A Global Approach for the Detection of Vanishing Points and Mutually Orthogonal Vanishing Directions

Michel Antunes and João P. Barreto
Institute of Systems and Robotics
University of Coimbra, 3030 Coimbra, Portugal
{michel,jpbar}@isr.uc.pt

Abstract

This article presents a new global approach for detecting vanishing points and groups of mutually orthogonal vanishing directions using lines detected in images of man-made environments. These two multi-model fitting problems are respectively cast as Uncapacited Facility Location *(UFL) and* Hierarchical Facility Location *(HFL) instances that are efficiently solved using a message passing inference algorithm. We also propose new functions for measuring the consistency between an edge and a putative vanishing point, and for computing the vanishing point defined by a subset of edges. Extensive experiments in both synthetic and real images show that our algorithms outperform the state-of-the-art methods while keeping computation tractable. In addition, we show for the first time results in simultaneously detecting multiple Manhattan-world configurations that can either share one vanishing direction (Atlanta world) or be completely independent.*

1. Introduction

A set of parallel lines in the scene project into a pencil of lines intersecting in the so-called vanishing point (VP). The VP is the image of the point at infinity where the parallel lines intersect and encodes their common direction. In the case of man-made environments, the sets of parallel lines are usually orthogonal to each other, and the detection of the corresponding VPs enables to accomplish different tasks. Applications include intrinsic camera calibration [9], estimation of the camera rotation with respect to the scene [1, 12], 3D reconstruction [20], and recognition [2].

The automatic detection of VPs using sparse edges [6] or edge gradients [19] is a problem of multi-model fitting where the models are line pencils. It is in general a "chicken-and-egg" problem because we neither know the number and parameters of the models (the vanishing points), nor the edges that belong to each model (the mem-

<div align="center">(a) Manhattan World   (b) Multiple orthogonal triplets</div>

Figure 1. Two images of man-made environments.

bership). The first attempt of automatic detection of VPs goes back to the 80's when Barnard proposed to use the Hough transform on a quantized Gaussian sphere [3]. It was latter shown that the accuracy of such an approach highly depends on the choice of the voting bins, and that the detection results are often spurious. In [1], Antone and Teller suggests to carry the VP detection using Expectation-Maximization (EM) with the E-step computing the probability distributions of the input lines passing through the hypothesized VPs, and the M-step refining the VP models by maximizing the likelihood of the observed data. Latter on, the EM framework was successfully extended to the case of uncalibrated cameras [12, 19]. However, the process is iterative and requires a good initial estimate that is typically accomplished by clustering the edges assuming a world dominated by either 3 (Manhattan) [1, 12] or 5 (Atlanta) [19] mutually orthogonal vanishing directions (VDs). In [18], Rother combines RANSAC search with several heuristics for recovering the VPs of Manhattan directions, but the final algorithm is computationally expensive and requires distinguishing between finite and infinite VPs. Finally, Tardif has recently proposed a new image-based consistency metric to be used with J-Linkage for clustering the edges into pencils of lines [21]. The algorithm is fast, robust, and accurate, being one of the best performing VP detectors that are currently available.

The works above perform the separate estimation of the VPs in the image, which, in many cases, is followed by

grouping the result into directions that are mutually orthogonal [21]. A different approach is to consider *a priori* that the scene follows the Manhattan world assumption and determine the rotation that is aligned with the 3 dominant VDs. In this case, the VP detection is no longer a problem of multiple model fitting, but the problem of fitting a single triplet of mutually orthogonal VPs in the presence of edges that are outliers. Such fitting can be accomplished through EM [6], by using minimal solutions as hypothesis generator in a RANSAC paradigm [17], or by applying Branch-and-Bound to solve a consensus set maximization that assures global optimality [4]. The disadvantages of this type of approach are that additional VDs that might exist are passed undetected, and the methods cannot handle images with more than one set of Manhattan-world directions for which multi-model fitting is again required (see Fig. 1(b)).

1.1. Contributions

This article addresses the problem of detecting VPs in uncalibrated images using either edges or line segments, and (given the intrinsic calibration) the problem of grouping the detection results into sets of mutually orthogonal VDs. The contributions with respect to the state-of-the-art are:

1. *A global solution for VP detection:* It has been recently argued that methods that greedily search for models with most inliers (within a threshold) while ignoring the overall classification of data are in general a flawed approach to multi-model fitting, and that formulating the fitting as an optimal labeling problem with a global energy function is usually preferable [11]. Our article goes towards this direction and formulates for the first time the detection of VPs as an *Uncapacited Facility Location* (UFL) problem [14] that can be solved using a local message passing approach [14, 15]. Experiments show that such a global approach is very competitive with the state-of-the-art algorithm [21] that relies in J-Linkage and EM. Very recently, Tretyak et al. [22] presented a method that integrates the estimation of line segments, lines, VPs, the horizon and zenith in a single energy optimization framework. Besides of being complex and time consuming, this formulation already assumes that a discrete number of accurate VPs has been obtained.

