UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Manhattan Frame Detection in Lens Distorted Images

Author:
Kostiantyn LIEPIESHOV

Supervisor:
Rostyslav HRYNIV
James PRITTS

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2020

Declaration of Authorship

I, Kostiantyn LIEPIESHOV, declare that this thesis titled, "Manhattan Frame Detection in Lens Distorted Images" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Manhattan Frame Detection in Lens Distorted Images

by Kostiantyn LIEPIESHOV

Abstract

Camera auto-calibration from a single image with radial distortion is a prevalent task in computer vision. Most of the existing approaches are based on the same process of extraction of features, such as circles, from the image. Since those features are noisy, the error is propagated to the higher level, and the final estimations are inaccurate.

We incorporate the constraints imposed by the division model of radial distortion and suggest a simple approach that gives soft estimates of three Manhattan directions. For this task, we adapt a well-known Expectation Maximisation algorithm. We combine it with different initialization and filtering steps that we form based on the division model and Manhattan world assumptions.

We illustrate the performance of the proposed approach on YORK Urban Database (YUD) and AIT Dataset of indoor and outdoor scenes. Besides, we experiment with the proposed initializations and filtering procedures.

Acknowledgements

I am highly grateful to James Pritts for inspiring and directing this work, Rostyslav Hryniv for consistent help having the structured Mathematical explanation to every problem, Oles Dobosevych for all the motivation during this long way.

Also, I want to thank the Ukrainian Catholic University, Faculty of Applied Sciences and everybody connected along these four years for helping me into building my solid knowledge and directing me into this challenging but highly interesting sphere.

Contents

List of Figures

List of Tables

List of Abbreviations

LCC	**L**ine of **C**ircle **C**enters
VP	**V**anishing **P**oint
EM	**E**xpectation **M**aximisation algorithm
MAP	**M**aximum **A** **P**osteriori
RANSAC	**RAN**dom **SA**mple **C**onscensus
FOV	**F**ield **O**f **V**iew
CS	**C**oordinate **S**ystem

List of Symbols

d	euclidean distance
$\mathbf{X}$	matrix/vector
x	scalar value
π	projection plane
Π	scene plane

Dedicated to my family

Chapter 1

Introduction

1.1 Motivation

A camera with a wide lens is of much use nowadays. It can enclose much more information, and people tend to use it more often due to a much more beautiful visual effect. It became practical to shoot the 360° videos and choose what angle to leave in a final version afterward. Practically every camera in use produces radial distortion. Even if this distortion is small enough for our perception, it creates a massive problem for computer vision pipelines. The reason is that after adding a fisheye effect to the image, the latter does not agree with the pinhole camera model anymore and an additional preprocessing is required to undistort it before any further processing.

Several approaches were proposed to describe and solve the mathematical model of radial distortion, which used complicated polynomial equations [5]. Currently, numerous approaches exist for estimating the parameters of radial distortion and removing it from the image. However, most of those approaches introduced several simplifications, which helped them to solve the objective. The most common ones are:

- square pixels assumption or skew of the lens is zero ($f_x = f_y = f$ or $s = 0$);
- the principal point of the image is in the center of the image;
- Manhattan frame assumption - orthogonality of the three Manhattan directions;
- the center of radial distortion is at the center of the image;

Those were used in most state-of-the-art approaches [30, 21]. Also, Fitzgibbon [13] and Brauer-Burchardt and Voss [4] proposed a simplified model of radial distortion called division model. Its single-parameter variant was used in many works [30, 21, 3, 2]. Even with such underparameterization, it can describe radial distortion of different levels with very good accuracy Barreto [3]. We follow the same strategy and make the same assumptions in our approach, except our solution does not rely on the Manhattan frame assumption. We only assume three dominant vanishing points in the image. An additional assumption that they correspond to the three mutually orthogonal directions defining a Manhattan frame could give an opportunity to use the vanishing points estimated by the proposed approach for auto-calibration setups such as focal length estimation [29].

Most of the approaches [6, 17, 31, 18] use the radial distortion property under the division model, in which parallel lines in the 3D scene are imaged as circles with a small curvature [24]. Based on this fact, the initial step of their methods is to fit multiple circles to the distorted edges on the image. Given the set of noisy circles,

there is a variety of solvers that find the position of three orthogonal vanishing points and parameter of distortion λ [6, 31].

We build our approach based on the structure formed by the circles that belong to the same vanishing point, the so-called line of circle centers (LCC).

1.2 Structure

The thesis is organized in the following way:

Background chapter gives the reader an introduction into the field of projective geometry and the geometry of a pinhole camera. The radial distortion formation with all its properties is thoroughly explained, followed by the Expectation Maximisation algorithm general probabilistic statement.

Related Work chapter gives an overview of previous approaches for camera auto-calibration and distortion parameter estimation in different setups. The proposed approach is compared with them.

Proposed Method chapter describes the way the circle creation is performed. The adaptation of the Expectation Maximisation (EM) algorithm with initialization and outlier detection is then reported. Several ways for λ approximation are also introduced.

Datasets chapter gives an overview of used datasets with a detailed description of the problems and advantages of each of them. The synthetic creation process from one of the datasets is introduced with graphical examples.

Experiments chapter discusses the achieved results with a statistical exploration of numerical stability, outlier robustness for different initialization and outlier detection approaches. The results are provided for fully synthetic scenes as well as for synthetic and completely real images.

Conclusions chapter summarises the pros and cons of the proposed algorithm according to the obtained results in the experiments chapter. The next steps as build-ups to this work are addressed.

Chapter 2

Related Work

Auto-calibration of a camera from a single image usually use Manhattan frame assumption, and then with the known principal point the auto-calibration task is solved using three vanishing points from the image. Coughlan and Yuille [8] were the first to propose a method to tackle this problem in their works [8, 9]. They started with a bottom-up approach that uses Bayesian inference on very low-level features, i.e., gradients of edges. They used the idea of Manhattan assumption of three orthogonal directions frame. This inspired other works, which tried to improve the results by using the Expectation Maximisation algorithm [10] for more accurate identification of positions of vanishing points [1, 23, 25, 16, 32]. Antone and Teller [1] used EM for VPs detection based on the line segments extracted from the edges. All of the segments were normalized using Gauss sphere. Outlier segments were modeled using Gaussian with a large variance on a sphere. It should have approximated the uniform distribution. The initial guess was made by thresholding the peaks in Hough space.

Then Kosecka and Zhang [16] tried to advance Antone and Teller [1] results incorporating the next changes. They tried to avoid Gaussian sphere representation using normalization that does not rely on the camera parameters. They proposed a new, more robust initialization from modes found on the histogram of line segments directions.

Schindler and Dellaert [23] returned to a lower level. They tried to adapt EM for Low-level Edge Grouping and Vanishing Point estimation by labeling each pixel as outlier or an edge that belongs to one of the VPs. They estimated the VPs by a nonlinear optimization technique, e.g., Levenberg-Marquardt. It is noteworthy that the only assumption they used was the existence of vanishing points. So they tried to generalize to not only the Manhattan world model but also proposed their Atlanta world model and concluded that the algorithm Atlanta world assumption.

Later, Wildenauer and Vincze [32] continued with the line segment-based approach. They tried to fix such weaknesses as initialization by proposing Multi Ransac [12] based initialization; outlier detection by modeling the outlier distribution as uniform with density set as the density of other models evaluated at a value of 3σ, where σ denotes the standard deviation of the models. They also add a new merging indicator based on the similarity metric and new distance measure between a VP and a line segment.

Tardif [25] followed Wildenauer and Vincze [32] with an improvement of results using the newly proposed J-linkage algorithm [26] for better initialization. J-linkage worked so good, that the authors considered the EM as an optional refinement step in their method.

The problem of radial distortion in lenses was researched by Brown [5]. The problem was that his approach used polynomials of higher order. Thus, Fitzgibbon

[13] and Brauer-Burchardt and Voss [4] proposed the simplified single parameter radial distortion model called division model.

After that, it became essential to include the division model into the solutions for single view auto-calibration methods [24, 27, 2, 31] to keep up with other approaches. With the introduction of the division model, it was proven that line segments are arcs of circles with small curvature [4]. Strand and Hayman [24] proposed technique to auto-calibrate based on circles extracted from the edges. After that, the methods diverged into different branches: creating solvers based on the circles and contrary [2, 31] or trying to create sophisticated structures that are possible to extract from the image (e.g., repeated patterns), and constructing corresponding solvers based on these features [19, 20, 22]. One of the exciting properties appearing under the division model is the Lines of Circle Centers. Antunes et al. [2] used it for constraining the extracted circles. Based on them, he created a much robust solver using Plücker coordinates.

