Facilitating and Exploring Planar Homogeneous Texture for Indoor Scene Understanding (Supplementary Material)

Shahzor Ahmad and Loong-Fah Cheong
shahzor.ahmad@gmail.com, eleclf@nus.edu.sg

Department of ECE, National University of Singapore

A Texture Frequency Projection Model — Back-Projection Error

[1]Besides the ***re-projection error*** (Eqn. 5), we also define an error measure in the affine space (i.e., on the intermediate plane, where constant world-plane frequency is projected to another constant frequency via an affine transform). Consider H_P^{-1} that projects the intermediate plane to the image. The corresponding transposed Jacobian:

$$\mathbf{J}^{\mathbf{T}}_{\mathbf{H_P^{-1}}} = \begin{pmatrix} \frac{\partial x_i}{\partial x_s} & \frac{\partial y_i}{\partial x_s} \\ \frac{\partial x_i}{\partial y_s} & \frac{\partial y_i}{\partial y_s} \end{pmatrix} = \frac{1}{(1-h_7x_s-h_8y_s)^2} \begin{pmatrix} 1-h_8y_s & h_7y_s \\ h_8x_s & 1-h_7x_s \end{pmatrix} \tag{A.1}$$

back-projects the variable image frequency u_i, v_i to constant intermediate plane frequency u_s, v_s. We thus arrive at the ***back-projection error***:

$$\begin{aligned} E_{BP}(h_7,h_8,u_s,v_s) = &\sum_{x_s}\sum_{y_s}\left(\frac{(1-h_8y_s)\tilde{u_i}+h_7y_s\tilde{v_i}}{(1-h_7x_s-h_8y_s)^2}-u_s\right)^2 \\ &+\sum_{x_s}\sum_{y_s}\left(\frac{(1-h_7x_s)\tilde{v_i}+h_8x_s\tilde{u_i}}{(1-h_7x_s-h_8y_s)^2}-v_s\right)^2 \end{aligned} \tag{A.2}$$

To compute Eqn.A.2, back-projected coordinates $x_s = x_s(x_i)$ and $y_s = y_s(y_i)$ are obtained via the following transformation by H_P:

$$x_s = \frac{x_i}{h_7x_i+h_8y_i+1} \tag{A.3a}$$

$$y_s = \frac{y_i}{h_7x_i+h_8y_i+1} \tag{A.3b}$$

In practice, we minimize (using the Levenberg-Marquardt algorithm) the sum of both error measures for improved estimates:

$$E(h_7,h_8,u_s,v_s) = E_{RP}+E_{BP} \tag{A.4}$$

[1] Published in ECCV 2016, Part II, LNCS 9906 ©Springer. The final publication is available at link.springer.com.

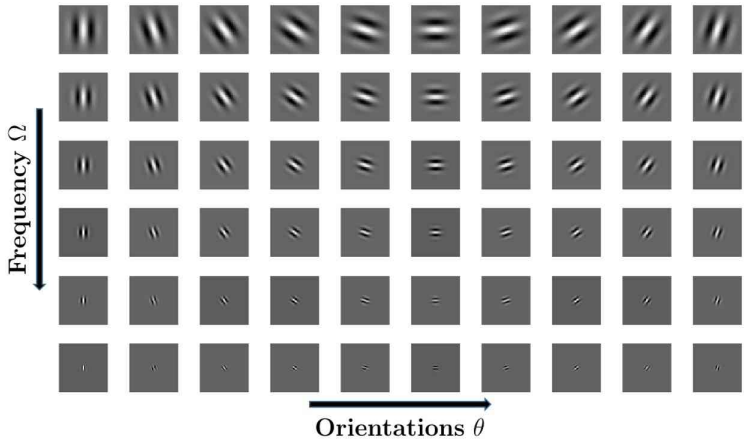


Fig. B.1: Visualization of the Gabor filter bank used in all experiments for this paper. Only the real parts of the complex-valued functions are shown. The radial frequencies increase along a geometric progression from 3 to 16.9706, while the orientations are uniformly spaced from $-90°$ to $72°$.

Our rationale for combining the two error measures E_{RP} and E_{BP} is as follows. One can view the operation performed by Eqns. A.3 to obtain error E_{BP} as some kind of data normalization, and, analogous to the case of estimating epipolar geometry in [1], we empirically evaluate which data normalization yields the best results and arrive at Eqn. A.4. For computational stability, the pixel coordinates are also normalized such that the top-left of an image patch is given by (-1,-1) and the bottom right by (1,1).

B Computing the Gabor Filter Bank

A Gabor filter with effective width γ, and spatial center frequency $\mathbf{u} = (u, v)$ is defined as:

$$h(\mathbf{u}; \mathbf{x}) = \frac{1}{2\pi\gamma^2} \exp\left\{-\frac{\mathbf{x}.\mathbf{x}^2}{2\gamma^2}\right\} \exp\left\{2\pi j \mathbf{u}.\mathbf{x}\right\} \tag{B.5}$$

The above form of the Gabor function is as in [2–4]. It can be easily shown that it is equivalent to the parameterization proposed in [5–7] if the spatial aspect ratio of the filter is fixed to 1 (i.e., the filters have a circular rather than an elliptical shape).

For all experiments in this paper, the filter bank was constructed based on the following parameters, which differ somewhat from [2–4] since they were experimentally fine-tuned to our setting. Filters sized 45x45 pixels are generated via Eqn. B.5. Six (6) radial center frequencies Ω are sampled along a geometric progression from 3 to 16.9706 cycles/image, with a common ratio $\sqrt{2}$. As suggested in [3, 4], the bandwidth is fixed so that the effective width γ varies

proportionally with the center frequency Ω. The proportionality constant may be computed via [5]:

$$\frac{\gamma}{\lambda} = \frac{1}{\pi}\sqrt{\frac{ln2}{2}}.\frac{2^b+1}{2^b-1} \tag{B.6}$$

where b is the half-magnitude spatial bandwidth of the Gabor filter, set to 1 in all experiments. Ten radial orientations θ spanning quadrants IV and I are used, spaced uniformly by 18° i.e., $-90°$ to 72°. Finally, the relationship between the polar form (Ω, θ) and the cartesian form $\mathbf{u} = (u, v)$ of spatial frequency is defined as:

$$\mathbf{u} = (u, v) = (\Omega sin\theta,\ \Omega cos\theta) \tag{B.7}$$

The filter bank constructed above is illustrated in Fig. B.1 by visualizing the real part of the complex-valued functions. The imaginary parts simply consist of a 90° offset relative to their real counterparts.

C Filtering — Implementation Details

Extensive experiments have helped to fine-tune parameters that yield the overall best results. The filter bank has been described in detail in Sec. B, with the filter kernel size fixed to 45x45 pixels. Meanwhile, the image patch to be filtered should be resized such that the smaller dimension is 80 pixels (using bicubic interpolation), and the aspect ratio is retained.

