Vanishing Points Estimation by Self-Similarity

Hadas Kogan, Ron Maurer and Renato Keshet
HP-Labs
HP-Labs Israel, Technion city, Haifa, Israel 32000
{hadas.kogan,ron.maurer,renato.keshet}@hp.com

Abstract

This paper presents a novel self-similarity based approach for the problem of vanishing point estimation in man-made scenes. A vanishing point (VP) is the convergence point of a pencil (a concurrent line set), that is a perspective projection of a corresponding parallel line set in the scene. Unlike traditional VP detection that relies on extraction and grouping of individual straight lines, our approach detects entire pencils based on a property of 1D affine-similarity between parallel cross-sections of a pencil. Our approach is not limited to real pencils. Under some conditions (normally met in man-made scenes), our method can detect pencils made of virtual lines passing through similar image features, and hence can detect VPs from repeating patterns that do not contain straight edges. We demonstrate that detecting entire pencils rather than individual lines improves the detection robustness in that it improves VP detection in challenging conditions, such as very-low resolution or weak edges, and simultaneously reduces VP false-detection rate when only a small number of lines are detectable.

1. Introduction

Under a pinhole camera model, a set of parallel lines in the 3D scene is projected to a set of concurrent lines which meet at a single point, known as a vanishing point (VP). Each VP is associated with a unique 3D orientation, and hence can provide valuable information on the 3D structure of the scene. VPs are used for a variety of vision tasks such as camera calibration, perspective rectification, scene reconstruction and more [3].

The most commonly used approach for VP-detection follows the VP definition above and searches for groups of concurrent straight lines in the image. One typical approach uses the RANSAC principle: intersecting randomly selected couples of lines and choosing those that are consistent with a large group of lines [14]. Another line-grouping approach pioneered by Bernard [1], involves the detection of meaningful line-segments in the image, mapping lines or meeting points of line-pairs to an appropriate search space (Hough transform), finding salient clusters there, and estimating from theses clusters corresponding VP locations.
One challenge of the clustering is the unbounded search space and various parameterizations were proposed to make the search space bounded and as uniform as possible (e.g., the standard Gaussian sphere representation proposed by Bernard [1]). A second challenge related to line-based approaches is the many sources of uncertainty in the estimation process, such as quantization of the search space [2], edge-detection and line-detection errors [8], and VP candidate confidence [4],[14].

A fundamental drawback of line-grouping based methods regardless of their sophistication level is their absolute dependency on the explicit detection of straight features in the image. In practice, straight features are hard to detect in many cases, either due to low image quality (low resolution or blurred), or due to low contrast. In addition, one often finds in man-made environments planar patterns of repeating shapes or objects such as ornamented tiles or wallpapers, whose VPs cannot be extracted by line-based methods described above.

The case of approximately ordered patterns with small element size and spacing is handled well by Shape-from-Texture (SfT) methods, that estimate surface normals for smooth surfaces with flat textures (see e.g. [7] and references therein). In particular, for a planar texture, SfT methods can extract the single normal to the plane and hence corresponding vanishing points [9]. Unfortunately, SfT methods are computationally slow, and work well only if the texture is rich enough. They cannot work on textureless planar shapes, like traditional line-based methods. Hence neither of the main VP-detection approaches described so far can detect alone *all* VPs in many man-made scenes.

We found two works on VP estimation that are based on different principles from the main methods discussed above, and extract perspective information from finding similarities in the image. Schaffalitzky and Zisserman [10] proposed planar grouping of repeating equally spaced similar

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

elements (such as bricks, tiles, fences, etc.). Stentiford [12] proposed a visual-attention measure by matching random subsets (forks) of neighboring pixels to the scaled down forks with respect to the candidate pixel as the origin. Although Stentiford didn't state it explicitly, the VP corresponds to the point around which the image is locally self-similar under scaling. The main appeal of Stentiford's approach is its non-parametric nature and independency of specific feature detection such as straight lines. On the other hand, Stentiford's particular implementation of the self-similarity principle is limited to essentially central VPs that lie within the image.

