D-Nets: Beyond Patch-Based Image Descriptors

Felix von Hundelshausen[1] Rahul Sukthankar[2]

felix.v.hundelshausen@live.de rahuls@cs.cmu.edu

[1]Institute for Autonomous Systems Technology (TAS), University of the Bundeswehr Munich

[2]Google Research and The Robotics Institute, Carnegie Mellon University

https://sites.google.com/site/descriptornets/

Abstract

Despite much research on patch-based descriptors, SIFT remains the gold standard for finding correspondences across images and recent descriptors focus primarily on improving speed rather than accuracy. In this paper we propose Descriptor-Nets (D-Nets), a computationally efficient method that significantly improves the accuracy of image matching by going beyond patch-based approaches. D-Nets constructs a network in which nodes correspond to traditional sparsely or densely sampled keypoints, and where image content is sampled from selected edges in this net. Not only is our proposed representation invariant to cropping, translation, scale, reflection and rotation, but it is also significantly more robust to severe perspective and non-linear distortions. We present several variants of our algorithm, including one that tunes itself to the image complexity and an efficient parallelized variant that employs a fixed grid. Comprehensive direct comparisons against SIFT and ORB on standard datasets demonstrate that D-Nets dominates existing approaches in terms of precision and recall while retaining computational efficiency.

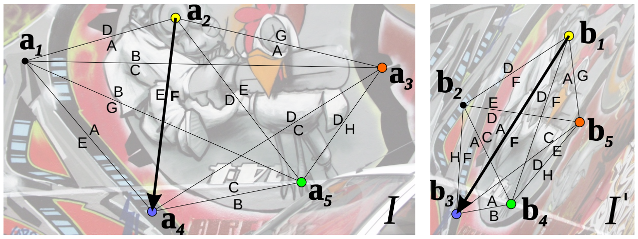

Figure 1. **Illustrative example:** D-Nets are graphs formed by pairwise connections over nodes. In practice, a D-Net may contain thousands of nodes, of which only a few are shown. The two directed connections between each pair of nodes are shown as a single line. Correct connection matches are marked with colored terminal nodes. The quantized "d-tokens" describing the image content along each strip are denoted with capital letters. One directed connection is highlighted for further discussion in the text.

1. Introduction

Image matching is a fundamental building block for a variety of computer vision tasks, including multi-view 3D reconstruction, tracking, object recognition and content-based image retrieval. In the last decade, keypoint-based methods employing patch-based descriptors, exemplified by the SIFT algorithm [9], have emerged as the standard approach to the problem. Extensive quantitative experiments using a variety of detectors [11] and descriptors [10] suggest that consistently outperforming SIFT in terms of precision and recall is extremely difficult. Consequently, the focus of research on descriptors has largely shifted to matching SIFT's accuracy under much stricter computational constraints. Examples of this trend include SURF [2], FERNS [14], and most recently, BRIEF [3], ORB [15] and BRISK [8]. Notably, ORB, which combines a FAST [17] corner detector on an image pyramid with a rotation-invariant version of BRIEF achieves a 100× speed-up over SIFT while approximating the accuracy of the original algorithm.

In our work, we explore alternatives to patch-based descriptors for image matching and aim to show significant improvements in terms of precision and recall in direct comparisons against SIFT-like algorithms. Rather than representing an image using descriptors computed over a set of disconnected patches, we advocate an approach that extracts low-level information over the edges in a network of connected nodes (Figure 1). This enables our features not only to capture local properties such as image gradients and texture information, but also to place these measurements in a relative spatial context defined by pairwise node connectivity. Our approach, termed Descriptor-Nets (D-Nets) differs from existing work in two fundamental respects. First, we abandon patches entirely in favor of "strips" (paths connecting nodes). Second, we express spatial information in a topological rather than geometric manner, which enables our approach to be highly robust to nonlinear image distortions. We summarize our paper's contributions as follows:

- A novel image representation that exhibits significant improvements over patch-based descriptors, such as SIFT and ORB, in both precision and recall, on standard datasets.
- A method for image matching that dynamically adapts to the difficulty of each case, such that simpler matching tasks can be solved with less computational effort than difficult cases.
- A keypoint-free (i.e. dense feature) representation that, unlike existing approaches [18], maintains invariance to reflection, scale and rotation. It is also noteworthy in that our dense network can be precomputed and is particularly amenable to fast and massively-parallel implementation.

2. Descriptor-Nets (D-Nets)

We present three variants of the Descriptor-Nets approach: In the first, the net is a fully connected graph over nodes generated using an interest-point operator (*clique D-Nets*); in the second, we show that full connectivity is not always required, leading to an iterative version (*iterative D-Nets*) that dynamically constructs links only as necessary; and in the third, we show that key-points are not required, so that nodes can simply be densely sampled over a regular grid that is independent of image content (*densely sampled D-Nets*). Due to space considerations, we detail only the first variant in this section and summarize the other two variants along with their respective experiments.

2.1. Fully Connected (Clique) Descriptor-Nets

We begin by introducing the most straightforward D-Nets variant, where nodes corresond to interest points and links connect each pair of nodes. Figure 1 presents a simple example (where only an illustrative subset of nodes and links are shown).

