Dealing with limited data and training blind spots in the Manhattan world

Anonymous WACV submission

Paper ID 453

Abstract

Leveraging Manhattan assumption we generate metrically rectified novel views from a single image, even for non-box scenarios. Our novel views enable the already trained classifiers to handle training data missing views (blind spots) without additional training. We demonstrate this on end-to-end scene text spotting under perspective. Additionally, utilizing our fronto-parallel views, we discover unsupervised invariant mid-level patches given a few widely separated training examples (small data domain). These invariant patches outperform various baselines on small data image retrieval challenge.

1. Introduction

Discovering and matching patterns in images is one of the fundamental problems in computer vision. One of the key challenges is the pattern variations –scale, viewpoint, illumination or intra-class among others. Classical solutions involve matching features, such as SIFT [22], that are partially robust to viewpoint change. These solutions encode local patterns in the image and do not attempt to correct for the viewpoint changes over the entire scene.

More recent methods (e.g. deformable part models [8], mid-level features [36] or deep neural networks [17]) have produced detectors with a higher degree of invariance by leveraging lots of training data to model the variation modes. These recent methods have shown to outperform those based on local descriptors in object and scene recognition tasks. But even in the current large data regime, handling these variations is still an open challenge [10]. This is specially true if the training data has blind spots or rarely occurring patterns [4, 44].

In addition, many scenarios, such as training models from one or few examples, have access to only small data. In this constrained data domain every possible 2D variation mode might not be present in the training set, and hence the approaches mentioned in the above paragraph have a poor performance. To put a few examples, imagine building a detector for every hotel using the few images on its website, for every apartment using its few sublet images, or for every lost luggage item using a few images from the trip album. Our work addresses pattern recognition in this small data domains by building viewpoint-invariant representations.

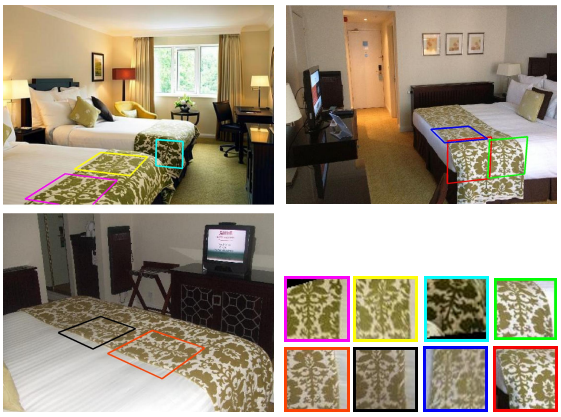


Figure 1: Detections of an unsupervised invariant mid-level detector in different views and different orientations per view.

In order to obtain such invariant representations we leverage the advances in single-image geometric scene understanding [14, 12, 13, 19, 42]. In a man-made scene, the entities are aligned mainly along three orthogonal directions (the Manhattan assumption); providing a cue about the observer's orientation or viewpoint [29, 31]. Furthermore, the majority of the man-made scenes can be summarized into a stage model [14, 23] composed of a few dominant planes surrounding the viewing camera. The indoor scenes, for example, have a box-stage with 5 planes –floor, ceiling and left-center-right walls.

This stage model, along with the cues about the viewing camera parameters, allows us to reason in the 3D space instead of the 2D image pixel space and undo the projective transformation. Specifically, our algorithm uses these 3D cues to generate several metrically rectified novel views per image. In each novel view, the entities aligned with its Manhattan direction are perspective-free. See Fig. 2b for

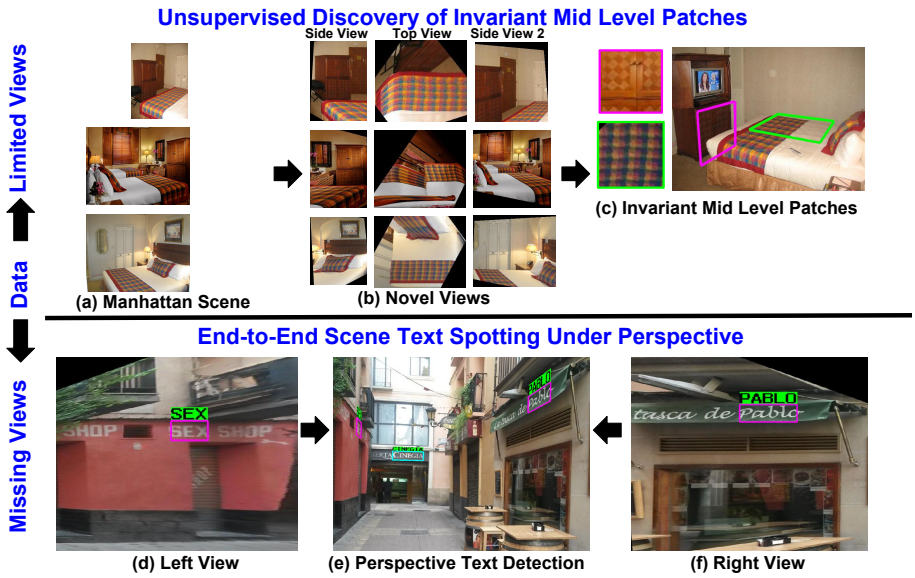


Figure 2: Overview of our approach. Top row: Unsupervised discovery of invariant mid-level patches (c) from widely separated viewpoints (a) using novel views (b). Bottom row: (e) Frontal text detected (CINEGIA) on the center wall (cyan box) using CNN based text detector [41]. (d+f) Our novel views help skewed text detection (SEX, PABLO) on the side walls (magenta boxes) without additional training.

examples of novel views.