1.1. Contribution and paper overview

This paper described a new self-similarity based approach for VP estimation, suitable to complex man made scenes. Our approach combines capabilities from the line-based, pattern-based and attention based methods reviewed in the introduction. To the best of our knowledge, our approach is the first that can extract VPs from both line information, texture information, and repeating object information by a single mechanism, thus forming a new link with SfT. The main novelty of our approach is the VPs estimation by the detection of entire pencils (sets of concurrent lines) rather than grouping individual lines. This gives our approach robustness in challenging conditions such as very low resolution, blurry or low-contrast edges and lines. This enables a considerable computation speed-up by severe downscaling of the input image, at the price of minor accuracy loss.

The paper is organized as follows: In section 2 we present the underlying principles of our approach. In section 3, we develop an algorithm that implements the general approach from section 2. Section 4 presents experimental results and a comparison to state-of-the-art line based algorithm [14]. Section 5 concludes the paper.

2. Vanishing point determination from self-similarity principles

2.1. Motivation

For easy exposition of the principles of our method, we start with the single central VP case, shown in Figure 1. The image manifests a global self-similarity property which can be used to estimate the VP as demonstrated in Figure 1. Since images exhibiting more that one VP do not posses such a global self-similarity property, we seek a more local self-similarity. We noticed that in Fig. 1c we can obtain the same VP prediction from matching pairs of parallel 1D cross-sections (or profiles) of the original and downscaled image. For example, the intensity profiles in the bottom and right boundaries of the original image, are approximately similar, up to downscaling and translation (1D-affine), to

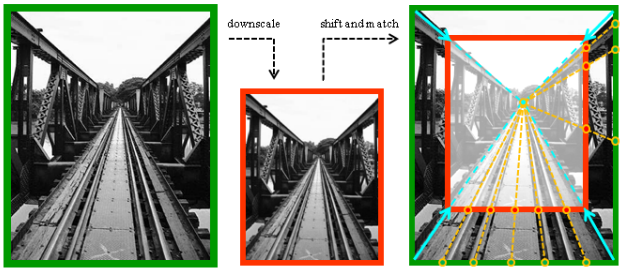

Figure 1. Central vanishing point estimation from global self-similarity or by corresponding 1D-profile similarity. Scaling down the original image in (a), gives a downscaled version (b) that is similar to a part of the original image, up to some global translation, which can be found by e.g. phase-correlation or Stentiford's method. Laying the scaled and translated image (thin green frame) on the original image (red thick frame) demonstrates this similarity (c). From the 2D self-similarity transformation the VP estimation can be estimated e.g. by the meeting of the virtual lines pointing from the outer to the inner corners (cyan arrows). Alternatively the VP can be extracted from similarity of image profiles. Pairs of orange circles connected by a dashed virtual line correspond to similar features along two parallel profiles, where all the virtual connecting lines converge at the VP.

the corresponding boundary profiles of the downscaled image (note the matching features in light circles). The arrows connecting the ends of the matching profiles (respective image corners) point towards, and meet at the VP. In other words, the VP position can be extracted from 1D-affine matching between a pair of parallel 1D-profiles. A complementary view is that the set of virtual straight lines that connect pairs of matching feature points (light circles) are concurrent and converge at the VP. Some of these virtual lines actually coincide with true straight edges, while others may connect feature points not on straight edges. We can view the process of obtaining a single VP from a global 2D self-similarity as equivalent to clustering large collection of VP candidates, each obtained from 1D-affine similarity between a pair of parallel 1D image profiles. Following this view, we next present the generalized self-similarity approach for detecting multiple VPs located anywhere in the image plane.

