°	0.08, 3.10°	0.07, 3.37°	0.03, 2.22°
DSAC++ [6]			**0.02, 0.50**°	**0.02, 0.90**°	**0.01, 0.80**°	**0.03, 0.70**°	**0.04, 1.10**°	**0.04, 1.10**°	**0.09, 2.60**°
Ours $\mathsf{NSO}()$	B	A	0.05, 1.47°	**0.05, 1.91**°	0.05, 2.54°	**0.06**, 1.60°	0.10, 2.46°	0.07, 1.73°	0.50, 9.18°
Ours $\mathsf{NSO}^{sym}()$	V	S	**0.04, 1.19**°	**0.05**, 2.05°	0.05, 2.84°	**0.06, 1.46**°	**0.10, 2.28**°	**0.06, 1.61**°	**0.22, 5.28**°
Ours Frustum	V	S	0.05, 1.25°	**0.05**, 2.02°	**0.04, 1.86**°	0.07, 1.73°	**0.10**, 2.40°	0.07, 1.71°	1.82, 12.0°
Box (Kitchen)	B	A	0.06, 2.19°	0.08, 2.94°	0.08, 5.42°	0.17, 4.87°	0.13, 3.21°	*	1.83, 50.1°
Vector (Kitchen)	V	S	**0.04**, 1.54°	0.06, 2.26°	0.04, 1.90°	**0.06**, 1.88°	**0.10**, 2.37°	*	0.55, 9.22°

3D point cloud. We use SIFT [28] features, and correspondences are filtered with Lowe's ratio test and matched using OpenCV's FLANN-based matcher. The pose is solved with RANSAC and PnP, and measured against the ground truth pose. We report median translation and rotation errors, with all test images as queries.

We compare three embeddings for this task, each trained with a different world-space measure: i) vector embeddings trained with frustum overlap, ii) vector embeddings trained with $\mathsf{NSO}^{sym}()$, and box embeddings trained with $\mathsf{NSO}()$. Ranking for (i) and (ii) is easy, and for (iii) we rank retrieved images according to $0.5(\mathsf{NBO}(\mathbf{b_x} \mapsto \mathbf{b_y}) + \mathsf{NBO}(\mathbf{b_y} \mapsto \mathbf{b_x}))$. This query function is used to show backwards compatibility with traditional metric embeddings.

7Scenes Table 2 shows the results of using (i) frustum overlap and both (ii) symmetric and (iii) asymmetric surface overlap. We also report the results of state-of-the-art work [3, 6, 22, 42] and two SOTA baselines, DenseVLAD and NetVLAD, that we generated by swapping our retrieval system with the ones of [58] and [1] respectively and leaving the rest of the pose estimation pipeline intact. Both surface overlap-based results are generally better than frustum overlap. Note, absolute differences between most recent methods on this dataset are relatively minor. For example, due to the use of PnP, we get better results using our frustum overlap than RelocNet. Ours-box is similar in performance to Ours-vector, except for the Stairs dataset, where we perform poorly because stairs have many repeated structures. This is a positive result, showing on-par performance while giving the benefit of an interpretable embedding. Although the localization task includes having a 3D reconstruction of the gallery/training images by its nature, we also compare two embeddings that were trained on Kitchen and tested generalization for retrieval on other scenes. Please see the supplementary material for further experimental results.

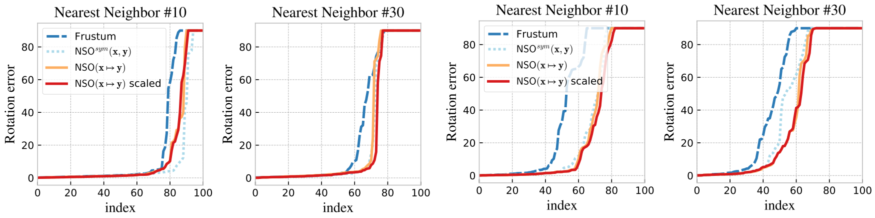
4.4 Predicting Relative Scale Difference for Image Matching