Instead of computing the partial derivatives of the Gabor filter $h(\mathbf{u}; \mathbf{x})$, the associativity property of convolution $[f * (g * h) = (f * g) * h]$ is invoked, and $B(\mathbf{u}; \mathbf{x})$, $C(\mathbf{u}; \mathbf{x})$ (Eqns. 8) defined as the responses of the partial derivatives f_x, f_y of the texture image $f(\mathbf{x})$ to the Gabor $h(\mathbf{u}; \mathbf{x})$. These partial derivatives were obtained via a simple forward difference approximation on the texture image. A Central difference approximation, or the use of filter masks involving it — e.g., Sobel and Fri-Chen — can only successfully recover half of the otherwise maximum measurable frequency, leading to aliasing in texture containing high frequency, where measuring changes over each pixel counts (see, e.g., Fig. 6(i)).

Following [3], the filter responses are smoothened by a Gaussian low-pass filter, also sized 45x45 pixels, and having a standard deviation $1/12^{th}$ its size.

D Robust Parameter Estimation via RANSAC

In our setting of Eqns. 5 and A.2, a minimum of 2 points are sufficient to estimate the 4 parameters h_7, h_8, u_s, v_s. For affine-rectification results (Sec. 4.1), RANSAC was run for 50 iterations with an error tolerance of 0.001, applied to Eqn. A.4, and using an anisotropic multiscale representation (Sec. E). For detection results (Sec. 4.2) — which, in turn, also influence the layout (Sec. 4.3) and scene recognition results (Sec. 4.4) — the threshold is 0.01, and the maximum number of iterations to run is adapted continuously based on the proportion of outliers in a given iteration [8]. RANSAC can then terminate in much fewer

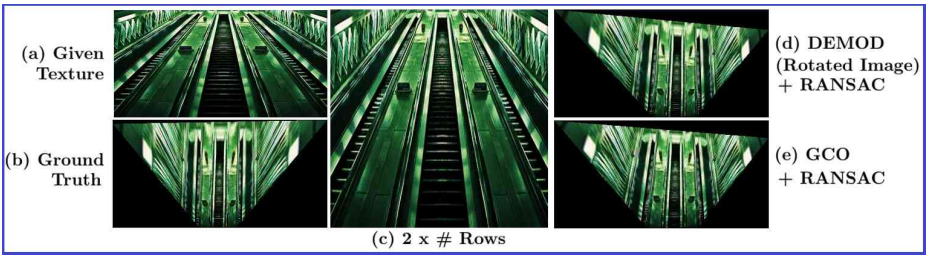


Fig. E.2: An anisotropic multi-scale approach, combined with carefully normalized error measure for choosing the best scale, improves texture rectification. Rotation may be allowed for DEMOD to automatically resolve quadrant ambiguity, if any.

iterations. While this speeds up the process, using more RANSAC iterations will certainly improve performance. Since we evaluate a large number of overlapping patches, however, we may choose to make this trade-off. For the same reason, an anisotropic representation for detection is also foregone, but the error normalization (see Sec. 4.2) is performed.

E Anisotropic Multiscale Representation

The following anisotropic approach was observed to improve performance. The given image is represented at three scales — one where the smaller dimension is 80 pixels, second where the rows are doubled while columns stay the same, and third where columns are doubled and rows stay the same (bicubic interpolation is used for the required resizing). For e.g., the `subway` patch (Fig. 6(q)) is originally 200x400 pixels. It is resized to give three representations: 80x160 pixels (shown in Fig. E.2(a)), 160x160 pixels (shown in Fig. E.2(b)) and 80x320 pixels. Parameters are obtained for each multiscale representation, and the winning one is obtained based on the normalized root mean squared error:

$$\mathcal{RMSE} = \sqrt{\sum_i \mathcal{E}(\mathbf{x_i})} \tag{E.8}$$

Table 1: Robust estimated projective parameters for the example texture in Fig. E.2(a) using an anisotropic multi-scale approach for DEMOD+ROT, GCO and QPBO. RANSAC error tolerance = 0.001.

	DEMOD+ROT+RANSAC	**GCO+RANSAC**	**GT**
h7	0.0437	0.0431	-0.0064
h8	0.6563	0.6087	0.7011
% outliers	36.21%	30.01%	N/A

where $\mathcal{E}(\mathbf{x_i})$ is as given in Eqn. 11. The qualitative (Fig. E.2) as well as the quantitative results (summarized in Table 1) indicate a marked improvement over a uni-scale approach. In each case (DEMOD, GCO), the winning representation happened to be case# 2 — i.e., doubling of rows (Fig. E.2(b)). For DEMOD, rotated versions of each of the three representations were also included to handle quadrant ambiguity, giving six representations in total (the winning one happened to be a rotated version with double the rows). Anisotropic scaling essentially makes the scale of the relevant image features (track rails in our example) more pertinent with respect to the size of the Gabor filters used (45x45 pixels).

The affine-rectification results presented in Sec. 4.1 employ the above anisotropic multiscale representation. Note that both the compared methods — TILT and REM — also employ multiscale representations.

F Detection in the Wild — Implementation Details

An approach similar to object detection [9, 10] is taken, wherein a given image is represented at multiple scales, and patches of fixed size extracted and processed at each scale. This provides for a **space and scale invariant detection**.

Specifically, a given image, in grayscale, is first resized to a reference scale, such that the smaller dimension is 400 pixels, and the aspect ratio preserved. Patches, sized 80x80 pixels, are extracted on a regular grid with a spatial stride of 16 pixels. This gives the number of octaves, such that at least one such patch may be extracted at the coarsest scale, as $log_2(400/80) = 2.3$. Fixing the number of scales per octave to 3.5, the total number of levels in our multiscale pyramid is then $N = floor(2.3 \times 3.5) + 1 = 9$. The corresponding scales to resize the image to (via bicubic interpolation) are given by a geometric progression with common ratio $r = 2^{-1/3.5}$, i.e., r^l, where $l = 0, 1, ..., N-1$. Following [10, 11], a patch containing very little image variation, i.e., gradient energy (average gradient norm over all pixels) smaller than a certain threshold (fixed to 50% of the average gradient energy over all image patches) are discarded at the outset. This results in a total of around 1500 patches per image on average.

A smaller grid spacing may be used at higher computational expense (e.g., a spacing of 8 pixels can result in four times the number of patches). Also, a non-unit aspect ratio for patches (e.g., sizes of 80x160 or 160x80, etc) can often be more representative of the homogeneous texture occurring in scenes, and sampling such additional patches to improve detection and recognition performance may be done at higher computational expense, but this was not implemented.