In our work, we adopt the Expectation Maximisation algorithm, which was previously shown to give good results [1, 16, 32, 25] for estimation of vanishing points based on line segments, to accurately estimate the Lines of Circle Centers [2]. In addition to that, an unsupervised approximation of radial distortion parameter λ is proposed. That gives the possibility to find the precise positions of vanishing points and can be used to affinely rectify the image.

We define two main contributions of this work:

- the adaptation of simple, fast and well-known iterative approach (Expectation Maximisation algorithm) for the soft assignment of circles to the three dominant vanishing points;
- estimation of distortion parameter λ using the circle formation properties under the division model to filter the inlier circles.

Chapter 3

Background

3.1 Homogeneous coordinates properties

For simpler explanation of the subsequent transformations, it is convenient to use 2D homogeneous coordinates introduced by August Ferdinand Möbius [14]. Simply, it is a way of representing 2D coordinates in more flexible way. To transform a point in the Cartesian coordinate system (X, Y) into homogeneous coordinates we simply add a third coordinate $Z \neq 0$ and get the point (XZ, YZ, Z) in homogeneous coordinates; accordingly a point (X, Y, Z) in the homogeneous coordinates corresponds to $(X/Z, Y/Z)$ in the Cartesian ones, where $Z \neq 0$. What is important for us in these coordinates is the special interpretation: the mapping process from homogeneous coordinates to Cartesian can be described as the process of perspective projection of points from 3D space onto the plane $Z = 1$, can also be interpreted as 2D space (Figure 3.1 A). The trick is that this transformation is non-linear in inhomogeneous coordinates, but it is a linear one in homogeneous.

3.2 Camera Model

The process of photography can be described as some way of transformation (3.1) of 3D real-world space points, X_{space}, into 2D photo plane pixels, X_{photo}

$$X_{photo} = f(X_{space}) \tag{3.1}$$

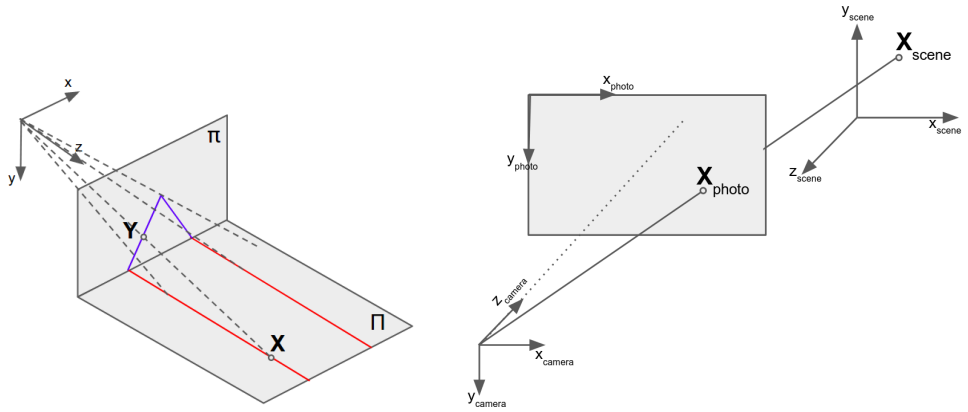
The common practice in computer vision is that our camera follows the simplest model of a pinhole camera, also known as a camera obscura. It consists of two parts [14]: camera location (location of a pinhole) and orientation with respect to the world frame, referred to as extrinsic parameters; intrinsic parameters - mapping between camera coordinates and pixels in the image. In (3.2) intrinsic and extrinsic parameters are denoted as K and [R t] respectively

$$\alpha(x_{photo}, y_{photo}, 1)^\top = [\mathbf{K}][\mathbf{Rt}](x_{space}, y_{space}, z_{space}, 1)^\top \tag{3.2}$$

The geometry of the pinhole camera is depicted in Figure 3.1 (B).

3.2.1 Extrinsic parameters

Extrinsic part is described by a 3×4 matrix and plays the role of changing the coordinate system (CS) [14] from Scene CS to Pinhole CS (Figure 3.1 (B)). This transformation can be decomposed into two steps: rotation and translation. Rotation is defined by a 3×3 rotation matrix which is invertible and can be decomposed into



(A) (B)

FIGURE 3.1: (A) an example of perspective projection of points from 3D space onto plane π; red lines with the same direction from Π are projected onto π as blue lines that intersect in a finite point; Y is the projection of point X. (B) Pinhole camera model depiction with X_{photo} being the image of the point X_{scene} in a scene.

three rotational matrices (3.3).

$$\mathbf{R} = \mathbf{R}(\gamma_x)\mathbf{R}(\gamma_y)\mathbf{R}(\gamma_z) = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{3.3}$$

where $\gamma_x, \gamma_y, \gamma_z$ stand for the rotation angles in each of the axis respectively. The translation $\mathbf{t} \in \mathbb{R}^3$ is responsible for change of the origin to the position of pinhole.

3.2.2 Intrinsic parameters

Intrinsic part does the actual image formation process [14] and is defined by 3×3 matrix $\mathbf{K}$ (3.4). It defines the transformation that projects the 3D points onto the image pixels plane.

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

where f - is the focal length in pixels with $f_x = fk_x, f_y = fk_y$ where k_x, k_y are the parameters of the aspect ratio that is introduced due to the real nature of lens; s - skew of the sensor; $(c_x, c_y)^\top$ - is the principal point offset relative to the image origin (in computer vision it is commonly a top left corner).

In most of the works the assumption of orthogonal raster ($s = 0$) and unary aspect ratio ($f_x = f_y = f$) is used, also referred to as ORUA. We do follow this assumption in our work as well. Under this assumption $\mathbf{K}$ simplifies to

$$\mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

3.2.3 Properties that appear in the image (Vanishing Point, Vanishing Line)

After understanding the camera matrix, it is possible to see that it can be simplified as a projection of 3D points on the image plane. There are several important structures for the later usage that follow from this camera formulation, such as vanishing points and vanishing lines. Parallel lines in the world scene appear to intersect in the image plane (3.7). The point of their intersection is called a vanishing point; denoting by $\mathbf{v}$ its homogeneous coordinates and by $\mathbf{l_j} = (l_{j1}, l_{j2}, l_{j3})^\top \; j = 1, 2$, the coordinates of the lines intersecting in the image plane, we get

$$\mathbf{v} = \mathbf{l_1} \times \mathbf{l_2} \tag{3.6}$$

$$\begin{bmatrix} \mathbf{l_1} & \mathbf{l_2} \end{bmatrix}^\top \mathbf{v} = 0^\top \tag{3.7}$$

The line connecting two vanishing points is called a vanishing line (3.8).

$$\mathbf{l}' = \mathbf{v_1} \times \mathbf{v_2} \tag{3.8}$$

where $\{\mathbf{v_1}, \mathbf{v_2}\}$ - two different vanishing points in the image in homogeneous coordinates; $\mathbf{l}'$ - vanishing line created with the set of vanishing points. Depending on the number of planes in the original scene, there could appear different numbers of vanishing points.

3.3 Radial distortion

In real-world cameras, there always appears to be an unintentional radial distortion created by the nature of the camera lens. Brown [5] did a comprehensive research of radial distortion and proposed the mathematical model that uses higher order polynomials (3.9).

$$\begin{aligned} x_u &= x_d + \tilde{x_d}(\lambda_1 r_d^2 + \lambda_2 r_d^4 + \lambda_3 r_d^6 + \cdots) + [P_1(r_d^2 + 2\tilde{x_d}^2) + 2P_2\tilde{x_d}\tilde{y_d}][1 + P_3 r_d^2 + \cdots] \\ y_u &= y_d + \tilde{y_d}(\lambda_1 r_d^2 + \lambda_2 r_d^4 + \lambda_3 r_d^6 + \cdots) + [2P_1\tilde{x_d}\tilde{y_d} + P_2(r_d^2 + 2\tilde{y_d}^2)][1 + P_3 r_d^2 + \cdots] \end{aligned} \tag{3.9}$$

with

$$\begin{aligned} \tilde{x_d} &= x_d - x_c \\ \tilde{y_d} &= y_d - y_c \\ r_d &= \sqrt{(x_d - c_x)^2 + (y_d - c_y)^2} \end{aligned}$$

where $(x_u, y_u)^\top$ is point in the undistorted space; $(x_d, y_d)^\top$ is the image of this point in the distorted space; $(c_x, c_y)^\top$ - the location of the center of distortion; $\lambda_1, \lambda_2, \lambda_3$ - are the coefficients of radial distortion; P_1, P_2, P_3 - are coefficients of decentering distortion.