Given a pair of images $\mathbf{x}$ and $\mathbf{y}$, we can compare the predicted $\mathsf{NBO}(\mathbf{b_x} \mapsto \mathbf{b_y})$ and $\mathsf{NBO}(\mathbf{b_y} \mapsto \mathbf{b_x})$ to estimate the relative scale of the images, so

$$\frac{\mathsf{NBO}(\mathbf{b_x} \mapsto \mathbf{b_y})}{\mathsf{NBO}(\mathbf{b_y} \mapsto \mathbf{b_x})} \approx \frac{\mathsf{NSO}(\mathbf{x} \mapsto \mathbf{y})}{\mathsf{NSO}(\mathbf{y} \mapsto \mathbf{x})} = \frac{\mathsf{overlap}(\mathbf{x} \mapsto \mathbf{y})/N_{\mathbf{x}}}{\mathsf{overlap}(\mathbf{y} \mapsto \mathbf{x})/N_{\mathbf{y}}} \approx \frac{1}{s^2}\frac{N_{\mathbf{y}}}{N_{\mathbf{x}}}. \tag{9}$$

This estimates the scale factor s to be applied to image $\mathbf{x}$, so that overlaps $\mathsf{overlap}(\mathbf{x} \mapsto \mathbf{y})$ and $\mathsf{overlap}(\mathbf{y} \mapsto \mathbf{x})$ occupy approximately the same number of pixels in each of the two images.

Figure 7 shows qualitative examples of pairs of images from the test set, with predicted normalized box overlaps and estimated scale factor applied to one of the images. We observed that this relative scale estimate is in general accurate if the images have zoom-in/zoom-out relationships. However, if the images are in crop-out/oblique-out relation to one-another, then the rescaling may not necessarily make matching easier, as the overlap is already quite small. However, we can detect if images are in such a relationship by looking at predicted box overlaps (see Section 4.2). The estimated scale factor can be applied to the images before local feature detection and description extraction, to improve relative pose accuracy for pairs of images with large scale differences.



Notre-Dame

Big Ben

Fig. 6. Each plot shows (sorted) rotation error (capped at 90°) when each test image is matched against 10-th and 30-th closest retrieved image for pose estimation. As we can see, box embeddings with surface overlap measure tend to outperform alternatives, especially when rescaling images according to estimated relative scale.

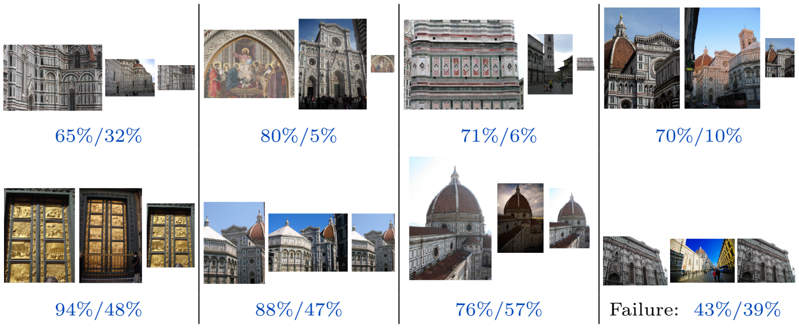


Fig. 7. Qualitative results of relative scale estimation on Florence test set. For each pair, the enclosure and concentration are calculated from which the relative estimated scaled can be derived. Based on that scale, the first image is resized and shown in the third position. If the relative scale is accurate, the content in the second and third images should match in size (number of pixels). The resized images are sometimes small, so the reader is encouraged to zoom into the images. The two numbers below each image pair show the estimated enclosure and concentration values.

We simulate this scenario by retrieving $k = 10$ and $k = 30$ closest matches for each embedding, and solving for the pose. Additionally, for box embeddings, we do feature matching with and without pre-scaling of images according to the predicted overlaps. Figure 6 shows results on Notre-Dame and Big Ben scenes.

Please see supplementary materials for further details.

5 Conclusions

We found surface overlap to be a superior measure compared to frustum overlap. We have shown that normalized surface overlap can be embedded in our new box embedding. The benefit is that we can now *easily* compute interpretable relationships between pairs of images, without hurting localization. Further, this can help with pre-scaling of images for feature-extraction, and hierarchical organization of images.

Limitations and Future work An obvious limitation is that we rely on expensive depth information for training. This could be addressed in two ways: either approximate the visual overlap with sparse 3D points and their co-visibility in two views, or train box embeddings with homography overlap of single images. Both of these approaches could also help with learning box embeddings that generalize across scenes, for image and object retrieval applications, as larger datasets could potentially be used for training.

Acknowledgements Thanks to Carl Toft for help with normal estimation, to Michael Firman for comments on paper drafts and to the anonymous reviewers for helpful feedback.