For the *qualitative* results presented in Figs. 2, H.4, H.5 an **intra-scale non-max suppression (NMS)** is performed as follows. Candidate patches (those with $< 50\%$ outliers) are sorted and processed in ascending order of percentage of outliers. Then, a patch is admitted as a detection only if some previously admitted patch (detected at the same scale) does not overlap 50% of its area. NMS across scales tends to discourage detections at coarse scales, hence suppression only within a given scale is carried out. However, the *quantitative* evaluation pre-

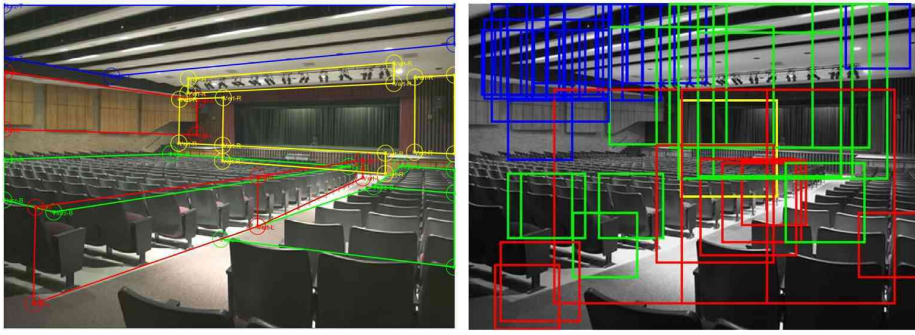


Fig. F.3: **Left:** We annotate images with quadrilaterals specifying left (red) / right (yellow) walls, ceiling (blue) and floor (green), using our GUI written for the purpose. **Right:** We assign a geometric class ID to a detection based on its estimated vanishing line (Sec. 4.3), and perform a quantitative evaluation based on precision and recall computed against our annotated ground truth (Sec. 4.2).

sented in Sec. 4.2. employs the semantic class-aware (walls/ceiling/floor) NMS as described in Sec. 4.3, and discussed in Sec. 4.2.

Given the experimental set-up as described above, and in Secs. B and C, processing one image takes around 15 – 20 mins per CPU core running a MATLAB implementation at 3GHz.

Fig. F.3(left) illustrates how we annotate our subset of 300 MIT Indoor67 images, for conducting the quantitative evaluation in Sec. 4.2. All homogeneous textured regions in the image are annotated with their vanishing points and semantic / geometric class ID.

G Indoor Scene Recognition — Implmentation Details

In what follows, we disclose the implementation details of our scene recognition pipeline from Sec. 4.4, which are common for all 4 descriptors used (except when stated otherwise). A given image is resized to a reference scale, such that the smaller dimension is 400 pixels, and the aspect ratio preserved.

Feature Extraction — Regular Rep. For regular representation (no rectification), patches sized 16x16 pixels are extracted on a regular grid with a spatial stride of 8 pixels (4 pixels for SIFT), with the reference image, in grayscale, represented at the *same* set of 9 scales as determined in Sec. F.

Feature Extraction — Rectified Rep. For a rectified representation, 80x80 pixel regions are detected as in Sec. F, fixing a decision threshold of 50% RANSAC outliers. No NMS is performed, as that would result in a significantly sparse representation, losing discriminative power of the resulting image representation. A detection is always warped (using bilinear interpolation for speed) to a *fixed* size of 80x80 pixels (essentially, retaining the scale at which a homogeneous texture is detected). Given a warped textured region, patches, sized 16x16

pixels, are densely extracted on a regular grid with a spatial stride of 8 pixels (4 pixels for SIFT). A patch in the warped image which is not fully visible in the original non-rectified image is rejected.

Feature Encoding. Best practices for dense local feature based classification, as suggested in [12, 11] are followed. Specifically, the descriptor dimensionality is first reduced to 80 features via PCA, followed by learning a 256-component GMM. Separate dictionaries for regular and rectified features are learned, using a sample of 10^6 features, obtained equally over the entire training set. A 2-level spatial pyramid ([13]) is constructed, wherein a Fisher Encoding with sum pooling [14] is performed over each of the 5 spatial bins, obtaining a 40,960-dimensional descriptor per bin. Different from [12] (who normalize each bin separately), descriptors at each *level* of the spatial pyramid are l_2-normalized separately (i.e., 1 at the first level and the concatenated 4 at the second level), since this was observed to give a better performance (for both regular and rectified representation). Hellinger kernel mapping is then performed on the descriptors, followed by an l_2-normalization (as before) again, thereby obtaining a so-called Improved Fisher Vector (IFV). The 5 descriptors are then concatenated to obtain a 204,800-dimensional image representation.

Classification. Linear one-vs-all SVMs are employed (having already incorporated a non-linear Hellinger mapping), using the library made available by [15]. Classification performance is reported as an average of 3 runs using the standard train-test split for the MIT Indoor 67 [16]. As is standard practice on this dataset, classification accuracy is defined as the average of the diagonal of the confusion matrix (i.e., average of per-class rates rather than average over all dataset). For obtaining the classification performance of a combined representation, softmax transformed SVM scores of individual representations are multiplied, as proposed in [17].

H Detection in the Wild — Additional Qualitative Results

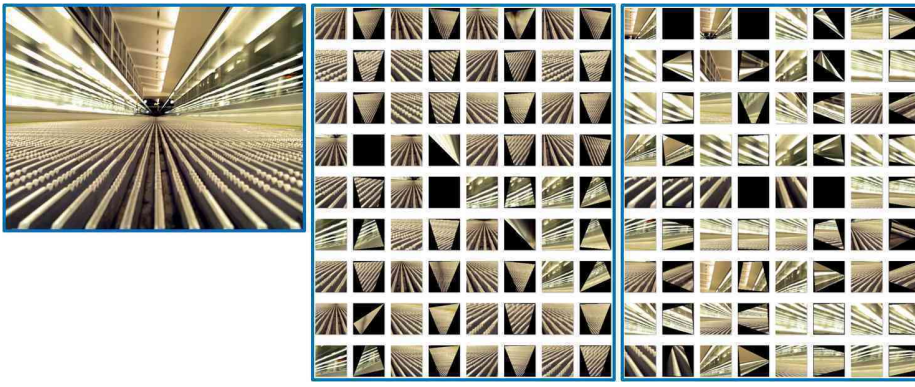
Fig. H.4 presents a qualitative comparison of our homogeneous texture detection vs. that performed by TILT [18]. The decision score for TILT used is a rank ratio of 0.5 (i.e., ratio of final to initial rank), along with the intra-scale NMS described in Sec. F. It can be observed that TILT is able to localize texture only in a few cases (e.g., b), when the low-rank assumption is satisfied. Correct rectifications are usually obtained when a patch is free from outliers (e.g., some patches in f). By contrast, the proposed approach is seen to perform remarkably well. Fig. H.5 presents additional qualitative results for our method in representative images from various scene categories.