Consider evaluating a match between two images, I and I', whose D-Nets consist of the *directed* graphs $G = (\mathcal{V}, \mathcal{E})$ and $G' = (\mathcal{V}', \mathcal{E}')$, respectively. Let $\mathcal{V} = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ and $\mathcal{V}' = \{\mathbf{b}_1, \ldots, \mathbf{b}_{n'}\}$ denote nodes in the respective images, and $\mathcal{E}, \mathcal{E}'$ the edges of those graphs. To avoid confusion with image edges, we refer to these edges as "connections". In the simplest D-Nets variant, the nodes are fully connected, with $\mathcal{E} = \{(\mathbf{a}_i, \mathbf{a}_j) | i \neq j\}$ and $\mathcal{E}' = \{(\mathbf{b}_i, \mathbf{b}_j) | i \neq j\}$.

We refer to the image region (consisting of raw pixels) under such a connection as a "strip". The simplest formulation for a strip is the directed straight line segment between two interest points in the image. In the D-Nets approach, image matching is built up by matching connections (rather than nodes) across images. Two connections are defined to be a correct match, if and only if their start and end nodes correspond to the same physical points in the two images. In Figure 1, the connections $(\mathbf{a}_2, \mathbf{a}_4)$ and $(\mathbf{b}_1, \mathbf{b}_3)$ are correct matches because nodes $\mathbf{a}_2 \leftrightarrow \mathbf{b}_1$ and $\mathbf{a}_4 \leftrightarrow \mathbf{b}_3$ match.

A traditional patch-based approach would independently infer matches between corresponding keypoints and optionally perform geometric verification on sets of keypoints. In D-Nets, we determine matches using image content in their corresponding strips and directly aggregate this information at the image level using hashing and voting, as described later.

2.2. Image pyramids

We employ image pyramids to construct a multi-scale representation of the image content, which not only reduces computational complexity for longer strips but enables them to have a broader swath. A pyramid consists of L levels, where the first level is simply the original image smoothed with a Gaussian kernel with σ=1.0. Subsequent levels are scaled versions of the first level, with a scaling factor of $f = (1/L)^{\frac{1}{L-1}}$. Thus, the final level of the pyramid is $1/L$ the size of the original image.[1]

We also define a level index for each strip according to its length l in the original image as: $i(l) = \langle \log_f(8s/l) \rangle$, where $\langle . \rangle$ discretizes its argument into the valid range of indices $i(l) \in \{0, \ldots, L-1\}$ and s is a parameter described in the next subsection. Thus, long strips map to coarse levels of the pyramid and are thus also correspondingly broader than their short counterparts.

2.3. Describing image content in a strip

We define a discrete descriptor (termed a "d-token") for each $e \in \mathcal{E}$ based on its strip in the image. The pixel intensities at the strip's level index are used to generate a d-token $d \in \mathcal{D}$.

We have experimented with a variety of descriptors, including binary comparisons among pixels in the strip (similar to those in Ferns and ORB), quantization using binary frequency comparisons on a 1D Fourier transform of strip pixels and wavelet-based approaches. Given space limitations, we detail only our best descriptor, which (like SIFT) is manually engineered. It is important to note that the D-Nets approach does not require this particular descriptor and that employing more straightforward descriptors is almost as good.

Consider the (directed) connection from node $\mathbf{a}_i \to \mathbf{a}_j$, whose length in the original image is given by $l = ||\mathbf{a}_i - \mathbf{a}_j||_2$. As shown above, we determine the appropriate pyramid level $i(l)$, whose associated scale factor is given by $f^{i(l)}$. At that pyramid level, the strip corresponding to this connection has length $\bar{l} = l f^{i(l)}$ and goes between the scaled points $\bar{\mathbf{a}}_i$ and $\bar{\mathbf{a}}_j$. The d-token is constructed as follows:

[1] Our implementation uses L=8 and interpolates during downsizing.

1. Sample pixel intensities from $\bar{l}$ equally spaced points along the 10% to 80% portion of the strip at this pyramid level, i.e., from $\bar{\mathbf{a}}_i + 0.1(\bar{\mathbf{a}}_j - \bar{\mathbf{a}}_i)$ to $\bar{\mathbf{a}}_i + 0.8(\bar{\mathbf{a}}_j - \bar{\mathbf{a}}_i)$. Briefly, the reasoning behind this unusual operation is that we seek to make the representation less sensitive to the positions of start- and end-nodes. Our experiments showed that omitting the ends of the strip is beneficial, particularly when keypoint localization is noisy, and that asymmetry is also desirable.
2. Group this sequence of values into a smaller set of s uniform chunks, averaging the pixel intensities in each chunk to reduce noise and generate an s-dimensional vector.
3. Normalize this vector using scaling and translation s.t. $\min_i s_i = 0$ and $\max_i s_i = 1$. In the improbable event that $\forall_{i,j}(s_i = s_j)$, we set $\forall_i s_i = 0.5$.
4. Uniformly discretize each value in the normalized s vector using b bits.
5. Concatenate the s sections, each with b bits to obtain a discrete d-token.

Three subtle points in the steps above merit further discussion: 1) unlike patch-based methods, the d-tokens descriptor samples intensities from image regions both near and far from the interest point and encodes this information in a form that is independent of scale and robust to lighting; 2) the asymmetry beween the two d-tokens for the same pair of nodes is intentional and enables each to capture different parts of the image (with some overlap); 3) d-tokens are very fast to compare and amenable to efficient hashing. Summarizing, each directed connection is expressed by a d-token, which (in the presented case) is a simply a $s \cdot b$ bit string that can be represented as an integer.

