Real-time robust estimation of vanishing points through nonlinear optimization

Marcos Nieto[a] and Luis Salgado[a]

[a]Grupo de Tratamiento de Imágenes - E.T.S. Ing. Telecomunicación
Universidad Politécnica de Madrid

ABSTRACT

Vanishing points are elements of great interest in the computer vision field, since they are the main source of information about the geometry of the scene and the projection process associated to the camera. They have been studied and applied during decades for plane rectification, 3D reconstruction, and mainly auto-calibration tasks.

Nevertheless, the literature lacks accurate online solutions for multiple vanishing point estimation. Most strategies focalize on the accuracy, using highly computational demanding iterative procedures. We propose a novel strategy for multiple vanishing point estimation that finds a trade-off between accuracy and efficiency, being able to operate in real time for video sequences. This strategy takes advantage of the temporal coherence of the images of the sequences to reduce the computational load of the processing algorithms while keeping a high level of accuracy due to an optimization process.

The key element of the approach is a robust scheme based on the MLESAC algorithm, which is used in a similar way to the EM algorithm. This approach ensures robust and accurate estimations, since we use the MLESAC in combination with a novel error function, based on the angular error between the vanishing point and the image features. To increase the speed of the MLESAC algorithm, the selection of the minimal sample sets is substituted by a random sampling step that takes into account temporal information to provide better initializations. Besides, for the sake of flexibility, the proposed error function has been designed to work using as image features indiscriminately gradient-pixels or line segments. Hence, we increase the range of applications in which our approach can be used, according to the type of information that is available.

The results show a real-time system that delivers real-time accurate estimations of multiple vanishing points for online processing, tested in moving camera video sequences of structured scenarios, both indoors and outdoors, such as rooms, corridors, facades, roads, etc.

Keywords: Vanishing points, robust estimation, MLESAC, Levenberg-Marquardt, projective plane

1. INTRODUCTION

Vanishing points are elements of great interest in the computer vision field, since they are the main source of information for structured environments. The simplest definition is that a vanishing point is a point in the image plane on which parallel lines of the scene seem to converge. The importance of vanishing points comes from the fact that these points of the image correspond to three-dimensional directions in the space. Vanishing points are studied and applied in applications such as plane rectification,[1] 3D reconstruction,[2] and for autocalibration algorithms.[3,4]

The estimation of vanishing points has been studied for decades and a wide number of works have been published offering different type of solutions. Former approaches follow the sphere normalization proposed by Barnard,[5] which work with accumulation spaces on the surface of the unit sphere, in a two-steps fashion: (i) image features (such as gradient pixels or line segments) vote in the accumulator, and (ii) searching algorithms find the most voted accumulator cells. Tuytelaars[6] uses the Hough transform in this sense for the detection of lines, vanishing points and vanishing lines. Further improvements have introduced error models, such as a swath model[7] or the vanishing hull,[8] which were used to consider measurement errors and uncertainties in the accumulation process.

More recent works face the problem as an optimization task that minimizes a global error value given by the set of image features, instead of as a voting and searching problem as done by previous approaches. For this purpose, an error or cost function is defined between a vanishing point and an image feature. Three main families of functions have been used, which can be classified as: (i) point-line distance in the image plane, in inhomogeneous[9] or homogeneous coordinates;[10] (ii) its extension to point-line distance after calibration,[11] which implies data normalization according to the camera calibration matrix; and (iii) end-points to line distances in the projective plane.[12, 13] The latter group operates under the assumption that a set of noisy line segments has been detected, which are corrupted by bivariate Gaussian noise affecting the position of the end-points of the line segments in the image. These approaches, which require the utilization of non-linear optimization algorithms for the estimation of vanishing points, provide the most accurate and reliable results of the literature.

Nevertheless, none of these groups offer a solution for an accurate and fast estimation of vanishing points in online applications with moving cameras. On the one hand, simpler approaches (such as those based on point-line distances) lack the required accuracy, as point-line distances have no geometrical sense as pointed out by Rother[14] so that their performance vary according to the actual position of the vanishing point. The second group requires the knowledge of the calibration matrix, which is an important limitation, and the third group, although very accurate and reliable, needs the implementation of robust initialization approaches, such as RANSAC,[15] which are iterative procedures than can be very time consuming.

In this work, we propose a new vanishing point estimation method specifically designed for online processing and to provide a trade-off between accuracy (due to the use of non-linear optimization methods), speed (using predictions in a tracking framework), flexibility (being able to work directly with gradient pixels or line segments if available), and robustness (through the use of an EM-like variation of the RANSAC algorithm, the so-called MLESAC[16]). It is based on the definition of a new type of error function, based on the relative orientation between the image feature and the line joining the vanishing point and the reference point of the image feature. We demonstrate that the proposed error function is better adapted than other approaches to the noise of the image features extracted from the image. Besides, we show that our error function can be used successfully in a robust scheme, the MLESAC algorithm, achieving more robustness against outliers that other robust schemes such as the simple RANSAC,[15] or MSAC.[10]

The use of our approach for tracking purposes allows us to provide predictions of the vanishing points to the MLESAC algorithm, so that the iterative nature of the algorithm is dramatically alleviated, requiring typically just one iteration to converge for low to moderate camera movement. The hypothesis is that vanishing points typically do not change much their position from one frame to the following, such that we can use previous estimations to initialize the MLESAC, which rapidly converge into the refined, current position of the vanishing points. Our strategy has been successfully applied for multiple vanishing point tracking in structured scenarios, both indoors and outdoors.

The paper is organized as follows: Section 2 describes the proposed error function, and compares it with other approaches, showing its advantages and the fast convergence of the related non-linear optimization procedure. Section 3 introduces the MLESAC algorithm and the formulation of the tracking procedures that reduces the number of required iterations to converge. Section 4 presents the results obtained for different tests done with both synthetic and real data.

2. ORIENTATION-BASED ERROR FUNCTION

Several error functions have been proposed in the literature to characterize the error between an image feature (typically a line segment or a gradient pixel) and a vanishing point. Point-line distances in the image plane[9,10] are the simplest ones, but only can be applied when the vanishing point is close to the image frame. Their major drawback is that these distances are not valid for vanishing points at the infinity, as the further a vanishing point is from the image, the larger the error will grow. Though, if the calibration matrix is known, the coordinates of the image plane can be normalized to the unit sphere. Hence, the point-line distance becomes a relative difference between two 3D orientations,[11] and infinity vanishing points can be handled.

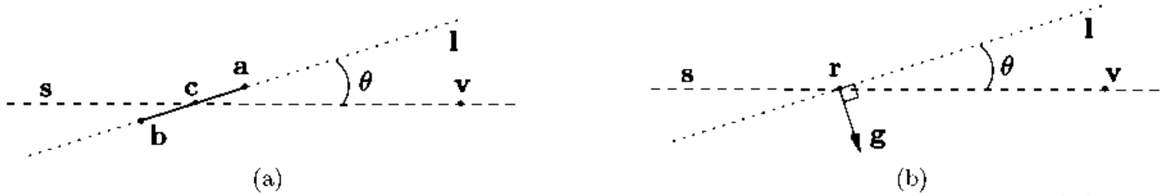


Figure 1. Relative orientation between the orientation of the measured feature and the vanishing point ((a) line segments and (b) edge-points). When the vanishing point lies on l, then the cost is minimum ($d(\mathbf{v}, \mathbf{l}) = |\sin\theta| = 0$).