We demonstrate the effectiveness of this novel representation in two applications. The first one is unsupervised mid-level pattern learning and recognition in *small data domain*. The aim is to discover discriminative detectors of a scene from a few widely separated examples (for example, the images in Fig. 2a). Although there are a few repeating elements in this scene (the bed sheet or the cabinet), the viewpoint variation makes them appear different. Our novel views reveal the similarity of these elements (Fig. 2b). Fig. 1 shows detections of one such invariant mid-level detector over multiple views. We evaluate our invariant mid-level detectors on a novel Hotels dataset, which we provide with this work. We outperform several baselines including Places (deep) [43] for image retrieval on this small data Hotels dataset.

Our second application involves enabling the supervised detectors to cope with the *missing training views* without additional training. To demonstrate this we have selected text spotting under perspective distortion. Fig. 2e shows a street scene with text on the center and the side walls. A CNN-based text detector trained on lots of synthetic plus real data [40] is able to detect only the frontal text CINEGIA (cyan box). Our novel images (Fig. 2d and 2f) helped detecting the text with greater skew, i.e., SEX, PABLO (magenta box), on the side walls. Our evaluation shows that our representation improves end-to-end scene text spotting under perspective over two baselines on a public dataset.

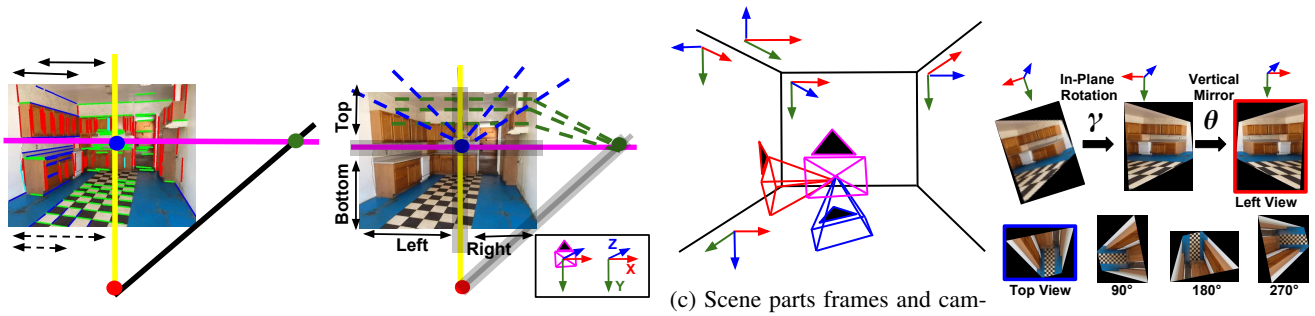
Contributions

1. Generation of metrically-rectified and world-aligned novel views from single image even for non-box Manhattan scenarios.

2. Unsupervised discovery of discriminative and 3D-invariant mid-level patches given limited training examples.

2. Related Work

Scene Based Reasoning. Much of the work on scene-level view-invariant representations focused on recovering the geometric layout of the scene from the image. In their pioneering work, Hoiem *et al*. [14] estimated the qualitative scene geometry from a single image. This 3D scene geometry lifted the reasoning from the 2D image space to the 3D physical world. Introducing physical rules helped to remove implausible detections in [15]. More recent works follow the similar goal of recovering 3D geometry by enforcing geometric and physical constraints [33, 37, 5, 34]. The work closest to ours is of Hedau *et al*. [13]. In addition to 3D physical reasoning, Hedau *et al*. generated affine rectified views to detect object cuboids. Differently from them, we generate metrically rectified plus aligned views. This world alignment enables us to correspond the content coming from different crowd sourced images. Satkin *et al*. [31, 30] registered the 2D images within 3D CAD models using single image layout. This allowed to match wide-



(a) VPs and horizons. (b) Scene parts. (c) Scene parts frames and cameras. (d) Novel views.

Figure 3: Novel views from a single image. The vanishing points and vanishing lines (horizons) are first extracted (a). The image is divided into scene parts (b). (c) shows the reference frame for camera and each scene part. d shows the rectified views for the left and bottom scene parts of the image in a.

baseline views by registering them to the same CAD model. Our view invariance reasoning is entirely based on the image content, without requiring any meta-data like EXIF tags or scene CAD models. Savarese et al. [32] generate a subset of the unseen object views at test time by morphing the available training set views. Our approach is not limited to this subset of morphed novel views. The above mentioned invariance methods work in supervised settings. Our novelty lies in achieving the invariance in an unsupervised setting. Srajer *et al*. [7] proposed wide-baseline image stitching by using the image layout to improve local feature matching. Our approach is more flexible, retrieving images from a scene class instead of a single scene instance. Similarly, Wu *et al*. [2] improved wide baseline image stitching by rectifying local patches before matching. These view-invariant patches belong to single scene instance whereas we use discriminative patches belonging to a scene class to train scene class detectors.