Although the proposed d-token descriptor is not localized to a small patch in the scene, it possesses all of the desirable invariance properties. Translation invariance is achieved because the strip is anchored to interest points rather than absolute coordinates. Rotational invariance is automatically ensured because the descriptor extracts information over a 1D strip rather than a 2D patch. This is a subtle but important advantage over patch-based schemes, where patches require explicit derotation so as to align dominant orientations to a canonical direction (and where incorrect derotations are harmful). Additionally, and in contrast to patch-based descriptors, d-tokens are automatically invariant to reflections since they operate on 1-D strips of pixels. Scale and affine invariance is ensured because every connection is represented using s segments and robustness to illumination through the use of a normalized b bit quantization for the average intensity of a segment. While the method is not intrinsically invariant to perspective, the fact that we do not explicitly enforce geometric consistency between connections enables very high robustness to globally nonlinear transformations. Indeed, as seen in our experimental results, D-Nets is particularly robust to image pairs with significant perspective distortions.

To better explain the preceding concepts, we continue discussing the scaled-down illustrative example shown in Figure 1. To keep numbers low, let us employ a small d-token generation scheme with s=3 sections per connection, each quantized to just a single b=1 bit. This gives us a vocabulary of 2^{sb}=8 possible d-tokens, which we denote as $\mathcal{D} = \{A, B, C, \ldots, H\}$.

Each connection in Figure 1 is shown as annotated with its d-token from this set; since the D-Nets connections are directed and the descriptor is asymmetric, we show two d-tokens (one for each directed connection). The next section details how D-Nets uses these d-tokens for efficient image matching.

2.4. Efficient Matching using Hashing and Voting

Continuing the example in Figure 1, we now show how the d-token descriptors (described above) enable efficient image matching. Given the nodes $\mathcal{V} = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ in image I and the nodes $\mathcal{V}' = \{\mathbf{b}_1, \ldots, \mathbf{b}_{n'}\}$ in image I', we seek to identify node correspondences. This is done by matching the directed connections between nodes in an image, each described by a discrete d-token, using a voting procedure in conjunction with a specialized hash table.

Our hash table has $|\mathcal{D}|$ bins — i.e., one for each distinct d-token $d \in \mathcal{D}$, which serves as the discrete index for the table. Conceptually, the table hashes a given D-Nets connection (based on its image content) to its appropriate bin. However, unlike a standard hash table, each bin d in our table contains two separate lists, one for each image to be matched. These lists simply enumerate the connections from each image that hash to the given bin (i.e., those connections with d-token=d). We limit the lengths of each lists to n_L=20, discarding any connections that hash to a full list. This design has two benefits: 1) it bounds the memory usage of our table, even when an image contains millions of connections; 2) analogous to the use of stop-word lists in information retrieval, it limits the impact of frequently repeating but uninformative content.

Continuing the example, Figure 2 shows our hash table (with 8 bins) after all connections from Figure 1 have been inserted.

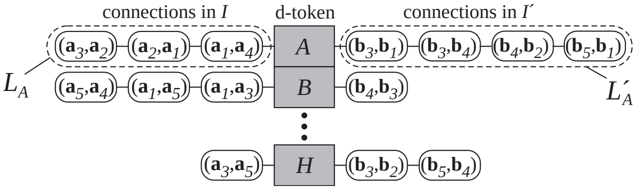
connections in I
d-token
connections in I'
$(\mathbf{a}_3,\mathbf{a}_2)$
$(\mathbf{a}_2,\mathbf{a}_1)$
$(\mathbf{a}_1,\mathbf{a}_4)$
A
$(\mathbf{b}_3,\mathbf{b}_1)$
$(\mathbf{b}_3,\mathbf{b}_4)$
$(\mathbf{b}_4,\mathbf{b}_2)$
$(\mathbf{b}_5,\mathbf{b}_1)$
L_A
$(\mathbf{a}_5,\mathbf{a}_4)$
$(\mathbf{a}_1,\mathbf{a}_5)$
$(\mathbf{a}_1,\mathbf{a}_3)$
B
$(\mathbf{b}_4,\mathbf{b}_3)$
L'_A
$(\mathbf{a}_3,\mathbf{a}_5)$
H
$(\mathbf{b}_3,\mathbf{b}_2)$
$(\mathbf{b}_5,\mathbf{b}_4)$

Figure 2. Hash table containing D-Nets connections from I and I'.

Using this hashtable, our voting algorithm casts votes into a two-dimensional *hypothesis correspondence grid*, $G[i,j] \in \Re$, where the cell $G[i,j]$ accumulates votes supporting the hypothesis that node $\mathbf{a}_i$ in image I corresponds to $\mathbf{b}_j$ in image I'. That is, G has $|\mathcal{V}| \times |\mathcal{V}'|$ cells.