It can describe the radial distortion of different levels, but uses a lot of parameters and includes even-order terms, that makes it hard to solve.

The much simpler description of a radial distortion was proposed by Fitzgibbon [13] and Brauer-Burchardt and Voss [4], later referred to as the division model (3.10).

$$\begin{cases} x_u = \frac{x_d}{1 + \lambda_1 r_d^2 + \lambda_2 r_d^4 + \cdots} \\ y_u = \frac{y_d}{1 + \lambda_1 r_d^2 + \lambda_2 r_d^4 + \cdots} \end{cases} \tag{3.10}$$

In this model, the center of distortion is assumed to be known and the coordinate system is recentered according to it. The significant advantage of this model in comparison with the Brown model [5] is that even significant distortion is possible to describe at a much lower order. It was shown [28], even with one parameter, it is possible to achieve good estimation results. In this work the one parameter setup of the division model is used, so the point transformation under division model distortion is stated as

$$\gamma(x_u, y_u, 1)^\top = (x_d, y_d, 1 + \lambda(x_d^2 + y_d^2))^\top \quad (\gamma \neq 0) \tag{3.11}$$

Here and furthermore in this work we assume that the center of distortion is at the center of the image and the coordinate system is recentered according to it.

3.3.1 Lines under division model distortion

It appears that lines under the Division Model with parameter λ are transformed into circles, e.g., line $(l_1, l_2, l_3)^\top$ is transformed into a circle $(x_c, y_c, R)^\top$ with center (x_c, y_c) and radius R [4], where

$$x_c = \frac{-l_1}{2\lambda l_3};\ y_c = \frac{-l_2}{2\lambda l_3};\ R = \sqrt{x_c^2 + y_c^2 - \frac{1}{\lambda}} \tag{3.12}$$

The above relations give a simple dependence between x_c, y_c, R and distortion parameter λ that will be used to estimate λ estimation in Proposed Method.

3.3.2 Line of Circle Centers

Lemma 3.3.1. *The group of lines that pass through the same vanishing point, create the pencil of circles such that the centers of these circles form the line in a distorted image.*

Proof. In this proof the origin is located at the center of distortion. Let us take a single line with parameters $(l_1, l_2, l_3)^\top$. We are given a constraint that it should pass through a vanishing point, i.e. $(v_x, v_y, 1)^\top$. It means that

$$l_1 v_x + l_2 v_y + l_3 = 0 \implies l_3 = -(l_1 v_x + l_2 v_y) \tag{3.13}$$

Then the line $(l_1, l_2, l_3)^\top$ can be parametrized using l_1, l_2, v_x, v_y as $(l_1, l_2, -(l_1 v_x + l_2 v_y))^\top$. Under the division model with parameter λ we obtain that the line will be transformed into a circle with center

$$x_c = \frac{l_1}{2\lambda(l_1 v_x + l_2 v_y))};\ y_c = \frac{l_2}{2\lambda(l_1 v_x + l_2 v_y)} \tag{3.14}$$

If we calculate the projection of the circle center onto vector created by origin $(0, 0, 1)^\top$ and the vanishing point, we will obtain (3.15).

$$x_c v_x + y_c v_y = \frac{l_1 v_x + l_2 v_y}{2\lambda(l_1 v_x + l_2 v_y)} = \frac{1}{2\lambda} \tag{3.15}$$

It turns out that the size of this projection is not dependent on the parameters $\{l_1, l_2\}$. Thus, the centers of the circles that pass through the point $(v_x, v_y, 1)^\top$ form a line with parameters $(v_x, v_y, -\frac{1}{2\lambda})^\top$. □

3.3.3 Properties that appear under division model distortion

Several properties appear in the image after the application of the division model of radial distortion. The circles formed with lines that pass through the same vanishing point create a pencil of circles. Therefore, for each of the vanishing points before distortion, there are two symmetric vanishing points after. Those are the two intersection points of the pencil of circles. Since the LCC is the line that goes through the centers of the circles of the pencil, it is by definition the line that splits the segment created by two symmetric vanishing points equally and is also perpendicular to it.

3.4 Expectation Maximisation Algorithm

Maximum likelihood estimation is the most commonly used method in statistics for estimating the parameters θ of the probability distribution $p(\cdot|\theta)$ in the parameter space Θ. For the given observed data X from the statistical model, it searches for the set of parameters that maximises the likelihood $L(\theta|X)$,

$$L(\theta|X) = p(X|\theta) \tag{3.16}$$

In other words, the maximum likelihood estimate $\hat{\theta}$ for the given X is equal to

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, L(\theta|X) \tag{3.17}$$

Most commonly, the observed data are assumed to be independent and identically distributed. Then the estimate of θ is found as the extremum point of the log-likelihood as a function of θ. The problem with maximum likelihood approach is that it assumes all relevant random variables are observed, i.e., that we have the complete dataset. In the case the observed data X is dependent on another unobserved random variable Z, also referred to as *latent variable*, the likelihood function (3.16) is marginalised and takes the following form:

$$L(\theta|X) = p(X|\theta) = \int p(X, Z|\theta) dZ \tag{3.18}$$

In this case, the maximum likelihood estimation approach is intractable, as we cannot integrate over Z.

Expectation maximisation algorithm is proposed in [10] to deal with the problem of latent variables. It is an iterative algorithm for iterative optimisation of θ that consists of taking expectation followed by the maximisation steps. The authors introduce the function $Q(\theta|\theta^{(r)})$,

$$Q(\theta|\theta^{(r)}) = E_{Z|X,\theta}(\log L(\theta|XZ)), \tag{3.19}$$

which is the expectation with respect to the current conditional distribution of Z given X and current estimate of θ of marginalised log-likelihood of θ on latent variable Z. This allows to iteratively obtain the $(r+1)^{th}$ step estimate of θ based on the r^{th} step estimate as the extremum point of $Q(\theta|\theta^{(r)})$. It is also proved in [10] that $Q(\theta|\theta^{(r)})$ is the lower bound to $p(X|\theta)$.

The steps of the algorithm are then formulated as follows. The **expectation** step:

- the current value of Z is computer based on the current estimate $\theta^{(r)}$ of θ;
- $Q(\theta|\theta^{(r)})$ is formulated based on $\theta^{(r)}$ and Z;

The **maximisation** step:

- the new estimate of θ is computed as the maximisation of the formulated expectation function (3.20):

$$\theta^{(r+1)} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta|\theta^{(r)}) \tag{3.20}$$

Then the new estimate of $\theta^{(r+1)}$ is used in the next iteration and the algorithm is looped until the convergence criteria is met.

The Expectation Maximisation is proven to find the (local) maximum, due to maximisation of the likelihood of parameters at each step.

Most commonly the latent variables are used in computer vision as an unobserved part of the data, the random variable that represents the membership of data points to one of the models in multi-model parameter estimation, e.g., mixture models. We adapt the same representation of latent variables in 4.1.

Chapter 4

Proposed Method

In Section 3.3.2, we showed that each vanishing point can create a Line of Circle Centers with the circles that pass through it. Under the assumption of three dominant directions in the image, there are three dominant line models in the space of circle centers. Estimating these three lines from the centers of the extracted circles will give us the λ-dependent positions of three orthogonal vanishing points. Using the dependency between a circle center, its radius, and λ, we can apply a Kernel Density Estimation algorithm to obtain the λ parameter. Then the λ-dependent equations can be solved.

Based on the set of all circles extracted from the distorted image, it is possible to find out the location of vanishing points and distortion parameter λ. The next subsections are to describe each of these steps and follow the next structure:

- Adaptation of Expectation Maximisation algorithm for the detection of LCCs;
- Kernel density estimation for lambda approximation from extracted circles;
- Curvature Based Circle Consistency Filtering.

4.1 Adaptation of Expectation Maximisation algorithm for the detection of LCCs

The problem states that given the set of centers of the circles, we have to determine three dominant lines that most accurately fit them. More generally formulated, it is a task of multi-instance line detection in the arbitrary set of points in R^2.

4.1.1 Random displacement model

We have $J = 3$ line models $L_1, L_2, \ldots, L_J$ described by the line equations

$$\ell_j(x, y) := -\sin(\alpha_j)x + \cos(\alpha_j)y + \beta_j = 0, \qquad j = 1, \ldots, J, \tag{4.1}$$

where α_j and β_j are the direction angle and intercept of the corresponding line, respectively. Also, there are n points (x_i, y_i), $i = 1, \ldots, n$, which may belong to any of these models but are subject to random displacements. Our task is to suggest an optimal soft labelling that for each point gives its probability of being a member of each of the line models.