2. *New functions $D(\mathrm{e}, \mathbf{v})$ and $W(\mathcal{S})$:* Independently of the multi-model fitting approach, the detection of VPs always requires a consistency function $D(\mathrm{e}, \mathbf{v})$, which measures the likelihood of the edge e being in a line l passing through the putative VP $\mathbf{v}$, and a function $W(\mathcal{S})$ that computes the most likely VP given a set of edges $\mathcal{S}$. Many prior works formulate the consistency function in the Gaussian sphere after back-projecting the edges and VPs [3, 1, 12, 4, 19]. However, and as argued in [21], measurements in the image space are usu-

ally preferred because the non-linear mapping into the sphere changes the statistics of noise ultimately leading to biased estimation results [18]. Therefore, Tardif proposes to formulate $D(\mathrm{e}, \mathbf{v})$ and $W(\mathcal{S})$ using the geometric distance measured in the image [21]. However, and in order to avoid iterative non-linear minimization, he works with the maximum orthogonal distances to the edge endpoints rather than considering the mean distance to all points. We show that this minimization problem can be solved in closed-form and propose new functions $D(\mathrm{e}, \mathbf{v})$ and $W(\mathcal{S})$ that improve the overall fitting results while keeping computation tractable.

3. *A global solution for detecting multiple sets of mutually orthogonal VDs:* The existing methods for detecting mutually orthogonal VDs assume that the image depicts a single Manhattan-world configuration [17, 4]. In practice these algorithms often become unstable and/or inaccurate whenever there is no image evidence for one of the Manhattan directions, and cannot cope with frames like the one of Fig. 1(b) showing more than one group of mutually orthogonal directions. We propose for the first time an algorithm that, given an initial set of VPs, it is able to detect multiple Manhattan-world configurations that can either be complete or incomplete (two directions), and be independent or have one direction in common (Atlanta-world). The multi-model fitting is solved in a global manner by casting the problem as an *Hierarchical Facility Location* (HFL) problem [8].

1.2. Notation and Organization

We denote scalars in italic, e.g. s, vectors in bold characters, e.g. $\mathbf{p}$, and matrices in sans serif font, e.g. M. Geometric entities are represented in homogeneous coordinates, e.g. $\mathbf{q} = (q_1 \ q_2 \ 1)^\mathsf{T}$, and the skew-symmetric matrix is denoted by $[\ ]_\times$. Finally, the orthogonal image distance between a point $\mathbf{q}$ and a line $\mathbf{l}$ is given by

$$d_\perp(\mathbf{q}, \mathbf{l}) = \frac{|\mathbf{q}^\mathsf{T}\mathbf{l}|}{\sqrt{\mathbf{l}^\mathsf{T}\mathsf{l}_s\mathbf{l}}} \ \text{ with } \ \mathsf{l}_s = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The article is organized as follows: Sec. 2 presents the FL and HFL problems, and shows how they can be solved using a message passing approach; Sec. 3 presents the algorithm for the detection of VPs; Sec. 4 concerns the detection of multiple orthogonal VP triplets; and finally, the experimental results are presented in Sec. 5 and 6.

2. The Facility Location Problems

This section briefly introduces the problems of *Uncapacited Facility Location* (UFL) and *Hierarchical Facility*

Location (HFL) that play a key role in the global approaches for detecting VPs and clustering mutually orthogonal VDs. To the best of our knowledge these frameworks were seldom used in the context of computer vision. In [15] and [16] the problems of subspace segmentation and two-view motion segmentation are formulated as UFL problems, respectively, while in [23] Xiao et al. formulated the simultaneous segmentation of registered 2D images and 3D points as a hierarchical exemplar-based clustering instance [7], a problem that is closely related to UFL, and that was solved using a greedy bottom-up affinity propagation approach [23]. The UFL is a classical NP-hard problem that can be solved by applying an optimization method based on the max-sum algorithm [14, 15]. This method is more robust than the greedy solver for UFL discussed by Delong et al. in [5], and has been recently extended for also handling the HFL problem [8]. Since this extension has never been applied in computer vision, we briefly outline the solver that relies in local message passing.

2.1. Uncapacited Facility Location (UFL) problem

Suppose that you need to open a set of facilities $\mathbf{v}_j^0$ to serve N customers $\mathbf{e}_i \in \mathcal{E}$ whose locations are known. Given a set $\mathcal{V}_0$ comprising M_0 possible facility locations, the cost $c_{ij}^0 : \mathcal{E} \times \mathcal{V}_0 \to \mathbb{R}$ for assigning the facility $\mathbf{v}_j^0$ to the customer $\mathbf{e}_i$, and the cost $v_j^0 : \mathcal{V}_0 \to \mathbb{R}$ for opening the particular facility $\mathbf{v}_j^0$, the goal of the UFL problem is to select a subset of $\mathcal{V}_0$ such that each customer is served by one facility, and the sum of the customer-facility costs plus the sum of facility opening costs is minimized. This leads to an integer programming problem that is usually formulated using unary indicator variables y_j^0 and binary indicator variables x_{ij}^0, and whose objective is to find the vector $\mathbf{x}^0 = \{x_{11}^0 ... x_{ij}^0 ... x_{NM_0}^0\}$ such that :

$$\min_{\mathbf{x}^0} \quad \sum_{i=1}^{N} \sum_{j=1}^{M_0} c_{ij}^0 x_{ij}^0 + \sum_{j=1}^{M_0} v_j^0 y_j^0$$

$$\text{subject to} \quad \begin{cases} x_{ij}^0, \, y_j^0 \in \{0,1\}, \, \forall_{i,j} \\ \sum_{j=1}^{M_0} x_{ij}^0 = 1, \, \forall_i \\ y_j^0 \geq x_{ij}^0, \, \forall_{i,j} \end{cases} \quad (1)$$

The equality in the second constraint ensures that each customer is assigned to exactly one facility, while inequality of the last constraint guarantees that each customer is only served by facilities that were opened.