In our example, we iterate over the d-tokens $d \in \{A, \ldots, H\}$ considering each bin in Figure 2, one at a time. In the first iteration (d=A), the bin contains 3 connections from image I and 4 connections from image I'. Any connection from the first list could correspond to any connection from the second list, resulting in the following set $\mathcal{B}_A$ of 12 hypotheses for correspondences across connections:

$$\mathcal{B}_A = \{((\mathbf{a}_3, \mathbf{a}_2), (\mathbf{b}_3, \mathbf{b}_1)), ((\mathbf{a}_3, \mathbf{a}_2), (\mathbf{b}_3, \mathbf{b}_4)), \\ \ldots, ((\mathbf{a}_1, \mathbf{a}_4), (\mathbf{b}_4, \mathbf{b}_2)), ((\mathbf{a}_1, \mathbf{a}_4), (\mathbf{b}_5, \mathbf{b}_1))\}.$$

Since our connections are directed, each correspondence hypothesis over connections implies a correspondence between nodes. For instance, the first element of $\mathcal{B}_A$, states $\mathbf{a}_3 \leftrightarrow \mathbf{b}_3$ and $\mathbf{a}_2 \leftrightarrow \mathbf{b}_1$. We represent this as votes in favor of $G[3,3]$ and $G[2,1]$. The strength of each vote is inversely proportional to the number of hypotheses in the given bin ($|\mathcal{B}_A|$=12). This normalization mutes the effect of popular strips and rewards rarer but more discriminative matches. It is also consistent with our design choice to limit the size of hash table lists (n_L). Votes for hypotheses involving large lists are low and thus limiting such lists enables efficient computation without substantially impacting matching accuracy. The procedure is formalized in Algorithm 1.

Algorithm 1 D-Nets Voter(G, L_i, L'_i)

```
clear $G$
for all $d \in D$ do
    $\mathcal{B}_d := \{(\mathbf{e}, \mathbf{e}') | \mathbf{e} \in L_d, \mathbf{e}' \in L'_d\}$
    $v \leftarrow 1/|\mathcal{B}_d|$
    for all $((\mathbf{a}_i, \mathbf{a}_j), (\mathbf{b}_k, \mathbf{b}_l)) \in \mathcal{B}_d$ do
        $G[i,k] \leftarrow G[i,k] + v$
        $G[j,l] \leftarrow G[j,l] + v$
    end for
end for
```

In practice, $|\mathcal{D}|$ can be quite large and many bins in the hashtable contain at least one empty list. Since such bins contain no valid hypotheses, they are skipped without casting any votes. The same occurs if the two images are very different, such that few d-tokens occur in both images. Thus, the running time of Algorithm 1 is highly dependent on the difficulty of the matching task. Image pairs that are unlikely to match (the vast majority) are quickly eliminated while computation time is focused on the harder cases.

2.5. Quality measure for ranking matches

In order to evaluate D-Nets in terms of precision-recall, we must associate a quality metric for each match, by which matches can be sorted. Denoting the values in row i^* of the correspondence grid as $g(j) := G[i^*, j]$, we select $j^* := \arg_j(g^* := \max g(j))$ as the best match for i^*. We define a quality measure q for a chosen correspondence $i^* \leftrightarrow j^*$ as its entropy-normalized vote: $q := \frac{g^*}{-\sum_{\forall j} p_j \log_2(p_j)}$, where $p_j := \frac{g(j)}{\sum_{\forall k} g(k)}$.

3. Experimental Results

This section summarizes results from three sets of experiments. All of these experiments employed the eight standard datasets[2] that have also been used in recent papers [10, 15] to enable direct comparisons with current approaches. Each data set consists of 6 images of the same scene under different ground-truthed conditions, such as translation, rotation and scale (*boats* and *bark*), viewpoint changes with perspective distortions (*graffiti* and *wall*), blurring (*bikes* and *trees*), lighting variation (*leuven*) and JPEG compression artifacts (*ubc*).

In each set the first image is matched against the 5 others, with increasing matching difficulty. In our discussion, specific cases are denoted as "*bark1-6*", meaning that image 1 is matched against image 6 in the *bark* dataset. We have made the source code of our basic D-Nets implementation publicly available[3] to enable others to replicate our results.

3.1. Evaluation Criteria: Strict/Loose Overlap

Following standard practice, we evaluate using 1-precision vs. recall graphs. Correspondences are sorted according to their feature-space distance (SIFT, ORB) or quality measure (D-Nets) and the fraction of correct correspondences[4] is plotted against the fraction of incorrect ones,[5] while iterating through this list. To determine whether a correspondence $\mathbf{a} \leftrightarrow \mathbf{b}$ is correct, we define two matching criteria based on the overlap error [11]. In *loose matching*, the correspondence $\mathbf{a} \leftrightarrow \mathbf{b}$ is deemed correct as long as the overlap error falls under the threshold ϵ_o, regardless of other matches $\mathbf{a}' \leftrightarrow \mathbf{b}$. *Strict matching* enforces a one-to-one constraint so that a match is correct if $\mathbf{b}$ is geometrically the closest point with sufficient overlap, and (if ties exist) the one closest in feature space/quality measure. Both schemes are detailed below; in all cases, we use an overlap threshold of ϵ_o=0.4.