There are some *prior* probabilities π_j that any given point has come from the respective model. Conditioned on the event that the point (x_i, y_i) belongs to the j^{th} model L_j, we assume that $x_i = \hat{x}_i + \eta_1$ and $y_i = \hat{y}_i + \eta_2$, where $(\hat{x}_i, \hat{y}_i)$ is some point on the line L_j and (η_1, η_2) is a random displacement. In contrast to the regression model, in which the displacements are only possible in the y-direction,

here (η_1, η_2) is a two-dimensional random variable. However, since goodness-of-fit of the point (x_i, y_i) to the model L_j only depends on the orthogonal distance from that point to the line L_j—i.e., on the orthogonal projection of (η_1, η_2) on the normal vector $\mathbf{n}_j := (-\sin(\alpha_k), \cos(\alpha_j))$ to the line,— it does not matter what particular point $(\hat{x}_i, \hat{y}_i)$ of the model L_j was moved off to the point (x_i, y_i) as long as $-\eta_1 \sin(\alpha_j) + \eta_2 \cos(\alpha_j)$ stays constant. As a result, it makes sense to assume that the noise (η_1, η_2) only displaces points of the model L_j in the direction $\mathbf{n}_j$ orthogonal to the line L_j. In other words, we can assume that (η_1, η_2) is a degenerate two-dimensional Gaussian random noise whose covariance matrix degenerates along the direction vector $(\cos(\alpha_j), \sin(\alpha_j))$ of the line, and we can model (η_1, η_2) as $\xi_j \mathbf{n}_j$ with a univariate normal random variable ξ_j of mean 0 and some variance σ_j^2.

To sum up, the conditional density of the point (x_i, y_i) given the model L_j is equal to

$$\phi(x_i, y_i \mid L_j) = \phi_j(x_i, y_i) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp\left\{ -\frac{d_j^2(x_i, y_i)}{2\sigma_j^2} \right\},$$

where d_j is the (signed) orthogonal distance to the line. Observe that

$$\begin{aligned} d_j(x_i, y_i) &= -\eta_{1i} \sin(\alpha_j) + \eta_{2i} \cos(\alpha_j) = -(x_i - \hat{x}_i) \sin(\alpha_j) + (y_i - \hat{y}_i) \cos(\alpha_j) \\ &= -\sin(\alpha_j) x_i + \cos(\alpha_j) y_i + \beta_j = \ell_j(x_i, y_i), \end{aligned}$$

so that finally

$$\phi(x_i, y_i \mid L_j) = \phi_j(x_i, y_i) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp\left\{ -\frac{\ell_j^2(x_i, y_i)}{2\sigma_j^2} \right\}$$

4.1.2 Setting of the mixture model

We have the following setting:

- points $\mathbf{p}_i = (x_i, y_i)$, $i = 1, \ldots, n$, center of circles from the image;
- line models L_j, $j = 1, \ldots, J$, each fixed by the angle α_j and the intercept β_j;
- $\xi_{ij} \mathbf{n}_j$ is a random displacement of the point $\hat{\mathbf{p}}_{i,j}$ of the model L_j in the orthogonal direction resulting in $\mathbf{p}_j$; here $\hat{\mathbf{p}}_{i,j}$ is the orthogonal projection of $\mathbf{p}_i$ on the line L_j, $\mathbf{n}_j = (-\sin(\alpha_j), \cos(\alpha_j))$ is the normal to the line L_j and ξ_{ij} is a random scaling factor, realization of $\mathcal{N}(0, \sigma_j^2)$.

The soft point assignment for a mixture line models suggests that each point $\mathbf{p}_i = (x_i, y_i)$, $i = 1, \ldots, n$, belongs to a model L_j with some *prior* probability $\pi_j \in (0, 1)$, $j = 1, \ldots, J$ (so that $\sum_j \pi_j = 1$). In view of the above discussions, we get the following mixture model:

$$\mathbf{p}_i = \hat{\mathbf{p}}_{i,j} + \xi_{ij} \mathbf{n}_j \qquad \text{with probability } \pi_j.$$

Under a natural assumption that ξ_{ij} are independent random variables, the total probability rule gives the joint density of $\mathbf{p}_1, \ldots, \mathbf{p}_n$ in the form

$$\prod_{i=1}^{n} \Big(\sum_{j=1}^{J} \pi_j \phi_j(x_i, y_i) \Big).$$

Denote by $\theta := (\alpha_1, \ldots, \alpha_J, \beta_1, \ldots, \beta_J, \sigma_1^2, \ldots, \sigma_J^2)$ the parameter vector of the mixture model. Then the log-likelihood of θ is

$$\log L(\theta|X) = \sum_{i=1}^{n} \log\Big(\sum_{j=1}^{J} \pi_j \phi_j(x_i, y_i)\Big),$$

maximizing which we update the parameters of the mixture model.

To maximise the log-likelihood, we use the Expectation Maximisation algorithm. First, the parameter vector θ is initialised as explained in Subsection 4.1.3. Denoting by $\theta^{(r)}$ the estimate of the parameters after the r^{th} iteration, on the $(r+1)^{th}$ iteration, the E-step of the EM algorithm involves the formulation of Q-function as described in Expectation Maximisation Algorithm. In our setting it takes the form

$$Q(\theta|\theta^{(r)}) = \sum_{j=1}^{J} \sum_{i=1}^{n} w_{ij}^{(r)} \log(\pi_j^{(r)} \phi_j(x_i, y_i)), \tag{4.2}$$

where

$$w_{ij}^{(r)} = P(z = j|x_i, y_i) = \frac{\pi_j^{(r)} \phi_j^{(r)}(x_i, y_i)}{\sum_{j=1}^{J} \pi_j^{(r)} \phi_j^{(r)}(x_i, y_i)}$$

is the distribution of a latent membership variable or estimate of the posterior probability that the i^{th} observation belongs to the j^{th} component of the mixture after the r^{th} iteration.

At the M-step, we find the maximum likelihood estimates $\theta^{(r+1)}$ and $\pi_j^{(r+1)}$ with respect to the updated Q-function (4.2). For the model probabilities π_j, this amounts to computing the sample proportion

$$\hat{\pi}_j^{(r+1)} = \frac{\sum_{i=1}^{n} w_{ij}^{(r)}}{n} \quad (j = 1, \ldots, 3)$$

To maximise in θ, it is convenient to use the vector notations $\mathbf{p}_i$ for the point (x_i, y_i) and $\mathbf{n}_j$ for the normal vector $(-\sin\alpha_j, \cos\alpha_j)$ to the line model L_j. In this notations, the line model equation becomes

$$\ell_j(\mathbf{p_i}) = \mathbf{n_j}^\top \mathbf{p_i} + \beta_j.$$

Keeping in mind the constraint $\|\mathbf{n}_j\| = 1$, we apply the Lagrange multiplier method and maximise the Lagrangian

$$\mathscr{L}(\theta, \boldsymbol{\omega}) = Q(\theta|\theta^{(r)}) + \sum_j \omega_j(\mathbf{n}_j^\top \mathbf{n}_j - 1).$$

The partial derivative in ω_j reproduces the constraints, viz.

$$\frac{\partial \mathscr{L}}{\partial \omega_j} = \mathbf{n}_j^\top \mathbf{n}_j - 1 = 0 \tag{4.3}$$

Since the Q-function is the sum of separate functions of $\alpha_j, \beta_j, \sigma_j^2$, each summand can be optimised separately. The derivative in β_j gives

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \beta_j} &= -\sum_{i=1}^{n} w_{ij}^{(r)} \frac{\mathbf{p_i}^\top \mathbf{n_j} + \beta_j}{\sigma_j^2} \\ &= -\sigma_j^{-2}\Big((\sum_{i=1}^{n} w_{ij}^{(r)} \mathbf{p_i}^\top)\mathbf{n_j} + \beta_j \sum_{i=1}^{n} w_{ij}^{(r)} \Big)\end{aligned}$$

Equating the derivative to zero, we find the relation between β_j and $\mathbf{n_j}$

$$\beta_j = -\frac{1}{\sum_{i=1}^{n} w_{ij}^{(r)}} \Big(\sum_{i=1}^{n} (w_{ij}^{(r)} \mathbf{p_i}^\top) \Big)\mathbf{n_j}$$

