Non-Iterative Approach for Fast and Accurate Vanishing Point Detection

Jean-Philippe Tardif
McGill University, Montréal, QC, Canada
tardifj@cim.mcgill.ca

Abstract

We present an algorithm that quickly and accurately estimates vanishing points in images of man-made environments. Contrary to previously proposed solutions, ours is neither iterative nor relies on voting in the space of vanishing points. Our formulation is based on a recently proposed algorithm for the simultaneous estimation of multiple models called J-Linkage. Our method avoids representing edges on the Gaussian sphere and the computations and error measures are done in the image. We show that a consistency measure between a vanishing point and an edge of the image can be computed in closed-form while being geometrically meaningful. Finally, given a set of estimated vanishing points, we show how this consistency measure can be used to identify the three vanishing points corresponding to the Manhattan directions. We compare our algorithm with other approaches on the York Urban Database and show significant performance improvements.

1. Introduction

Computing vanishing points in an image has applications ranging from camera calibration, pose estimation, single-view reconstruction, autonomous navigation and rectangular structure estimation and matching. Since the early 90's, the usefulness of estimating the vanishing points is well understood in camera calibration and pose estimation [6]. For that reason, most efforts have been put into algorithms for their automatic detection in images.

Recently, vanishing points have been used for detecting dominant rectangular structures located on building facades [14]. Indeed, the structures formed by two pairs of lines on a plane and corresponding to orthogonal vanishing points can be rectified into rectangular patches. This has been shown to improve the matching of these rectangular structures [15] or the matching of the features located onto the building facades [21].

In addition, with the appearance of video cameras on cellphones and personal device assistants, applications for single-view reconstruction and pose estimation are becoming of very high interest, for example, for augmented reality. But since the resources on these devices are limited, algorithms should be efficient and avoid sophisticated optimization techniques.

1.1. Contributions

In this work, we are concerned with the computational aspect of the vanishing point detection. Our first contribution is a non-iterative solution for simultaneously estimating the vanishing points in an image given a set of sparse edges. Our solution is based on an approach recently proposed by Toldo and Fusiello, called J-Linkage [19]. Similarly to RANSAC, it estimates vanishing point hypotheses using minimal sets of edges and computes consensus sets. The J-Linkage algorithm requires a measure of consistency between hypothesized vanishing points and edges that is geometrically meaningful, threats all vanishing points equally and does not require distinguishing finite and infinite vanishing points. Our second contribution is to give such measure in closed-form. We will discuss how this measure is also useful to define a distance between two vanishing points. Our third contribution is an algorithm that uses that notion of distance to verify which of the vanishing points corresponds to Manhattan directions [9, 10].

Our algorithms do not operate on the Gaussian sphere. Specifically, the consistency and distance measures have clear geometric interpretation and are given in image pixels.

1.2. Organization

Related works are described in Section 2 followed by our approach for detecting edges in images in Section 3. Our algorithms are described in detail in Sections 4 to 6. Experimental validation and concluding remarks are presented in Sections 7 and 8, respectively.

1.3. Notation

Matrices are in *sans-serif*, *e.g.* M and vectors in bold, *e.g.* $\mathbf{v}$. The 3×3 matrix $[\mathbf{v}]_\times$ is the skew-symmetric matrix corresponding to $\mathbf{v}$ and relates to the cross product with

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

$[\mathbf{v}]_\times \mathbf{w} = \mathbf{v} \times \mathbf{w}$. The orthogonal distance of a point $\mathbf{p}$ to a line $\mathbf{l}$ is

$$\text{dist}(\mathbf{l}, \mathbf{p}) = \frac{|\mathbf{l}^\top \mathbf{p}|}{\sqrt{l_1^2 + l_2^2}}$$

where $\mathbf{l}$ and $\mathbf{p} \in \mathbb{P}^2$ and $p_3 = 1$, *i.e.* $\mathbf{p}$ is not at infinity. In this work, $\mathbf{p}$ will be a pixel of the image.

2. Related work

2.1. Vanishing point detection

There is a large amount of work on detecting vanishing points in different contexts and for different applications. In this section, we present the ones that are most related to the present work. Techniques for estimating vanishing points can be roughly divided into three categories. The first two require the knowledge of the internal parameters of the camera and the last one operates in an uncalibrated setting.

Given the internal parameters, image lines as well as vanishing points, including the ones at infinity, can be represented by normalized 2d homogeneous coordinates, *i.e.* unit vectors on the Gaussian sphere centered on the optical center of the camera [7, 5]. Since Barnard *et al.* [4], detection was performed on a quantized Gaussian sphere using a Hough transform, until this was shown to lead to spurious vanishing points [18]. Since then, most of the works rely on 3D parallelism or orthogonality of dominant structures in the scene to avoid any false detection [2].

More recent techniques, requiring the knowledge of internal parameters, are based on the "Manhattan assumptions" that the prominent structures of the scene are orthogonal to each other [9, 10]. The algorithms directly estimate the so-called Manhattan directions or equivalently the camera orientation. In this setting, algorithms using either edge gradients [17] or edges [11] have been successfully demonstrated.