Let $\mathbf{a} \leftrightarrow \mathbf{b} \in \mathcal{M} := \{\mathbf{a}_i \leftrightarrow \mathbf{b_j} | e_{\mathbf{H}}(\mathbf{a_i}, \mathbf{b_j}) < \epsilon_o\}$, where $e_{\mathbf{H}}(\mathbf{a_i}, \mathbf{b_j})$ denotes the overlap error using the ground-truth homography $\mathbf{H}$ relating images I and I'. Note that $\mathcal{M}$ can contain several pairs $\mathbf{a}_i \leftrightarrow \mathbf{b}$ that map to the same point $\mathbf{b}$,

[2] http://www.robots.ox.ac.uk/~vgg/research/affine/

[3] https://sites.google.com/site/descriptornets/

[4] The fraction is computed in relation to the maximum number of possible correct correspondences, determined using the ground truth.

[5] The fraction is computed against the number of matches returned by the matching procedure, including both correct and incorrect ones.

even though SIFT/ORB/D-Nets are restricted to returning only the single best match for a point $\mathbf{a}_i$ in image I.

Under loose matching, if several keypoints are geometrically very close to the ideal correspondence in terms of overlap, a match is counted as correct if it fulfills the overlap criterion, even if it is not geometrically the closest.

In contrast, for the strict evaluation scheme, we only consider the correspondence with the best overlap to be a possible match, i.e., we verify that: $\mathbf{a} \leftrightarrow \mathbf{b} \in \mathcal{M}_{\text{strict}}$, where

$$\mathcal{M}_{\text{strict}} := \left\{ \mathbf{a}_k \leftrightarrow \mathbf{b} \in \mathcal{M} | k = \arg\min_i (e_{\mathbf{H}}(\mathbf{a}_i, \mathbf{b})) \right\}.$$

In cases where multiple points $\mathbf{a}_i$ all map to the same point $\mathbf{b}$, strict matching specifies that we accept only the $\mathbf{a}_i$ with smallest feature distance (SIFT, ORB) or highest quality (D-Nets). For computing the 1-precision vs. recall graphs, the number of returned correct matches needs to be related to the total number of possible correct matches, which is $|\{\mathbf{a} \in \mathcal{V} | \exists \mathbf{b} \in \mathcal{V}' : \mathbf{a} \leftrightarrow \mathbf{b} \in \mathcal{M}\}|$ for loose matching and $|\mathcal{M}_{\text{strict}}|$ for strict matching.

We generalize the overlap error [11] to non-patch-based descriptors as follows. Consider $\mathbf{a}_i$ in I and $\mathbf{b}_j$ in I' (related via ground-truth homography $\mathbf{H}$). First, we map $\mathbf{b}_j$ to I using $\mathbf{H}^{-1}$. Next, we take circular[6] regions (30 pixel radius) around each of the two points in I and map them to I' using $\mathbf{H}$; this is done as a finely sampled polygon since $\mathbf{H}$ may not be affine. Finally, the overlap error (1 - ratio of areas of intersection to the union) between the corresponding polygons in I' is computed and compared against ϵ_o.

3.2. Experiment 1: Comparison vs. SIFT and ORB

We directly compare D-Nets against SIFT [9] and the very recent ORB [15] algorithms. To ensure that we fairly compare descriptors, we use the same interest points, extracted using SIFT, for all three methods. For D-Nets, we discard the scale and orientation information and simply use the keypoint location as our nodes. In this set of experiments, we employ the d-tokens representation described earlier on a fully connected D-Nets graph.

Results Figures 3 (a) and (b) show results for strict and loose matching, respectively. D-Nets (blue) significantly outperforms both SIFT (red) and ORB (black) in recall and precision for the strict matching. The difference under the loose matching criterion (Fig. 3b), is even more dramatic. We attribute D-Nets' success under such conditions to its ability to exploit image information from a larger portion of the image. Unlike patch-based descriptors, which are myopic and very sensitive to the patch location, the strip over which D-Nets extracts image content is resilient to changes in endpoint location, generating d-tokens that continue to match. The fact that D-Nets' more "global" features can thrive under conditions where patches fail may seem surprising, given that aggregates of local features have long been lauded for their resilience to drastic image changes. The explanation is that D-Nets features are also local, but the region over which they extract information (long 1-D strips between keypoints) enables us to overcome patch myopia. Robustness to spatial imprecision is highly desirable and this experiment shows the substantial benefits of switching from patch-based descriptors to the D-Nets representation.

Implementation Details To enable replication of our results, we provide the full details of our experimental setup. For ORB and SIFT we use the OpenCV 2.3.1 implementation with the following parameter settings.

SIFT-keypoints	SIFT descriptors	ORB descriptors	D-Nets
nOctaves=4	magnification=3	firstLevel=0	σ=1
nOctaveLayers=3	isNormalized=true	n_levels=8	L=8
firstOctave=-1	recalculateAngles=true	scale_factor=1.346	s=13
angleMode=0			b=2
threshold=0.04			q_0=0.1
edgeThreshold=10			q_1=0.8

To enable the ORB descriptors to work with the SIFT keypoints, some care had to be taken, because the OpenCV 2.3.1 code discards keypoints at the stage of ORB-descriptor extraction. Hence, we made sure to also exclude such keypoints from the other approaches. We also had to discard SIFT keypoints within a 15 pixel distance of the image border (scaled respectively for the other scale levels of the pyramid) to allow the ORB-descriptors work on SIFT-keypoints and we computed the pyramid index for ORB from the size of the respective SIFT patch in the non-scaled image. Although D-Nets can employ nodes at the border of the image (unlike patch-based descriptors), we choose to enforce consistency over maximizing the performance of our method and restrict ourselves to exactly the same keypoints as SIFT and ORB.