I Indoor Scene Geometric Layout — Additional Qualitative Results

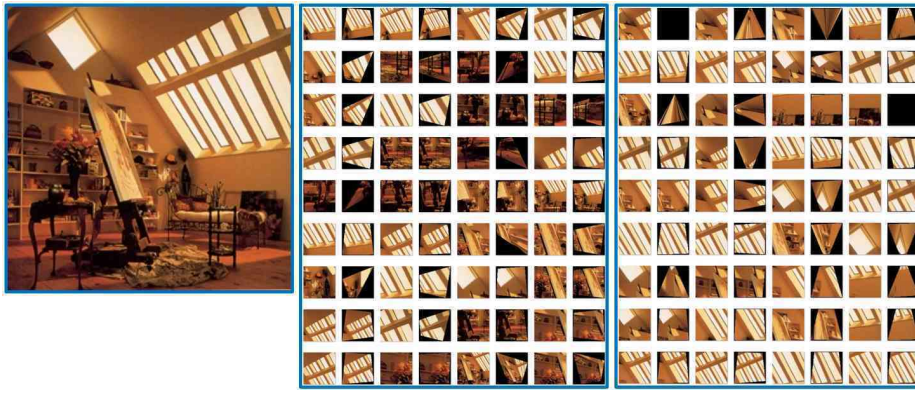
Fig. I.6 gives additional scene layout estimation results for the proposed approach (Sec. 4.3). Apart from many success cases, some limitations of ours may also be identified. For e.g., the middle column for (c) (respectively, d) reveals some correct detections on the right wall (respectively, ceiling) were obtained. However, the right colum demonstrates that across-category NMS (as described in Sec. 4.3) adversely rejects these detections. So while our NMS scheme is instrumental in "cleaning up" bad detections (e.g., a, b, etc) to produce a good layout, it can also suppress valid detections. Another limitation is that the proposed method naturally requires the scene / room faces to be textured to be able to produce an exhaustive layout, for e.g, (g) and (r) where only the office blinds or the patterned table cover can be correctly detected.

The floor in case (i) is incorrectly assigned the ceiling category — the algorithm has mistaken the muddy footsteps for homogeneous texture. The spurious right-wall detections in case (q) are likely due to the person in this `museum` image, interfering with the otherwise regularly spaced stands and their shadows.

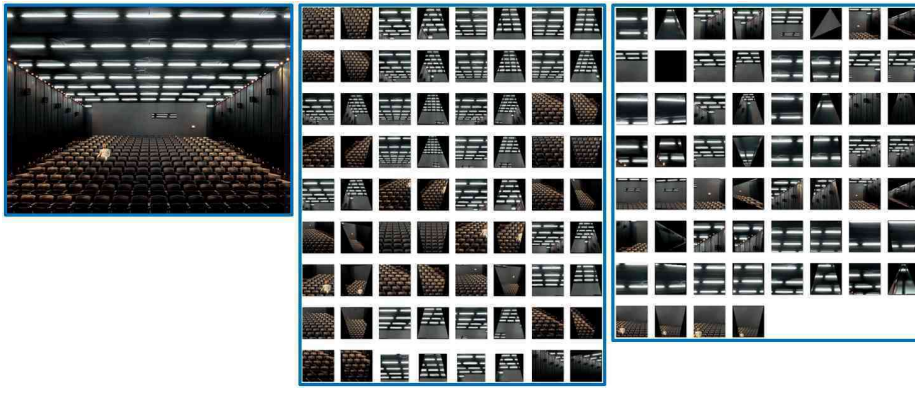
In order to address these shortcomings, future work is intended that shall investigate a combined approach involving appearance based assignment of geometric classes to superpixels, as in [19, 20], while also leveraging our texture cues. Global optimization, possibly with some ordering constraints, as in [21], can also potentially provide a more principled approach to rejecting bad detections, as opposed to our current greedy NMS scheme.



(a)

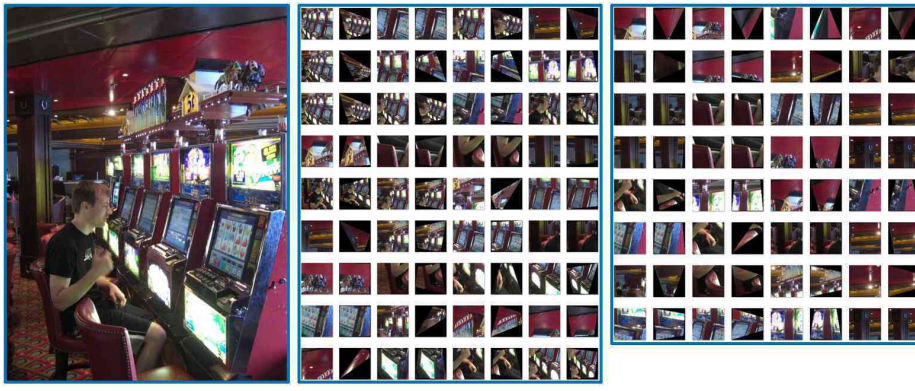


(b)



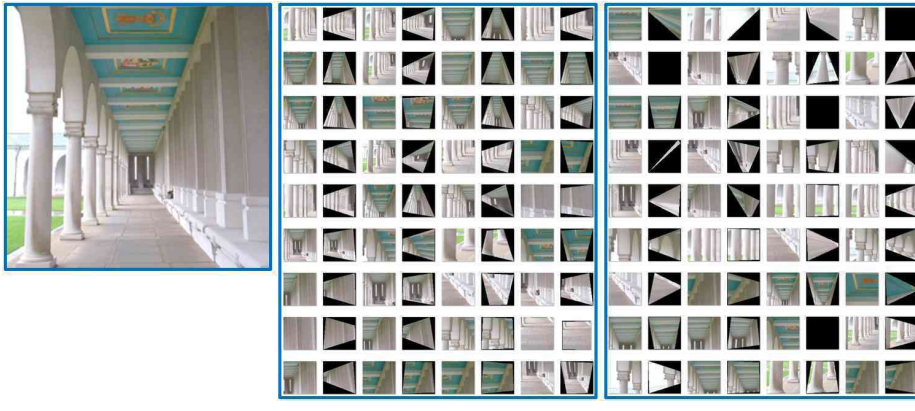
(c)

Fig. H.4: Detection of Homogeneous Texture: comparing PROPOSED method **(CENTER)** with TILT **(RIGHT)**. Images **(LEFT)** sampled from (a) `airport_inside`, (b) `art_studio`, (c) `auditorium` — 1/3



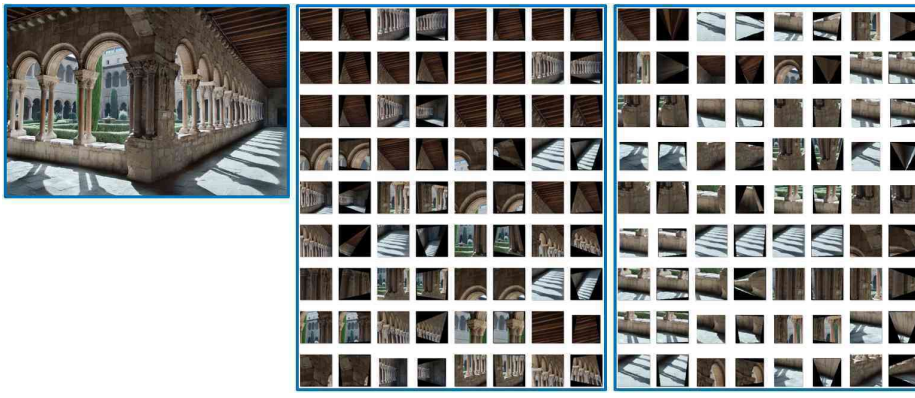
(d)

(e)