2.2. Hierarchical Facility Location (HFL) problem

Let's now imagine that the facilities $\mathbf{v}_j^0$ need to be stocked by storage facilities $\mathbf{v}_k^1$, which in turn need to be stocked by larger warehouses $\mathbf{v}_m^2$, and so forth till the graph of the UFL problem is extended by L additional levels
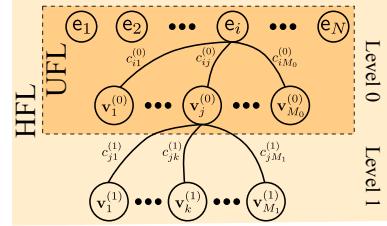

Figure 2. A HFL problem with two layers is shown. The first layer $l = 0$ corresponds to the UFL problem, where the customers $\mathbf{e}_i$ need to be served by the facilities $\mathbf{v}_j^0$. The second layer $l = 1$ contains the storage facilities $\mathbf{v}_k^1$ that are the suppliers of $\mathbf{v}_j^0$.

(Fig. 2). Given a set of potential M_l facility locations $\mathcal{V}_l$ at layer l, the cost $v_j^l : \mathcal{V}_l \to \mathbb{R}$ for opening the facility $\mathbf{v}_j^l$, and the cost $c_{jk}^l : \mathcal{V}_{l-1} \times \mathcal{V}_l \to \mathbb{R}$ for the facility $\mathbf{v}_k^l$ supplying the facility $\mathbf{v}_j^{l-1}$, the goal of HFL is to find the vector $\mathbf{x} = \{\mathbf{x}^0 ... \mathbf{x}^l ... \mathbf{x}^L\}$ that minimizes the following function:

$$\min_{\mathbf{x}} \quad \sum_{i=1}^{N} \sum_{j=1}^{M_0} c_{ij}^0 x_{ij}^0 + \sum_{l=1}^{L} \sum_{j=1}^{M_{l-1}} \sum_{k=1}^{M_l} c_{jk}^l x_{jk}^l + \sum_{l=0}^{L} \sum_{j=1}^{M_l} v_j^l y_j^l$$

$$\text{s.t.:} \quad \begin{cases} x_{ij}^l, y_j^l \in \{0,1\} \\ \sum_{j=1}^{M_0} x_{ij}^0 = 1, \forall_i \quad \bigwedge \quad \sum_{k=1}^{M_l} x_{jk}^l = y_j^{l-1}, \forall_{j, \, l>0} \\ y_j^0 \geq x_{ij}^0, \forall_{i,j} \quad \bigwedge \quad y_k^l \geq x_{jk}^l, \, \forall_{j, \, k, \, l>0} \end{cases}$$

The additional restrictions compared to Eq. 1 are that (i) if a facility $\mathbf{v}_j^{l-1}$ is closed in layer $l-1$, then $\mathbf{v}_j^{l-1}$ will not need to be stocked by a storage facility $\mathbf{v}_j^l$, whereas (ii) if a facility $\mathbf{v}_j^{l-1}$ is open, then it must be stocked by a facility in the next layer l. Note that in the case of a single layer, the HFL problem reduces to the UFL problem (see Fig. 2).

2.3. Solving the UFL and HFL problems using the max-sum algorithm

In [15, 14] Lazic et al show how to solve the UFL problem using a local message passing approach. They formulate the UFL problem as a maximum-a-posteriori (MAP) problem and represent it using a factor graph [13]. The MAP estimates for x_{ij}^0 can then be inferred using the max-sum algorithm [13], which is a log-domain equivalent of the max-product solver [13]. More recently, Givoni et al. [8] extended this message passing framework for solving the HFL problem. The basic idea is to iteratively update the

following messages until convergence[1]:

$$\eta_{ij}^l = -c_{ij}^l + \min(\tau_i^l, -\max_{k \neq j}(\alpha_{ik}^l - c_{ik}^l)), l > 0$$

$$\eta_{ij}^l = -c_{ij}^l - \max_{k \neq j}(\alpha_{ik}^l - c_{ik}^l), l = 0$$

$$\alpha_{ij}^l = \min[0, -v_j^l + \phi_j^l + \sum_{k \neq i} \max(0, \eta_{kj}^l)], l < L+1$$

$$\alpha_{ij}^l = \min[0, -v_j^l + \sum_{k \neq i} \max(0, \eta_{kj}^l)], l = L+1$$

where the messages

$$\tau_k^{l+1} = \sum_{j=1}^{M^l} \max(0, \eta_{jk}^l) - v_k^l \quad , \quad \phi_j^{l-1} = \max_k(\alpha_{jk}^l - f_{jk}^l)$$

are required for connecting successive layers. The message τ_k^{l+1} is passed upwards from layer l to layer $l+1$, while the message ϕ_j^{l-1} goes down from layer l to layer $l-1$. The max-sum algorithm is guaranteed to converge on tree graphs, and has shown good performance for $L=1$ on graphs with cycles in many applications, e.g. [15]. It is important to mention that a practical way of dealing with message oscillations is to *damp* the messages at each iteration [14]

$$\eta = \gamma \eta_{\text{prev}} + (1 - \gamma)\eta$$

where $\gamma \in [0, 1[$ is the damping factor and η_{prev} is the previous message. Upon convergence, the set of facilities $\mathcal{F}^l$ in layer l that are open are $\mathcal{F}^l = \{\mathbf{v}_j^l | (\alpha_{ij}^l + \eta_{ij}^l) > 0\}$. The optimal MAP estimation for $\mathbf{x}$ is given by

$$x_{ij}^l = \begin{cases} 1 & \text{if } c_{ij}^l \leq c_{ik}^l \bigwedge \mathbf{v}_j^l, \mathbf{v}_k^l \in \mathcal{F}, \forall j, k \\ 0 & \text{otherwise} \end{cases}$$

3. Algorithm for vanishing point detection

This section shows that the detection and estimation of VPs can be formulated as an instance of the UFL problem discussed in Sec. 2.1. Such formulation requires defining a consistency metric $D(\mathbf{e}_i, \mathbf{v}_j^0)$ that measures the consistency of an edge $\mathbf{e}_i$ with a putative VP $\mathbf{v}_j^0$, and a function $W(\mathcal{S}, \mathbf{w})$ that, given a subset of edges $\mathcal{S}$, computes the most likely vanishing point $\mathbf{v}$.