Finally, some algorithms assume no knowledge of the internal parameters and their main goal is to estimate possibly non-orthogonal vanishing points. These are especially useful for calibration of the camera using one or more views, but also to recover building facades. Almansa *et al.* use the Helmholtz principle to partition the image plane into *Meaningful vanishing regions* and use Minimum Description Length to reject spurious vanishing points [2]. The Expectation Maximization (EM) approach of Antone and Teller requiring entities represented on Gaussian sphere [3] was extended to the uncalibrated case by Kosecka and Zhang [13]. The EM approach requires an initial estimate of the vanishing points. A typical solution is to cluster edges based on their orientation and to compute vanishing points for each of the clusters. Heuristics based on RANSAC have also been suggested [1]. Finally, Rother proposes heuristics to recover the vanishing points corresponding to Manhattan directions [16]. Unfortunately, it is computationally expensive and requires a criteria to distinguish finite from infinite vanishing points.

Our work relates to the last two categories of approaches. Our main method is to detect possibly non-orthogonal vanishing points. In addition, we propose an algorithm for determining the ones corresponding to Manhattan directions in the case of known internal parameters.

2.2. Multiple model estimation

The problem of simultaneous multiple model estimation within a dataset containing outliers is very important in computer vision. A popular method, called *Randomized Hough Transform* (RHT), is based on random sampling over minimal sets and voting in a discretized parameter space. The models can be recovered by finding peaks in this histogram. The shortcoming of such a Hough transform is that the accuracy is limited and computational efficiency is inversely proportional to the number of bins used to sample the parameter space. Other techniques do not require the partitioning of the parameter space [20]. They proceed similarly to RHT, but the models correspond to modes in the parameters space, which can be identified using the mean shift algorithm [8]. In this sort of technique, the parameterization of the solution space is critical.

Multi-RANSAC was introduced by Zuliani *et al.* as an extension to RANSAC [22]. However it requires knowledge of the number of models in the image. Recently, Toldo and Fusiello presented an algorithm called J-Linkage that overcome the weakness of Multi-RANSAC [19]. Each data point is represented by its "characteristic function of the set of models prefered by that point". The models are built using minimal sample sets similar to RANSAC. Our approach for detecting vanishing points relies on the J-Linkage algorithm. We will describe it in more details in Section 4.1.

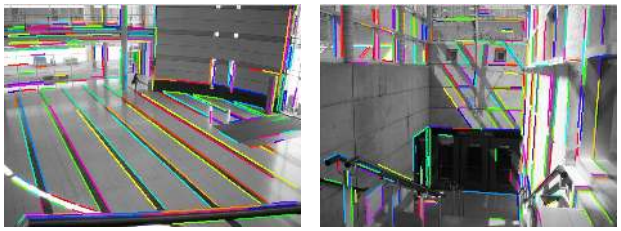
3. Edge detection

The algorithms tested in this work use a sparse set of edges, as this was shown to be more accurate than edge gradients [11]. This section gives a brief description of our algorithm for detecting them in an image. Examples are show in Figure 1.

The first step uses the Canny edge detector followed by non-maximal suppression to obtain a map of one pixel thick edges. The following steps are done to recover straight edges within this edge map. First, we eliminate junctions within the edge map. Then, connected components are found using flood fill. Each component is then divided into straight edges by browsing the list of coordinates and splitting when the standard deviation of fitting a line becomes larger than a one pixel. Finally, we obtain sub-pixel accuracy by fitting a line to each edge.

Table 1. Summary of the notation.

Entities	Definition
$\mathcal{E}_n$	Edge indexed n
$\mathbf{e}_n^1, \mathbf{e}_n^2$	The two end points of $\mathcal{E}_n$, $\in \mathbb{P}^2$
$\bar{\mathbf{e}}_n$	Centroid of the end points, $\in \mathbb{P}^2$
$\mathbf{l}_n$	Implicit line passing by $\mathcal{E}_n$, $\in \mathbb{P}^2$
$\mathcal{S}_m$	Subset of edges of $\mathcal{E}$
$\lvert\mathcal{S}_m\rvert$	Size of the set $\mathcal{S}_m$



(a) Indoor scene (b) Outdoor scene

Figure 1. Example of edge map given by the procedure described in Section 3. Images can be found in the York Urban Database [11].

We denote $\mathcal{E}$ the set of all the edges. The n^{th} edge of the set is given by $\mathcal{E}_n$ and consists of its pixels. In addition, the following entities corresponds to each edge. The end points are given by $\mathbf{e}_n^1$ and $\mathbf{e}_n^2$ and their centroid is given by $\bar{\mathbf{e}}_n$. The implicit representation of the line passing by the $\mathcal{E}_n$ is given by $\mathbf{l}_n$. Finally, a subset of edges is given by $\mathcal{S}$ and the size of a set is given by $|\mathcal{S}|$. This notation is summarized in Table 1.