As abovementioned, the best results are obtained with specific error functions defined for line segments. Liebowitz[12] proposed an error function based on the sum of the point-line distance of each end-point of the line segment with respect to a line that joins the vanishing point and the line segment. This error function, that has been modified by other authors,[13,14] requires a non-linear optimization process to obtain the vanishing point that minimizes the total error.

Nevertheless, more general approaches[17,18] consider the orientation error as the workspace on which perform optimization, such that the error function can be directly applied indifferently with pixel gradients and line segments. For instance, Schindler[17] considers the angle between the normal orientation of a pixel gradient and the line joining the vanishing point and that pixel. Inspired by this approach, we propose to use the sine of this orientation, so that our approach also can be applied for both line segments or gradient-pixels.

2.1 Proposed error function

Let $\mathbf{r}_i = (x_i, y_i, 1)^\top$ be the homogeneous coordinates of a point of the image plane with significant gradient magnitude, defined by the components of the gradient vector $\mathbf{g}_i = (g_{xi}, g_{yi})^\top$. Typically this can be obtained in images applying any operator that generates an approximation to the gradient vector at each pixel of the image. Besides, a line segment is defined by a pair of end-points, $\{\mathbf{a}, \mathbf{b}\}$, such that it defines the line $\mathbf{l}_i = \mathbf{a} \times \mathbf{b}$ in homogeneous coordinates. Its associated reference point is the mid-point $\mathbf{r}_i = \frac{\mathbf{a}+\mathbf{b}}{2}$. Analogously, any gradient pixel can be expressed as a line $\mathbf{l}_i = \mathbf{r}_i \times (\mathbf{r}_i + \lambda \mathbf{m}_i)$, where $\mathbf{m}_i$ is the the normal to the gradient vector, and λ is any real number.

Let consider from here on a generic data sample as $\mathbf{x}_i = \{\mathbf{r}_i, \mathbf{l}_i\}$, independently if it comes from a gradient-pixel or a line segment.

The distance between a vanishing point $\mathbf{v} = (v_1, v_2, v_3)^\top$, in homogeneous coordinates, and a data sample is defined as the absolute value of the sine of the angle between $\mathbf{l}$ and a line, $\mathbf{s} = (s_1, s_2, s_3)^\top$, which joins $\mathbf{v}$ and $\mathbf{r}$, and can be computed as $\mathbf{s} = \mathbf{v} \times \mathbf{r}$:

$$d(\mathbf{x}, \mathbf{v}) = \left| \frac{-l_2 s_1 + l_1 s_2}{\sqrt{l_1^2 + l_2^2}\sqrt{s_1^2 + s_2^2}} \right| \tag{1}$$

These concepts are illustrated in Fig. 1, for the two defined types of data samples. The absolute value is considered because we are only interested in the relative deviation between the orientation of the lines, not in their sign, so that the angle between them is limited between 0 and π. The use of the sine function is motivated by the fact that, for low values of the deviation angle, which is the case for inliers, $\sin\theta \simeq \theta$, thus avoiding the arctangent computation for large amounts of data.

To qualify the contributions of each data sample to the global error minimization process, our approach is extended by adding a scale factor to (1) that weights the importance of image features in the optimization process. This way, the error orientation-based scaled function can be defined as $d'(\mathbf{x}, \mathbf{v}) = S \cdot d(\mathbf{x}, \mathbf{v})$, where S is a factor that weights the importance of image features. For instance, in the case of using edge-points, S is the magnitude of their gradient, so that more intense edge-points are more important. For line segments, the scale could be related to their length or any quality measurement obtained in the feature extraction process. As it will be shown, the use of this information makes the system to perform better against outliers.

2.2 Global optimization

Considering Gaussian noise in the defined error function, the likelihood of a data sample to meet a given vanishing point is:

$$p(\mathbf{x}_i|\mathbf{v}) \propto \exp\left(-\frac{1}{2\sigma^2}d^2(\mathbf{x}_i,\mathbf{v})\right) \tag{2}$$

Given a set of i.i.d. data samples $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, actually meeting a single vanishing point (i.e., all data are inliers)* the global likelihood function is:

$$p(\mathcal{X}|\mathbf{v}) = \prod_{i=1}^{N} p(\mathbf{x}_i|\mathbf{v}) \tag{3}$$

Provided that the conditional distribution (2) is normal, the maximum likelihood estimation is solved by minimizing the sum of squared costs $\mathcal{C} = \sum_{i=1}^{N} d^2(\mathbf{x}_i, \mathbf{v})$. However, this is a difficult problem to solve analytically as $\mathcal{C}$ is highly non-linear. Therefore, approximate numerical methods for non-linear optimization are required. In this work we propose to use the Levenberg-Marquardt algorithm,[3] which has been already used by other authors whose error functions are non-linear.[12, 13]

2.3 Image features

This subsection compares the characteristics of the noise of detected gradient pixels and line segments as far as it concerns the computation of vanishing points. Since our strategy is flexible enough to work with any of these features, we compare the benefits of using them regarding different aspects. For instance, it has been shown that generic sparse sets of edges are more accurate than gradient maps for the whole image.[18] This way, the use gradient pixels for vanishing point estimation (which is reasonable for many applications and platforms since the Sobel operator is typically easy to implement in an efficient way), is subject to the introduction of a sub-selection stage that selects just a few pixels of the image. Otherwise, the robust approaches that compute vanishing points, such as RANSAC or EM, would suffer a great decrease of their performance due to the excessive number of data sample to handle in their iterative procedures. In that sense, we propose to use the Sobel operator as the approximation to the gradient, and only those pixels whose gradient magnitude is below a certain threshold are selected. The resulting set of gradient-pixels is then ordered by magnitude. Following, gradient pixels are selected in descending order and the neighborhood of each selected pixel with a $r \times r$ window is removed. The result is a sparse set of gradient-pixels, whose cardinality is configurable with the threshold value and the window size.

Nevertheless, line segments detection strategies typically provide better sparse sets of features. For the extraction of the line segments we have designed a fast method that is based on a random sampling strategy that sequentially select pixels in the image according to their gradient magnitude. For each selected pixel a non-maximal suppression and a growing algorithm based on orientation comparisons between neighbor pixels are applied in order to determine the line segment. We have seen that the results we obtain are very similar, in terms of speed and performance to those of the line segment detector that claims to be one of the fastest and more accurate on the literature.[19] For a generic image, we have found that the line segment detector is not more than two or three times slower than the obtention of an sparse set of gradient pixels.

Besides, to compare the error distribution of the different sets, we have conducted a test on the set of images of the York Urban Data Base[18] for which ground truth vanishing points and calibration are available. In summary, for the 102 images of the database, provided 3 vanishing points per image, we observed that, as expected, the gradient-pixels set actually suffer more outliers proportion than line segments sets; i.e. the outliers proportion is understood as the ratio between the number of image features than do not meet any ground truth vanishing point in an image and the total number of image features. In average, this proportion is 8% for line segments and 19% for gradient-pixels. On the other hand, the orientation error with respect the ground truth vanishing

*The classification between inliers and outliers is provided in Section 3.

points is also higher for gradient pixels. The reason is simple: one long line segment got less orientation error than its corresponding set of gradient pixels. Analogously, the average orientation error of inlier gradient pixels is of about 12, whereas only 5 for line segments.