References

1. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: CVPR (2016) 2, 4, 5, 12
2. Arandjelovic, R., Zisserman, A.: All about VLAD. In: CVPR (2013) 4
3. Balntas, V., Li, S., Prisacariu, V.: RelocNet: Continuous metric learning relocalisation using neural nets. In: ECCV (2018) 1, 2, 4, 5, 9, 12
4. Balntas, V., Riba, E., Ponsa, D., Mikolajczyk, K.: Learning local feature descriptors with triplets and shallow convolutional neural networks. In: BMVC (2016) 4
5. Bonin-Font, F., Ortiz, A., Oliver, G.: Visual navigation for mobile robots: A survey. Journal of intelligent and robotic systems (2008) 1
6. Brachmann, E., Rother, C.: Learning less is more-6D camera localization via 3D surface regression. In: CVPR (2018) 4, 12
7. Buehler, C., Bosse, M., McMillan, L., Gortler, S., Cohen, M.: Unstructured lumigraph rendering. In: Computer graphics and interactive techniques (2001) 5
8. Cakir, F., He, K., Xia, X., Kulis, B., Sclaroff, S.: Deep metric learning to rank. In: CVPR (2019) 4
9. DeTone, D., Malisiewicz, T., Rabinovich, A.: Deep image homography estimation. arXiv preprint arXiv:1606.03798 (2016) 3
10. Dufournaud, Y., Schmid, C., Horaud, R.: Image matching with scale adjustment. Computer Vision and Image Understanding (2004) 2
11. Erlik Nowruzi, F., Laganiere, R., Japkowicz, N.: Homography estimation from image pairs with hierarchical convolutional networks. In: ICCVW (2017) 3
12. Frahm, J.M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.H., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M.: Building rome on a cloudless day. In: ECCV (2010) 2, 3
13. Gálvez-López, D., Tardós, J.D.: Bags of binary words for fast place recognition in image sequences. IEEE Transactions on Robotics (2012) 3
14. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: Deep image retrieval: Learning global representations for image search. In: ECCV (2016) 4
15. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: ACM SIGMOD international conference on Management of data (1984) 8
16. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVPR (2006) 4
17. Hartley, R.I.: In defense of the eight-point algorithm. TPAMI (1997) 2
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 8
19. He, K., Lu, Y., Sclaroff, S.: Local descriptors optimized for average precision. In: CVPR (2018) 4
20. Heinly, J., Schönberger, J.L., Dunn, E., Frahm, J.M.: Reconstructing the world* in six days. In: CVPR (2015) 3
21. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person reidentification. arXiv:1703.07737 (2017) 4
22. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: A convolutional network for realtime 6-DOF camera relocalization. In: ICCV (2015) 4, 5, 12
23. Kulkarni, T.D., Whitney, W.F., Kohli, P., Tenenbaum, J.: Deep convolutional inverse graphics network. In: NeurIPS (2015) 5
24. Laskar, Z., Melekhov, I., Kalia, S., Kannala, J.: Camera relocalization by computing pairwise relative poses using convolutional neural network. In: ICCV (2017) 4