Unsupervised Pattern Discovery. Unsupervised pattern discovery has been found useful in various vision tasks, e.g., scene classification [36], single-view geometry [9], scene dynamics prediction [39] and style correspondence over time and space [20]. In these works the pattern discovery takes place mainly in the 2D image space, where the scene structure and viewpoint influences the image patterns. Ths requires training data per view. In our approach the patterns are discovered in the novel rectified images, where the perspective effects are gone. This makes our patches applicable in limited training data domain.

Scene Text Detection. The state-of-the-art scene text detectors have mainly focused on frontal text detection [40, 41, 1, 24, 16] with special emphasis on cropped word detection [25]. Recently, Phan *et al*. [28] proposed cropped word detection on scene text under perspective. We extend the latest and show how image rectification improves end-to-end scene text spotting without additional training.

3. Generating Novel Views from Single Images

3.1. Manhattan Structure

In man-made scenes, entities are aligned mainly along three orthogonal directions (for example, the alignment of the room walls and the objects in Fig. 3a.) This is commonly known as the Manhattan assumption.

The Manhattan directions can be estimated by clustering the detected 2D line segments in an image into three dominant clusters (the red, green and blue line clusters in Fig. 3a). The intersection of the lines of each cluster gives the three *vanishing points* (VPs) [12, 29]. Each VP is associated with one of the Manhattan directions and constrains the scene structure. For example, the two axes of each plane segment are aligned with two of the VPs, e.g., the center wall in Fig. 3a is aligned with the red and green VPs. A line joining two VPs gives a scene horizon. For example, the horizontal horizon (magenta line in Fig. 3a) is defined by the green and blue VPs. The upward facing planes, e.g., floor, and the downward facing planes e.g., ceiling, cannot exist on the same side of the horizontal vanishing line. Therefore, the horizontal vanishing line forms a boundary between the top and the bottom parts of the scene (Fig. 3b). Similarly, the vertical horizon (yellow line in Fig. 3b) and the front horizon (black line in Fig 3b) divide the scene into left-right and front-back parts respectively. This gives us at most five parts (top, bottom, left, right, front). The number of scene parts changes with the location of the VPs. For example, if the blue VP exists outside the image on the left, there is no left part.

3.2. Aligned Fronto-Parallel Views

For every scene part from the previous section, our aim is to generate a metrically rectified view. In metric rectifica-

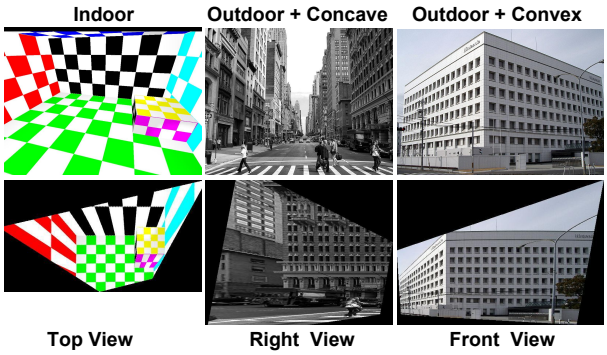


Figure 4: Novel views including non-box Manhattan scenes (last column).

tion, the aspect ratio of the 2D quadrilaterals remains constant [11]. For example, we want the checkers on the floor in Fig. 3a to be squares in the novel view of the bottom part.

Viewing the same 3D plane from two calibrated views is constrained by the planar homography given in Eq. 1 [11]:

$$\mathbf{H} = \mathbf{K}(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^\top}{d})\mathbf{K}^{-1}\,, \tag{1}$$

where $\mathbf{R}$ and $\mathbf{t}$ are the relative rotation matrix and the translation vector between the two views and $\mathbf{n}$ and d are the normal and distance to the origin of the plane. $\mathbf{K}$ stands for the internal calibration of the cameras, that we will take the same for the original and the novel views.

The image in Fig. 3a is taken from the magenta camera in Fig. 3c. In order to see the checkered pattern as squares, we want to place the camera facing the floor, i.e., the blue camera in Fig. 3c. Only a rotation is required to align the magenta and the blue camera. Setting $\mathbf{t} = 0$ in Eq. 1 gives the following

$$\mathbf{H} = \mathbf{K}\mathbf{R}_z^\gamma\mathbf{R}_y^\beta\mathbf{R}_x^\alpha\mathbf{K}^{-1}\,, \tag{2}$$

where $\mathbf{R} = \mathbf{R}_z^\gamma\mathbf{R}_y^\beta\mathbf{R}_x^\alpha$; $\mathbf{R}_i^\phi$ standing for a rotation of a ϕ-angle around the i-axis. The product $\mathbf{K}\mathbf{R}_z^\gamma$ is a similarity transform affecting the scale, in-plane rotation and the reflection that we will address later. The remaining part gives us the metric rectification in Eq. 3.

$$\mathbf{H}_m(\alpha,\beta) = \mathbf{R}_y^\beta\mathbf{R}_x^\alpha\mathbf{K}^{-1} \tag{3}$$