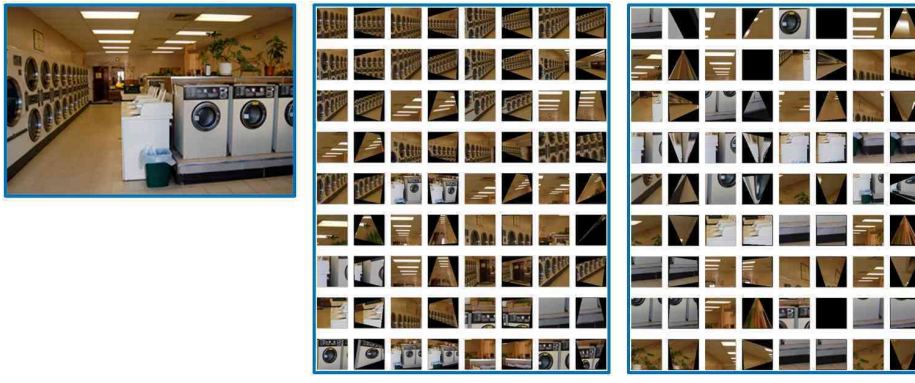


(f)

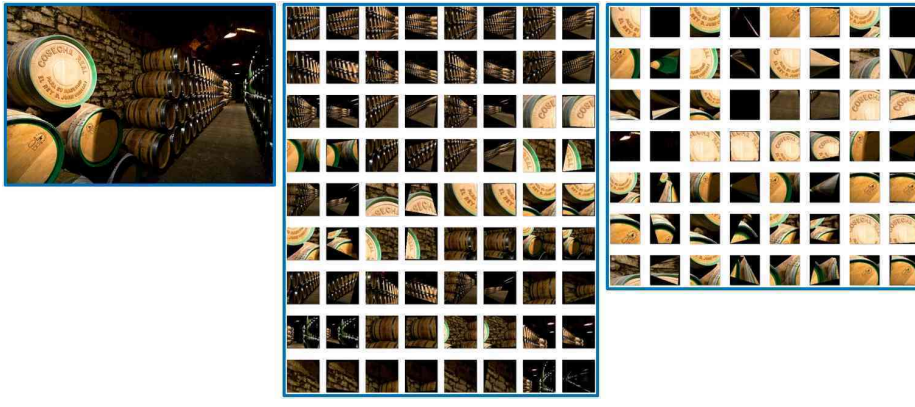
Fig. H.4: Detection of Homogeneous Texture: comparing PROPOSED method **(CENTER)** with TILT **(RIGHT)**. Images **(LEFT)** sampled from (d) `casino`, (e) `classroom`, (f) `cloister` — 2/3



(g)

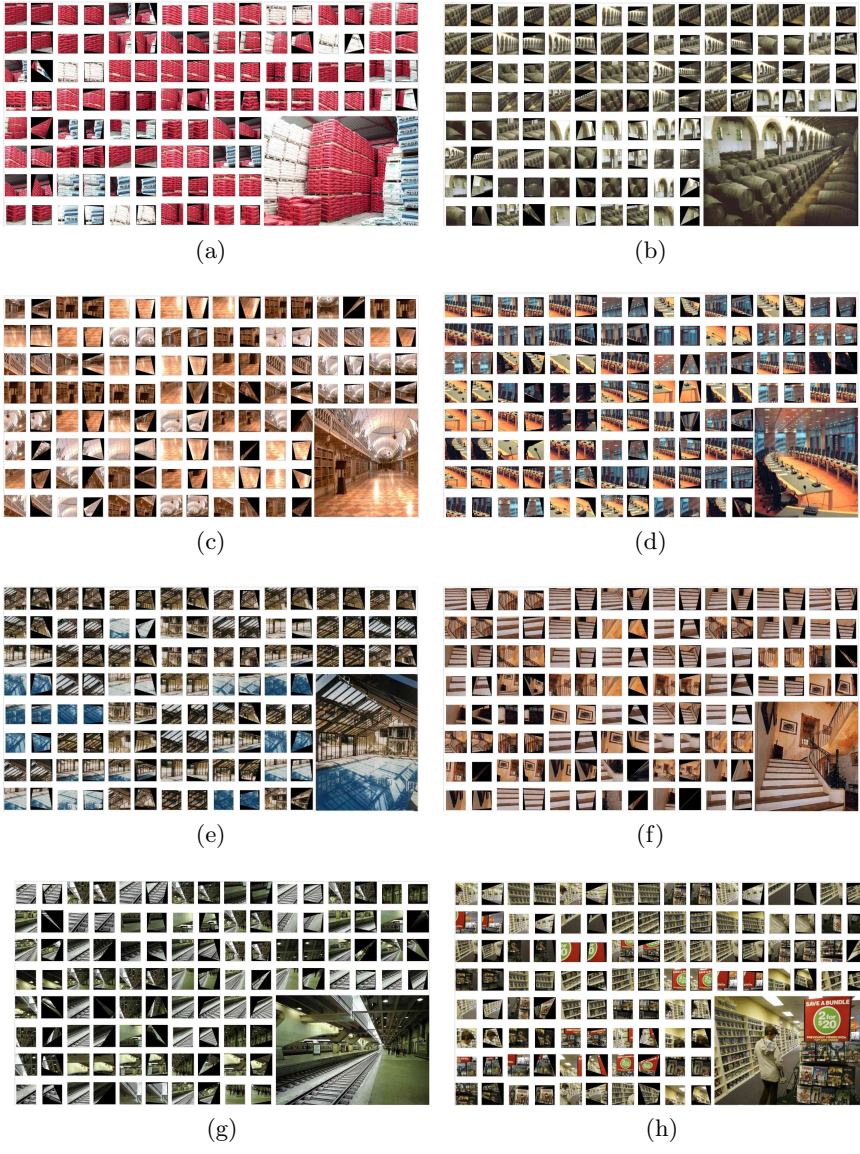


(h)



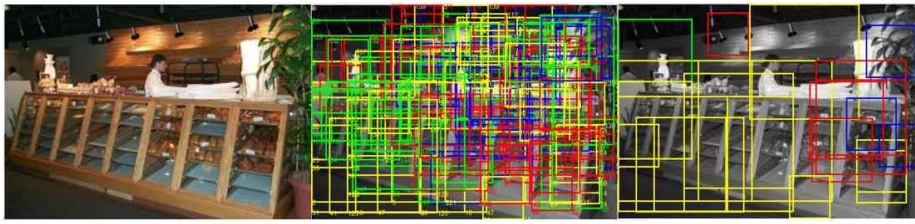
(i)

Fig. H.4: Detection of Homogeneous Texture: comparing PROPOSED method **(CENTER)** with TILT **(RIGHT)**. Images **(LEFT)** sampled from (g) `cloister`, (h) `laundromat`, (i) `winecellar` — 3/3