2.2. Pencil detection from similarity of cross-sections

Assume the image contains a pencil consisting of K line segments with finite slopes $a_k (k = 1..K)$ that converge at a common vanishing point VP=(x_v, y_v), as shown in Figure 2. A vertical cross-section of the pencil at some horizontal position x crosses the pencil lines at the points with y-coordinates $y_k(x) = y_v + a_k(x - x_v)$. All such cross-sections are similar up to scaling with respect to the VP. In

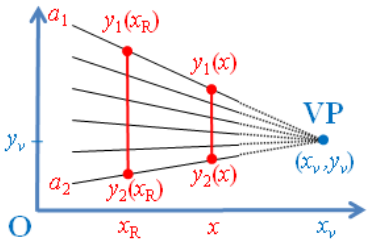

Figure 2. Relations between parallel cross sections of a pencil and its vanishing point.

particular the relation between two cross sections at x and x_R respectively is

$$\forall k : \frac{y_k(x_R) - y_v}{y_k(x) - y_v} = \frac{x_R - x_v}{x - x_v} \equiv s_{[x_R,x]} \tag{1}$$

From the general scaling similarity relation of (1), the VP location can be estimated by expressing (1) as an affine similarity relation relative to the origin:

$$\forall k : y_k(x_R) = s_{[x_R,x]} y_k(x) + \tau_{[x_R,x]} \tag{2}$$

$$\tau_{[x_R,x]} = y_v(1 - s_{[x_R,x]}) \tag{3}$$

If we find the affine transformation parameters (s, τ) between a matching pair of cross sections at x_R, x, we get the pencil's vanishing point by

$$x_v = x_R + \frac{x - x_R}{1 - s_{[x_R,x]}}, \quad y_v = \frac{\tau_{[x_R,x]}}{1 - s_{[x_R,x]}}. \tag{4}$$

Hence instead of searching for a common meeting point of many straight line features like in traditional approaches, we can search the image for pairs of similar parallel cross sections, and estimate from each such pair a potential vanishing point. Such estimates are based on characteristics of an entire pencil of *virtual* lines that connect matching features, and hence do not rely on straight-feature detection.

2.3. Pencil based accumulator space

In practice each matching pair of cross-sections (x, x_R) produces a VP candidate and the VPs are estimated by candidate accumulation and clustering. Since in typical man-made scenes many VPs lie far away from the image boundaries or at infinity, we propose a parametric accumulator space that is designed to deal with distant VPs. We first establish from (1) and (3) that the transformation parameters relating a fixed reference cross-section at x_R to another cross section at x is a linear function of x. Hence the rate of change in the affine matching parameters $a_s \equiv \partial_x s, \quad a_\tau \equiv \partial_x \tau$ is fixed. We denote the parameter pair a_s, a_τ as the *pencil slope coordinates*. They have an invertible relation to the VP coordinates (x_v, y_v), and can be computed directly from the transformation parameters of a single cross-section match:

$$a_s(x_R) = \frac{1}{x_R - x_v} = \frac{s_{[x_R,x]} - 1}{x - x_R} \tag{5}$$

$$a_\tau(x_R) = \frac{-y_v}{x_R - x_v} = \frac{\tau_{[x_R,x]}}{x - x_R} \tag{6}$$

Since different matched cross-section pairs may have different reference positions x_R, we transform the pencil slope coordinates to a common reference position x_o of our selection, e.g. the image center. From (6,5) we get the following transformation rule:

$$[a_s(x_o), a_\tau(x_o)] = \frac{[a_s(x_R), a_\tau(x_R)]}{1 + a_s(x_R)(x_o - x_R)} \tag{7}$$

In this parametrization infinite vanishing points are mapped to $[a_s(x_o), a_\tau(x_o)] = (0, \tan\theta_v)$, where θ_v is the VP direction. The main limitation of the pencil-slope parametrization is that VPs at x_o are mapped to infinity. This limitation is resolved by using two reference positions, e.g. at opposite boundaries of the image. Alternatively, like in [11], one may search for VPs located inside the image in bounded image space (x_v, y_v), and for VPs located outside the image in pencil slope space (with x_o at the image center).

2.4. Selection of pencil cross-section orientations

The derivation so far can be applied of course to cross-sections in arbitrary orientations. In general a single cross-section orientation is insufficient, since it cannot detect pencils which have most lines almost parallel to the cross-section orientation. Two perpendicular cross-section orientations can jointly detect VPs located at all orientations. The question is what are the best orientations.