3.1. Vanishing point detection as a UFL problem

Let $\mathbf{e}_i \in \mathcal{E}$ with $i = 1...N$ be the ith edge extracted from an image. The objective is to assign to each edge $\mathbf{e}_i$ a VP $\mathbf{v}_j^0 \in V^0$ using as few unique VP models as possible. This multi-model fitting problem can be casted as an instance of the UFL problem as follows: consider that the edges $\mathbf{e}_i$ are the customers and the putative VPs $\mathbf{v}_j^0$ are the facilities. Let the cost c_{ij}^0 be given by the function $D(\mathbf{e}_i, \mathbf{v}_j^0)$ that evaluates

---

[1]Remark that initially $\eta = 0$ and $\alpha = 0$.

the consistency between $\mathbf{e}_i$ and $\mathbf{v}_j^0$, and let v_j^0 be the cost for adding $\mathbf{v}_j^0$ in the final VP assignment. The goal is to select a subset of VPs in $\mathcal{V}^0$ such that sum of the consistency measures c_{ij}^0 and the costs $\mathbf{v}_j^0$ is minimized, which corresponds exactly to the minimization of Eq. 1. There are however some issues that must be addressed: (i) the selection of the set $\mathcal{V}^0$ of VP hypotheses, (ii) the definition of the function $D(\mathbf{e}_i, \mathbf{v}_j^0)$ that provides the client-facility cost c_{ij}^0 by measuring the consistency between edge $\mathbf{e}_i$ and a putative VP $\mathbf{v}_j^0$, and (iii) the choice of the function $W(\mathcal{S}, \mathbf{w})$ that, after clustering a subset $\mathcal{S}$ of edges, updates the VP location.

The set $\mathcal{V}^0$ containing the initial VP hypotheses depends mainly on the time constraints of the particular application. In the case of no time limitations, $\mathcal{V}^0$ can comprise all the point intersections between pairs of lines $\mathbf{l}_i, \mathbf{l}_j$ fitting every possible pair of image edges $\mathbf{e}_i, \mathbf{e}_j$, respectively. Otherwise, a fast RANSAC procedure can be used for quickly extract model hypotheses using minimal sample sets. The issues (ii) and (iii) are addressed in the next sections.

3.2. The consistency function $D(\mathbf{e}, \mathbf{v})$

Given an edge $\mathbf{e}_i$, comprising P_i points $\mathbf{e}_k$ with $k = 1...P_i$, and a putative VP $\mathbf{v}_j^0$, the objective is to find a cost function $D(\mathbf{e}_i, \mathbf{v}_j^0)$ that evaluates how well a line $\mathbf{l}_j$ in the pencil centered in $\mathbf{v}_j^0$ can fit the edge points $\mathbf{e}_k$. We propose to determine the line $\mathbf{l}_j$ that minimizes the sum of the squares of the geometric distances to the points, and use the root mean value of this sum as the client-facility cost c_{ij}^0. Contrary to what is suggested in [21], the minimization problem can be solved in a closed-form manner. Any line $\mathbf{l}_j$ going through $\mathbf{v}_j^0$ can be parametrized as follows

$$\mathbf{l}_j(\lambda) \sim (1 - \lambda)[\mathbf{a}]_\times \mathbf{v}_j^0 + \lambda[\mathbf{b}]_\times \mathbf{v}_j^0,$$

with $\mathbf{a}, \mathbf{b}$ being any two points non-collinear with $\mathbf{v}_j^0$, and λ being a free parameter. For the sake of convenience, the points $\mathbf{a}, \mathbf{b}$ are typically chosen as being the endpoints of a line segment orthogonal to the edge $\mathbf{e}_i$ and passing through its midpoint. We want to find the λ value such that:

$$\min_\lambda \sum_{k=1}^{P_i} d_\perp^2 (\mathbf{e}_k, \mathbf{l}_j(\lambda))$$

From the formula for the orthogonal distance $d_\perp$, it comes after some algebraic manipulations that

$$\sum_{k=1}^{P_i} d_\perp^2 (\mathbf{e}_i^k, \mathbf{l}_j) = \frac{(\mathbf{v}_j^{0\mathsf{T}} \mathsf{A}_2 \mathbf{v}_j^0)\lambda^2 + (\mathbf{v}_j^{0\mathsf{T}} \mathsf{A}_1 \mathbf{v}_j^0)\lambda + \mathbf{v}_j^{0\mathsf{T}} \mathsf{A}_0 \mathbf{v}_j^0}{(\mathbf{v}_j^{0\mathsf{T}} \mathsf{B}_2 \mathbf{v}_j^0)\lambda^2 + (\mathbf{v}_j^{0\mathsf{T}} \mathsf{B}_1 \mathbf{v}_j^0)\lambda + \mathbf{v}_j^{0\mathsf{T}} \mathsf{B}_0 \mathbf{v}_j^0}$$