3.3. Experiment 2: Iterative D-Nets

The nodes of the D-Nets in the first experiment were fully connected. But is this degree of connectivity required? Motivated by this question we propose an iterative version of D-Nets that dynamically increases its connection density. One of the important contributions of this paper is that we can provide a stopping criterion that automatically determines the optimal connection density for any given matching task, such that simple cases are matched with less effort.

Our iterative implementation of D-Nets starts with the edges of a triangulation over the nodes as its initial set of connections. In each iteration, we expand each node's connections by one hop transitions (i.e. k-hops in k iterations). The hash table and voting grid are incrementally updated through the iterations.

[6] The motivation for using isotropic regions in I is because the first image in the datasets is typically a frontal view of a quasi-planar scene.

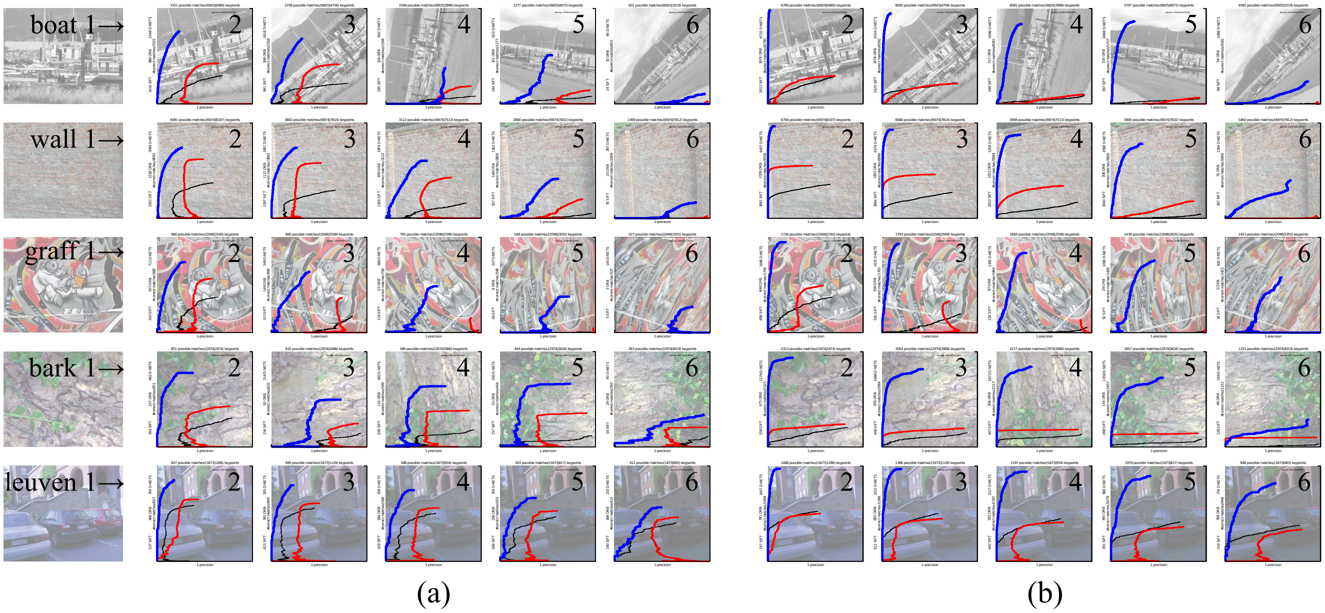

Figure 3. **Experiment 1.** 1-precision vs. recall for SIFT (red), ORB (black) and D-Nets (blue) using strict (a) and loose (b) matching on standard datasets. Image I shown on left and I' under respective graph. D-Nets clearly dominates patch-based methods. (View in color.)

Each of the two lists for a bin in the hash table, L_d and L'_d, is divided into two portions: the first part holding entries from all previous iterations ($P_{d,t}$ and $P'_{d,t}$), and the second part holding only those new entries from the current iteration ($N_{d,t}$ and $N'_{d,t}$). Figure 4 illustrates how those portions can be tracked using the two indices i_0 and i_1.

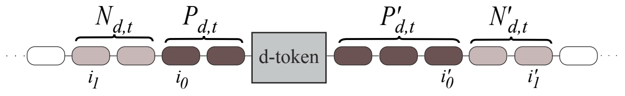

Figure 4. Illustration of the modified list structure for the iterative version corresponding to a single bin in the hash table. New connections added in the current iteration are drawn gray. Cells from previous iterations t=0 are drawn black. White cells are empty.

For voting in iteration t, a set $\mathcal{B}_{d,t}$ of hypotheses is defined, analogous to $\mathcal{B}$ in Algorithm 1:

$$\mathcal{B}_{d,t} = (N_{d,t} \times P'_{d,t-1}) \cup (N_{d,t} \times N'_{d,t}) \cup (P_{d,t-1} \times N'_{d,t})$$

Similar to the original algorithm, votes are cast for $\mathcal{B}_{d,t}$ with a strength of $v = 1/|\mathcal{B}_{d,t}|$. To prepare the next iteration, the indices i_0, i_1, i'_0 and i_1 are updated such that the recent lists are integrated into the old ones and emptied. That is $P_{d,t+1} := P_{d,t} \cup N_{d,t}$ and $N_{d,t+1} := \emptyset$. The same applies for lists $P'_{d,t+1}$ and $N'_{d,t+1}$. Then, a new iteration starts, increasing the network density to include new connections reachable from the old nodes by an additional hop.