Most man made scenes contain mainly horizontal and vertical surfaces. In addition, images of man made scenes are usually taken with the optical axis essentially parallel to the horizontal support surface (ground,floor). Under these conditions vertical and horizontal cross-sections are best suited to detect pencils of real or virtual lines. In addition, these orientations are most compatible with the image pixel grid, so that accuracy reduction due to aliasing is avoided.

So far we have presented the general principles of our scheme, next we present a corresponding algorithm.

3. Algorithm

Our algorithm consists of the following conceptual stages: first, representation of 1D cross sections that is suitable for robust matching; then, affine matching of 1D cross sections based on similarity between perceptually significant points; then, estimating VP candidates from the affine matching parameters and finally, clustering those candidates, e.g., by mean-shift procedure, to obtain the final VPs.

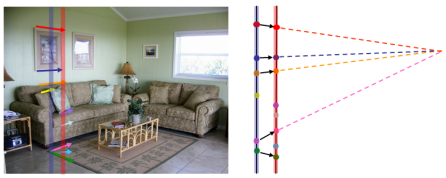

Figure 3. Similar features between matching columns.

3.1. Representation of 1D cross sections

Consider for example vertical cross sections. The most simple representation of 1D cross sections consists of single columns of the intensity values of the image. However, this simple representation is sensitive to illumination and noise and is not restrictive enough to assure perceptually correct matches. For perceptually correct matches we need to replace the image values by a feature map that is insensitive to slow illumination changes and noise and to include spatial context.

We choose to generate the feature map by a - Laplacian of Gaussian (LoG) filter with kernel size set to 5×5 as a compromise between locality and performance. For each image point (c, r) we collect the feature-map values in a rectangular spatial context into a feature vector $V_{c,r}$. The context width should be larger than 1 in order to have informative matching and avoid a high rate of false matchings. Yet it should not be too large as this would increase the sensitivity to foreshortening effects, or in other words invalidate the 1D cross-section approximation. The context height should be larger than the width, again to keep the 1D approximation valid, but not too large so as to make the context non-local. A context size of 3×5 was found to strike the right balance between all the above trade-offs.

The columns (rows) of the feature vector map can now be used to represent the parallel sets of $1D$ cross-sections discussed in Section 2.

3.2. Matching of 1D cross-sections

From now on we will use the term column (row) matching instead of cross-section matching. Our matching procedure first creates a map of similarity (denoted as Structural CORrelation Evidence - $SCORE$ matrix) between each two columns of the feature-vector map. Then, we estimate the similarity transformation parameters directly from the $SCORE$ matrix. A candidate VP is estimated directly from the similarity transformation as described in section 2.

Figure 3 demonstrates the challenge of matching columns in cluttered scenes. Some of the virtual lines connecting matched points between the two columns form a pencil that flows towards the right VP, while lines connecting other matching points do not belong to that pencil. There are also non-matching feature points.

We can characterize the challenge as the problem of finding a global affine match that is consistent with the majority of the perceptually significant point-matches, but is not affected by mis-matches due to occlusions. For that we introduce a new perceptual matching criterion that favors visually significant matches (positive evidence), while effectively ignoring visually insignificant matches (little evidence) and mismatches (negative evidence). We call this criterion $SCORE$ (Strcutural CORrelation Evidence). The form of $SCORE$ is related to the SSIM perceptual error-measure [13]. The $SCORE$ value of pair of feature vectors v_1, v_2 is given by:

$$SCORE(v_1, v_2) = \left. \frac{\langle v_1, v_2 \rangle}{|v_1|^2 + |v_2|^2 + T\sqrt{|v_1|^2 + |v_2|^2}} \right|_{0+} \quad (8)$$

Note that the $SCORE$ is very small if either $|v_1|$ or $|v_2|$ are much smaller than the characteristic activity threshold T. For two vectors larger than T, $SCORE$ is determined mainly by the smaller of the two. Hence large $SCORE$ are only obtained for two large aligned vectors. Large activity threshold T in equation 8 increases robustness to noise but also decreases the $SCORE$ grade for weak matching features.