Let $\boldsymbol{\ell}_i$ and $\boldsymbol{\ell}_j$ be the line segments forming the two axes of a checker. Their transformed counterparts using Eq. 3, i.e., $\mathbf{H}_m^{-\top}\boldsymbol{\ell}_i$ and $\mathbf{H}_m^{-\top}\boldsymbol{\ell}_j$, must be perpendicular to each other. Using this orthogonality constraint, the rotation angles, i.e., α and β in Eq. 3, are estimated using the following minimization.

$$\underset{\alpha,\beta}{\operatorname{argmin}} \sum_{i,j} |\hat{\boldsymbol{\ell}}_i'^\top\hat{\boldsymbol{\ell}}_j'|\,, \tag{4}$$

where $\boldsymbol{\ell}' = \mathbf{H}_m^{-\top}\boldsymbol{\ell}$, $\hat{\boldsymbol{\ell}}$ is the normal vector to $\boldsymbol{\ell}$. Note that the lines $\boldsymbol{\ell}_i$ and $\boldsymbol{\ell}_j$, are not the detected line segments from Fig. 3a. Unlike the approach taken by [42], we generate the line segments using the two VPs forming the horizon related to the current part. In Fig. 3b, the dashed lines (green and blue) are generated for the top part. This helps in rectifying the parts with little local line content, e.g., the ceiling in Fig. 3b. Finally, we calibrate the camera focal length (f) from the three VPs, using the code provided by [3, 9, 12]. If available, EXIF data can also be used for camera calibration [42]. The optimization in Eq. 4 is repeated for all the scene parts.

So far, the homography in Eq. 3 only aligns the Z-axis of the novel camera view and the normal of its corresponding scene part (both shown as blue axis in Fig. 3c c). This suffices to define the geometry of the problem up to a similarity transform; and approaches similar to the above have been used for single-view reconstruction (SVR) [21, 42]. As our aim is not SVR but obtaining invariant image representations, we still have to estimate the similarity transform (in-plane rotation and reflection) that aligns the image content of different images. Notice, for example, how the novel views of figure 3d are fronto-parallel but different (up to a similarity transform).

Our proposal for such alignment is as follows. We align the X and Y axes of the novel view with the X (horizontal) and the Y (vertical) axes of the corresponding scene part (X and Y axes correspond to red and green axes respectively in Fig. 3c c). In other words, the X-axis is orthogonal to the vertical direction vector ([1 0]) and Y-axis is orthogonal to the horizontal direction vector([0 1]). This transformation is given by Eq. 5:

$$\mathbf{H}_r(\gamma,\theta_1,\theta_2) = \mathbf{R}_z^\gamma\mathbf{R}_{yflip}^{\theta_1}\mathbf{R}_{xflip}^{\theta_2}\,, \tag{5}$$

where γ is the in-plane rotation angle. θ_1 and θ_2 account for the mirror reflection, i.e., horizontal and vertical flipping (see first row in Fig. 3d for γ and θ_2). At this point $\mathbf{K}$ only effects digital zoom, so we drop it without loss of generality. These parameters are estimated using the following minimization:

$$\underset{\gamma,\theta_1,\theta_2}{\operatorname{argmin}} \left(\sum_i |\hat{\boldsymbol{\ell}}_i''^\top [0\,,1]\,| + \sum_j |\hat{\boldsymbol{\ell}}_j''^\top [1\,,0]\,| \right) \tag{6}$$
$$s.t.\ \gamma \in [0\,, 2\pi]\,; \theta_1, \theta_2 \in \{0\,, \pi\},$$

where $\boldsymbol{\ell}'' = \mathbf{H}_r^{-\top}\boldsymbol{\ell}'$. $\boldsymbol{\ell}_i$ and $\boldsymbol{\ell}_j$ are the lines corresponding to the two VPs associated with the current scene part. The complete transformation is $\mathbf{H}_c = \mathbf{H}_r\mathbf{H}_m$. Finally, the novel views for the top and the bottom part are rotated by 0°, 90°, 180°, 270° to make them rotationally invariant. The bottom row in Fig. 3d, shows the 4 versions of the top view.

Algorithm 1 Discovering Invariant Patches

INPUT: $\mathcal{P}$ {postitive set}, $\mathcal{N}$ {negative set}

$\mathcal{P} \Rightarrow \mathcal{P}^r, \mathcal{N} \Rightarrow \mathcal{N}^r$
$\mathcal{P}^r \Rightarrow \{\mathcal{P}_1^r, \mathcal{P}_2^r\}, \mathcal{N}^r \Rightarrow \{\mathcal{N}_1^r, \mathcal{N}_2^r\}$
$S \Leftarrow$ sample_patches $(\mathcal{P}^r)$
$K \Leftarrow$ Kmeans (S)
while not converged() **do**
  for all i such that size$(K[i] \leq 3)$ **do**
    $C_{new}\ [i] \Leftarrow$ svm_train $(K[i], \mathcal{N}_1^r)$
    $K_{new}\ [i] \Leftarrow$ get_topDetections $(C[i], \mathcal{P}_2^r, m)$
  end for
  $K \Leftarrow K_{new}\ C \Leftarrow C_{new}$
  swap$(\mathcal{P}_1^r,\ \mathcal{P}_2^r)$, swap$(\mathcal{N}_1^r,\ \mathcal{N}_2^r)$
end while
$A[i] \Leftarrow \Sigma$ scores$(K[i])$
return select_top(C, A, n)

These rotations are not required for the vertical parts (left, right and front) because in the crowd sourced data, the view variations in the vertical plane are $<45°$. Fig. 4 shows a few qualitative examples. For each view, the entities at only one orientation, irrespective of depth, are correctly rectified. This can be seen by comparing the green and yellow checkers (first Col. in Fig. 4) with the remaining multi-colored ones.