$$(2)$$

where

$$A_0 = \sum_{k=1}^{P_i} [a]_\times \mathbf{e}_k \mathbf{e}_k^T [a]_\times, \quad A_1 = \sum_{k=1}^{P_i} ([a]_\times \mathbf{e}_k \mathbf{e}_k^T [b]_\times + [b]_\times \mathbf{e}_k \mathbf{e}_k^T [a]_\times) - 2A_0$$

$$B_0 = \sum_{k=1}^{P_i} [a]_\times \mathsf{I}_s [a]_\times, \quad B_1 = \sum_{k=1}^{P_i} ([a]_\times \mathsf{I}_s [b]_\times + [b]_\times \mathsf{I}_s [a]_\times) - 2B_0$$

$$B_2 = \sum_{k=1}^{P_i} ([b]_\times \mathsf{I}_s [b]_\times) - B_0 - B_1, \quad A_2 = \sum_{k=1}^{P_i} ([b]_\times \mathbf{e}_k \mathbf{e}_k^T [b]_\times) - A_0 - A_1$$

The minima and maxima of the objective function are the λ values for which the derivative is zero. By differentiating the expression of Eq. 2, it comes that these extrema can be easily computed by solving a second order equation. Given the particular arrangement between $\mathbf{a}$, $\mathbf{b}$, and the edge $\mathbf{e}_i$ we choose the root λ_0 that is closest to 0.5, and replace the result in the equation below:

$$D(\mathbf{e}_i, \mathbf{v}_j^0) = \sqrt{\frac{\sum_{k=1}^{P_i} d_\perp^2(\mathbf{e}_k, \mathbf{l}_j(\lambda_0))}{P_i}} \tag{3}$$

3.3. The function $W(\mathcal{S})$ for updating the VP estimate

After solving the UFL problem, the edges sharing the same label are clustered into a subset $\mathcal{S}$, and the objective is to determine the most likely intersection point $\mathbf{v} \sim W(\mathcal{S})$ for the lines $\mathbf{l}_i$ fitting the edges $\mathbf{e}_i \in \mathcal{S}$. We propose to update the VP by finding the point location that minimizes the weighted sum of the square of geometric distances to the lines $\mathbf{l}_i$. Taking into account the formula of the orthogonal distance, it comes after some algebraic manipulations that

$$W(\mathcal{S}) = \min_{\mathbf{v}} \quad \mathbf{v}^\mathsf{T} Q \mathbf{v}$$
$$\text{subject to}: \quad \mathbf{v}^\mathsf{T} \mathbf{p} = 1$$

with

$$Q = \sum_{i=1}^{N_i} w_i^2 \frac{\mathbf{l}_i \mathbf{l}_i^\mathsf{T}}{\mathbf{l}_i^\mathsf{T} \mathsf{I}_s \mathbf{l}_i}$$

where w_i is the length of each edge $\mathbf{e}_i$, and $\mathbf{p} = (0\ 0\ 1)^\mathsf{T}$. Remark that the purpose of the constraint is to assure that $v_3 = 1$ complies with the formula for computing the orthogonal distance $d_\perp$. We can rewrite the constrained minimization problem as an unconstrained one:

$$W(\mathcal{S}) = \min_{\mathbf{v}, \lambda} \quad \mathbf{v}^\mathsf{T} Q \mathbf{v} - \lambda(\mathbf{v}^T \mathbf{p} - 1).$$

with λ being a Lagrange multiplier. By differentiating the objective function, it comes that the minima can be determined by solving following matrix equation

$$\underbrace{\begin{bmatrix} 2Q & -\mathbf{p} \end{bmatrix}}_{Q'} \begin{pmatrix} \mathbf{v} \\ \lambda \end{pmatrix} = \mathbf{0},$$

Note that if the lines $\mathbf{l}_i$ are quasi-parallel, the problem becomes undetermined, which can be observed by the matrix Q' becoming poorly conditioned. In this case, the VP $\mathbf{v}$ is at infinity, and its direction can be computed by simply averaging over the directions of $\mathbf{l}_i$.

4. Detection of multiple orthogonal triplets

We assume in this section that a set of VPs has already been extracted using any type of VP detection approach e.g. the approach proposed in Sec. 3, and the objective is to detect multiple mutually orthogonal directions in the scene. As will be shown, this problem can be easily cast as a HFL problem.

Given the intrinsic calibration matrix K, two VPs $\mathbf{v}_j^0$ and $\mathbf{v}_k^0$ are orthogonal if the following relation is verified

$$\mathbf{v}_j^{0\mathsf{T}} \omega \mathbf{v}_k^0 = 0, \tag{4}$$

where $\omega = K^{-\mathsf{T}} K^{-1}$ is the image of the absolute conic [10]. Let the set

$$\mathbf{v}_m^1 = \{\mathbf{v}_j^0, \mathbf{v}_k^0, \mathbf{v}_l^0\}$$

be a mutually orthogonal triplet, meaning that each pair of VPs in $\mathbf{v}_m^1$ verifies Eq.4. Given a set of edges $\mathbf{e}_i \in \mathcal{E}$, a set of VPs $\mathbf{v}_j^0 \in \mathcal{V}^0$, and a set of orthogonal triplets $\mathbf{v}_k^1 \in \mathcal{V}^1$, whose individual VP elements are known and are contained in $\mathcal{V}^0$, the objective is to assign a VP to each edge $\mathbf{e}_i$, minimizing not only the number of VPs, but also the number of orthogonal triplets. This problem is cast as a HFL instance with two different layers (see Fig. 2): at the bottom layer $l = 0$ we have the edges $\mathbf{e}_i$ and the VPs $\mathbf{v}_j^0$, and at the top layer $l = 1$ we have the orthogonal triplets $\mathbf{v}_k^1$. In addition to the costs c_{ij}^0 and v_j^0 described in Sec. 3, there is a new penalization $v_k^1 : \mathcal{V}^1 \to \mathbb{R}$ for $\mathbf{v}_k^1$ being contained in the scene. The connection costs c_{jk}^1 between $\mathbf{v}_j^0$ and $\mathbf{v}_k^1$ are given by

$$c_{jk}^1 = \begin{cases} 0 & \text{if } \mathbf{v}_j^0 \in \mathbf{v}_k^1 \\ \infty & \text{otherwise} \end{cases}$$