Assume we try to match the two columns c_1, c_2, we generate a $SCORE$ matrix by calculating a $SCORE$ value between every two point combination:

$$SCORE_{c_1,c_2}(i, j) = SCORE(v_{c_1}(i), v_{c_2}(j))$$

If column c_2 is a perfect affine transformation of the column c_1, then $c_1(s \cdot i + \tau) = c_2(i)$, where s, τ are the transformation parameters. In the corresponding $SCORE_{c_1,c_2}$ matrix, that relation would appear as high intensity areas (that correspond to strong matching points) spread along a straight line determined by the transformation parameters. However in practice, there is a large percentage of outliers due to accidental matches, occluding features (as shown in Fig. 3), etc. Therefore, a robust line-fitting procedure is required. We propose to improve the robustness of fitting by incorporating a-priory assumptions. Our model prefers global matches because perspective effects are global in the image. In other words, we prefer lines that pass through a set of high $SCORE$ points that are as spatially spread as possible. Such global matches are also less sensitive to noise and digitization artifacts in low-resolution images. Our proposed procedure for global line fit involves solving the following non-linear maximization problem:

$$[s, t] \quad = \quad argmax \sum_i |i - CM| * SCORE(i, s \cdot i + \tau)$$

where $CM = \sum_i i \cdot SCORE(i, s \cdot i + \tau)$ is the center of mass of the line, $SCORE(i, s \cdot i + \tau)$ is the linear interpolation of the $SCORE$ matrix at the coordinates $(i, s \cdot i + \tau)$

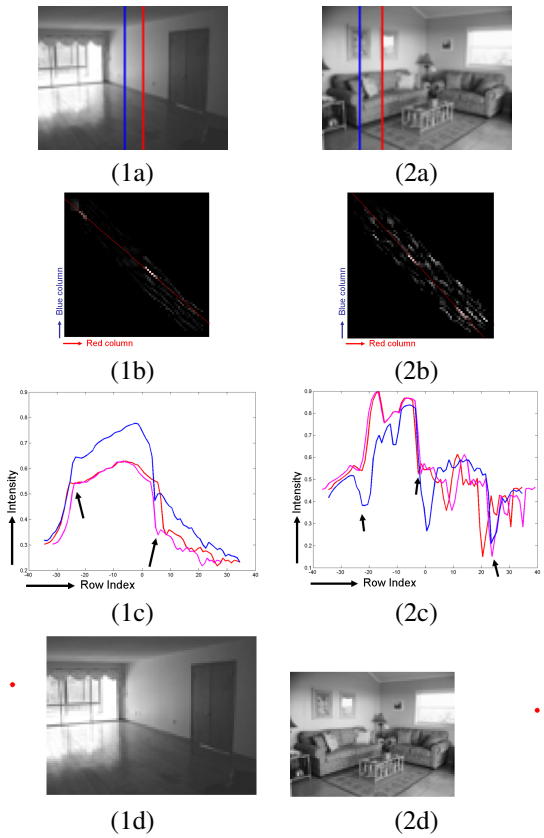

Figure 4. (a) Two columns marked on the image: red and blue. (b) The $SCORE$ matrix between the columns with the robustly fitted line in red (the line parameters characterizing the 1D-affine matching transformation). (c) Cross sections of the image luminance along the reference column (blue) and matched column before (red) and after (magenta) the 1D-affine transformation. The arrows highlight significant correspondences between the reference and matched column after transformation. (d) The VP candidate derived from the transformation parameters by Eq. 4 (red dot).

and $|i - CM|$ is a weight term that increases linearly with the distant of each point from the center of mass. We used a hierarchical exhaustive search to solve that maximization problem, but other optimization methods could be applied as well. Figure 4 demonstrates the estimation of the transformation parameters in a simple scene (on the left) and cluttered scene (on the right).

3.3. Algorithm summary

Our algorithm consists of the following steps:

1. Compute a scalar feature-map, e.g. 5×5 LoG.

2. For *vertical* cross-section matching: Form a feature vector map, such that the feature-vector at each point is a collection of feature map values in an elongated vertical local context of size 3×5.