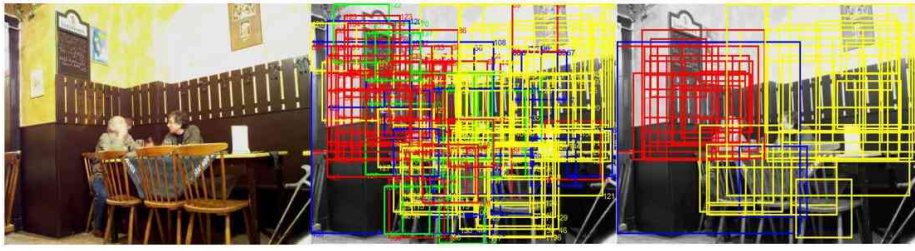


(a) (b) (c) (d) (e) (f) (g) (h)

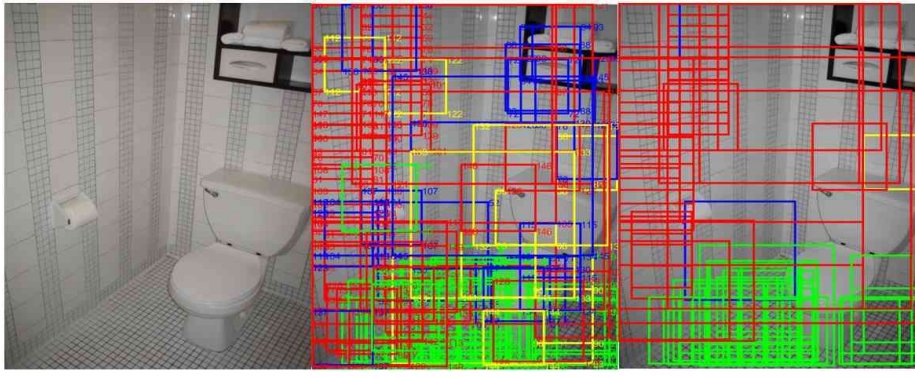
Fig. H.5: Detection of Homogeneous Texture by the proposed method. Images sampled from (a) `warehouse`, (b) `winecellar`, (c) `library`, (d) `meeting_room`, (e) `pool_inside`, (f) `staircase`, (g) `train_station`, (h) `video_store`.



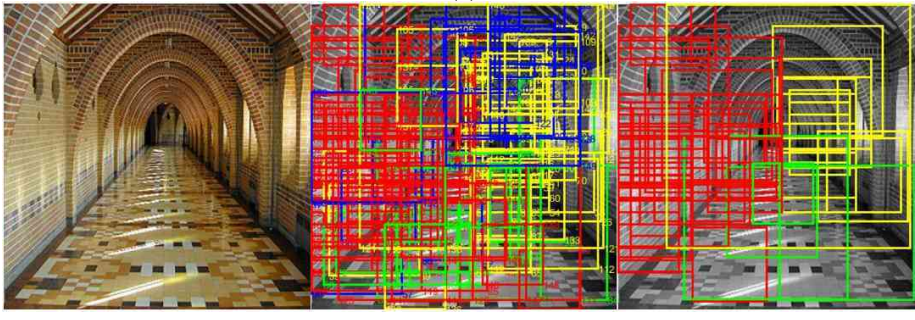
(a)



(b)

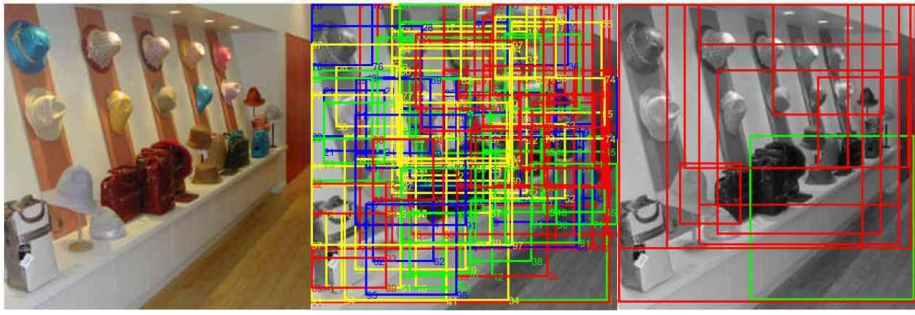


(c)

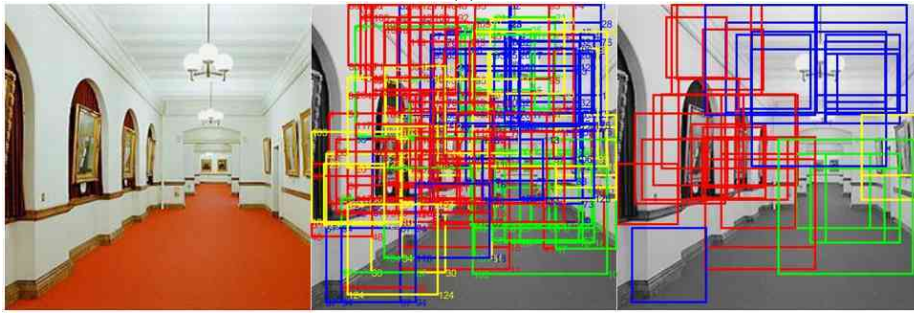


(d)

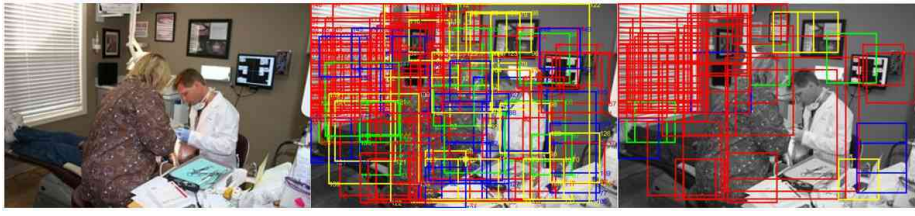
Fig. I.6: Indoor Scene Geometric Layout Estimation. **LEFT:** Image, **CENTER:** top 150 detections with geometric class assigned, **RIGHT:** after performing across-category NMS. **Red:** Left Wall, **Yellow:** Right Wall, **Blue:** Ceiling, **Green:** Floor. **Best viewed in color.** — 1/6



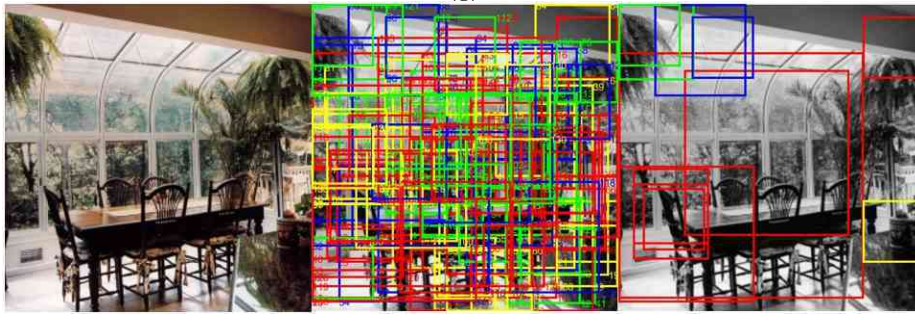
(e)



(f)