Stopping criterion We propose a stopping criterion that does not involve geometric verification, considering how correspondences change at each step. We terminate if a fraction q_{stop} of $\max(n, n')$ correspondences does not change within m_{stop}=10 consecutive iterations, or if the maximum density of the network is reached. Here, n and n' are the number of nodes in the respective two images, I and I'.

Results Figure 5 shows the results for q_{stop} and m_{stop} using loose matching. Importantly, the iterative D-Nets algorithm automatically determines the required number of iterations and the resulting network density for a given matching case. Comparing the resulting last iteration index t_{fin} as labeled in Figure 5 with the respective two images of each matching case, it can be seen that t_{fin} reflects the matching difficulty very well. For instance, *wall1-6* is well known to be the most difficult matching case, because it involves a strong perspective distortion and has many repeating image structures, due to the bricks of the wall. Accordingly, it has the highest t_{fin}. The important contribution is, that the stopping criterion does not require a geometric verification step, which would involve the use of background knowledge about the application-domain and further computations in each iteration. Note, that of course the recall is smaller than e.g., in Figure 3, because the iterations stop intentionally before a maximum density is reached. This has no drawback for applications that only need a few precise matches to seed a RANSAC-based verification [6]. As expected, recall increases as q_{stop} is increased.

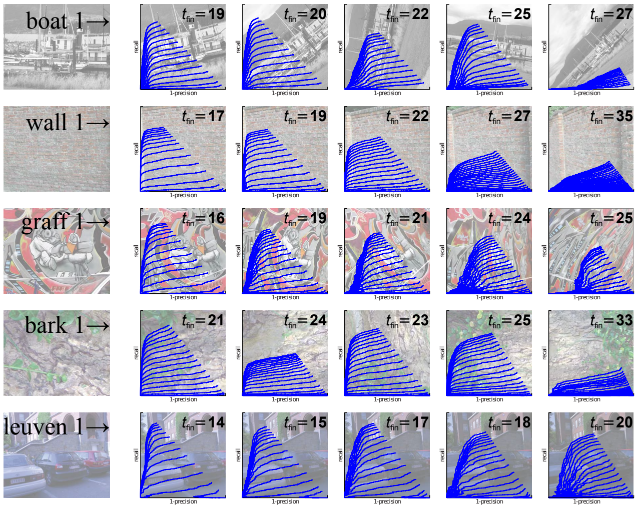

Figure 5. **Experiment 2**. Iterative D-Nets (q_{stop}=0.2 and m_{stop}=10): 1-precision vs. recall as connection density is dynamically increased. Our stopping criterion automatically determines the optimal connection density for the given matching task. Matching quality is comparable to that of clique D-Nets, but at much lower computational cost.

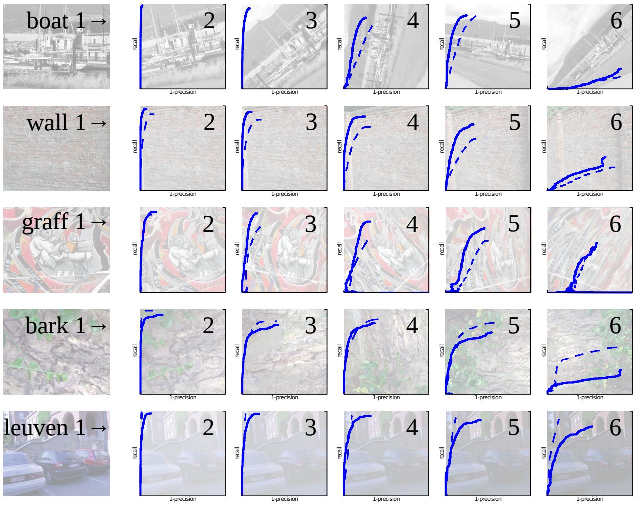

Figure 6. Comparison of D-Nets using sparse keypoints (solid blue) vs. densely sampled D-Nets (dashed blue), under loose matching. Both strongly outperform SIFT, ORB (cf. Figure 3b).

3.4. Experiment 3: Is keypoint extraction required?

Experiment 1 shows that D-Nets generate better matches than patch-based methods and are also much more robust to misregistered keypoints. The latter observation motivates a D-Nets variant that eschews keypoints entirely in favor of a dense grid of nodes with average spacing of 10-pixels. Since a completely regular grid can exhibit sampling pathology, we add Gaussian noise (σ=3) to the position of each node. In all other respects, the variant is identical to that in Experiment 1.

Results and Implications The above sampling procedure produces 5780 nodes, which is comparable to the number of nodes found by the SIFT-keypoint detector generated in Experiment 1. Figure 6 compares D-Nets on densely sampled poins (dashed blue) against the original D-Nets on sparse keypoints from Experiment 1 (i.e., solid blue, from Figure 3b). We see strong matching performance on all cases. Due to space limits, we restrict ourselves to three key observations:

- Unlike patch-based descriptors, which derive their scale and rotation invariance from the interest point detector (and lose these under dense sampling [13]), the dense variant of D-Nets retains all of its original invariance properties. This is because the D-Nets descriptor is defined according to pairwise connections, which specify rotation and scale regardless of how the (dimensionless) nodes are generated.
- Dense sampled patch-based descriptors are poor at establishing point-to-point correspondences [18]. Instead, dense SIFT is primarily used to characterize and aggregate local texture (e.g., for Bag-of-Words in object recognition). By contrast, dense sampling with D-Nets provides high-quality correspondences in addition to characterizing image content.
- Dense SIFT is computationally much more expensive than sparse SIFT. In contrast, dense D-Nets is faster than the original version; additional acceleration based upon precomputation using a fixed grid is possible.