4. Algorithm

The input of our algorithm is a set of N edges. The output is a set of vanishing points and a classification for each edge: assigned to a vanishing point or marked as an outlier. Our solution relies on the J-Linkage algorithm to perform the classification. Then, the results can be refined using Expectation Maximization or an other iterative algorithm. These algorithms require two functions. The first one, denoted $\mathrm{D}(\mathbf{v}, \mathcal{E}_j)$, provides a measure of the consistency between a vanishing point $\mathbf{v}$ and an edge $\mathcal{E}_j$. The second one is a function that computes a vanishing point using a set of edges $\mathcal{S}$. We denote it $\mathrm{V}(\mathcal{S}, \mathbf{w})$ where $\mathbf{w}$ is a vector of weights. The weights are only useful for the EM algorithm. We postpone the description of the functions to Section 5 as our algorithm is not tailored to any specific formulation.

In the case of the J-Linkage algorithm, the parameters are the *consensus threshold* ϕ and the number of vanishing point hypotheses M. In our work, we used $\phi = 2$ pixel and $M = 500$.

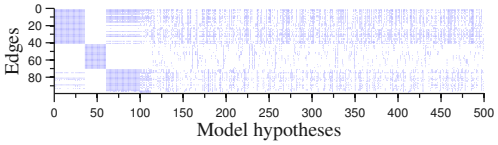


Figure 2. Preference matrix for $N = 100$ edges and $M = 500$ vanishing point hypotheses.

4.1. J-Linkage

In this section, we give a brief overview of the J-Linkage algorithm in the context of our work. More details on J-Linkage in general can be obtained in [19].

The first step is to randomly choose M minimal sample sets of 2 edges $\mathcal{S}_{1\ldots M}$ and to compute a vanishing point hypothesis $\mathbf{v}_m = \mathrm{V}(\mathcal{S}_m, \mathbf{1})$ for each of them ($\mathbf{1}$ is a vector of ones, *i.e.* the weights are equal). The second step consists of constructing the *preference matrix* P, a $N \times M$ Boolean matrix. Each row corresponds to an edge $\mathcal{E}_n$ and each column to a hypothesis $\mathbf{v}_m$. The consensus set of each hypothesis is computed and copied to the m^{th} column of P. An example of matrix P is given in Figure 2. Each row r of P is called the *characteristic function of the preference set* of the edge $\mathcal{E}_n$: the m^{th} entry is 1 if $\mathbf{v}_m$ and $\mathcal{E}_n$ are consistent, *i.e.* when $\mathrm{D}(\mathbf{v}_m, \mathcal{E}_n) \leq \phi$, and 0 otherwise.

The J-Linkage algorithm is based on the assumption that edges corresponding to the same vanishing point tend to have similar preference sets. Indeed, any non-degenerate choice of two edges corresponding to the same vanishing point should yield solutions with similar, if not identical, consensus sets. The algorithm represents the edges by their preference set and clusters them as described below. Note that at this point, the hypothesized vanishing points are completely forgotten by the algorithm. The algorithm defines the preference set of a cluster of edges as the intersection of the preference sets of its members. It also uses the Jaccard distance between two clusters, given by

$$d_j(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

where A and B are the preference sets of each of them. It equals 0 if the sets are identical and 1 if they are disjoint. The algorithm proceeds by placing each edge in its own cluster. At each iteration, the two clusters with minimal Jaccard distance are merged together. The operation is repeated until the distance between all clusters is equal to 1. Typically, between 3 and 7 clusters are obtained. This procedure is generally quite fast if the number of edges and hypotheses if not too high. For example, for around 150 edges and 500 hypotheses, clustering takes around one-tenth of a second on an Intel Core 2 cpu. Once clusters of edges are formed, a vanishing point can be computed for each of them. Outlier edges appear in very small clusters, typically of two edges. If no refinement is performed, we classify small clusters as

(a) J-Linkage (b) Refinement using EM

Figure 3. Detection using our J-Linkage approach and refinement using EM. Blue and purple clusters were merged after three EM iterations.

Algorithm 1 Summary of our algorithm.

INPUT
$\mathcal{E}_{1..N}$: Set of N edges and associated entities (Table 1)
ϕ: consensus threshold (2 pixels)
M: number of vanishing point hypotheses
K: internal parameters (optional)

OUTPUT
$\mathcal{V}$: set of vanishing points
$\mathcal{V}'$: Three most orthogonal amongst them (given K)

ALGORITHM
Build the preference matrix $\mathsf{P} \in \{0,1\}^{N \times M}$:
For: $m \leftarrow 1$ to M //columns of P
Randomly select a subset of 2 edges $\mathcal{S}$ from $\mathcal{E}$
$\mathbf{v}_m \leftarrow \mathrm{V}(\mathcal{S}, \mathbf{1})$, //weights are set to 1
For: $n \leftarrow 1$ to N //rows of P
$\mathsf{P}_{n,m} \leftarrow \mathrm{D}(\mathbf{v}_m, \mathcal{E}_n) \leq \phi$
Classify edges using J-Linkage clustering on P (§4.1)
$\mathcal{V} \leftarrow$ Re-compute vanishing points for each cluster (§5.2)
$\mathcal{V} \leftarrow$ Refine $\mathcal{V}$ using EM (Optional) (§4.2)
If K is known: (Optional)
$\mathcal{V}' \leftarrow$ find 3 most orthogonal vanishing points (§6)

outliers. Finally, the vanishing points and classification can be refined using an iterative approach, such as EM.