Setting

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} w_{ij}^{(r)} \mathbf{p_i}}{\sum_{i=1}^{n} w_{ij}^{(r)}}$$

to be the weighted sample mean, we get

$$\beta_j = -\boldsymbol{\mu}_j^\top \mathbf{n_j} \tag{4.4}$$

Using this relation in the derivative in $\mathbf{n_j}$, we obtain

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \mathbf{n_j}} &= -\frac{\partial}{\partial \mathbf{n_j}} \Big(\sum_{i=1}^{n} w_{ij}^{(r)} \frac{\ell_j^2(\mathbf{p_i})}{2\sigma_j^2} \Big) + 2\omega_j \mathbf{n_j} \\ &= -\frac{\partial}{\partial \mathbf{n_j}} \Big(\sum_{i=1}^{n} w_{ij}^{(r)} \frac{(\mathbf{p_i}^\top \mathbf{n_j} - \boldsymbol{\mu}_j^\top \mathbf{n_j})^2}{2\sigma_j^2} \Big) + 2\omega_j \mathbf{n_j} \qquad \text{(from (4.4))} \\ &= -\frac{\partial}{\partial \mathbf{n_j}} \Big(\frac{1}{2}\sigma_j^{-2} \sum_{i=1}^{n} w_{ij}^{(r)} ((\mathbf{p_i} - \boldsymbol{\mu}_j)^\top \mathbf{n_j})^2 \Big) + 2\omega_j \mathbf{n_j}\end{aligned}$$

Now let us stack the centered points $(\mathbf{p_i} - \boldsymbol{\mu}_j)$ into matrix $\mathbf{X_j}$ of size $2 \times n$ and weights $\{w_{ij}^{(r)}\}_{i=1}^{n}$ into a diagonal matrix $\mathbf{W}_j$ of size $n \times n$. Then

$$\frac{\partial \mathscr{L}}{\partial \mathbf{n_j}} = -\frac{\partial}{\partial \mathbf{n_j}} \Big(\frac{\mathbf{n_j}^\top \mathbf{X_j} \mathbf{W_j} \mathbf{X_j}^\top \mathbf{n_j}}{2\sigma_j^2} \Big) + 2\omega_j \mathbf{n_j}$$

The matrix $\mathbf{X_j} \mathbf{W_j} \mathbf{X_j}^\top$ is a weighted covariance matrix of the points $\mathbf{p}_i$. Let us denote it by $\mathbf{C_j}$ and set the derivative to zero:

$$\frac{\partial \mathscr{L}}{\partial \mathbf{n_j}} = -\mathbf{C_j} \mathbf{n_j} \frac{1}{\sigma_j^2} + 2\omega_j \mathbf{n_j} = 0$$

i.e.,

$$\mathbf{C_j} \mathbf{n_j} = 2\sigma_j^2 \omega_j \mathbf{n_j} \tag{4.5}$$

On account of (4.3), we see that $\mathbf{n_j}$ is an eigenvector of matrix $\mathbf{C_j}$ corresponding to the eigenvalue $2\sigma_j^2 \omega_j$.

Finally, the derivative with respect to σ_j produces

$$\frac{\partial \mathscr{L}}{\partial \sigma_j^2} = \sum_{i=1}^{n} w_{ij}^{(r)} \Big(-\frac{1}{2\sigma_j^2} + \frac{\ell_j(\mathbf{p_i})^2}{2\sigma_j^4} \Big)$$

so that

$$\sigma_j^2 = \frac{\sum_{i=1}^{n} w_{ij}^{(r)} \ell_j(\mathbf{p_i})^2}{\sum_{i=1}^{n} w_{ij}^{(r)}} \tag{4.6}$$

Summarizing, we conclude that the optimal point $\widehat{\theta}$ has $\mathbf{n}_j$, β_j, σ_j^2, ad ω_j satisfying the relations (4.3)–(4.6). We observe that the covariance matrix $\mathbf{C}_j$ has two positive eigenvalues, say r_j^+ and r_j^-, and two corresponding eigenvectors. To decide which of them gives the maximum of $\mathscr{L}$, we just calculate the corresponding values. Using the equations (4.5) and (4.6), we derive the equalities

$$\sum_{i=1}^{n} w_{ij}^{(r)} \ell_j(\mathbf{p_i})^2 = \mathbf{n}_j^\top \mathbf{C_j} \mathbf{n}_j = r_j^{\pm}$$

and

$$\sigma_j^2 = \frac{r_j^{\pm}}{\sum_{i=1}^{n} w_{ij}^{(r)}}$$

and

$$\frac{\sum_{i=1}^{n} w_{ij}^{(r)} \ell_j(\mathbf{p_i})^2}{2\sigma_j^2} = \frac{\sum_{i=1}^{n} w_{ij}^{(r)}}{2}.$$

As a result,

$$\begin{aligned} \mathscr{L}(\hat{\theta}, \hat{\omega}) = Q(\hat{\theta}|\theta^{(r)}) &= \sum_{j=1}^{J} \sum_{i=1}^{n} w_{ij}^{(r)} \Big[\log(\pi_j^{(r)}) - \frac{1}{2} \log(2\pi\sigma_j^2) - \frac{\ell_j(\mathbf{p}_i)^2}{2\sigma_j^2} \Big] \\ &= \frac{1}{2} \sum_{j=1}^{J} \sum_{i=1}^{n} w_{ij}^{(r)} \log \frac{(\pi_j^{(r)})^2 \sum_{i=1}^{n} w_{ij}^{(r)}}{2\pi r_j^{\pm}} - \frac{1}{2} \sum_{j=1}^{J} \sum_{i=1}^{n} w_{ij}^{(r)}. \end{aligned}$$

Therefore, we see that the larger value of $\mathscr{L}(\widehat{\theta}, \widehat{\omega})$ and thus of $Q(\hat{\theta}|\theta^{(r)})$ is obtained if we take the smaller eigenvalue r_j^- of the matrix $\mathbf{C}_j$. Thus

- $\mathbf{n}_j$ must be taken to be the normalized eigenvector of $\mathbf{C}_j$ corresponding to the smaller eigenvalue r_j^-;
- $\alpha_j^{(r+1)}$ is set to be the corresponding direction angle of the vector $\mathbf{n}_j$;
- $\beta_j^{(r+1)} = -\boldsymbol{\mu}_j^\top \mathbf{n}_j$ is given by (4.4);
- $\sigma_j^{(r+1)2} = r_j^- / \sum_{i=1}^{n} w_{ij}^{(r)}$.

Then the algorithm is looped with the new value $\theta^{(r+1)}$ until the convergence criterion is met.

4.1.3 Initialisation

Expectation Maximisation is considered to be very init-dependent algorithm. That is why it is very important to do the initialization correctly. We propose two simple approaches for this task.

Random

We start with the most primitive way of initialization: random initialization. Two points per each model are randomly sampled from the set input data. Then the initial models are considered valid if any two of the line models do not overlap to some numerically small level. In case of overlapping the random process is repeated. One can argue, that this approach can create very bad models from the outlier points, but on average it is working as good as more sophisticated approaches 6.3.

Spectral clustering

A more comprehensive approach tries to incorporate the structural patterns in the input data to give a good initial setup for the algorithm. Minimum of 3000 or all unique lines are generated from pairs of input points. Let us consider M being the number of these lines and N being the number of points in the data. In the next step the points are considered to have a connection if they have a line to which both of them have distance lower than predefined outlier threshold. Based on it the matrix of similarity of this graph is built and is fed into spectral clustering algorithm with the specified 3 as a number of clusters. The output are the hard labels of clusters per each point. This data is then used to fit each of the lines to the belonging points using RANSAC algorithm [12].

4.1.4 Outlier Class Modeling

The circles provided from real world images are very noisy. That is why it is very important to make the algorithm noise tolerant. In our work we choose a very simple way to incorporate outlier detection into the pipeline.

We decided to specify the outlier class as an additional model that has a uniform density over the squared area created by the points. The density remains constant throughout the iteration process, but the prior is always updated, starting from the value given at the initialization step. The prior value is given based on the fraction of outlier points after hard labeling using initial models described in Initialisation.

4.1.5 Model Selection

In our work we do not deal with the problem of choosing the correct number of models or the ways to merge similar models. We leave this as an important task for the future work. That said, we restrict our current algorithm to find 3 models plus additional outlier class in the given set of circles. That restriction can lead into oversegmentation in some cases, but works numerically good on average 6.3. Furthermore, it is discussed in Experiments.

4.1.6 Convergence criteria

We decided to follow one of the basic setups for this part of the algorithm. After each iteration the incomplete log-likelihood is calculated and compared with the

value from the previous step. If the relative error between the steps is not significant for several epochs in a row, it means the algorithm achieved its saddle point.