Before applying $\mathbf{H}_c$ to each scene part, we remove the area close to horizon lines (the gray area around horizons in Fig. 3b). The points on the horizon lines are mapped to infinity by definition [11]. Additionally, the area near the horizon is magnified as compared to scene content away from horizon line. Therefore, in order to provide consistent levels of resolution, we subdivide each part into 2 subparts as shown as double arrows in Fig. 3a. Instead of using our rectification and applying random 3D rotations (using random parameters in Eq. 2) distorts the texture (Fig. 6). Using random 2D affine transformation provides some invariance (Fig. 6). However, as shown in the experimental section, these 2D transformations are not helpful.

4. Unsupervised Discovery Of Invariant Mid-Level Patches

After the geometry estimation of the previous section the scene can be divided into several parts, each one tentatively corresponding to a homography that rectifies that scene part to a canonical fronto-parallel view. We then generate a large collection of patches from these rectified views and we extract a number of "good" patches, i.e., patches that are discriminative to be used in classification and retrieval tasks. This approach is summarized in **Algorithm 1**, inspired by [36]. For the details read the referenced source.

Our goal is to recover these discriminative patches from a few widely separated views. For example in the two images (Fig. 5a, second row), it is difficult to match patches due to the high viewpoint variance. Our novel images (Fig. 5c) undo this variation. Therefore, we convert the training set (positive class $\mathcal{P}$ and negative world $\mathcal{N}$) into the rectified set $(\mathcal{P}^r, \mathcal{N}^r)$ using $\mathbf{H}_c = \mathbf{H}_r\mathbf{H}_m$. We perform the process (given in **Algorithm 1**) to recover now the invariant mid-level detectors.

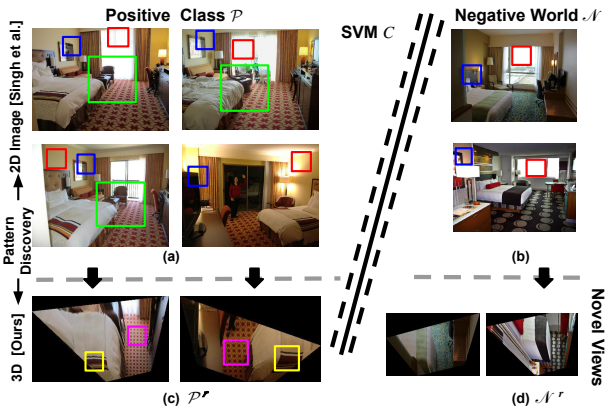


Figure 5: Unsupervised discriminative patch discovery. [36] samples in the 2D image space, ours in the 3D space.

5. Implementation Details

For fronto-parallel views, the image is separated into five parts (left, right, top, bottom, front). Each part is subdivided into two parts. We use different versions of this subdivision for the patch discovery (solid double arrows in Fig. 3a) and the text detection (dashed double arrows in Fig. 3a). The bottom part is rotated at $0°$, $90°$, $180°$, $270°$ for rotational invariance. We drop the top part, as its mostly non informative. This gives at most 14 fronto-parallel views per image. For text spotting we only used the vertical views (left,right and center view) as the text existed mainly on the walls. For patch discovery, the minimum sampled patch size is 80x80 pixels and the max size is up to image size. The HoG feature, per patch, is 1984 dimensional, i.e., 8x8x31. Linear SVM is used with $C = 0.1$.

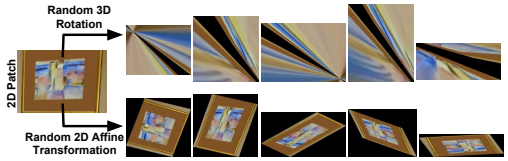


Figure 6

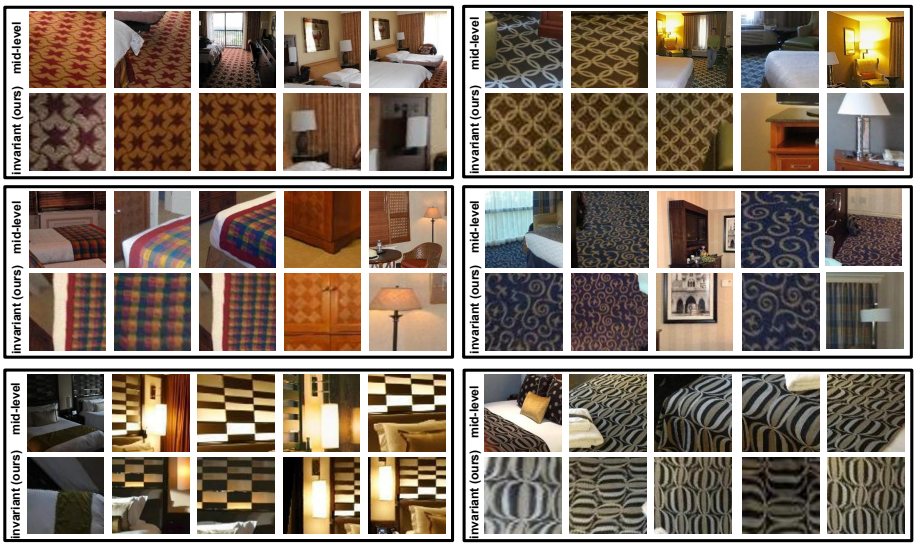


Figure 7: Complementary nature of the mid-level patches [36] (top rows) and our invariant patches (bottom rows). Best viewed in color and magnification. Mid-level patches are viewpoint-specific and capture spatial arrangements of objects. Our patches are fronto-parallel and aligned and mostly capture distinctive planar patterns.