There are three issues that must be addressed: (i) how to propose an initial set of orthogonal triplets $\mathcal{V}^1$, (ii) there might exist VPs in $\mathcal{V}^0$ that are not part of any orthogonal triplet $\mathbf{v}_k^1$, and (iii) the orthogonal triplets can share a common VD.

The issue (i) is solved as follows: for each pair of VPs $\mathbf{v}_j^0, \mathbf{v}_k^0$ in $\mathcal{V}^0$ whose angle between the corresponding VDs is in the range $[\frac{\pi}{2} - \theta, \frac{\pi}{2} + \theta]$, we obtain an exact orthogonal triplet $\mathbf{v}_m^1$ computed as follows

$$\mathbf{v}_m^1 = \{\mathbf{v}_1^0, \mathbf{v}_2^0, \mathbf{v}_3^0\}$$
$$\mathbf{v}_1^0 = \mathbf{v}_j^0, \mathbf{v}_2^0 = \text{Null}(\omega(\mathbf{v}_1^0\ \mathbf{v}_k^0)), \mathbf{v}_3^0 = \text{Null}(\omega(\mathbf{v}_1^0\ \mathbf{v}_2^0)),$$

which is added to $\mathcal{V}^1$, and where the operator $\text{Null}(M)$ returns the left nullspace of the matrix M. Note that the additional created VPs are also added to $\mathcal{V}^0$, which implies having very similar or even equal VPs in $\mathcal{V}^0$. This problem is easily handled by the HFL solver that prefers assignments with less VPs. For solving (ii), we add the vanishing groups $\mathbf{v}_m^1 = \{\mathbf{v}_j^0\}$ containing a single VP to $\mathcal{V}^1$ whenever there is no other VP in $\mathcal{V}^0$ whose VD makes up an angle in the

range $\left[\frac{\pi}{2}-\theta, \frac{\pi}{2}+\theta\right]$ with the VD of $\mathbf{v}_j^0$. The costs v_m^1 for sets $\mathbf{v}_m^1$ containing a single VP are always less than for orthogonal triplets, keeping these VPs in the final labeling. Finally, the issue (iii) is solved by noting that since we construct each orthogonal triplet $\mathbf{v}_m^1$ individually, we can keep track of similar VPs in $\mathbf{v}_m^1$ after the HFL labeling.
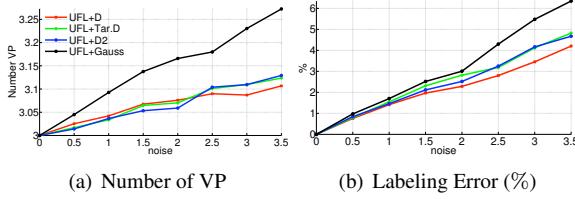
5. Experiments with synthetic data

(a) Number of VP   (b) Labeling Error (%)

Figure 3. Clustering of line pencils in synthetic data. We compare UFL using four different consistency metrics.

In this section we conduct a set of experiments in a simulation environment that considers an image of size 640×480 and the intrinsic parameters provided by the York Urban Database (YUD) [6]. We randomly generate 3 VDs in the Gaussian sphere, with the angles between them being always less than $20°$. For each VP $\mathbf{v}_j$, we generate a pencil of N line segments in the image, which are sampled into a discrete set of points $\mathbf{e}_k$ with $k = 1...P_i$. Each set $\mathbf{e}_i$ has a length between 20 to 200 pixels. The points are then perturbed with Gaussian noise of different magnitudes and 200 trials are run for each noise level.

Fig. 3 compares four different consistency metrics for quantifying c_{ij}^0 for the UFL clustering method: (i) **UFL+D** - our measure $D(\mathbf{e}_i, \mathbf{v}_j^0)$ described in Eq.3 using all the points in $\mathbf{e}_i$, (2) **UFL+D2** - the same measure $D(\mathbf{e}_i, \mathbf{v}_j^0)$ using only the two end points the edge $\mathbf{e}_i$; (3) **UFL+Tar.D** - the consistency metric of Tardif described in [21], and (4) **UFL+Gauss** - operate on the Gaussian sphere by analyzing the angle between the normal to the line $\mathbf{l}_i$ and $\mathbf{v}_j^0$. Clearly, UFL operating on the Gaussian sphere provides the worst labeling results with increasing magnitude of noise. The performance of the three metrics operating in the image plane are similar for low noise, but our metric $D(\mathbf{e}_i, \mathbf{v}_j^0)$, which uses all the points in $\mathbf{e}_i$ being clearly the top-performer for higher noise magnitudes. The consistency metric $D(\mathbf{e}_i, \mathbf{v}_j^0)$ operating on the two end points of $\mathbf{l}_i$ performs slightly worse, but with a high increase in computational efficiency (less input data). By taking this results in consideration, we decided to select **UFL+D2** for measuring the consistency between edges and VPs, being a good trade-off between accuracy and computational efficiency.