4.2 Kernel density estimation for lambda approximation

Recall that the equation of the radius of a circle with center at $(x_c, y_c)^\top$ that was formed from a line $(l_1, l_2, l_3)^\top$ using division model distortion with parameter of distortion λ is

$$R = \sqrt{\frac{l_1^2 + l_2^2}{4\lambda^2 l_3^2} - \frac{1}{\lambda}} = \sqrt{x_c^2 + y_c^2 - \frac{1}{\lambda}} \tag{4.7}$$

From this dependency λ can be expressed as

$$\lambda = \frac{1}{x_c^2 + y_c^2 - R^2}$$

Multiple samples of λ from all extracted circles of the distorted image will have a significant amount of noise. However, the mode of the distribution of these values will be close to the real value of λ. Example of this distribution for the real image is shown in Figure **??**.

For the estimation of the distribution of λ we use a technique called Kernel Density Estimation [15], that estimates the distribution of λ based on the samples we pass to it. In the experiments we choose the normal density as a kernel and bandwidth value selected from tests on synthetic data as 1.0. Then for the set of λ estimates $\lambda_1, \lambda_2, \ldots, \lambda_i \; i \in (1, n)$ the kernel density function takes form

$$kde(\lambda) = \frac{1}{n\sqrt{2\pi}} \sum_{i=1}^{n} \exp\left\{ -\frac{1}{2}(\lambda - \lambda_i)^2 \right\}$$

From the obtained distribution we can get the mode value and the confidence of the estimation based on the variance of the distribution.

Furthermore, we are able to use this approximation as a filtration step for the input circle centers we pass to EM. Mean-Shift Clustering [33] algorithm with the same bandwidth is used to find the clusters in the input set of λ. Then the the cluster that contain the λ-value chosen at the previous step is kept. The results of the proposed filtering is shown in Figure **??**.

4.3 Curvature Based Circle Consistency Filtering

Having the set of all circles in the image, a useful constraint can be derived based on the formation process of these circles. It follows from (4.7) that $R^2 = x_c^2 + y_c^2 - \frac{1}{\lambda}$, which implies the following statement.

Property 4.3.1. *For division model of radial distortion under assumption that that the center of distortion is located at the origin, the more distant the center of the circle is from the origin the bigger the radius of this circle should be.*

This property holds for the noiseless case. However, not all the circles from the real image will be consistent with it. The circles obtained from the image are extracted in several steps. Each of these steps can introduce additional level of noise to the circle parameters estimates. Besides, some of the circles may be outliers, and would increase the error in the next steps of vanishing point and λ estimation.

To get rid of the outliers we can setup a two step procedure that will do the filtering based on the consistency with the outlined property 4.3.1. Based on the estimated parameters we can sort our sequence of circle estimates into increasing order by the distance from the origin. Then the goal is to find a subsequence that satisfies 4.3.1. We can achieve it by finding the longest non-decreasing by radius subsequence of our sorted sequence. It is a standard task of algorithmic programming and is solved using dynamic programming in $\mathcal{O}(n \log n)$ time.

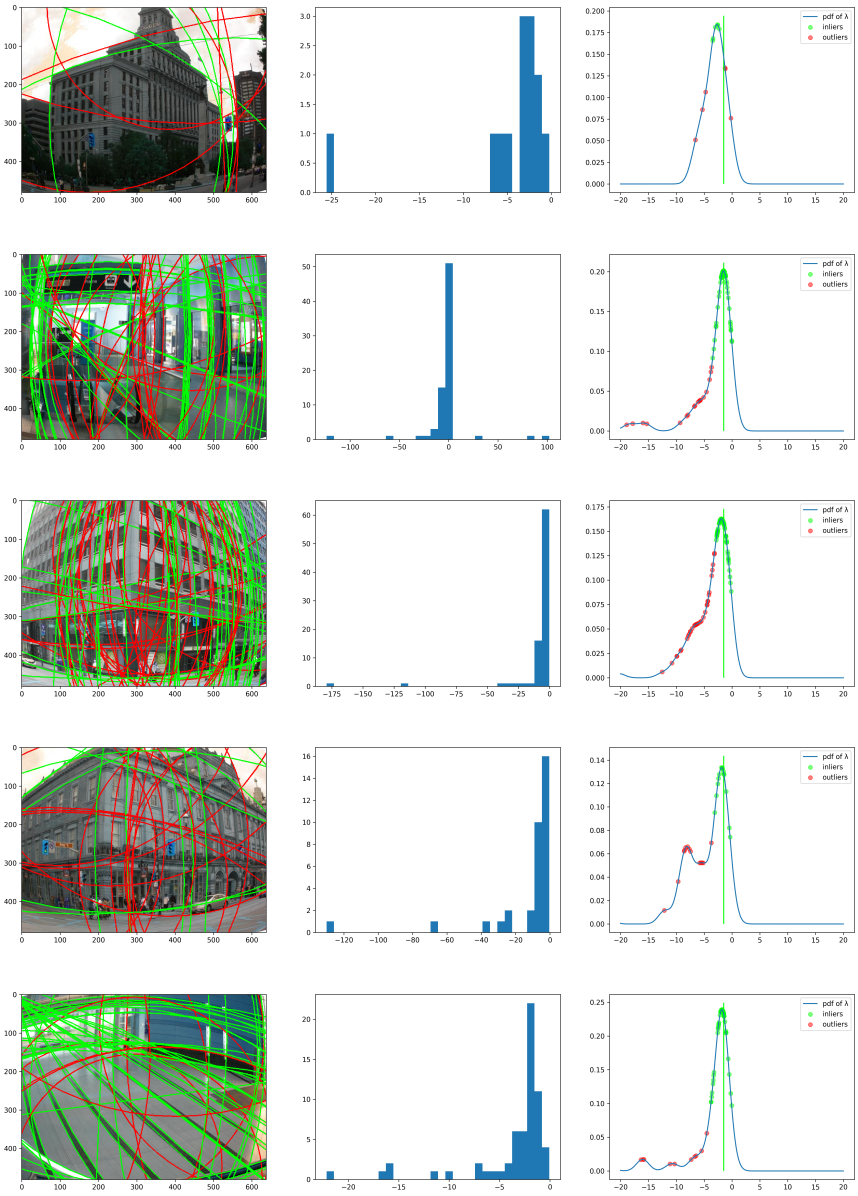


FIGURE 4.1: KDE based λ filtering on Synthetic YORK. Columns from left to right: original image with green and red colors marking the inlier and outlier circles, respectively; histogram of original λ estimates; estimated density of λ using KDE with green points representing inlier estimates of λ, red - outlier and green vertical line - the real value of λ.

Chapter 5

Datasets

The tasks like the one we are working on are hard to be evaluated. You should have the ground truth for the distortion parameter λ and the positions of the three vanishing points. The problem is that it is hard to explicitly label the vanishing point's position or the value of the parameter of distortion lambda by hand. Thus, the only way to annotate the images is to do it in a stepped semi-supervised way. Each circle should be checked by hand and labeled to the vanishing direction it belongs. Then there are multiple approaches to precisely estimate the vanishing points and λ parameter, but they will still have a small fraction of error. Another way to create a dataset for radial distortion is to synthesise it. It is far more easier to label images with no distortion, because you will need to label lines instead of circles. Lines have less degrees of freedom, so their parameter estimation can be more accurate. Then you can create radially distorted images with your own parameter of distortion. One can argue, that there is a problem that we are modeling the distortion following a certain mathematical model, which only approximates the model of real lens distortion, but it is still a great improvement in the amount of data that can be obtained.

In my work the combination of both approaches is used using two datasets: one synthetically generated based on real photos; the other - with natural distortion, but semi-supervisely labeled.

5.1 YORK Urban Database

York Urban Database was proposed by Denis, Elder, and Estrada [11]. The primary purpose of the database was to create a means for comparing different approaches for image auto-calibration. It consists of 102 images (45 indoor, 57 outdoor) of Toronto scenes with pixel resolution of 640×480. The images were taken on a digital camera Panasonic Lumix DMC-LC80.

The image annotation process consisted of three steps: manual annotation of ground truth lines using author-proposed Matlab program; estimation of vanishing points and camera parameters using Gauss Sphere method of Collins & Weiss [7]; re-estimation of those parameters using non-linear search preserving the orthogonality of Manhattan directions. The authors used an ORUA assumption for the simplicity of the camera model. As a result each image in the database has labels for intrinsic camera parameters (focal length, principal point position) and a bunch of line segments with the specified Manhattan scene direction, one of the three orthogonal.