This way, it is always advisable to use line segments if it possible to implement the detector efficiently, such that its computational cost is affordable for real-time operation. If not, the sparse set of gradient-pixels can still be be used for real-time vanishing point estimation in a reliable way, although the performance of the system would probably slightly decrease.

3. MLESAC STRATEGY

The following subsection introduces the robust estimation of vanishing points with the RANSAC algorithm, and its extensions MSAC and MLESAC algorithms.[16] At the end of the section, the extension of MLESAC to the tracking system is described in detail for video sequences considering camera motion.

3.1 MLESAC algorithm

The robust estimation of vanishing points using RANSAC can be done iteratively, in a similar way to other works in the literature.[15] At each iteration, given the complete set of observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, a single dominant vanishing point is hypothesized, such that each data sample is modeled to have been drawn from a mixture model of two classes, that are labeled as $c = \{ON, OFF\}$, indicating respectively that the data sample meets the vanishing point or not (i.e. it is an outlier). The likelihood model is defined as follows:

$$p(\mathbf{x}_i|\mathbf{v}) = \sum_{c} p(c)p(\mathbf{x}_i|\mathbf{v}, c) \tag{4}$$

where the density $p(\mathbf{x}_i|\mathbf{v}, c = ON)$ is modeled as a normal distribution as in (2), and $p(\mathbf{x}_i|\mathbf{v}, c = OFF)$ is a uniform distribution. The value of the uniform distribution is computed as the inverse of the highest expected error, so that, for the defined error function, the maximum error corresponds to the case in which the orientation of $\mathbf{s}$ is orthogonal to $\mathbf{l}$ and the sine of the angle is 1. Hence, $p(\mathbf{x}_i|\mathbf{v}, c = OFF) = 1$.

After the estimation, the inliers are removed from the data set and the presence of another dominant vanishing point is hypothesized.

Summarizing, RANSAC proceeds iteratively in two main steps: (i) a minimal sample subset (MSS) of data is randomly selected in order to generate a hypothesis of the vanishing point. In this case, the minimal set is composed by two data samples whose best estimation is provided by their line parameters as $\mathbf{v}^* = \mathbf{l}_i \times \mathbf{l}_j$; and (ii) the consensus set, $CS(\mathbf{v}^*)$, i.e. the subset of data samples that are coherent with the hypothesis, defined as:

$$CS(\mathbf{v}^*) \triangleq \{\mathbf{x}_i \in \mathcal{X} : d^2(\mathbf{x}_i, \mathbf{v}^*) \leq \delta\} \tag{5}$$

where δ is a certain error threshold governed by the statistics of the data. These two steps are repeated until the probability of finding a better consensus set is below the convergence threshold. Here resides the difference between MSAC and RANSAC: while basic RANSAC ranks the consensus sets according to their cardinality, i.e. the number of inliers, the MSAC version ranks the consensus set according to a global cost function that gathers individual contributions of inliers and outliers. The total cost of an hypothesis of the MSAC iterative procedure is given by $\mathcal{C}(\mathcal{X}, \mathbf{v}^*) = \sum_{i=1}^{N} \rho(\mathbf{x}_i, \mathbf{v}^*)$, where $\rho(\mathbf{x}_i, \mathbf{v}^*)$ is defined as:

$$\rho(\mathbf{x}_i, \mathbf{v}^*) = \begin{cases} d^2(\mathbf{x}_i, \mathbf{v}^*) & , d^2(\mathbf{x}_i, \mathbf{v}^*) \leq \delta \\ \delta & , \text{otherwise} \end{cases} \tag{6}$$

As a final remark, the minimal sample subset selection can be carried out more efficiently for both RANSAC and MSAC by using any type of sampling strategy that is more peaked than a purely random process. Thinking in sampling as the procedure in which each data sample is given a certain probability of being selected, random sampling assigns the same probability to all samples. A more efficient approach is to assign to each sample a value that encodes its importance. For instance, line segments are assigned their length, and gradient pixels their gradient magnitude, since longer line segments and stronger gradient pixels are more likely to be less noisy and, therefore, more likely to be inliers.

The MLESAC[16] defines a more accurate ranking system: provided that the likelihood of data samples with respect to the hypothesized vanishing point is available, the total cost is computed as the global likelihood for all data samples as $p(\mathcal{X}|\mathbf{v}^*) = \prod_{i=1}^{N} p(\mathbf{x}_i|\mathbf{v}^*)$. The only problem with this approach is that the a priori probability, $p(c = ON)$, is unknown and must be estimated[†]. Nevertheless, this estimation can be done with a one-dimensional EM optimization procedure. Typically, with an initialization of the a priori probability as $p(c = ON) = 0.5$, convergence is achieved in just 3 or 4 iterations.

The convergence of the MLESAC algorithm is related to the mean log-likelihood value, which is given by:

$$\log\left(p(\mathcal{X}|\mathbf{v}^*)\right) = \frac{1}{N}\sum_{i=1}^{N} \log \sum_{c} p(c)\; p(\mathbf{x}_i|\mathbf{v}^*, c) \tag{7}$$

At each iteration, indexed as k, the improvement of mean log-likelihood is checked as:

$$\Delta \log\left(p(\mathcal{X}|\mathbf{v}_k^*)\right) = \frac{\log\left(p(\mathcal{X}|\mathbf{v}_k^*)\right) - \log\left(p(\mathcal{X}|\mathbf{v}_{k-1}^*)\right)}{\log\left(p(\mathcal{X}|\mathbf{v}_{k-1}^*)\right)} \tag{8}$$

When $\Delta \log\left(p(\mathcal{X}|\mathbf{v}_k^*)\right)$ is below a certain threshold ϵ, the process is considered to have converged. Typically it is accurately enough to use $\epsilon = 10^{-5}$.

3.2 Extension to tracking

The number of iterations of the RANSAC-like approaches can be dramatically reduced if the random sampling process that selects the MSS is substituted by a deterministic algorithm that uses the information of the estimations of the previous time instant.

If the motion of the camera is not too high, it can be assumed that vanishing points between consecutive images of a sequence do not show significant changes in their position. This way, we can compute the error between the data samples obtained in time t, and the previous estimations of vanishing points, $\mathbf{v}_{t-1}$, in time $t-1$. The two data samples with lower error are selected as MSS, and a new vanishing point candidate, $\mathbf{v}_t^*$, is generated. Typically, this candidate is accurate enough to obtain a very high log-likelihood value (computed as in (7)), thus not requiring new iterations to be carried out.

In our experience, we have observed that, in average, very few iterations are required for a very accurate estimation of the vanishing points, reducing dramatically the cost of the MLESAC algorithm and allowing real-time processing.

4. TESTS AND RESULTS

In this section we want to demonstrate the ability of our approach to provide accurate estimation of multiple vanishing points and real-time operation. For this purpose we conduct a set of tests, using both synthetic and real data examples. These tests will show that our approach, on the one hand, provide very accurate estimations due to the use of nonlinear optimization processes, and that the proposed error function provides faster convergence to these processes. On the other hand, the performance of the proposed tracking-oriented variant of the MLESAC algorithm for is presented for different video sequences.