4.2. Expectation-Maximization

The most frequent mistake made by the J-Linkage algorithm is to divide a cluster of edges of a vanishing point into two groups. However, this is easily corrected by refining the solution using EM [3, 13]. An example is shown in Figure 3. We refer the reader to these references for a description of the algorithm. In the original work, the functions D and V are defined on the Gaussian sphere, but using our own functions instead provides significant improvements (details in Section 7).

The summary of our algorithm is given in Algorithm 1. At this point, the only pieces missing are the definition of the consistency measure D and function V for estimating vanishing points.

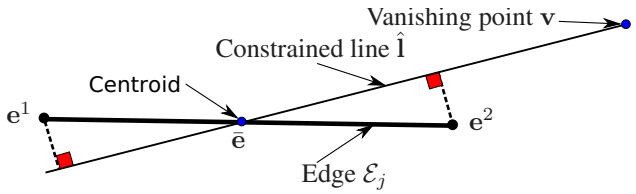


Figure 4. The proposed approximation for estimating $\hat{\mathbf{l}}$ used in the computation of our consistency measure given in (1).

5. Vanishing points estimation

In this section, we give the two functions required by the J-Linkage and the EM algorithms:

- $\mathrm{D}(\mathbf{v}, \mathcal{E}_j)$: measure of consistency between a vanishing point and an edge;
- $\mathrm{V}(\mathcal{S}, \mathbf{w})$: function performing a weighted estimation of a vanishing point using a set of edges $\mathcal{S}$, where $\mathbf{w}$ is a vector of the weights.

Since these operations are performed a large number of times, they must be efficient. We will first describe D and then V in Sections 5.1 and 5.2, respectively. We proceed in this order because a result of the first section will be used in the second one.

5.1. Consistency measure

In this section we consider only one edge $\mathcal{E}_j$ ($j \in \{1...N\}$) and one vanishing point $\mathbf{v}$. We will describe the function $\mathrm{D}(\mathbf{v}, \mathcal{E}_j)$ and discuss its differences with the one of [3, 13] which operates on the Gaussian sphere.

When edges and vanishing points are represented on the Gaussian sphere, a typical error is given by $(\mathbf{l}_j^\top \mathbf{v})^2$ or $\sin^{-1}(\mathbf{l}_j^\top \mathbf{v})$ where $\mathbf{l}_j$ is the implicit line representation of $\mathcal{E}_j$ (see Table 1). This formulation has two inconveniences. Firstly, this quantity is not geometrically meaningful in the uncalibrated case. Secondly, the length and position of the edge is not taken into account so that any two co-linear edges will have the same error even if their distances to the vanishing point or their lengths are different.

In computer vision, consistency measures in the image space are usually prefered because that is where the uncertainty originates [12]. In this context, one example would be the average orthogonal distance of a line $\hat{\mathbf{l}}$ to the end points of $\mathcal{E}_j$, such that $\hat{\mathbf{l}}$ intersects $\mathbf{v}$ and minimizes that distance. Note that this distance is given in pixels, even for a vanishing point located at infinity. The computation of $\hat{\mathbf{l}}$ is non-linear and requires iterative refinement of an initial estimate.

For the J-Linkage algorithm, the consistency measure must be computed a few hundred thousand times. We seek a closed-form approximation for $\hat{\mathbf{l}}$. Instead of finding the line minimizing the average distance to end points, we find the one minimizing the maximal distance. It is given in closed-form by

$$\mathrm{D}(\mathbf{v}, \mathcal{E}_j) = \mathrm{dist}(\mathbf{e}_j^1, \hat{\mathbf{l}}), \text{ where } \hat{\mathbf{l}} = [\bar{\mathbf{e}}_j]_\times \mathbf{v}. \quad (1)$$

This is illustrated in Figure 4. The proof is given in Appendix A.

5.2. Vanishing point estimation

In this section, we describe the function $\mathrm{V}(\mathcal{S}, \mathbf{w})$ that computes a vanishing point using a set of edges. In this work, computing vanishing points occurs in two contexts:

- for the construction of preference matrix using pairs of edges;
- after J-Linkage classification and in EM with usually more than two edges.

The case of two edges $\mathcal{E}_1$ and $\mathcal{E}_2$ is straightforward: the vanishing point is given by $\hat{\mathbf{v}} = \mathbf{l}_1 \times \mathbf{l}_2$. Thus, every step of the construction of the preference matrix involves solutions that can be obtained in closed-form. In the rest of the section, we give our solution in the case of more than two edges.

On the Gaussian sphere, V is typically defined as

$$\mathrm{V_{GS}}(\mathcal{S}, \mathbf{w}) = \arg\min_{\mathbf{v}} \sum_{\mathcal{E}_j \in \mathcal{S}} (w_j \mathbf{l}_j^\top \mathbf{v})^2, \text{ s. t. } \|\mathbf{v}\| = 1$$

which can be computed in closed-form, but which optimizes an algebraic error [3, 13].