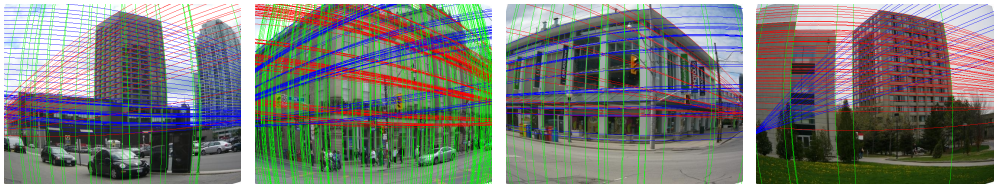


FIGURE 5.1: Example of synthetic images from YORK Urban Database with labeled circles. Circles of the same color correspond to the same vanishing point.

5.2 Synthetic YORK Urban Databse

We adapt the YORK Urban Database [11] for the purpose of our problem in a synthetic way. We create a Synthetic Radially Distorted YORK Urban Database. Following the division model we can add a radial distortion to the existing images with an arbitrary parameter of distortion λ. That creates a big space for experimentation with different level of distortion.

To get the desired result we separately distort images and according lines segments in the normalised space using camera intrinsic matrix. The image is distorted using per point transformation under division model following equation 5.1 obtained as the inverse mapping from equation 3.10 [4].

$$\mathbf{X_d} = \mathbf{X_u} \cdot \frac{1 - \sqrt{1 + 4\lambda \|\mathbf{X_u}\|^2}}{2\lambda \|\mathbf{X_u}\|^2} \tag{5.1}$$

The output image will be much smaller and will have undesired radial borders with empty regions in the corners. To fix both of these issues we scale the obtained image and then cut the borders to eliminate white regions appearance.

Each of line segments we convert into lines. The distorted mapping of those lines are circles as it was described above. Obtained circles are accordingly scaled and shifted to match the transformations applied to distorted image when eliminating distortion artifacts. The set of obtained images with labeled circles are shown in Figure 5.1.

5.3 AIT Dataset

The AIT Dataset of urban images was proposed by Wildenauer and Micusik [31] for comparison of approaches in natural lens distorted images. It contains 102 images of mostly outdoor scenes. All images were taken using a Canon EOS 500D mounting a Walimex Pro 8mm fish-eye lens. The lens has $\approx 170^\circ$ horizontal field of view. Every image was scaled to the resolution of 1188×792 pixels.

For the estimation of focal length and λ, the center of distortion is fixed at the center of the image, zero skew and square pixels are assumed. The standard calibration process results in ground truth estimates $f = 446$ pixels and $\lambda = -0.296$. In addition to this, the estimated positions of vanishing points are provided, that makes this dataset a perfect match for the purpose of evaluation of our approach.

Chapter 6

Experiments

In this section, we want to compare different combinations of the proposed algorithm that can be set up using modifications described in Proposed Method. There were introduced two main extensions: initialization and λ-filtering. Based on this, we want to compare six different approaches:

- **Rand+NoFilt** - random initialisation and no filtration;
- **Spec+NoFilt** - spectral clustering based initialization and no filtration;
- **Rand+KDE** - random initialization and filtration based on Kernel Density Estimation of λ distribution (4.2);
- **Spec+KDE** - spectral clustering initialization and filtration based on Kernel Density Estimation of λ distribution (4.2);
- **Rand+LIS** - random initialization and Curvature Based Circle Consistency Filtering;
- **Spec+LIS** - spectral clustering initialization and Curvature Based Circle Consistency Filtering;

All of them are tested on numerical stability, noise sensitivity and compared on real images.

6.1 Labeling Metric

The main contribution of our work is the method that clusters the passed circle centers into four classes: three vanishing points or outlier class. Besides, this labeling is soft. We need to define some particular metric to compare different modifications of the proposed approach. The task we want to measure can also be referred to as a multi-class classification of the points. The only difference is that at each particular image EM clusters the points and not classifies them. So we need also to solve the correspondence problem between ground truth classes and predicted classes. We choose the correspondence that maximizes the objective function.

The appropriate metric for measuring the quality of multi-class classification is considered to be mean F1-score. In our work, we first start with the formulation of the global confusion matrix for four classes. Then the F1-score can be formulated as

$$F_1 = \frac{1}{4}\sum_{i=1}^{4}\frac{2 \cdot TP_i}{2 \cdot TP_i + FP_i + FN_i}$$

where TP, FP, FN are True Positive, False Positive and False Negative counts obtained from earlier mentioned confusion matrix.

Name	F1-score
Rand+NoFilt	0.9125
Rand+KDE	0.9125
Rand+LIS	0.9172
Spec+NoFilt	**0.9694**
Spec+KDE	**0.9694**
Spec+LIS	0.9681

TABLE 6.1: Numerical Stability scores on YORK Synth Dataset

Name	$\sigma = 0.5$	$\sigma = 1.0$	$\sigma = 2.0$
Rand+NoFilt	0.8419	0.7962	0.7586
Rand+KDE	0.866	0.794	0.7635
Rand+LIS	0.8366	0.7822	0.7065
Spec+NoFilt	0.8926	**0.807**	**0.7681**
Spec+KDE	**0.8941**	0.7954	0.7458
Spec+LIS	0.87	0.8043	0.7021

TABLE 6.2: Noise sensitivity F_1 scores on YORK Synthetic Dataset

6.2 Numerical Stability

This test is used to find out how good the methods are on the data without any noise. In most cases, fully synthetic data is used for this purpose. However, in our work, we decided to harden the task by using YORK Synthetic Dataset. To stay consistent with the idea of the test, we use only ground truth set of circles. That way, we will eliminate any noise in the values, that can appear during the circle extraction since the ground truth circles are formed from manually labeled line segments.

The results for all of the approaches are shown in Table 6.1. It is clear why the results are not perfect. Most of the images do not have the circle representations for three of the Manhattan directions. However, we always over-segment and try to fit all three models. Based on the results, Spectral initialization shows the best performance.

6.3 Noise Sensitivity

We also need to check how three methods deal with different amounts of noise. For this test, we will use the same data as in Numerical Stability. However, we need to divide the inference process for each image into two steps: train and evaluation. The train stage is the process of fitting the EM to ground truth circles with added noise, and evaluation is the process of finding the membership probabilities for each of the same ground truth circles without any noise using fitted EM. In our experiments, we decided to use a white noise sampled from Gaussian with $\mu = 0$ and $\sigma = \{0.5, 1, 2\}$. It is added to measurements of the circles to model the random displacement noise. From results in Table 6.2, we see that all of the modifications degrade with an increase in the level of added noise, but spectral initialization always stands out.

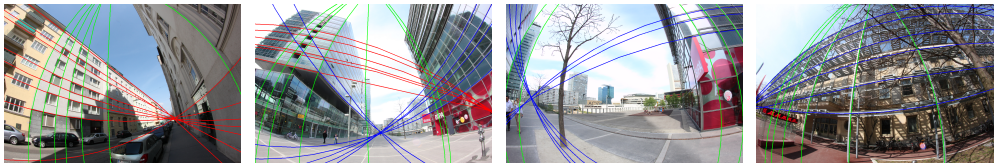


FIGURE 6.1: Example of images from AIT dataset with labeled circles

Name	AIT	YORK
Rand+NoFilt	0.70	0.73
Rand+KDE	**0.76**	0.74
Rand+LIS	0.45	0.77
Spec+NoFilt	0.71	**0.79**
Spec+KDE	0.75	0.66
Spec+LIS	0.64	0.74

TABLE 6.3: Real data F_1 scores on YORK Synthetic Dataset and labeled AIT subset

6.4 Real Data

We decided to evaluate the performance on real data, e.g., using two previously described datasets: York Synthetic and AIT. We follow the same two-step evaluation technique. Except for the EM fitting training data, the circles are extracted from the images in an unsupervised manner: edge detection, edge merging, and circle fitting. Then a metric evaluation is made on ground truth ideal circles.

York Synthetic Dataset has all the information needed to set up the evaluation process: ground truth circles, memberships per each circle, and distorted image. However, for the AIT dataset, we do not have the circles with their memberships labeled. Indeed we have the distortion parameter λ provided with the camera parameters, i.e., focal length and principal point. Our intend is to label some circles with their memberships to vanishing directions. It is quite a hard task to do in distorted space. So we undistorted the image using the provided parameters, labeled line segments with their memberships and transformed them back to distorted space under the division model. Even using such simplification, it is hard and time-consuming work. That is why we labeled only ten chosen images. Some of them are presented in Figure 6.1.

Surprisingly, the best results for AIT dataset appeared to be random initialization and KDE filtering in comparison with spectral initialization and no filtering on YORK dataset (Table 6.3).