Given a cluster $\mathcal{S}$ containing N lines $\mathbf{l}_i$, we need to compute a better VP estimation. As in the previous experiment, we randomly generate a pencil $\mathcal{S}$ containing N lines, sam-
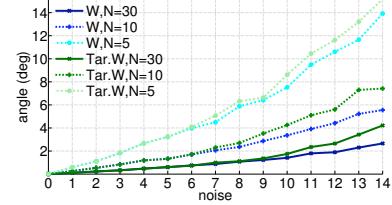

Figure 4. Accuracy of the estimation of VPs given a pencil of N lines. The error corresponds to the angle in the Gaussian sphere between the ground truth VP and the estimated VP.

ple the lines into a discrete set of points, perturb the points using Gaussian noise of different magnitudes, and then fit a line $\mathbf{l}_i$ to these points in the least-squares sense. We compare in Fig. 4 our function $W(\mathcal{S})$ described in Eq. 3.3 (**W**) with the VP estimator proposed by Tardif in [21] (**Tar.W**). A careful analysis of the graphic shows that our VP estimator provides better estimates for the same pencil of lines, being considerably more robust to the noise level. These results justify our choice for selecting $W(\mathcal{S})$ as VP estimator.

6. Experiments in real images

The section presents experimental results carried in real data. Our algorithm was implemented in Matlab, being the UFL and HFL solver run in MEX files. We compute an initial set of 5000 VP hypothesis for UFL using RANSAC over a minimal set of two edges. In order to handle possible outlier edges detected in the images, we added the empty sets $\mathbf{v}_\emptyset^0$ and $\mathbf{v}_\emptyset^1$ to both UFL and HFL, which have the facility costs $\mathbf{v}_\emptyset^{0,1} = 0$ and constant connections costs.

6.1. Experiments in YUD using the supplied lines

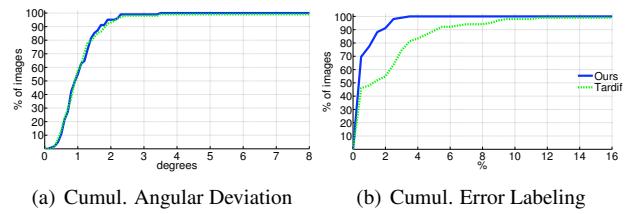(a) Cumul. Angular Deviation   (b) Cumul. Error Labeling

Figure 5. Comparison between our UFL approach with the method proposed by Tardif [21] for the detection of VPs.

We tested our algorithm for VP detection in the YUD [6], which consists in 102 calibrated images of man-made environments. Each image contains two or three VPs, line segments that were manually extracted, and whose VP membership is provided. Given the set of line segments, we run our UFL algorithm and compared the results against the ones obtained using the algorithm proposed by **Tardif** [21]. The results are shown in Fig. 5. The accuracy of the estimation of the VPs positions is very similar. However, our approach applies the UFL solver and computes for each
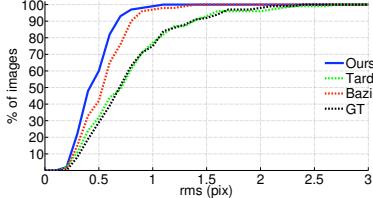
Figure 6. Cumulative consistency error computed using our D for the three groups of ground truth edges (belonging to orthogonal VDs). For each image we compute the root-mean-square (rms) consistency error across all lines fitting the estimated VPs; **GT** corresponds to the ground truth VPs provided by YUD.
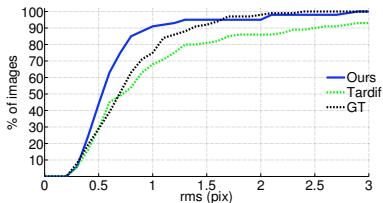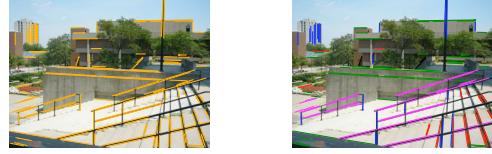

Figure 7. Cumulative consistency error computed using D for the three groups of ground truth edges. The results for **Ours** and **Tardif** were obtained using edges automatically detected.

output cluster a VP using $W(\mathcal{S}, \mathbf{w})$ (Eq. 3.3) without any refinement step, whereas Tardif's method re-estimates the position of the VP after the J-Linkage procedure using EM. Regarding the clustering of the lines, our approach shows some improvements, having in 92% of the images less than 2% of the lines wrongly labeled. In terms of computation time, **Tardif** takes on average 0.5 seconds on images of YUD, while our UFL approach needs 1 second (note that the number of initial VPs is the same for both).

Given the initial set of VPs obtained using the UFL algorithm, the objective now is to detect the manhattan directions, or similarly, a single rotation. We run our HFL method and compared it against (1) the globally optimal line clustering approach proposed by **Bazin** et al. [4], and (2) the rotation obtained using the three most orthogonal VDs of **Tardif** after fitting a perfect orthogonal frame to these VPs [6]. The results are shown in Fig.6. Despite of the close performance in terms of estimating the three orthogonal VDs, our method is computational more efficient, running more than 50 times faster than **Bazin**.

6.2. Experiments in YUD using extracted edges

In this section, we test our HFL algorithm for detecting the Manhattan frame in the YUD, but using edges extracted trough Tardif's detector [21] instead of the line segments supplied by the database (see Fig.8). The comparison with respect to **Tardif** is shown in Fig. 7. We consistently outperform **Tardif**, reaching 100% of success only approximately 1.5 pixels later than using the ground truth lines, which proves the robustness of our approach. The **Bazin**

(a) We simultaneously detected the Manhattan directions (red,green,blue) and 1 non-orthogoanl VD (magenta).