In this work, the measurements are kept in the image space. As a result, one could refine the solution provided by $\mathrm{V_{GS}}$ using a Maximal Likelihood Estimator (MLE). One would seek to determine a set of lines passing by a single point and such that the orthogonal distance to the end points is minimized [12]. The function can be written as

$$\mathrm{V_{MLE}}(\mathcal{S}, \mathbf{w}) = \arg \min_{\mathbf{v}, \hat{\mathbf{l}}_{1,,,n}} \sum_{\mathcal{E}_j \in \mathcal{S}} \sum_{i=1}^{2} w_j^2 \mathrm{dist}^2(\hat{\mathbf{l}}_j, \mathbf{e}_j^i)$$

such that $\|\mathbf{v}\| = 1$ and $\forall j : \hat{\mathbf{l}}_j^\top \mathbf{v} = 0, \|\mathbf{l}_j\| = 1$. This is a function which would have to be minimized iteratively. Proper parameterization of the variables avoids the bilinear constraints and the unity constraints. Overall, n variables are necessary to represent the $\hat{\mathbf{l}}_j$, two for $\mathbf{v}$ (in spherical coordinate) and the number of residuals is $2n$.

We propose a replacement for $\mathrm{V_{MLE}}$ which is also non-linear. However, it is more efficient to compute and gives virtually the same performance in practice (see Section 7). We find

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} \sum_{\mathcal{E}_j \in \mathcal{S}} w_j^2 \max\left(\mathrm{dist}^2(\hat{\mathbf{l}}_j, \mathbf{e}_j^1), \mathrm{dist}^2(\hat{\mathbf{l}}_j, \mathbf{e}_j^2)\right)$$

such that $\forall j : \hat{\mathbf{l}}_j^\top \mathbf{v} = 0, \forall j : \hat{\mathbf{l}}_j^\top \mathbf{e}_j = 0, \|\mathbf{v}\| = 1$. The difference with $\mathrm{V_{MLE}}$ is that the summation of the distance to the end points as been replaced by the maximal distance to end points, per edge. Using the result of Section 5.1, we replace $\hat{\mathbf{l}}_j$ by $[\bar{\mathbf{e}}_j]_\times \mathbf{v}$, and also get rid of the $\max$ function. As a result, we can rewrite this problem without relying on extra parameters $\hat{\mathbf{l}}_j$:

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} \sum_{\mathcal{E}_j \in \mathcal{S}} w_j^2 \mathrm{dist}^2([\bar{\mathbf{e}}_j]_\times \mathbf{v}, \mathbf{e}_j^1) \quad (2)$$

where $\mathbf{v}$ is represented in spherical coordinates. Compared to $\mathrm{V_{MLE}}$, the number of variables is reduced from $n + 2$ to two and the number of residuals is divided by two. A dense solver rather than a sparse solver can thus be used.

Finally, our definition for V is given by

$$\mathrm{V}(\mathcal{S}, \mathbf{w}) = \begin{cases} \mathbf{l}_1 \times \mathbf{l}_2 & \text{if } \mathcal{S} \text{ contains 2 edges} \\ \hat{\mathbf{v}} \text{ given by (2)} & \text{otherwise} \end{cases} \quad (3)$$

In Section 7, we compare V with $\mathrm{V_{MLE}}$ and $\mathrm{V_{GS}}$ and confirm that iterative refinement of the vanishing points is useful in practice. On the other hand, we found that using a MLE has no significant advantage over this formulation. In addition, if both optimization are done using a dense solver, $\mathrm{V_{MLE}}$ is around 14 times slower than V.

6. Finding Manhattan directions

In this section, we assume that a set of vanishing points has been obtained using our method. Clusters of less than three edges are discarded so that each edge is assigned to one of the vanishing points. Given the knowledge of the internal parameters, our goal is to select the three vanishing points corresponding to the Manhattan directions. This is useful for computing the orientation of the camera or to give an initial solution to global methods relying on the Manhattan assumption. To this end, we will use our consistency criterion between a vanishing point and an edge to define a distance measure between two vanishing points given a set of edges.

Denote K the 3×3 matrix of the internal parameters. It is well known that two orthogonal vanishing points $\mathbf{v}$ and $\mathbf{v}_\perp$ satisfy

$$\mathbf{v}^\top \omega \mathbf{v}_\perp = 0 \quad (4)$$

where ω is the Image of Absolute Conic (IAC) given by $\mathsf{K}^{-\top}\mathsf{K}^{-1}$ [12]. Typically, this constraint is used for calibrating a camera by estimating ω given previously recovered vanishing points. Here, ω is given and we seek the three vanishing points that best satisfy (4) two by two. We will consider every possible triplet. For each of them, given by say $\mathbf{v}_{1,2,3}$, we need a measure of their orthogonality. The naive measure is given by the sum of squares of the constraint: $(\mathbf{v}_1^\top \omega \mathbf{v}_2)^2 + (\mathbf{v}_1^\top \omega \mathbf{v}_3)^2 + (\mathbf{v}_1^\top \omega \mathbf{v}_3)^2$. Instead, one should seek vanishing points $\mathbf{v}_i'$ as close as possible to the original $\mathbf{v}_i$'s, while exactly satisfying (4). We won't be solving this problem, but consider a similar criteria that avoids any optimization. The approach requires a measure of the distance between two vanishing points.