6. Datasets

6.1. The Hotels Dataset 1.0

Our aim is to study the image retrieval task given few training examples with high viewpoint difference. The existing available large datasets (Places [43], ImageNet [6]) do provide large viewpoint variance but with large intra-class variance. For example, in the bedroom class, the images mainly belong to different rooms. This is a more challenging problem. The NYU V2 dataset [35] has little intra-class variance. For example, in bedroom class, up to six images from the same bedroom are available but with little viewpoint variance.

For our dataset we collected data of 21 different hotel interiors from `http://www.tripadvisor.com/`. These images are uploaded by different visitors to these hotels. This provides large viewpoint variance. Additionally, the hotel interiors have little intra-class variance since they have repeating elements (e.g., bed type, bed sheets, carpets or curtains). Even with this specific style, image retrieval is difficult for the state of the art due to the specific attributes of this dataset, namely wide baseline viewpoints with little overlap and the small training size. We have 315 training images (15 per hotel) and 210 test images (10 per hotel).

6.2. The Hotels Dataset 2.0

For further evaluation, we extend the above dataset to 68 classes. We have 1360 training images (20 per hotel) and 680 test images (10 per hotel). Total **2040** images.

6.3. The Street View Text-Perspective (SVT-Perspective) Dataset [28]

SVT-Perspective [28] contains images of scenes with skewed text from Google Street View. It is composed of 238 images with 639 annotated words. Additionally, every image comes with a lexicon, i.e., names of nearby places (approx. 50) to reduce the search space at test time. The performance of our text detection baselines [40, 41] was noticeably degraded in this dataset compared with its fronto-parallel equivalent, the SVT (Street View Text) dataset [40]. Specifically, in our experiments, the F-scores of [40, 41] in the SVT dataset were respectively 0.380 and 0.460 and 0.100 and 0.114 in the SVT-Perspective one.

7. Experimental Results

7.1. Image Retrieval In Small Data Domain (Hotels Dataset 1.0)

This section presents first a qualitative overview of the view-invariant discriminative patches extracted by our algorithm, followed by quantitative image retrieval results. Extra results are provided in supplementary work.

Fig. 7 shows a comparison between the unsupervised mid-level patches of [36] (top row) and the view-independent ones using our algorithm (bottom row). The visual inspection of these patches reveals their complementary nature. Notice that mid-level view-specific patches include the spatial configuration of scene entities whereas our invariant model mainly captures distinctive planar patterns.

The invariance of the mid-level patches extracted by our

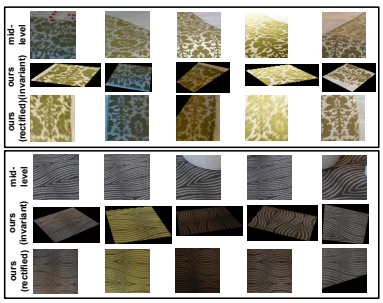


Figure 8: Comparison of our invariant mid-level patches and the traditional mid-level ones [36].

BoW [18]	0.26	BoW*	0.21
Scene DPM	0.36	Scene DPM + BoW + GIST [27]	0.37
GIST [26]	0.38	Places (deep- pretrained) [43] Places (deep- fine tuned)	0.61 0.66
Mid-level† + level 1 [36]	0.74	Mid-level† + level 1 + level 2	0.68
(Mid-level [36] + 2D affine)‡ + level 1	0.72	(Mid-level [36] + 2D affine)‡ + level 1 + level 2	0.67
Ours† + level 1	0.83	Ours† + level 1 + level 2	0.77
Ours‡ + level 1	**0.89**	Ours‡ + Mid-level† + level 1	**0.95**

Table 1: Average image retrieval rates on our Hotels dataset. BoW results are for the optimal dictionary size (200 words). * stands for rectified views only. † $\leq$ 80 patches. This is the max number of patches detected for [36] on our dataset. ‡ $\leq$ 200 patches. Level refers to the spatial grid units (see section 7.1 for details).

algorithm can be further evaluated in Fig. 8. The first and the second row show the image patterns firing for a given variant mid-level detector and an invariant mid-level one respectively. Notice the view rigidity of mid-level detections (row 1) as compared to view agility of our patches (row 2). The different looking patterns in the second row look alike once rectified, as shown in row 3.