(b) Our algorithm detected 2 orthogonal triplets (middle and right), and assigned the blue direction as being common for both.

Figure 8. Two cases from the YUD. The left images show the extracted edges (orange), while the detection results are shown on the right. Black lines were assigned to the empty set (no VP).
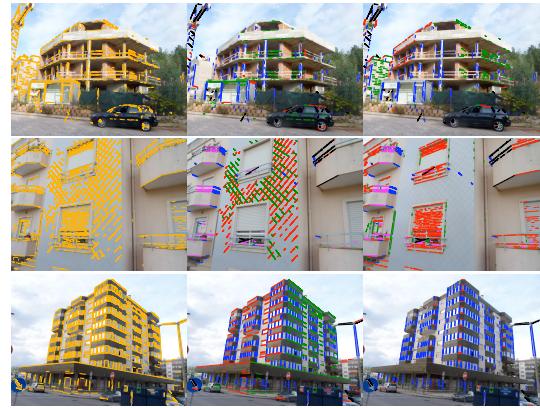

Figure 9. (Left) extracted edges, and (right) detection results. In each example (row) we detected 2 groups of orthogonal triplets with the blue VD in common. The VD in magenta (row 3) was detected but (incorrectly) not assigned to any triplet.

method was not included in this experiment due to its high computational cost when compared to **Ours** and **Tardif**.

Fig. 8 shows two particularly interesting results obtained by our approach. Using the HFL, we correctly identified in Fig.8(a) the Manhattan frame and simultaneously estimated the VDs corresponding to the handrails of the stairs. In Fig.8(b), we identified two different mutually orthogonal triplets (remark that for the analysis in Fig. 7, the orthogonal triplet with more lines was automatically selected), one corresponding to the Manhattan frame and the other is due to the squares on the floor. We also identified that both orthogonal frames share the same vertical direction.

6.3. Experiments in scenes containing multiple orthogonal triplets

This section shows experiments on real images containing more than one orthogonal triplet of VDs. The images

Figure 10. (Row 1) extracted edges (left) and clustering obtained using UFL (right); (Row 2) 3 groups of orthogonal triplets were detected using HFL, the 2 on the left have the blue VD in common

shown in Fig. 9 were obtained using a Panasonic DMC digital camera, while the image shown in Fig. 10 was downloaded from Flickr. We run Tardif's edge detector for obtaining the input edges for our UFL and HFL algorithms.

The results are quite encouraging. We are able to handle high-resolution images containing many edges, detecting simultaneously both multiple orthogonal triplets as well as single VDs. Fig. 10 shows results for both the UFL labeling (top,right) and the following HFL procedure (bottom). There is one error in the hierarchical clustering (bottom, middle). Our approach mistakenly assigns the edges on the roof to one orthogonal triplet, but this issue can be a consequence of either a poor estimation of the focal length or an ineffective tuning of the facility costs for HFL.

7. Conclusions

We presented an automatic and global approach for the detection of VPs and mutual orthogonal VDs. The core of the framework is the formulation of these multi-model fitting problems as UFL and HFL instances, which are solved using a message passing approach. The effectiveness of the framework is proved by challenging real scenarios containing multiple Manhattan-world configurations.

Acknowledgements

References

[1] M. E. Antone and S. J. Teller. Automatic recovery of relative camera rotations for urban scenes. In *CVPR*, 2000.

[2] G. Baatz, K. Kser, D. M. Chen, R. Grzeszczuk, and M. Pollefeys. Handling urban location recognition as a 2d homothetic problem. In *ECCV*, 2010.

[3] S. T. Barnard. Interpreting perspective images. *Artificial Intell.*, 1983.

[4] J. C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *CVPR*, 2012.

[5] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *IJCV*, 2012.

[6] P. Denis, J. H. Elder, and F. J. Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *ECCV*, 2008.

[7] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 2007.

[8] I. E. Givoni, C. Chung, and B. J. Frey. Hierarchical affinity propagation. In *UAI*, 2011.

[9] L. Grammatikopoulos, G. Karras, and E. Petsa. An automatic approach for camera calibration from vanishing points. *Journal of Photogrammetry and Remote Sensing*, 2007.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[11] H. N. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *IJCV*, 2012.

[12] J. Kosecka and W. Zhang. Video compass. In *ECCV*, 2002.

[13] F. Kschischang, S. Member, B. J. Frey, and H. andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 2001.

[14] N. Lazic, B. J. Frey, and P. Aarabi. Solving the uncapacitated facility location problem using message passing algorithms. *Journal of Machine Learning Research*, 2010.

[15] N. Lazic, I. E. Givoni, B. J. Frey, and P. Aarabi. Floss: Facility location for subspace segmentation. In *ICCV*, 2009.

[16] H. Li. Two-view motion segmentation from linear programming relaxation. In *CVPR*, 2007.

[17] F. M. Mirzaei and S. I. Roumeliotis. Optimal estimation of vanishing points in a manhattan world. In *ICCV*, 2011.

[18] C. Rother. A new approach for vanishing point detection in architectural environments. In *BMVC*, 2000.

[19] 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 *CVPR*, 2004.

[20] S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *ICCV*, 2009.

[21] J.-P. Tardif. Non-iterative approach for fast and accurate vanishing point detection. In *ICCV*, 2009.

[22] E. Tretyak, O. Barinova, P. Kohli, and V. Lempitsky. Geometric image parsing in man-made environments. *IJCV*, 2012.

[23] J. Xiao, J. Wang, P. Tan, and L. Quan. Joint affinity propagation for multiple view segmentation. In *ICCV*, 2007.