[†] Note that $p(c = OFF) = 1 - p(c = ON)$.

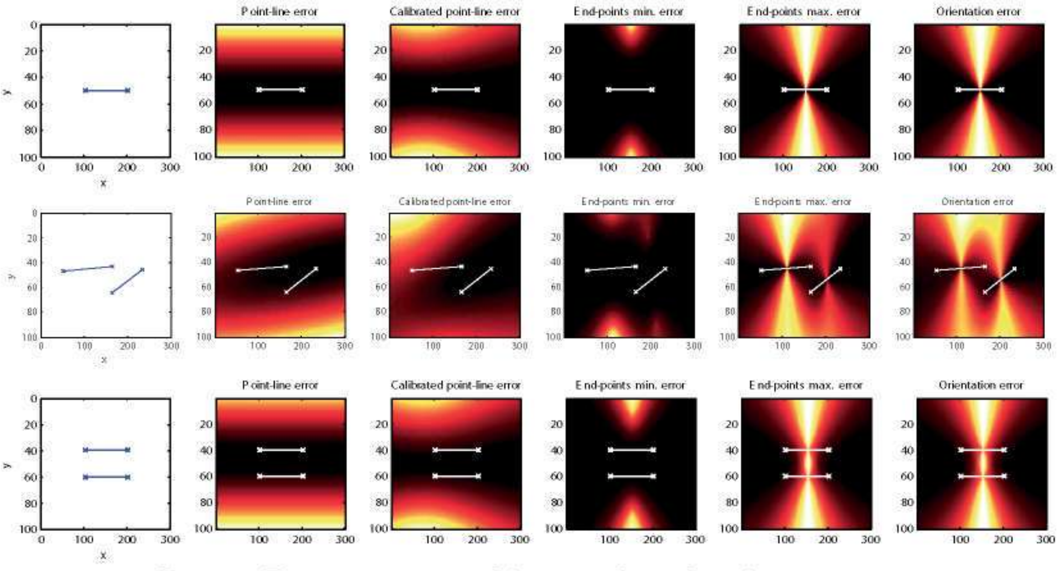


Figure 2. Error maps generated for example synthetic line segments.

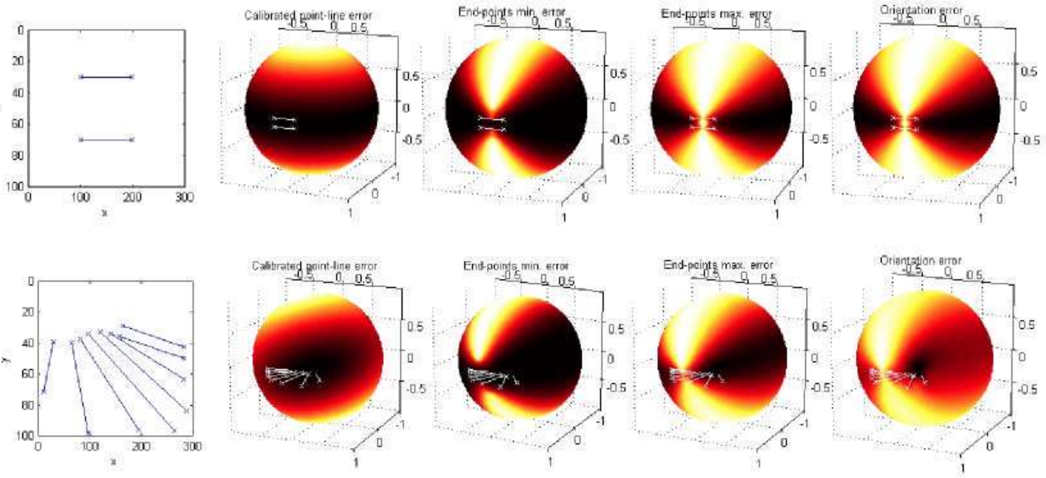


Figure 3. Error maps generated for synthetic examples in the unit sphere.

4.1 Synthetic data

The estimation accuracy of our approach (i.e. using the orientation-based error function plus non-linear optimization) is compared with respect to other well known approaches in the literature: (1) end-point distance plus non-linear optimization (that we will call EPM) by Liebowitz;[12] (2) the extension of this work by Tardif (EPx);[13] (3) the calibrated point-line (CPL) distance plus linear optimization estimation by Košecká,[11] and (4) point-line (PL) distance plus linear optimization. Our approach will be denoted as O (coming from orientation error function), and OS for its extension using the scale factor.

4.1.1 Qualitative comparison

We first want to evaluate how the error maps generated with the different error functions look like using some synthetic examples. Fig. 2 illustrates the error associated to each position of the space (contained inside the limits of the image frame) given different sets of line segments. In these error map representations, darker regions correspond to points in the image with low associated error (which means that are likely to be the vanishing points). Brighter regions depict high error positions. The first row illustrates the case of a single line segment, which shows quite clearly the differences between the methods. For instance, the point-line distance delivers an error map that grows with the orthogonal distance of the point to the line segment. The calibrated point-line

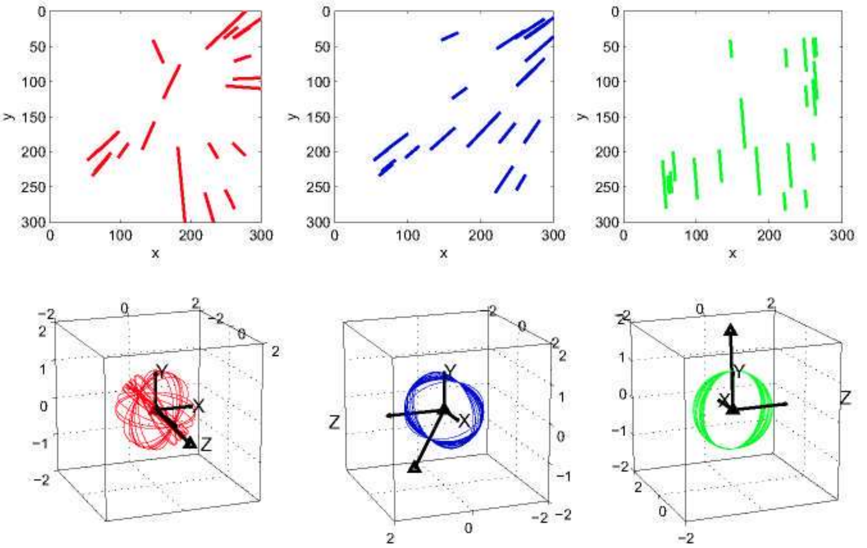


Figure 4. Ground truth line segments for three different vanishing points, and their representation in the unit sphere (note that we use here line segments instead of gradient pixels for a better visualization).

behaves similarly, though is better understood if we consider the unit sphere representation, shown in Fig. 3. The method EPx and our proposed orientation-based method result about the same error map for this simple case, as the error grows with the angle with respect the mid-point of the line segment. The EPM method is similar to these ones, but with a wider margin near the line segment. The second row includes two converging line segments, and the associated error maps. The third row has been intentionally included to depict the problems of this image-plane representation for lines converging outside the limits of the image, as is the case for parallel lines, that meet in the infinity.