Finally, table 1 shows a comparison of our algorithm and state-of-the-art approaches for image retrieval. Similar to [36], we select the top n ranked patch detectors. For a given image, the response of each detector is calculated for each rectified view (from section 3.2) at multiple scales s . This response is transferred to the original image space using the inverse rectification homography $\mathbf{H}_c^{-1}$. The response of all the detectors is max-pooled over different spatial grid levels at each scale. Level 1 has one grid unit, i.e., entire image. The subsequent levels divide each grid unit into 4 sub units. This gives a feature vector of size $\Sigma n s 2^{2(\text{level}-1)}$. A one-versus-all linear SVM is trained given these features.

Table 1 shows interesting results. BoW in the original, non-rectified images performs better than its counterpart trained on our rectified ones. This shows that local features are not able to take advantage of this rectification. The results for scene DPM [27] are worse than those of a simpler global descriptor like GIST [26]. Scene DPM learns the 2D spatial configuration of the scene parts, a difficult task with limited training examples with large viewpoint variations.

Mid-level patches [36] perform better than CNN based deep features [43]. We used the pre-trained CNN ($\sim$2.5 million images) provided by the authors to extract 4096 dimensional feature vector for one-versus-all SVM detection. Fine tuning the pre-trained net improved the performance but was still below the mid-level patches. Our Hotels dataset consists of repeating stylistic elements (carpet, bed sheets or bed type patterns). It is safe to conclude that, in this limited data domain, discriminative patches [36] better capture these mid-level elements than CNN [43].

Finally, our invariant detectors perform better than the mid-level viewpoint dependent detectors. For a fair comparison, we selected the same amount of patches, i.e., † $\leq$ 80. These are the maximum patches discovered for [36] on our dataset. The performance of our method improves once we increase the number of patches, i.e., ‡ = 200. Adding random 2D affine transformation to patches, helps in discovering more patches. However, they are not discriminative and they reduce the performance of standard mid level patches. The combination of mid-level patches and ours further improves the performance, indicating their complementary nature. For both methods, building a spatial pyramid over multiple grid levels does not help. The number of training parameters increases with the number of grid levels. Given the limited training set, this causes over-fitting. For level 1 we max-pooled over all scales to reduce the training parameters.

7.2. Image Retrieval In a Small Data Domain (Hotels Dataset 2.0)

We further evaluated our approach for image retrieval on a larger dataset (68 classes) comparing the two top performers (ours, mid-level). The results (Fig. 10) are consistent with the results from the last section. Our approach consistently outperforms the mid-level baseline. The performance gap is wider in the small training data domain and as expected, decreases as we increase the training data. The combination of the two approaches outperforms the baselines showing their complementary nature. In our opinion, in the absence of entities with specific line/edge characteristics, the performance of this fusion reduces to the mid-level baseline. Adding **Adaboost** [38]-based discriminative patch selection performs slightly better (perspective view 9%, our novel views 10%) than BoW 8%, given 5 training

Figure 9: Text spotting under perspective distortion. The ground truth text is shown in a red box and the spotted text in a blue box. The top row shows the original images. The bottom row shows the novel view, generated by our proposal, where the text spotting algorithm [41] originally detected the text.

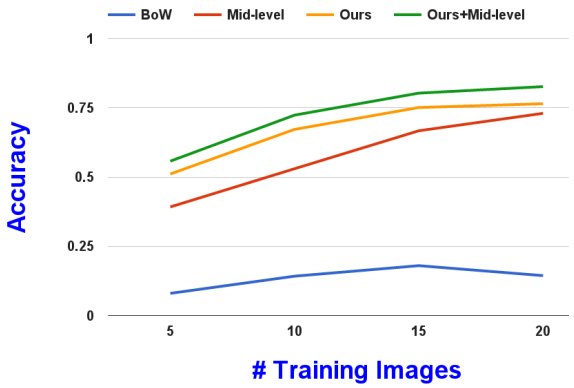


Figure 10: Image retrieval on Hotels 2.0.

examples. However, we achieve 51% and fusion with mid-level achieves 55%, given 5 training examples. See supplementary material for a visual comparison of boosting-based patches vs ours. We provide more examples with objects not having rich edge content in the supplementary material.

7.3. End-to-End Text Spotting under Perspective

In this section we show how our novel views help boosting the performance of already trained text spotters at test time and without additional training steps. The publicly available text spotters (pictorial-text detector [40], CNN-text detector [41]) have been trained on lots of real and synthetic data. However, they struggle to cope with the perspective distortion. Table 2 shows their F-scores.

The major feature in the **pictorial spotter** [40]) is the spacing between the detected individual characters. This spacing must be close to one character size. Under perspective distortion (some examples in Fig. 9, top row) the relative size of the characters and their spacing gets distorted, confusing the detector. Our novel views restore the scale and the spacing consistency (Fig. 9, bottom row).

The **convolutional spotter** [41] segments in a first step

Baseline	F-score	Baseline + Ours	F-score
Wang [40]	0.100	Wang [40] + Ours	**0.179**
Wang-Wu[41]	0.114	Wang-Wu [41] + Ours	**0.207**

Table 2: Text Detection F-scores (SVT-Perspective dataset)

the horizontal lines of foreground text in the image. These text lines are detected at multiple scales. However, at a single scale the characters must have similar size, resulting in rectangular boxes. Under perspective distortion the characters closer to the viewer appear at a different scale compared to the far field characters (Fig. 9, bottom row). Furthermore, the spacing between the characters decides the segmentation of the foreground text line into multiple text boxes. A distorted spacing might lead to an incorrect segmentation that degrades the performance. Our novel views restore this scale and character spaces and convert oblique perspective lines into horizontal ones, leading to better performance. Additionally note that the text spotting examples are not entirely box-like (Fig. 9, first column). Our novel views work even in outdoor quasi-Manhattan scenarios.