3.5. Memory and Computational Complexity

The memory requirements for D-Nets are $2n_L|\mathcal{D}| \cdot \log_2(|\mathcal{D}|)$ bits for the hashtable and $n{\cdot}n'$ floats for the voting grid, where n, n' are the number of nodes in images I and I', respectively. A precise computational analysis must consider image content, because images with many ambiguous patterns are slower to match than those with many unique patterns. An upper bound for the worst-case running time is $O(|\mathcal{E}|+|\mathcal{E}'|+n_L^2 \cdot |\mathcal{D}|+n \cdot n')$, since every strip is accessed once, and in the worst case, the lists L and L' are full for all d-tokens. The last term $n \cdot n'$ accounts for initializing and extracting candidates from the voting grid.

4. Related Work

The idea of exploiting pairwise relationships is in itself not new. For instance, the pairwise representation inherent in D-Nets resembles that of pairwise matching approaches to object recognition [7]. However, this similarity is superficial: unlike in pairwise matching, where the representation focuses on the geometric relationships between nodes, pairwise relationships in D-Nets only specify the regions of pixels (strips) from which the descriptor is computed. D-Nets eschews encoding pairwise geometric relationships, since those are not invariant to image transformations.

D-Nets is also distinct from pairwise representations such as compact correlation coding [12], which advocates quantization of pairs of keypoints in a joint feature space or pairwise shape representations [4, 20], which aggregate statistics from pairs of points in an image.

In general, the use of shape structures for matching is related to our work, because these go beyond matching local features. For instance, using line segments [1] for wide-baseline stereo or more complex shape features, such as k-adjacent contours [5].

Our dense sampling experiment suggests a similarity to DAISY [16]. But, aside from DAISY's different application domain (computing dense depth maps for wide-baseline stereo), DAISY needs to know the relative pose of the cameras, while D-Nets does not.

At first glance our voting and hashing scheme resembles geometric hashing [19], which also starts from pairs of points. However, in contrast to geometric hashing, the D-Nets voting space is always a 2-D space of node-indices. Furthermore, no knowledge about the space of possible geometric transformations is required for D-Nets.

5. Conclusions

This paper proposes D-Nets, a novel image representation and matching scheme that is based upon a network of interconnected nodes. Image content is extracted along strips connecting nodes rather than on patches centered at nodes and quantized in the form of descriptive tokens. Node-to-node correspondences between images are determined by a combined hashing and voting scheme that exploits spatial and topological relationships. Extensive experiments using standard datasets show that D-Nets achieves significantly higher precision and recall than state-of-the-art patch-based methods, such as SIFT and ORB. We also describe two extensions of D-NETS that offer additional computational benefits. The first dynamically adapts to the complexity of the matching task. The second eschews interest points entirely and demonstrates that comparable accuracy can be achieved using a fixed, dense sampling of nodes; we plan to explore massively parallelized implementations of this idea in future work. Because the connections in the dense variant are fixed, all image accesses can be precomputed and stored in lookup-tables to determine to which lines they contribute, making D-Nets attractive for efficient GPU- and hardware-based implementations.

We consider D-Nets to be an initial exploration in a broad class of net-based image representations that goes beyond patch-based approaches. This paper has focused on straight-line strips connecting nodes, but we have also observed promising results using strips that follow image edges. The robustness of our voting procedure, in combination with the distinctiveness of each d-token means that our net-based representations are very resilient to cropping and occlusion since confident matches can be achieved with only a small fraction of available tokens.

References

[1] H. Bay, V. Ferrari, and L. Van Gool. Wide-baseline stereo matching with line segments. In *CVPR*, 2005. 8

[2] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006. 1

[3] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary robust independent elementary features. In *ECCV*, 2010. 1

[4] A. Evans, N. Thacker, and J. Mayhew. Pairwise representations of shape. In *ICPR*, 1992. 8

[5] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *PAMI*, 30(1), 2008. 8

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6), 1981. 6

[7] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR*, 2009. 7

[8] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *ICCV*, 2011. 1

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004. 1, 5

[10] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10), 2005. 1, 4

[11] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1/2), 2005. 1, 4, 5

[12] N. Morioka and S. Satoh. Compact correlation coding for visual object categorization. In *ICCV*, 2011. 8

[13] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV*, 2006. 7

[14] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *PAMI*, 32(3), 2010. 1

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2011. 1, 4, 5

[16] E. Tola, V. Lepetit, and P. Fua. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *PAMI*, 32(5), 2010. 8

[17] M. Trajkovic and M. Hedley. Fast corner detection. *Image Vision Computing*, 16(2), 1998. 1

[18] T. Tuytelaars. Dense interest points. In *CVPR*, 2010. 2, 7

[19] H. J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE CISE*, 4(4), 1997. 8

[20] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. In *CVPR*, 2010. 8