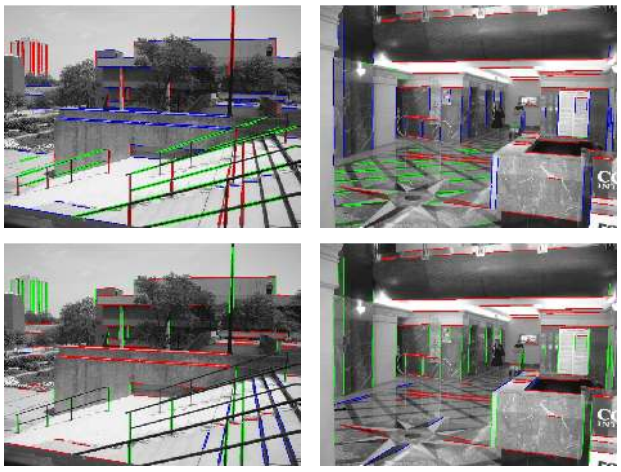


Figure 5. Determining Manhattan directions using the approach of Section 6. **Top)** Three largest clusters of edges corresponding to vanishing points. **Bottom)** Three sets of edges corresponding to Manhattan directions.

Assume that we want to compute the distance between $\mathbf{v}_i$ and $\mathbf{v}'_i$. Using the set of edges $\mathcal{S}_i$ that is known to be consistent with $\mathbf{v}_i$, we define it as

$$\mathrm{C}(\mathbf{v}'_i, \mathcal{S}_i) = \frac{1}{|\mathcal{S}_i|} \sum_{\mathcal{E}_j \in \mathcal{S}_i} \mathrm{D}(\mathbf{v}'_i, \mathcal{E}_j)$$

In other words, the distance between $\mathbf{v}'_i$ and $\mathbf{v}_i$ is indirectly given by the average consistency between the edges and $\mathbf{v}'_i$. This distance is given in pixels.

We come back to our triplet of vanishing points. According to (4), a vanishing point $\hat{\mathbf{v}}'_3$ that is perfectly orthogonal to the first ones is given by

$$\hat{\mathbf{v}}'_3 = \mathrm{Null}\left(\omega \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{pmatrix}\right), \tag{5}$$

where $\mathrm{Null}(\mathsf{M})$ returns the left nullspace of M. To measure how close $\hat{\mathbf{v}}_3$ is to $\hat{\mathbf{v}}'_3$, all we need to compute is $\mathrm{C}(\hat{\mathbf{v}}'_3, \mathcal{S}_3)$. Repeating this operation for $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$, we define the orthogonality measure as

$$\max_{i=1}^{3} \mathrm{C}(\hat{\mathbf{v}}'_i, \mathcal{S}_i) \tag{6}$$

where all $\hat{\mathbf{v}}'_i$'s are computed similarly to (5).

To summarize, our algorithm consists of evaluating (6) for every triplet of vanishing points. The triplet yielding the minimal value is selected. Typical results of this procedure are shown in Figure 5.

7. Experiments

We implemented our algorithms in MATLAB, with most of the computation being done in MEX files (compiled C code)[1]. We tested our algorithms using the York Urban Database provided by Denis *et al.* [11]. It consists of 102 indoor or outdoor images of man-made environments. For each image, edges were manually selected so that they correspond to one of the three Manhattan directions. Classification of the edges and estimation of the vanishing points was done using the approach of Collins and Weiss which operates on the Gaussian sphere [7]. Each image contains either two or three vanishing points. Contrary to the work of Denis *et al.* , ours does not involve training using a subset of the images, so we could use the whole database.

We tested the following algorithms:

- **JL**: J-Linkage clustering;
- **JL+EM**: the above followed by EM;
- **calib JL**: J-Linkage clustering and detection of Manhattan directions;
- **calib JL+EM**: the above followed by EM;
- **GS EM**: original EM approach operating on the Gaussian sphere [3, 13], using the internal parameters as normalization transformation;
- **EM**: a modified EM approach operating in image space.

The results using the ground truth are referred to **GT**. We use our own implementation of the EM algorithm. Both variants were initialized by grouping edges according to their orientation as proposed by Kosecka and Zhang. Note that we don't compare our work to the one of Denis *et al.* or any other approach that globally estimates the Manhattan directions. Indeed, our main goal is to detect all (possibly non-orthogonal) vanishing points, without knowing the camera internal parameters. For the J-Linkage algorithms, we used $\phi = 2$ pixels and $M = 500$ in all our tests.

We automatically detected straight edges in the 102 images of the database as described in Section 3. Results given by our approach for four of the images of database are shown in Figure 6. Two error measures were considered.