Fig. 3 show the error maps in the unit sphere, for all the methods except for the point-line distance, that can not be projected into the sphere since its error grows up to infinite for infinite vanishing points. A few words shall be said about this type of projection: the information of the image plane is projected into the unit sphere using the camera calibration matrix, K, so that points are calibrated as $\mathbf{v}' = K^{-1}\mathbf{v}$ subject to $\|\mathbf{v}'\| = 1$, and lines as $\mathbf{l}' = K^{\top}\mathbf{l}$. In the three-dimensional projective space, we can obtain the error value for each position in the space for each method by computing the corresponding sum $\sum_{i=1}^{N} d^2(\mathbf{v}', \mathbf{x}'_i)$ and plotting it into the sphere[‡].

The two parallel line segments case is shown in the first row of Fig. 3, where it can be observed that the low error regions are at the horizontal axis, which corresponds to the direction in the space that is orthogonal to the optical axis of the camera, or parallel to the image plane. As shown, better accuracy is provided by the EPM, EPx and orientation-based (O) method rather than the point-line (CPL), that share similar properties with the PL but in the projective plane. The second row includes a larger set of line segments, with some added noise. In this case, the error map of the proposed orientation-based method is clearly more peaked around the best vanishing point. Although it is clear from the figures that all the methods provide low error values in this region, we shall demand them to give also high error values for the rest of the space (as done by the orientation-based method). This is important since estimation methods based on iterative procedures may be initialized randomly in the sphere, and the sharper the error map is, the less number of iterations the process would require. This concepts are studied in the following subsection, which also includes examples of error maps for line segments obtained from real images.

4.1.2 Quantitative comparison

Fig. 4 (first row) illustrates the ground truth synthetic line segments in the image plane, meeting three different vanishing points, which have been defined to be inside the image boundaries (first vanishing point), outside

[‡] Details about sphere projection can be found in any vanishing point related work.[5, 11, 20]

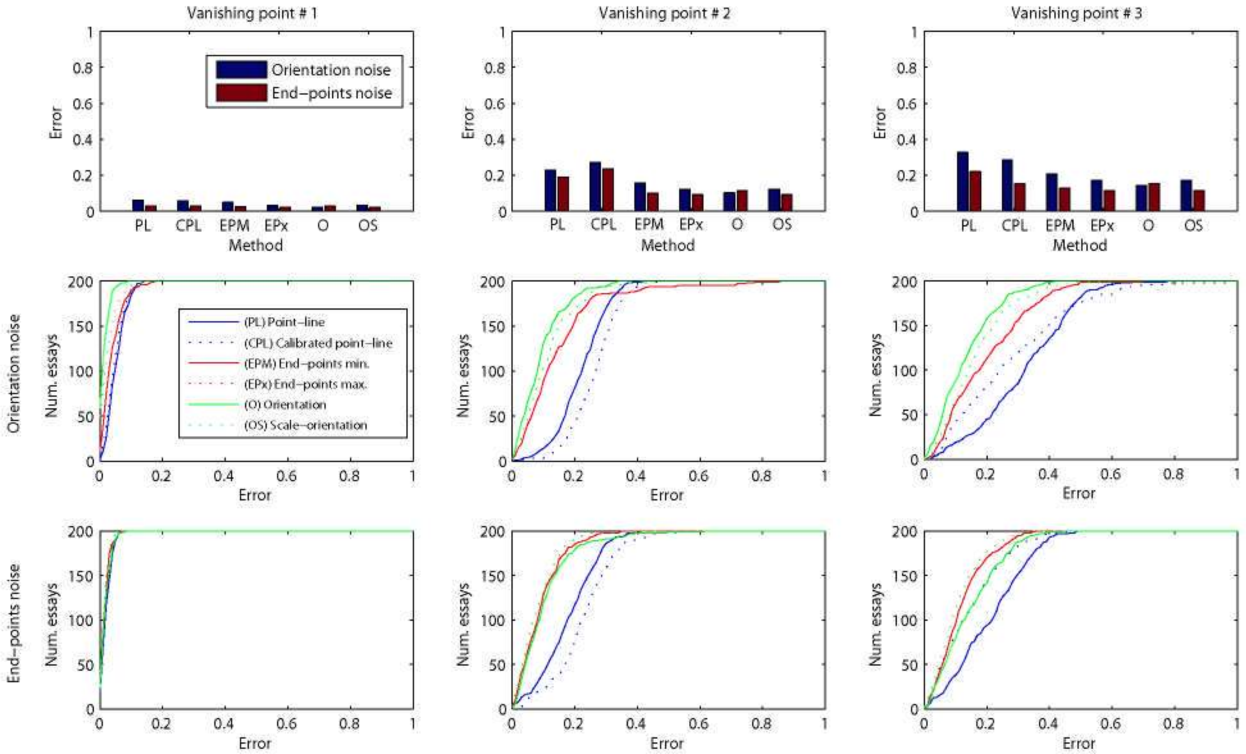


Figure 5. Results of the synthetic tests for 200 essays. The first row shows the histograms of the error for each vanishing point and each method. The second row contains cumulative error histograms for the orientation noise assumption. The third row shows the corresponding cumulative error histograms for the end-point noise model.

(second vanishing point), and in the infinity (third vanishing point). The second row shows the corresponding representation of these line segments in the unit sphere. Note that each line segment correspond in this space to planes that pass through the optical center, and thus can be represented by circumferences on the sphere surface. The ground truth vanishing point is shown as a solid black line. This representation helps to understand vanishing points as 3D vectors corresponding to space directions rather than 2D points on the image plane.

Additionally, Gaussian noise is added to corrupt the data samples, in two different ways: (a) the position of the end-points (which corresponds to the noise model hypothesized by Liebowitz); and (b) in the orientation of the data sample (corresponding to the noise model proposed in this paper). The values of the noise distribution are $\sigma_x = \sigma_y = 3$ (pixels) in the case of bivariate noise in the position of the end-points, and $\sigma_\theta = 0.1$ for the noise in the orientation.

All the estimation methods are applied to the different data sets, and the estimation error is computed as the sine of the relative angle between the detected vanishing point, $\hat{\mathbf{v}}$, and the ground truth vanishing point, $\mathbf{v}_{gt}$ (provided a ground truth calibration of the synthetic view, and a normalization of the data as described by Košecká[11]). Hence, this error ranges from 0 for perfect matching, to 1 for orthogonal directions.

The tests are run 200 times, and the obtained results are shown in Fig. 5. The first row shows the average error obtained for the different methods, using the two types of noise models, for the three considered vanishing points. As expected, the point-line methods offer the higher errors, for both noise types. The evaluation of the performance of the end-points-based methods and the proposed orientation-based methods is more clearly depicted in second and third rows. These figures illustrate the cumulative histograms of the committed error, respectively for orientation and end-points noise model. As shown, the vanishing point number 1 is well estimated by all the methods, and there is no significant difference between them. However, for vanishing points number 2 and specially the number 3, the end-points methods (both EPM and EPx) deliver more error than the proposed approaches O and OS. Nevertheless, under the hypothesis of bivariate Gaussian noise in the position of the end-points, the results for EPM, EPx and our approaches is not significant.