3. Select partial set of column pairs in the feature-vector map, such that the horizontal separation between the two columns is large enough for a reliable detection of 1D affine similarity, and such that the pairs cover the entire image in a uniform fashion.

4. For each pair of columns find the 1D affine transformation (s, τ) that maximizes their matching. This is computed from their mutual Structural CORrelation Evidence ($SCORE$) matrix, by fitting a line that passes through a set of high SCORE points that are as spatially spread as possible. Then compute the candidate VP by converting (s, τ) into pencil-slope coordinates as defined in Section 2, and accumulate in a candidate VP store.

5. Apply a similar process to *horizontal* cross-section matching, with the necessary adjustments.

6. Cluster all candidate VPs by a mean shift procedure. The largest clusters are the desired VP estimates.

4. Results

We have evaluated our algorithm relative to a classical line-based method. Our line-based method implementation follows [14] as they provided good results for complex man made world. We used the same line detection tools as [14], available from a Matlab toolbox by Kovesi [6]: straight segments are extracted from linked edge lists by a subdivision scheme, and line segment parameters are obtained by a least squares fit. The implementation in [14] includes an initialization stage that provide a set of candidate VPs, and a refinement stage, which validates candidates and improves their accuracy. The refinement stage can not find VPs that were not detected in the initialization stage [5].

Our evaluation set consists of 76 images downloaded from the internet, where 20 which contain a single horizontal VP, and 56 contain two horizontal VPs, (132 VPs in total). We have created a GUI for marking manually the "ground truth" horizontal VPs. While manual marking is not very accurate, it reflects well the human perspective perception and is not biased towards any one of the two tested methods. The evaluation was made in several image resolutions, pushing towards very low resolutions to evaluate the potential of considerable computational speed-up through severe downscaling of the source image.

Figure 5 demonstrates the differences between the performance of our method and the line-based method by a variety of examples. The advantage of of our method over the line-based method in low resolution images is demonstrated in Figures 5(a-d). The height of all these images is 40 pixels. Note that humans can perceive the 3D scene geometry instantly from these low resolution images, despite the very blurred edges. In all cases (a-d) our algorithm detected VP

candidates quite close to the location we manually marked, while the line-based algorithm failed to detect any correct candidates due to failure to detect lines. A second advantage of our method is demonstrated in Figures 5(e-g) (image height 70 pixels), showing uncluttered scenes with very few straight lines converging at the vanishing point. In these cases, the line-based method does not prefer any line-pair intersection over the other, hence it creates VP candidates from arbitrary line pairs. Our model detects groups of lines that share a common range of columns or rows, and in most cases these lines are consistent with pencils related to VPs. Note in particular 5(f) where the right VP is not detected by the line-based method because the upper and lower edges of the closet are not associated with the same candidate. On the down side, the very same argument can cause our method to fail where the line-based method succeeds: 5(h) (image column 150 pixels) shows a failure case: there are almost no vertical cross section of pencils converging at the right VP, that do not suffer from major interferences (the lines of the rug that are not directed towards the right VP). As a result, our method fails to detect this VP, contrary to the line-based method.

A third advantage of our method is its capability to detect VPs from regular patterns, as shown in Figure 5(i-j), (image height 90). Particularly note 5(i), where the left VP is detected based on the regular pattern on the wall (virtual pencil), while the right VP is detected based on real lines (real pencil). No prior method can achieve that.

We compared the correctness of the vanishing point candidates (largest clusters) derived with out method to the candidates derived with the line-based initialization method. Each VP candidate inside a relatively large window around a ground truth VP was declared as correct. As it is very hard to create an exact ground truth of the vanishing points on real images, we focused on detection rates rather than on the accuracy. We have conducted the following experiment to test the robustness of our method to low resolution images. Each image was downscaled to a variety of sizes: from a 40 to 160 pixels high images. We measured the numbers of correctly detected VP candidates by our method and by the line-based method, presented in Figure 6. Note that our method significantly outperforms the line-based method in low resolutions. For instance, for 70 pixels image height, our method detects 96 our of the 132 VPs and the line-based method detects only 50. The detection rate of our method deteriorates with image size increase, probably since the fixed feature context size is too small for high resolution images. In contrast, the detection rate of the line-based method increases with resolution due to better line detection.