Based on all these experiments, it is clear that in most cases, spectral clustering gives a boost. It takes into account the connection between the points and gives a very good initialization for the EM. The filtering step is a harder choice. For the data with the big group of consistent circles, it will give the wanted result. However, it is very dependent on the location and radius of the circles that are not always reliable in natural data setup. Maybe, it can be used as some soft filtering, but not in a way we showed in the experiments. With a moderate amount of noise, it may cut out a large number of circles that are not consistent enough with the chosen mode.

Chapter 7

Conclusions

In this work, we studied the possibility of using the lines of circle centers as a means for the detection of the thee Manhattan Directions. The adaptation of the Expectation Maximisation algorithm was proposed as the solution for this task. The experiments with six different modifications based on different initialization and filtering methods showed that the approach gives good results on real datasets, i.e., YORK Urban Database, AIT Dataset. The method with the outlined spectral clustering initialization achieved the average F_1 classification score of up to 0.79 on real datasets.

The main problem that we faced in our experiments was the over-segmentation of circle centers extracted from the image. Most of the scenes in real datasets do not have the representations of all three vanishing directions, or not all of them fall into the set of extracted circles. The model selection problem is a very difficult one, and we plan to study it in future work.

Bibliography

[1] M. E. Antone and S. Teller. "Automatic recovery of relative camera rotations for urban scenes". In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*. Vol. 2. 2000, 282–289 vol.2.

[2] Michel Antunes, Joao P Barreto, Djamila Aouada, and Björn Ottersten. "Unsupervised Vanishing Point Detection and Camera Calibration from a Single Manhattan Image with Radial Distortion". In: July 2017. DOI: 10.1109/CVPR.2017.708.

[3] João P. Barreto. "A unifying geometric representation for central projection systems". In: *Computer Vision and Image Understanding* 103.3 (2006). Special issue on Omnidirectional Vision and Camera Networks, pp. 208 –217. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2006.06.003. URL: http://www.sciencedirect.com/science/article/pii/S1077314206000701.

[4] C. Brauer-Burchardt and K. Voss. "A new algorithm to correct fish-eye- and strong wide-angle-lens-distortion from single images". In: *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*. Vol. 1. 2001, 225–228 vol.1.

[5] Duane C. Brown. "Close-range camera calibration". In: *PHOTOGRAMMETRIC ENGINEERING* 37.8 (1971), pp. 855–866.

[6] Faisal Bukhari and Matthew Dailey. "Automatic Radial Distortion Estimation from a Single Image". In: *Journal of Mathematical Imaging and Vision* 45 (May 2013). DOI: 10.1007/s10851-012-0342-2.

[7] R. T. Collins and R. S. Weiss. "Vanishing point calculation as a statistical inference on the unit sphere". In: *[1990] Proceedings Third International Conference on Computer Vision*. 1990, pp. 400–403.

[8] J. M. Coughlan and A. L. Yuille. "Manhattan World: compass direction from a single image by Bayesian inference". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 941–947 vol.2.

[9] James Coughlan and AL Yuille. "Manhattan World: Orientation and Outlier Detection by Bayesian Inference". In: *Neural computation* 15 (June 2003), pp. 1063–88. DOI: 10.1162/089976603765202668.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B* 39.1 (1977), pp. 1–38.

[11] Patrick Denis, James H. Elder, and Francisco J. Estrada. "Efficient Edge-Based Methods for Estimating Manhattan Frames in Urban Imagery". In: *Computer Vision – ECCV 2008*. Ed. by David Forsyth, Philip Torr, and Andrew Zisserman. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 197–210. ISBN: 978-3-540-88688-4.

[12] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Commun. ACM* 24.6 (June 1981), 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: https://doi.org/10.1145/358669.358692.

[13] Andrew Fitzgibbon. "Simultaneous linear estimation of multiple view geometry and lens distortion". In: vol. 1. Feb. 2001, pp. I–125. ISBN: 0-7695-1272-0. DOI: 10.1109/CVPR.2001.990465.

[14] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511811685.

[15] Ivana Horová. "Kernel Density Estimation". In: *Wiley StatsRef: Statistics Reference Online*. American Cancer Society, 2014. ISBN: 9781118445112. DOI: 10.1002/9781118445112.stat07186. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118445112.stat07186. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat07186.

[16] Jana Kosecka and Wei Zhang. "Video Compass". In: *Proceedings of the 7th European Conference on Computer Vision-Part IV*. ECCV '02. Berlin, Heidelberg: Springer-Verlag, 2002, 476–490. ISBN: 3540437487.

[17] R. Melo, M. Antunes, J. P. Barreto, G. Falcão, and N. Gonçalves. "Unsupervised Intrinsic Calibration from a Single Frame Using a "Plumb-Line" Approach". In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 537–544.

[18] Zhang Mi, Jian Yao, Menghan Xia, Kai Li, Zhang Yi, and Yaping Liu. "Line-Based Multi-Label Energy Optimization for Fisheye Image Rectification and Calibration". In: June 2015. DOI: 10.1109/CVPR.2015.7299041.

[19] James Pritts, Zuzana Kukelova, Viktor Larsson, and Ondrej Chum. "Radially-Distorted Conjugate Translations". In: (2017). arXiv: 1711.11339 [cs.CV].

[20] James Pritts, Zuzana Kukelova, Viktor Larsson, and Ondrej Chum. "Rectification from Radially-Distorted Scales". In: (2018). arXiv: 1807.06110 [cs.CV].

[21] James Pritts, Zuzana Kukelova, Viktor Larsson, Yaroslava Lochman, and Ondřej Chum. *Minimal Solvers for Rectifying from Radially-Distorted Conjugate Translations*. 2019. arXiv: 1911.01507 [cs.CV].

[22] James Pritts, Zuzana Kukelova, Viktor Larsson, Yaroslava Lochman, and Ondřej Chum. "Minimal Solvers for Rectifying from Radially-Distorted Conjugate Translations". In: (2019). arXiv: 1911.01507 [cs.CV].

[23] 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: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 1. 2004, pp. I–I.

[24] Rickard Strand and Eric Hayman. "Correcting Radial Distortion by Circle Fitting". In: Jan. 2005. DOI: 10.5244/C.19.9.

[25] J. Tardif. "Non-iterative approach for fast and accurate vanishing point detection". In: *2009 IEEE 12th International Conference on Computer Vision*. 2009, pp. 1250–1257.

[26] Roberto Toldo and Andrea Fusiello. "Robust Multiple Structures Estimation with J-Linkage". In: *Computer Vision – ECCV 2008*. Ed. by David Forsyth, Philip Torr, and Andrew Zisserman. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 537–547. ISBN: 978-3-540-88682-2.

[27] Aiqi Wang, Tianshuang Qiu, and L.-T Shao. "A Simple Method of Radial Distortion Correction with Centre of Distortion Estimation". In: *Journal of Mathematical Imaging and Vision* 35 (Nov. 2009), pp. 165–172. DOI: 10.1007/s10851-009-0162-1.

[28] Aiqi Wang, Tianshuang Qiu, and L.-T Shao. "A Simple Method of Radial Distortion Correction with Centre of Distortion Estimation". In: *Journal of Mathematical Imaging and Vision* 35 (Nov. 2009), pp. 165–172. DOI: 10.1007/s10851-009-0162-1.

[29] Horst Wildenauer and Allan Hanbury. "Robust camera self-calibration from monocular images of Manhattan worlds". In: June 2012, pp. 2831–2838. ISBN: 978-1-4673-1226-4. DOI: 10.1109/CVPR.2012.6248008.

[30] Horst Wildenauer and Branislav Micusik. "Closed form solution for radial distortion estimation from a single vanishing point". In: Jan. 2013, pp. 106.1–106.11. ISBN: 1-901725-49-9. DOI: 10.5244/C.27.106.

[31] Horst Wildenauer and Branislav Micusik. "Closed form solution for radial distortion estimation from a single vanishing point". In: Jan. 2013, pp. 106.1–106.11. ISBN: 1-901725-49-9. DOI: 10.5244/C.27.106.

[32] Horst Wildenauer and Markus Vincze. "Vanishing Point Detection in Complex Man-made Worlds". In: Oct. 2007, pp. 615–622. ISBN: 978-0-7695-2877-9. DOI: 10.1109/ICIAP.2007.4362845.

[33] Kuo-Lung Wu and Miin-Shen Yang. "Mean Shift-Based Clustering". In: *Pattern Recogn.* 40.11 (Nov. 2007), 3035–3052. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2007.02.006. URL: https://doi.org/10.1016/j.patcog.2007.02.006.