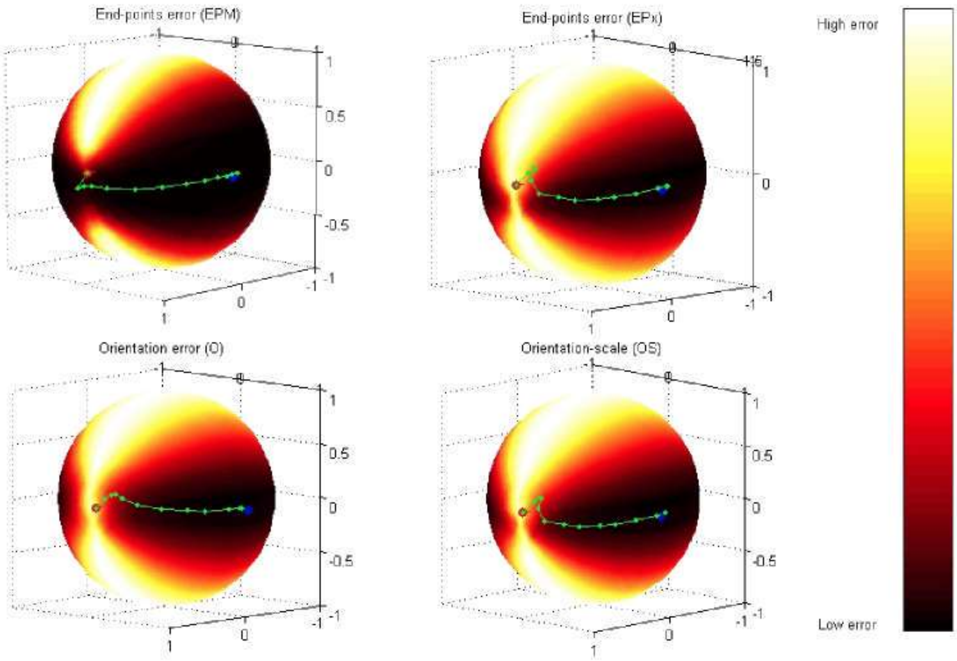
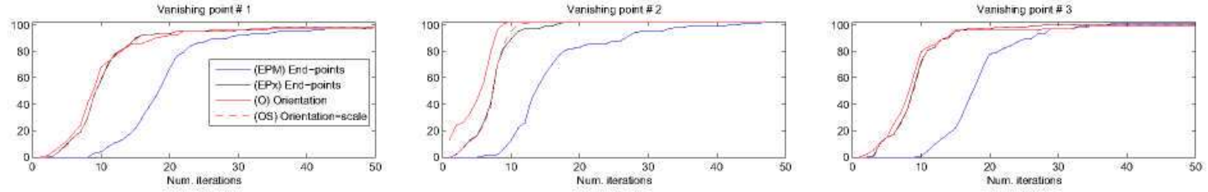


Figure 6. Error maps for an example line segment set. Darker regions correspond to the low error values of the vanishing point.



(a) (b) (c)

Figure 7. Cumulative histograms of the number of required iterations for the non-linear optimization procedures for the three different vanishing points of the set of images of the YUDB.

4.2 Real images

To test the performance of the proposed methods with real images we have chosen an available image database, called the York Urban Data Base (YUDB),[18] composed by 102 images of structured environments (corridor, facades, parkings, etc.) for which the calibration information is available as well as the ground truth of the three dominant vanishing points. We use the YUDB since it also includes the ground truth line segments meeting the dominant vanishing points. Although manually selected, these line segments are noisy, as well as the ground truth vanishing points, which are computed with the Collins method.[21]

We can generate error maps for each vanishing point of each image considering considering their corresponding sets of line segments as done in the synthetic data examples. This way we can easily find the position of the vanishing point that provides the lower error, and test the different non-linear iterative methods (EPM, EPx, O and OS) using the same randomly obtained initialization point. Specifically, we want to test the number of iterations of the Levenberg-Marquardt algorithm required by the different error functions to reach the ground truth vanishing points. Fig. 6 shows the unit sphere error map for one vanishing point of a real image. The darker regions are lower error regions, which correspond to values of the parameters of the vanishing point that produces lower global error. The initialization is shown as a red point on the sphere surface, and the iterations are shown as a green path. The final estimation is depicted in blue (last steps of the path are very close to the final estimation, so that they can not be seen in the sphere). As expected, all methods offer the same final

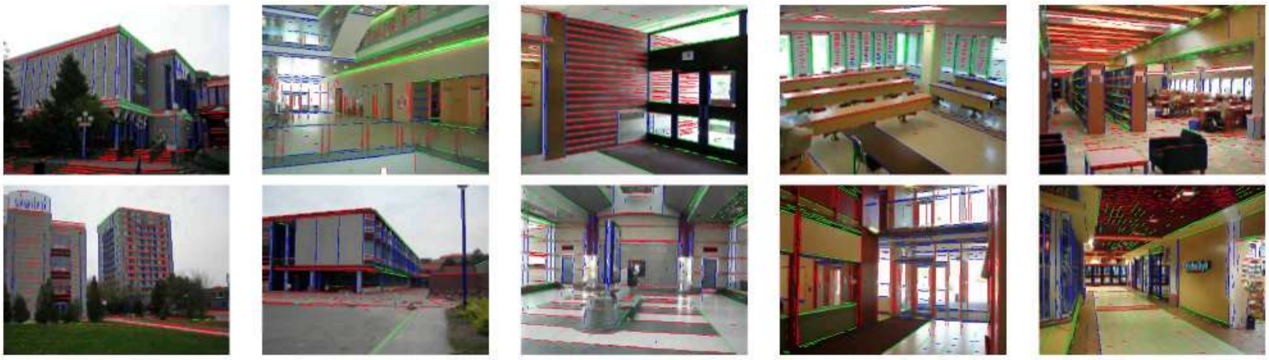


Figure 8. Examples of the use of the proposed MLESAC algorithm in images of the YUDB.

estimation. Nevertheless, our methods display narrower error valleys, which lead to faster convergence of the optimization procedures. This is a key advantage for our approach as it is devoted to real-time processing. For a reduced or fixed available number of iterations the error of our approach is lower than the others, since the error valleys are sharper and thus in few iterations we have already reached very low error regions, while the other methods typically spend some more iterations without significant error reductions.

Fig. 7 shows the cumulative histogram of the required number of iterations for the different methods for all the vanishing points of all the images of the data base.