The first one is the consistency of the ground truth edges with the estimated vanishing points. Each image contains three sets of ground truth edges, corresponding to the three orthogonal vanishing points: $\mathcal{S}_{i,1}$, $\mathcal{S}_{i,2}$ and $\mathcal{S}_{i,3}$ which we ordered so that $\forall_{i=1\ldots102} : |\mathcal{S}_{i,1}| \geq |\mathcal{S}_{i,2}| \geq |\mathcal{S}_{i,3}|$. Thus, the sets $\mathcal{S}_{i,1}$ very often consist of vertical edges, the easiest to detect. On the other hand, the smallest sets $\mathcal{S}_{i,3}$ are the most problematic. They sometimes contain just a few edges. Furthermore, the images sometimes contain larger sets of edges corresponding to non Manhattan directions (examples in Figure 5). We tested the consistency of the estimated vanishing points on each group of edges. The results are given in Figure 7. The advantages of our methods are clearly observed on the estimation of the third and most difficult vanishing points. Our method for detecting the Manhattan directions also improves the accuracy of the

[1] Available at `http://www-etud.iro.umontreal.ca/~tardifj/fichiers/VPdetection.tar.gz`

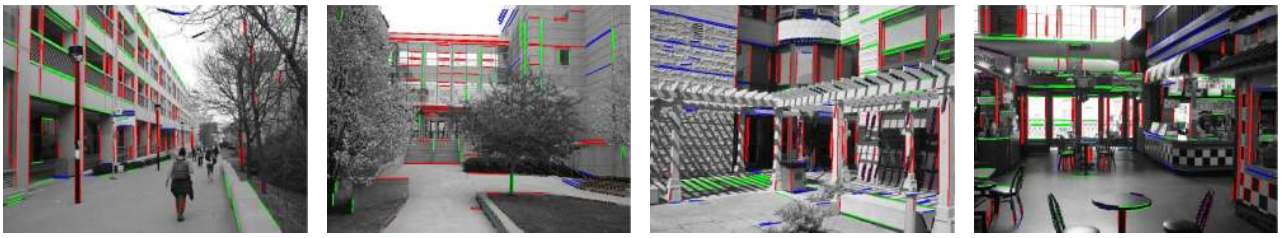
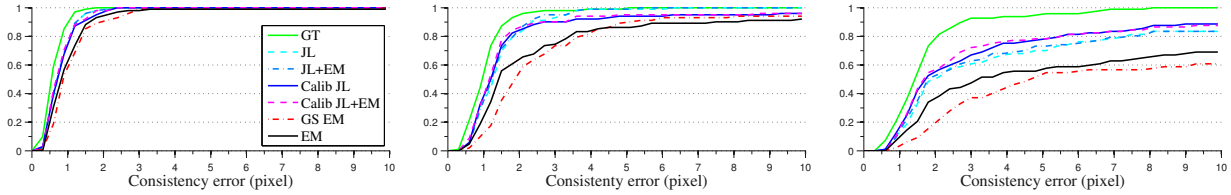
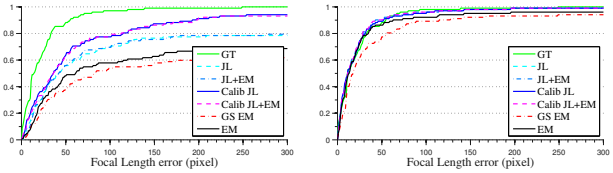


Figure 6. Results given by our algorithm on four images of the York Urban Database [11].



(a) $\mathcal{S}_{i,1}$ (b) $\mathcal{S}_{i,2}$ (c) $\mathcal{S}_{i,3}$

Figure 7. Cumulative histograms of the error for three groups of ground truth edges (See text for details). The vanishing points were estimated using edges automatically detected.



(a) Edges automatically detected (b) Ground truth edges

Figure 8. Cumulative histograms of the error on the focal length estimation.

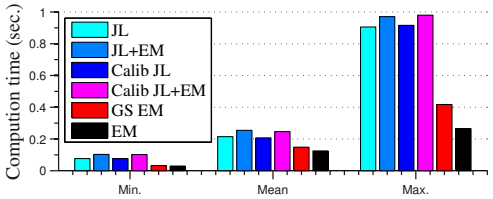


Figure 9. Computation time of the tested algorithms.

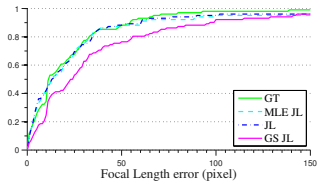


Figure 10. Cumulative histogram of the error of the estimated focal length for vanishing points estimated using V_{MLE}, V and V_{GS}, as discussed in 5.2.

vanishing points. Both EM algorithms are quite sensitive to the initial solution, although our variant seem to perform better.

The second error measure is the accuracy of the estimated focal length using the vanishing points. We compute the internal parameters using the linear solution based on (4) [12] and don't rely on bundle adjustment. The results are given in Figure 8(a). Finally, the algorithms were also tested on the ground truth edges. Results for the focal length estimation are in Figure 8(b).

Minimal, average and maximal computation time are reported in Figure 9 with average computation time of roughly two-tenth of a second. When initialized using J-Linkage clustering, the EM algorithm converged after just a few iterations.