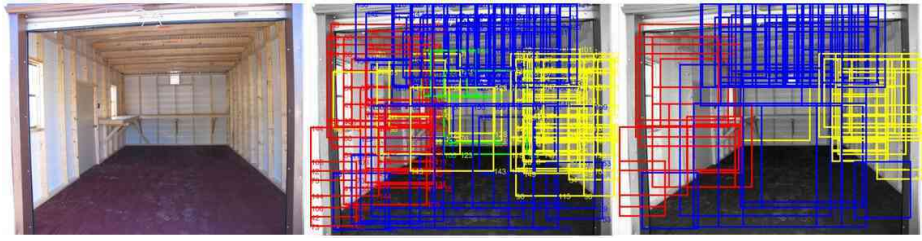


(g)

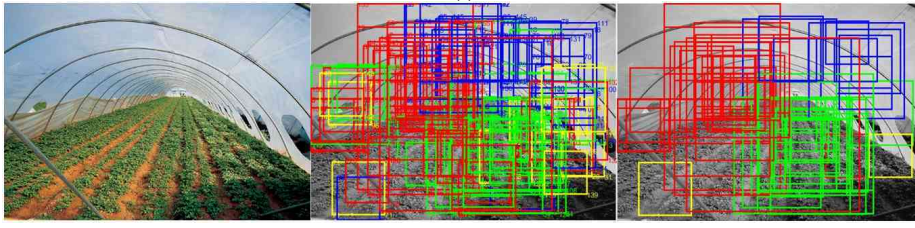


(h)

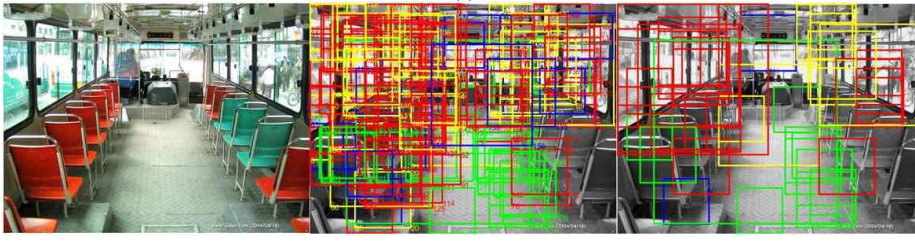
Fig. I.6: Indoor Scene Geometric Layout Estimation. **LEFT:** Image, **CENTER:** top 150 detections with geometric class assigned, **RIGHT:** after performing across-category NMS. **Red:** Left Wall, **Yellow:** Right Wall, **Blue:** Ceiling, **Green:** Floor. **Best viewed in color.** — 2/6



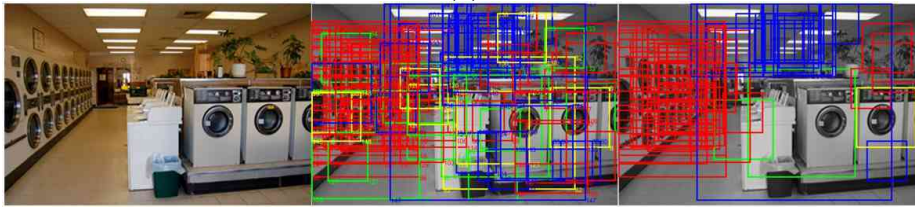
(i)



(j)

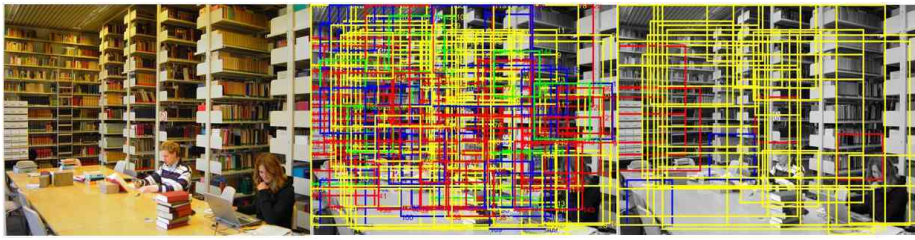


(k)

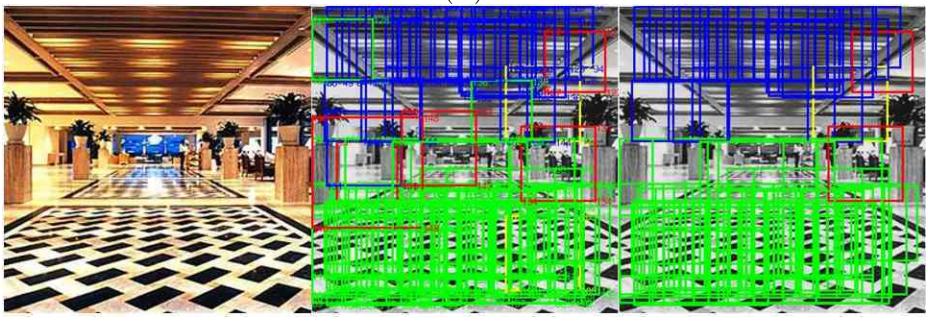


(l)

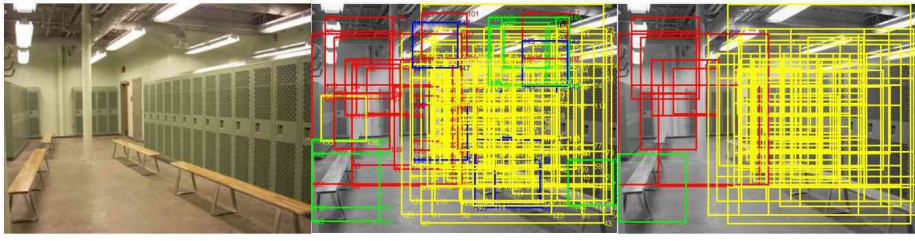
Fig. I.6: Indoor Scene Geometric Layout Estimation. **LEFT:** Image, **CENTER:** top 150 detections with geometric class assigned, **RIGHT:** after performing across-category NMS. **Red:** Left Wall, **Yellow:** Right Wall, **Blue:** Ceiling, **Green:** Floor. **Best viewed in color.** — 3/6



(m)



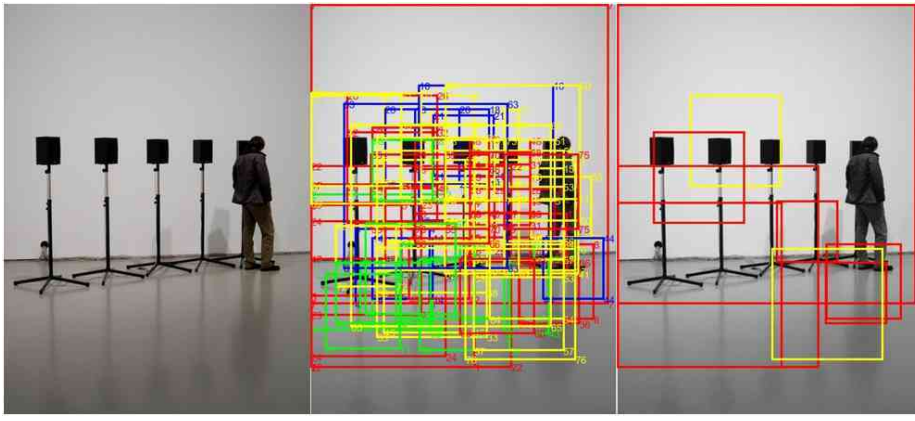
(n)