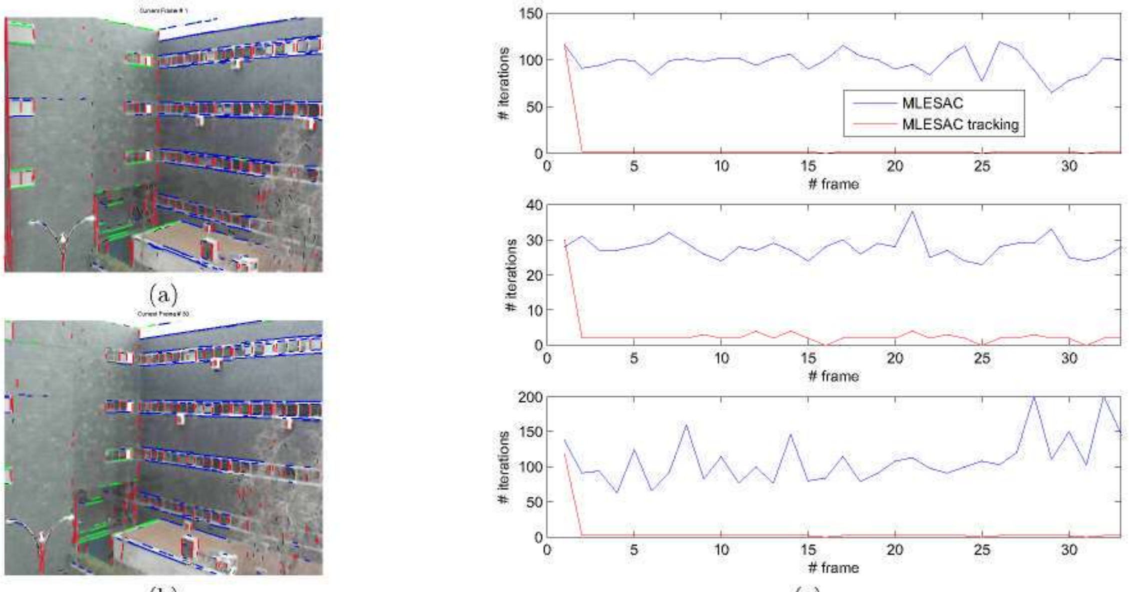
As shown, the number of iterations provided by the proposed orientation-based methods are lower in average for all vanishing points, specially for the second one (which corresponds to the vertical vanishing point of the images in the YUDB). Regarding the end-points methods, the one by Tardif[13] clearly improves the results of the method by Liebowitz.[12]

At this point, we can discuss about the two proposed methods, which only differ in the use of the scale factor. In general terms, as shown by the described tests, the performance of these methods is very similar in average, but slightly better for the orientation-based method without scale. In the case of synthetic data, all data samples have been corrupted by the same noise level, such that the hypothesis that longer or stronger image features are less noisy is not kept, hence the orientation-based with scale method can not provide better results. For the real data tests, the reasoning is similar, since the used data samples come from manual selection, which again shall produce noisy line segments incoherent with the mentioned hypothesis. Nevertheless, for real detected line segments and gradient pixels, we have observed that the use of scale factor indeed increase the performance of the methods. Therefore, the MLESAC is used, in the following examples in combination with the orientation-based method that uses the scale factor.

As far as we can compute our own ground truth vanishing point (provided by the lower error point on the sphere error map), which improves the accuracy of the ground truth vanishing points of the YUDB, we can test the robustness of the proposed MLESAC robust scheme given a real detection of line segments with presence of outliers. Although there are 102 images in the data base, not all of them contain three vanishing points. Thus, we have observed that 98 images have information about the first horizontal vanishing point, 102 about the vertical one, and 101 about the second horizontal vanishing point. For clarity in the results, we consider that a vanishing point has been correctly detected when the error angle is below 10 degrees with respect to the ground truth.

We have tested the MLESAC and it provides excellent detection results, with 91/98, 96/102 and 84/101 detection rates, respectively for the first, second and third vanishing point. The average detection value is 90.03% of the total number of vanishing points, and the average error is below 3 degrees.

Fig. 8 shows the results of the MLESAC approach, using the Orientation-scale error function defined in this paper, for some example images of the YUDB. For a better visualization line segments are colored according to the vanishing point they meet. As shown, excellent results are obtained for structured scenarios, which includes non-structured elements such as trees, or pedestrians, that increase the total number of outliers, and thus difficult the estimations.



(a)

(b)

(c)

Figure 9. Number of iterations of the MLESAC algorithm with and without using the prediction information.

4.3 Video data

As a tracking solution, the MLESAC has been extended to use previous estimations of vanishing points, so that the required number of iterations is dramatically reduced. To illustrate this advantage, we have tested the MLESAC algorithm, with the proposed error function for a sequence of an outdoor structured scenario, with and without using the selection of MSS. The results are shown in Fig. 9. In (a) and (b) a couple of frames of the sequence are shown, with the detected line segments colored according to the vanishing point that they meet (red for the vertical one, blue and green for the two horizontal vanishing points). The number of iterations required by the MLESAC algorithm are shown in (c). The first row correspond to the vertical vanishing point, and the second and third row correspond to the other two vanishing points. As can be observed, the number of iterations of the MLESAC algorithm are always very low, ranging between 2 and 4, except for the first frame, that requires a full MLESAC computation. Without the prediction information, the number of iterations is typically very high to obtain similar values of likelihood.

Fig. 10 shows an example of the application of the proposed strategy to detect and track three mutually orthogonal vanishing points in an indoor scene. Specifically, the scene corresponds to a surveillance environment, monitored with PTZ cameras (that stands for pan, tilt and zoom cameras). In this application, the motion of the camera is limited to rotation in two axes. For the example sequence shown in the figure, the rotation incurred by the camera is on the vertical axis, which is depicted in the figures of the bottom, where the two horizontal vanishing points, shown in red and blue, rotates along the green, vertical one. The proposed strategy has been also applied for other outdoor scenarios, such as those exemplified by Fig. 11, where the found vanishing point also typically identify the three main orthogonal directions of the imaged structured elements of the scenes. It is noteworthy the case of the road scenario, which, although being the one with highest speed camera motion, it corresponds to movement in the direction of the optical axis. This way, the vanishing point of the lane markings of the road does not vary significantly its position over time and can be easily tracked by our approach.

5. CONCLUSIONS

In this paper we have introduced a new robust strategy for real-time vanishing point tracking in structured scenarios. The key contribution is that we find a trade-off between the accuracy of non-linear optimization methods and the fast operation of robust techniques using the MLESAC algorithm, which is extended to use information from previous time instants to reduce the number of required iterations to converge.