We compare the approaches for estimating vanishing points using sets of edges discussed in Section 5.2: our approach V, a MLE estimator V_{MLE} and the algebraic on the Gaussian sphere V_{GS}. We used the ground truth edges of the York Urban Database and applied our algorithm with no EM refinement. Final vanishing points were obtained for each cluster using the three above functions and were used for estimating the internal parameters of the camera. Results are reported in Figure 10 and show a clear advantage of the two iterative approaches, but no clear winner amongst them. Consistency of the ground truth edges with the vanishing points estimated using our algorithms were always below two pixels and was sometimes even below than the consistency of the ground truth vanishing points.

8. Conclusion

The detection of vanishing points in images is a typical problem of robust simultaneous multiple model estimation. The J-linkage algorithm proposes a new perspective for solving this multiple model estimation problem. It

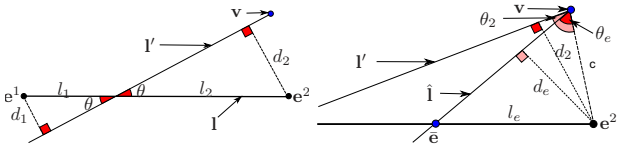


Figure 11. The line passing by the vanishing point $\mathbf{v}$ and minimizing the maximal distance to the end points of the edge intersect the edge at then centroid of the edge (see text for details).

avoids clustering on the Gaussian sphere and does not require a guess of the number of vanishing points. Based on this algorithm, our approach is fast and accurate. In addition, we gave an approach for determining which of the vanishing points correspond to Manhattan directions.

We plan to extend our approach to estimate vanishing points in images of fish-eyes and catadioptric cameras.

A. Proof for Section 5.1

The proof involves showing that the maximal distance to the end points is minimized when it is equal for both, which can only happen when $\hat{\mathbf{l}}$ intersects the centroid $\bar{\mathbf{e}}_j$. We prove this in two steps. Consider the top of Figure 11, where we drop the subscript j. The orthogonal distances of the end points to $\mathbf{l}'$ are given by d_1 and d_2, respectively. Since both triangles are identical up to scale factor, d_1 and d_2 are proportional to l_1 and l_2. Thus, the maximal distance occurs at the end point whose distance to intersection point of $\mathbf{l}'$ with $\mathbf{l}$ is the largest. Now consider the bottom of Figure 11 where in addition to $\mathbf{l}'$, we draw $\hat{\mathbf{l}}$. The distance between $\mathbf{e}^2$ and $\bar{\mathbf{e}}$ is given by $\mathbf{l}_e$. If l_2 is larger than l_e, d_2 is larger than d_e. Indeed, these are respectively given by $c \sin\theta_2$ and $c \sin\theta_e$. Since both θ's are between 0 and $\pi/2$, $\sin\theta$ is a monotonic function. Thus, $d_2 > d_e$.

Acknowledgments To Mike Langer for useful comments on the paper and to Patrick Denis for providing the York Database.

References

[1] D. G. Aguilera, J. G. Lahoz, and J. F. Codes. A new method for vanishing points detection in 3d reconstruction from a single view. *Proceedings of the ISPRS Commission V*, 2005.

[2] A. Almansa, A. Desolneux, and S. Vamech. Vanishing point detection without any a priori information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):502–507, 2003.

[3] M. E. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

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

[5] B. Brillault-O'Mahony. New method for vanishing point detection. *CVGIP: Image Understanding*, 54(2):289–300, 1991.

[6] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, 1990.

[7] R. T. Collins and R. S. Weiss. Vanishing point calculation as a statistical inference on the unitsphere. In *International Conference on Computer Vision*, pages 400–403, 1990.

[8] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[9] J. M. Coughlan and A. L. Yuille. Manhattan world: compass direction from a single image by bayesian inference. In *International Conference on Computer Vision*, 1999.

[10] J. M. Coughlan and A. L. Yuille. Manhattan world: Orientation and outlier detection by bayesian inference. *Neural Computation*, 15(5):1063–1088, 2003.

[11] P. Denis, J. H. Elder, and F. J. Estrada. Efficient Edge-Based methods for estimating manhattan frames in urban imagery. In *European Conference on Computer Vision*, pages 197–210, 2008.

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

[13] J. Košecká and W. Zhang. Video compass. *European Conference on Computer Vision*, pages 476–490, 2002.

[14] J. Košecká and W. Zhang. Extraction, matching, and pose recovery based on dominant rectangular structures. *Computer Vision and Image Understanding*, 100(3):274–293, 2005.

[15] B. Micusik, H. Wildenauer, and J. Kosecka. Detection and matching of rectilinear structures. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–7, 2008.

[16] C. Rother. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9-10):647–655, 2002.

[17] G. Schindler and F. Dellaert. Atlanta world: an expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[18] J. A. Shufelt. Performance evaluation and analysis of vanishing point detectiontechniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):282–288, 1999.

[19] R. Toldo and A. Fusiello. Robust multiple structures estimation with J-Linkage. In *European Conference on Computer Vision*, pages 537–547, 2008.

[20] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.

[21] C. Wu, F. Fraundorfer, J. M. Frahm, and M. Pollefeys. 3D model search and pose estimation from single images using VIP features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[22] M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multiransac algorithm and its application to detect planar homographies. In *International Conference on Image Processing*, 2005.