8. Conclusion and Future Work

Our novel view-invariant representation enables pattern discovery given limited training examples (as shown in the image retrieval task). It also enables the already trained detectors to cater for the training data blind spots (as shown in the text spotting task). Our 3D invariant mid-level detectors are a step forward towards unsupervised 3D scene understanding with limited data. Future work involves discovering 3D relations amongst our unsupervised patches. Such relations might lead to exciting algorithms, e.g., unsupervised 3D DPM and unsupervised single view geometry.

References

[1] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. Photoocr: Reading text in uncontrolled conditions. In *ICCV*,

2013.

[2] X. L. J.-M. F. C. Wu, B. Clipp and M. Pollefeys. 3d model matching with viewpoint-invariant patches (vip). In *CVPR*, 2008.

[3] B. Caprile and V. Torre. Using vanishing points for camera calibration. *IJCV*, 1990.

[4] C.-Y. Chen and K. Grauman. Inferring unseen views of people. In *CVPR*, 2014.

[5] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, 2013.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[7] M. P. F. Srajer, A. G. Schwing and T. Pajdla. Match-box: Indoor image matching via box-like scene estimation. In *3DV*, 2014.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[9] D. F. Fouhey, A. Gupta, and M. Hebert. Data-driven 3d primitives for single image understanding. In *ICCV*, 2013.

[10] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, 2014.

[11] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[12] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009.

[13] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*. 2010.

[14] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 2007.

[15] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 2008.

[16] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *ECCV*. 2014.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[18] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[19] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009.

[20] Y. J. Lee, A. A. Efros, and M. Hebert. Style-aware mid-level representation for discovering visual connections in space and time. In *ICCV*, 2013.

[21] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Computer Graphics Forum*, 1999.

[22] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[23] V. Nedovic, A. W. Smeulders, A. Redert, and J.-M. Geusebroek. Stages as models of scene geometry. *PAMI*, 2010.

[24] L. Neumann and J. Matas. Scene text localization and recognition with oriented stroke detection. In *ICCV*, 2013.

[25] T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky. Large-lexicon attribute-consistent text recognition in natural images. In *ECCV*. 2012.

[26] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in brain research*, 2006.

[27] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011.

[28] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. Tan. Recognizing text with perspective distortion in natural scenes. In *ICCV*, 2013.

[29] C. Rother. A new approach to vanishing point detection in architectural environments. *IVC*, 2002.

[30] S. Satkin and M. Hebert. 3dnn: Viewpoint invariant 3d geometry matching for scene understanding. In *ICCV*, 2013.

[31] S. Satkin, J. Lin, and M. Hebert. Data-driven scene understanding from 3D models. In *BMVC*, 2012.

[32] S. Savarese and L. Fei-Fei. View synthesis for recognizing unseen poses of object classes. In *ECCV 2008*.

[33] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 2009.

[34] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3d layout and object reasoning from single images. In *ICCV*, 2013.

[35] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*. 2012.

[36] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*. 2012.

[37] M. Sun, Y. Bao, and S. Savarese. Object detection with geometrical context feedback loop. In *BMVC*, 2010.

[38] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*, 2004.

[39] J. Walker, A. Gupta, and M. Hebert. Patch to the future: Unsupervised visual prediction. In *CVPR*, 2014.

[40] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *ICCV*, 2011.

[41] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *ICPR*, 2012.

[42] A. Zaheer, M. Rashid, and S. Khan. Shape from angle regularity. In *ECCV*. 2012.

[43] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. *NIPS*, 2014.

[44] X. Zhu, D. Anguelov, and D. Ramanan. Capturing long-tail distributions of object subcategories. In *CVPR*, 2014.

‘rebuttal and list of changes’

We would like to thank the reviewers for their time and insightful queries. The appreciation of the clarity of idea (R1: “ *paper is well written* “, R3: “ *idea is clear*”) and the thoroughness of the experiments (R1: “*experiments are carried out well*”, R2 *”experiments ... well support the claim*”) encouraged us.

We address below the main points recommended in the meta-review by the AE, summarizing the reviewers’ comments. For more details on some of our changes, please also see the rebuttal of the first round submission.

1. “state more clearly your contribution”
   We have changed the title, the abstract and the introduction to focus on the small data domain problem we are dealing. We made clear that our approach is specifically focused on the small data domain.
2. “compare your feature selection method against other approaches”
   As recommended by R2 we have added the discriminative feature selection using adaboost and its performance is similar to BoW based performance. We outperform these boosting based patches by a considerable margin (lines 753-792).
3. “test your method on larger dataasets”
   We have increased our dataset size (from 21 to 68 classes, section 6.2, Hotels dataset 2.0) and added a more detailed analysis (see for example the newly added figure 10). The new analysis and dataset further proves the advantage of using our representation over the state of the art baselines given limited training examples.
4. “give more details about the algorithm for text spotting”
   We have added the feature details in text spotting algorithm (section 7.3, lines 802-846).