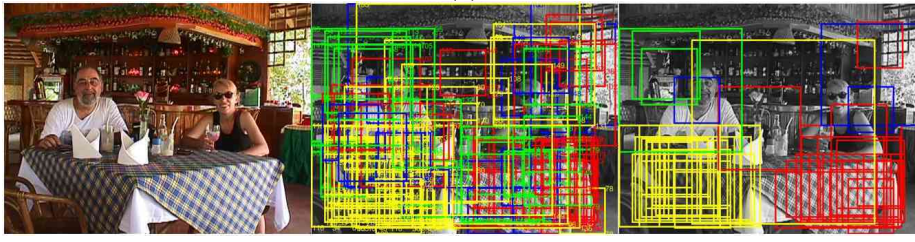
(o)

(p)

Fig. I.6: Indoor Scene Geometric Layout Estimation. **LEFT:** Image, **CENTER:** top 150 detections with geometric class assigned, **RIGHT:** after performing across-category NMS. **Red:** Left Wall, **Yellow:** Right Wall, **Blue:** Ceiling, **Green:** Floor. **Best viewed in color.** — 4/6

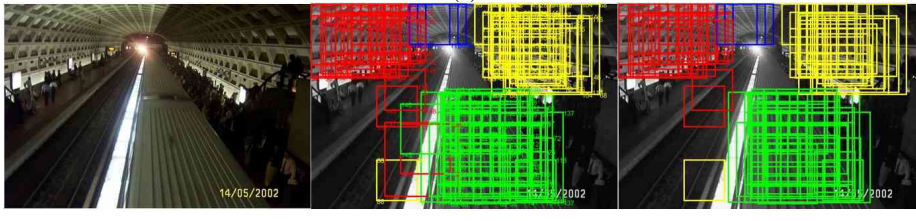


(q)



(r)

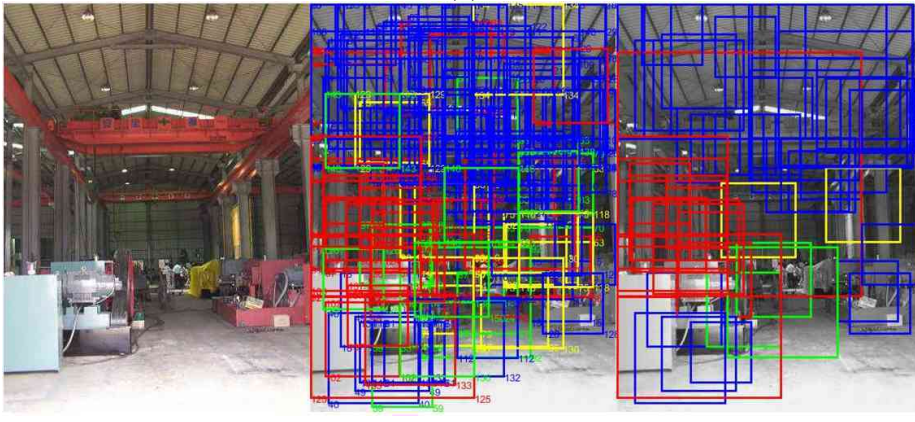
(s)



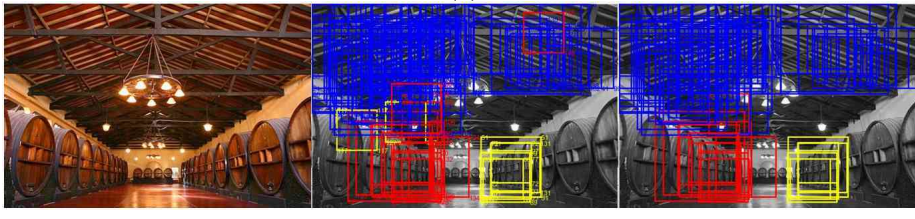
(t)

Fig. I.6: Indoor Scene Geometric Layout Estimation. **LEFT:** Image, **CENTER:** top 150 detections with geometric class assigned, **RIGHT:** after performing across-category NMS. **Red:** Left Wall, **Yellow:** Right Wall, **Blue:** Ceiling, **Green:** Floor. **Best viewed in color.** — 5/6

(u)



(v)



(w)

Fig. I.6: Indoor Scene Geometric Layout Estimation. **LEFT:** Image, **CENTER:** top 150 detections with geometric class assigned, **RIGHT:** after performing across-category NMS. **Red:** Left Wall, **Yellow:** Right Wall, **Blue:** Ceiling, **Green:** Floor. **Best viewed in color.** — 6/6

References

1. Zhang, Z.: Determining the epipolar geometry and its uncertainty: A review. IJCV **27**(2) (1998) 161–195
2. Super, B.J., Bovik, A.C.: Three-dimensional orientation from texture using gabor wavelets. In: Proc. SPIE Visual Communications and Image Processing '91: Image Processing. (1991)
3. Super, B.J., Bovik, A.C.: Planar surface orientation from texture spatial frequencies. Pattern Recognition **28**(5) (1995) 729–743
4. Super, B.J., Bovik, A.C.: Shape from texture using local spectral moments. TPAMI **17**(4) (1995) 333–343
5. Petkov, N., Kruizinga, P.: Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells. Biological Cybernetics **76**(2) (1997) 83–96
6. Serre, T., Wolf, L., Poggio, T.: Object recognition with features inspired by visual cortex. In: CVPR. (2005)
7. Mutch, J., Lowe, D.G.: Object class recognition and localization using sparse features with limited receptive fields. IJCV **80**(1) (2008) 45–57
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6) (1981) 381–395
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI **32**(9) (2010) 1627–1645
10. Singh, S., Gupta, A., Efros, A.A.: Unsupervised discovery of mid-level discriminative patches. In: ECCV. (2012)
11. Juneja, M., Vedaldi, A., Jawahar, C.V., Zisserman, A.: Blocks that shout: distinctive parts for scene classification. In: CVPR. (2013)
12. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. In: BMVC. (2011)
13. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. (2006)
14. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/ (2008)
15. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM TILT **2** (2011) 27:1–27:27 Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
16. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: CVPR. (2009)
17. Pandey, M., Lazebnik, S.: Scene recognition and weakly supervised object localization with deformable part-based models. In: ICCV. (2011)
18. Zhang, Z., Liang, X., Ganesh, A., Ma, Y.: TILT: Transform invariant low-rank textures. In: ACCV. (2010)
19. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. IJCV **75**(1) (2007) 151–172
20. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the spatial layout of cluttered rooms. In: ICCV. (2009)
21. Liu, X., Veksler, O., Samarabandu, J.: Order-preserving moves for graph-cut-based optimization. TPAMI **32**(7) (2010) 1182–1196