5. Conclusion

We presented a self-similarity based approach and a corresponding algorithm for vanishing points detection in man-made scenes. Our method does not rely on detection of explicit features such as straight lines or texels, and therefore can extract vanishing points from both lines or patterns perspective information. Our method relies on global structures and trends in the image rather then local details. This makes our method very robust against loss of image detail, for instance in low resolution images (much like the human perception). While the method handles well both indoor and outdoor man-made scene, more work is needed to extend it for general natural scenes. Another direction for future work is to explore the possibility that the human visual system uses principles similar to our method for perspective perception.

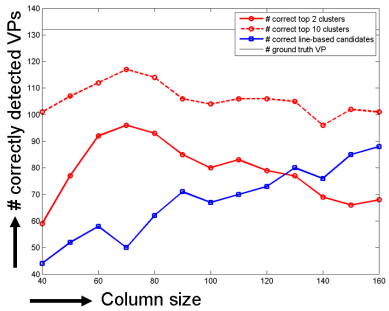

Figure 6. The detection rates vs. image size. The number of correct VP candidates with the line-based method (blue), the number of correct two largest clusters with our method (red), and the number of correct largest 10 clusters with our method (dashed red).

References

[1] S. Bernard. Interpreting perspective images. *Artificial Intel.*, 21:435–462, 1983.

[2] R. Collins and R. Weiss. Vanishing point calculation as a statistical inference on the unit sphere. *Third Inter. Conference on Computer Vision*, pages 400–403, Dec 1990.

[3] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *Inter. Journal of Comp. Vision*, 40(2):123–148, 2000.

[4] P. Gamba, A. Mecocci, and U. Salvatore. Vanishing point detection by a voting scheme. *Inter. Conference on Image Processing*, 1:301–304, Sep 1996.

[5] J. Kosecka and W. Zhang. Video Compass. *Lecture notes in comp. sceience*, pages 476–490, 2002.

[6] P. Kovesi. Matlab and octave functions for computer vision and image processing. In *School of Computer Science and Software Engineering, University of Western Australia.*

[7] A. Loh and R. Hartley. Shape from non-homogeneous, non-stationary, anisotropic, perspective texture. In *BMVC05: British Machine Vision Conf.*, pages 69–78, 2005.

[8] G. McLean and D. Kotturi. Vanishing point detection by line clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(11):1090–1095, 1995.

[9] E. Ribeiro and E. Hancock. Estimating vanishing point from the angular correlation of spectral texture distribution. In