25. Le, H., Liu, F., Zhang, S., Agarwala, A.: Deep homography estimation for dynamic scenes. In: CVPR (2020) 3
26. Li, X., Vilnis, L., Zhang, D., Boratko, M., McCallum, A.: Smoothing the geometry of probabilistic box embeddings. In: ICLR (2019) 2, 5, 7
27. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: CVPR (2018) 8, 9
28. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004) 12
29. Mikulik, A., Chum, O., Matas, J.: Image retrieval for online browsing in large image collections. In: International Conference on Similarity Search and Applications (2013) 2, 3
30. Mikulík, A., Radenović, F., Chum, O., Matas, J.: Efficient image detail mining. In: ACCV (2014) 1, 2, 3
31. Mishchuk, A., Mishkin, D., Radenovic, F., Matas, J.: Working hard to know your neighbor's margins: Local descriptor learning loss. In: NeurIPS (2017) 4
32. Mishkin, D., Matas, J., Perdoch, M.: Mods: Fast and robust method for two-view matching. Computer Vision and Image Understanding (2015) 4
33. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Transactions on Robotics (2017) 3
34. Nguyen, T., Chen, S.W., Shivakumar, S.S., Taylor, C.J., Kumar, V.: Unsupervised deep homography: A fast and robust homography estimation model. IEEE Robotics and Automation Letters (2018) 3
35. Nickel, M., Kiela, D.: Poincaré embeddings for learning hierarchical representations. In: NeurIPS (2017) 5
36. Perd'och, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: CVPR (2009) 3
37. Perronnin, F., Liu, Y., Sánchez, J., Poirier, H.: Large-scale image retrieval with compressed Fisher vectors. In: CVPR (2010) 4
38. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007) 4
39. Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csurka, G., Cabon, Y., Humenberger, M.: R2d2: Repeatable and reliable detector and descriptor. In: NeurIPS (2019) 4
40. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2D-to-3D matching. In: ICCV (2011) 1, 4
41. Sattler, T., Leibe, B., Kobbelt, L.: Improving image-based localization by active correspondence search. In: ECCV (2012) 4
42. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & effective prioritized matching for large-scale image-based localization. TPAMI (2016) 1, 4, 12
43. Sattler, T., Torii, A., Sivic, J., Pollefeys, M., Taira, H., Okutomi, M., Pajdla, T.: Are large-scale 3D models really necessary for accurate visual localization? In: CVPR (2017) 4
44. Sattler, T., Weyand, T., Leibe, B., Kobbelt, L.: Image retrieval for image-based localization revisited. In: BMVC (2012) 1, 4
45. Sattler, T., Zhou, Q., Pollefeys, M., Leal-Taixe, L.: Understanding the limitations of cnn-based absolute camera pose regression. In: CVPR (2019) 4
46. Schönberger, J.L., Berg, A.C., Frahm, J.M.: Paige: pairwise image geometry encoding for improved efficiency in structure-from-motion. In: CVPR (2015) 3
47. Schönberger, J.L., Radenovic, F., Chum, O., Frahm, J.M.: From single image query to detailed 3D reconstruction. In: CVPR (2015) 3

48. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016) 1, 3
49. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR (2015) 4
50. Shen, T., Luo, Z., Zhou, L., Zhang, R., Zhu, S., Fang, T., Quan, L.: Matchable image retrieval by learning from surface reconstruction. In: ACCV (2018) 1, 2, 3, 4, 7
51. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in RGB-D images. In: CVPR (2013) 8, 9, 12
52. Snavely, N., Garg, R., Seitz, S.M., Szeliski, R.: Finding paths through the world's photos. ACM Transactions on Graphics (2008) 1, 5
53. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. In: SIGGRAPH (2006) 1, 3, 5
54. Sohn, K., Lee, H.: Learning invariant representations with local transformations. In: ICML (2012) 5
55. Stewénius, H., Gunderson, S.H., Pilet, J.: Size matters: exhaustive geometric verification for image retrieval. In: ECCV (2012) 3
56. Subramanian, S., Chakrabarti, S.: New embedded representations and evaluation protocols for inferring transitive relations. In: SIGIR Conference on Research & Development in Information Retrieval (2018) 2, 5, 7
57. Tian, Y., Yu, X., Fan, B., Wu, F., Heijnen, H., Balntas, V.: Sosnet: Second order similarity regularization for local descriptor learning. In: CVPR (2019) 4
58. Torii, A., Arandjelovic, R., Sivic, J., Okutomi, M., Pajdla, T.: 24/7 place recognition by view synthesis. In: CVPR (2015) 12
59. Vendrov, I., Kiros, R., Fidler, S., Urtasun, R.: Order-embeddings of images and language. In: ICLR (2016) 5
60. Vilnis, L., Li, X., Murty, S., McCallum, A.: Probabilistic embedding of knowledge graphs with box lattice measures. In: ACL (2018) 2, 5, 7
61. Vilnis, L., McCallum, A.: Word representations via gaussian embedding. In: ICLR (2015) 5
62. Weyand, T., Leibe, B.: Discovering favorite views of popular places with iconoid shift. In: ICCV (2011) 3
63. Weyand, T., Leibe, B.: Discovering details and scene structure with hierarchical iconoid shift. In: ICCV (2013) 3
64. Witkin, A.P.: Scale-space filtering. In: IJCAI (1983) 2
65. Worrall, D.E., Garbin, S.J., Turmukhambetov, D., Brostow, G.J.: Interpretable transformations with encoder-decoder networks. In: ICCV (2017) 5
66. Zhou, L., Zhu, S., Shen, T., Wang, J., Fang, T., Quan, L.: Progressive large scale-invariant image matching in scale space. In: ICCV (2017) 2, 4