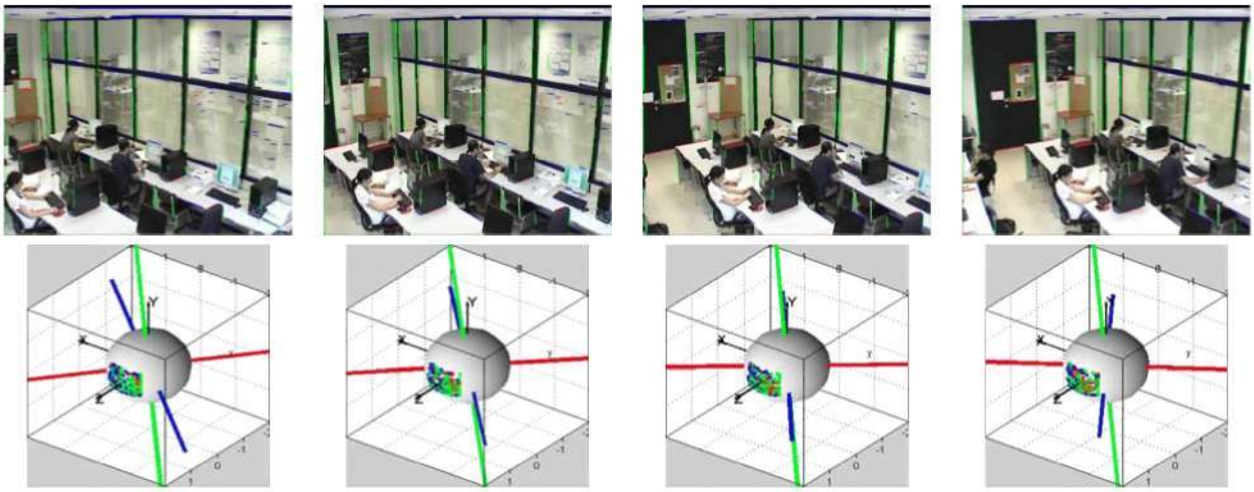
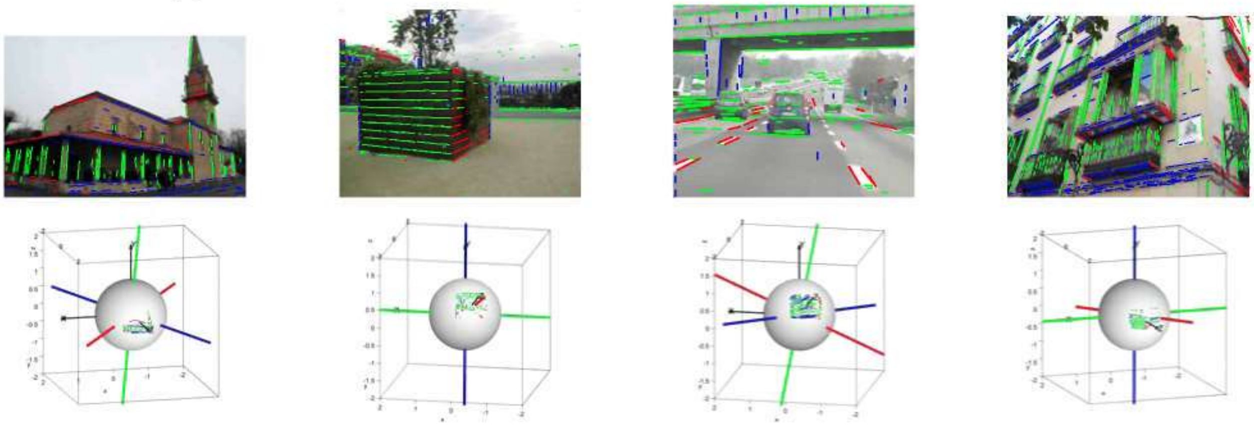


Figure 10. Example images of a typical surveillance video sequence in an indoor scenario. Line segments are colored according to the vanishing point they meet. In the lower row, a representation on the unit sphere is shown, where the three main vanishing points are shown as thick colored lines.



(a) (b) (c) (d)

Figure 11. More examples of the obtained three mutually orthogonal vanishing points for different indoor and outdoor sequences.

We have shown that the definition of a new error function, based on the scaled orientation of the image features (that can be both line segments or gradient pixels), achieves very high estimation accuracy as well as define more peaked error maps which leads to faster convergence for optimization algorithms like Levenberg-Marquardt.

The results, obtained with synthetic and real images, as well as sequences of images show that our approach is an excellent tool for real-time vanishing point tracking in structured scenarios, which can be used for applications such as plane rectification or camera autocalibration.

ACKNOWLEDGMENTS

Thanks to P. Denis for the York Urban Data Base. Besides, this work has been supported by the Ministerio de Educacin y Ciencia of the Spanish Government under project TEC2007-67764 (SmartVision) and TEC2006-26845-E (HIGHWAY: Intelligent co-operative system in cars for road safety).

REFERENCES

[1] Liebowitz, D. and Zisserman, A., "Metric rectification for perspective images of planes," in [*IEEE Computer Society Conference on Computer Vision and Pattern Recognition*], 482–488 (1998).

[2] Carballeira, P., Ronda, J. I., and Valdés, A., "3d reconstruction with uncalibrated cameras using the six-line conic variety.," in [*International Conference on Image Processing*], 205–208 (2008).

[3] Hartley, R. I. and Zisserman, A., [*Multiple view geometry in computer vision*], Cambridge University Press (2004).

[4] Liebowitz, D., Criminisi, A., and Zisserman, A., "Creating architectural models from images," in [*Computer Graphics Forum*], 39–50 (1999).

[5] Barnard, S. T., "Interpreting perspective images," *Artificial Intelligence Journal* **21**(4), 435–462 (1983).

[6] Tuytelaars, T., Proesmans, M., and Van Gool, L., "The cascaded hough transform," in [*IEEE International Conference on Image Processing*], 736–739 (1998).

[7] McLean, C. F. and Koyyuri, D., "Vanishing point detection by line clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence* **17**(11), 1090–1095 (1995).

[8] Hu, J., You, S., and Neumann, U., "Vanishing hull," in [*3DPVT'06: Int. Symposium on 3D Data Processing, Visualization, and Transmission*], 448–455 (2006).

[9] Minagawa, A., Tagawa, N., Moriya, T., and Gotoh, T., "Vanishing point and vanishing line estimation with line clustering," *IEICE Trans. Inf. & Syst.* **E83-D**(7), 1574–1582 (July 2000).

[10] Trinh, H.-H. and Jo, K.-H., "Image-based structural analysis of building using line segments and their geometrical vanishing points," in [*SICE-ICASE International Joint Conference*], 566–571 (2006).

[11] Košecká, J. and Zhang, W., "Video compass," in [*European Conference on Computer Vision, LNCS 2350*], 476–491 (2003).

[12] Liebowitz, D., *Camera calibration and reconstruction of geometry*, PhD thesis, University of Oxford (2001).

[13] Tardif, J.-P., "Non-iterative approach for fast and accurate vanishing point detection," in [*IEEE Int. Conf. on Computer Vision*], 1250–1257 (2009).

[14] Rother, C., "A new approach for vanishing point detection in architectural environments," in [*11th British Machine Vision Conference*], 382–391 (2000).

[15] Pflugfelder, R., *Self-calibrating cameras in video surveillance*, PhD thesis, Graz University of Technology (2008).

[16] Torr, P. and Zisserman, A., "Mlesac: A new robust estimator with application to estimating image geometry," *Journal of Computer Vision and Image Understanding* **78**(1), 138–156 (2000).

[17] Schindler, G. and Dellaert, F., "Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments," in [*Conf. on Computer Vision and Pattern Recognition*], 203–209 (2004).

[18] Denis, P., Elder, J. H., and Estrada, F. J., "Efficient edge-based methods for estimating manhattan frames in urban imagery," in [*European Conference on Computer Vision*], **2**, 197–210 (2008).

[19] von Gioi, R. G., Jakubowicz, J., Morel, J.-M., and Randall, G., "Lsd: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(4), 722–732 (2010).

[20] Shufelt, J. A., "Performance evaluation and analysis of vanishing point detection techniques," *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(3), 282–288 (1999).

[21] Collins, R., "Vanishing point calculation as statistical inference on the unit sphere," in [*International Conference on Computer Vision 1990*], 400–403 (1990).