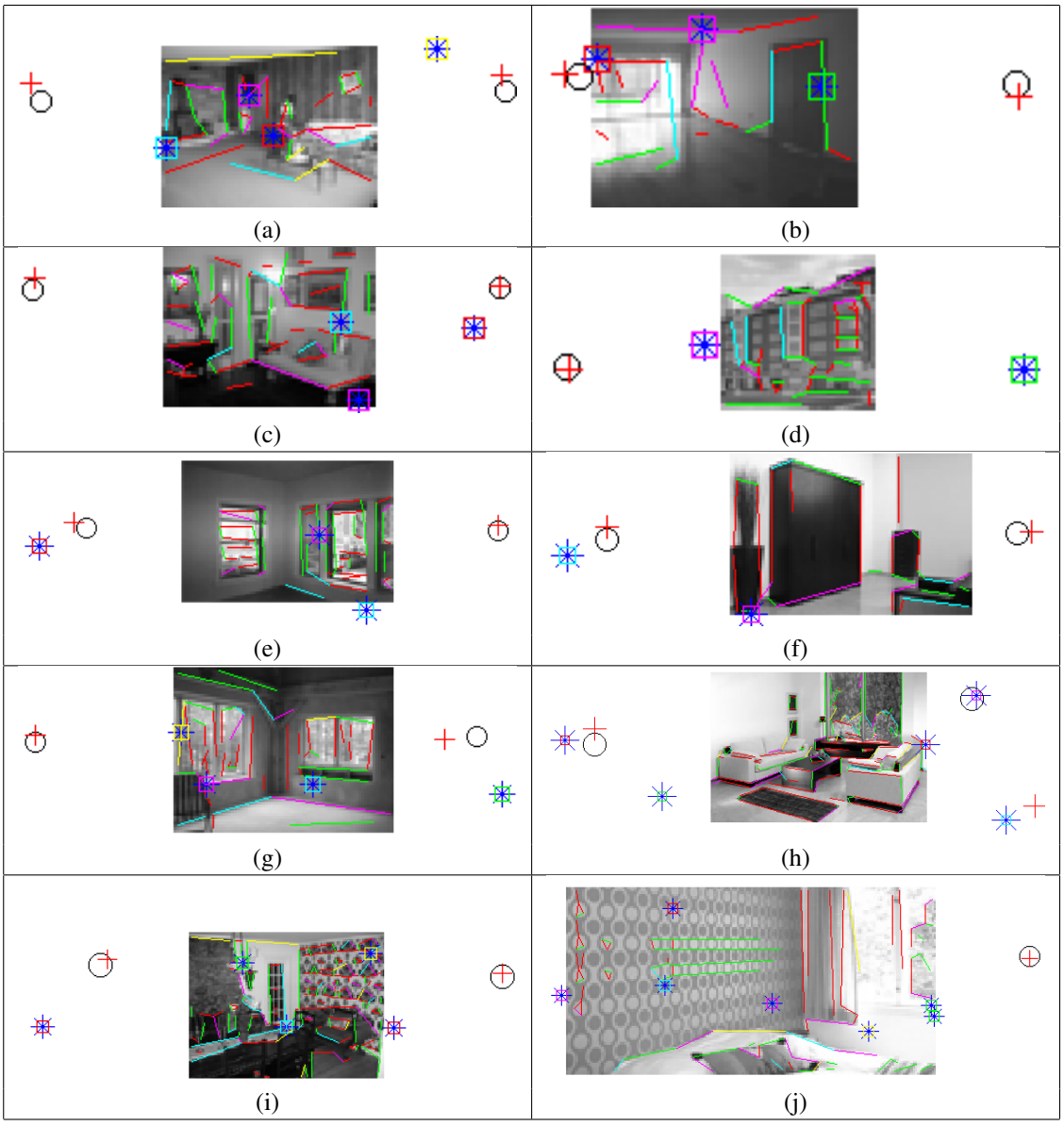

Figure 5. Various examples demonstrating the performance of our algorithm compared to the line-based algorithm. In all images, the red crosses are the two biggest clusters derived with our method (one on each side of the image), the black circles are our manual marking of the correct VPs location, and the blue stars are the result of the line-based initialization process. Each star is highlighted in a different color and the lines that are associated to that candidate VP are plotted with the same color. Note that the images were cropped for the sake of presentation, therefore the distant VP candidates are not shown.

Inter. Symposium on Computer Graphics, Image Processing and Vision, Brazil, pages 339–345, October 1998.

[10] F. Schaffalitzky and A. Zisserman. Planar grouping for automatic detection of vanishing lines and points. *Image and Vision Computing*, 18:647–658, 2000.

[11] K. Seo, J. Lee, and H. Choi. An efficient detection of vanishing points using inverted coordinates image space. *Pattern Recognition Letters*, 27(2):102–108, January 2006.

[12] F. Stentiford. Attention-based vanishing point detection. In *Int. Conf. on Image Proc. ICIP, Atlanta, 2006*, pages 417–420, Oct. 2006.

[13] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Tran. Image Proc.*, 13(4):600–612, Apr. 2004.

[14] H. Wildenauer and M. Vincze. Vanishing point detection in complex man-made worlds. *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 615–622, Sept. 2007.