



Mosaicing with Pan-Tilt-Zoom Cameras

Rodrigo Franco Serra

Thesis to obtain the Master of Science Degree in
Electrical and Computer Engineering

Supervisor: Prof. José António da Cruz Pinto Gaspar

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. José António da Cruz Pinto Gaspar
Member of the Committee: Prof. João António Madeiras Pereira

October 2020

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Declaração

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

Acknowledgments

First and foremost, I would like to express my gratitude to Prof. José Gaspar for the excellent academic guidance, supervision, advice, scientific discussions and freedom to pursue my ideas during this period of my degree.

Secondly, I would like to thank my family for their unconditional and unfailing support throughout this academic journey, and without whom this accomplishment would not have been possible. To my friends for continuous encouragement and never ending support, who were always there when I needed them.

Finally, I would also like to thank everyone whose name is not mentioned above but who always believed in me and inspired me to chase the best version of myself. To each and every one of you, thank you.

Abstract

Pan-tilt-zoom (PTZ) cameras become popular over the last years in activities such as traffic monitoring, sports analysis and surveillance systems due to its ability to cover a wide range of views at a high definition over multiple zoom levels. Having an overall perception of a scene become significantly important in such activities and subsequent applications, thus the need for its panoramic background representation.

This dissertation work is focused on the PTZ panoramic image generation process over multiple zoom levels. The work involves studying different camera calibration algorithms, whose results are then used as input of the panoramic transformation step. Performance metrics are established to assess pixel color errors and perceived quality of the given panoramic transformations in the cube and cylinder based representations. It is further proposed, a variance minimization approach aiming at correcting calibration errors, namely the focal length parameters and consequently, improve the final mosaics.

The last part of this work consists of studying and applying different image blending techniques. This step compensates for exposure differences and reduces generated artifacts in the final images. An image blending method is implemented where the final mosaic is reconstructed from its combined gradient. The main mathematical tool used in the blending method is the Poisson partial differential equation with Dirichlet boundary conditions.

The proposed methods allowed to obtain promising results on panoramic image mosaicing.

Keywords

PTZ camera, Panorama, Mosaicing, PTZ camera calibration, Image Blending

Resumo

Nos últimos anos, as câmaras PTZ tornaram-se ubíquas em aplicações como monitorização de trânsito, análise desportiva e vídeo-vigilância, devido à capacidade de cobrirem um cenário alargado com alta definição através dos seus múltiplos níveis de zoom. A perceção generalizada de um cenário é importante nestas atividades e consequentes aplicações, motivando investigação e desenvolvimento na criação de representações panorâmicas de *background*.

O trabalho desenvolvido nesta dissertação foca-se no processo de geração de imagens panorâmicas em câmaras PTZ a múltiplos níveis de zoom. São estudados vários algoritmos de calibração de câmara. Os resultados de calibração são utilizados no processo de transformação em imagem panorâmica, sendo consideradas métricas de performance de modo a avaliar erros na cor dos pixéis, bem como avaliar a qualidade dos mosaicos finais obtidos nas representações panorâmicas do tipo cubo e cilindro. Também é proposto, um processo de minimização da variância de *pasting* com o objetivo de corrigir erros obtidos no processo de calibração, nomeadamente nos parâmetros de distância focal e, consequentemente, melhorar os resultados obtidos nos mosaicos finais.

Na última parte deste trabalho são testadas várias técnicas de *blending* de imagens. O processo de blending compensa diferentes níveis de exposição entre imagens e reduz artefatos gerados nas imagens finais. É implementado um método de *blending* de imagem onde o mosaico final é construído a partir da combinação do gradiente das várias imagens, utilizando como ferramenta matemática principal a equação diferencial parcial de Poisson com condições fronteira Dirichlet.

Os métodos propostos permitiram obter resultados promissores no processo de *mosaicing*.

Palavras Chave

Câmara PTZ, Panorâma, *Mosaicing*, Calibração de câmara PTZ, *Blending* de imagens

Contents

1	Introduction	2
1.1	Related Work	3
1.2	Problem Formulation	4
1.3	Thesis Structure	5
2	Background	6
2.1	Camera Model	7
2.1.1	Intrinsic Parameters	7
2.1.2	Extrinsic Parameters	10
2.1.3	Radial Distortion	11
2.1.3.A	Simplified model	12
2.1.3.B	Division model	12
2.2	Panoramic Image Generation Process	13
2.3	Image Blending	15
2.3.1	Poisson Blending	16
3	Pan-Tilt-Zoom Camera Modeling	20
3.1	Pan-Tilt-Zoom Camera	21
3.1.1	Direct Kinematics	21
3.1.2	Inverse Kinematics	22
3.1.3	Zoom and Rotation Transformations	23
3.2	Camera Calibration	24
3.2.1	Matlab Calibration Toolbox	25
3.2.2	Image Based Calibration	26
3.2.3	Automatic Calibration Method	28
4	Pan-Tilt-Zoom Camera Panoramic Image Generation Process	30
4.1	Panoramic Transformation	31
4.1.1	Cube based representation	32
4.1.1.A	Back-Projection and Identification of the Correct Cube Face	32

4.1.1.B	Re-projection to Cube Face	34
4.1.2	Cylinder based representation	35
4.2	Panorama Construction Assessment Metrics	36
4.2.1	Variance and Standard Deviation of Pasting	37
4.2.2	Assessment based on the Structural Similarity Index	38
4.3	Optimize Mosaic Alignment	40
4.4	Seamless Mosaicing by Image Blending	41
4.4.1	Average and Median	41
4.4.2	Linear and Hard Blending	42
4.4.3	Blending based on a Gradient Approach	43
5	Experimental Results	45
5.1	Dataset	46
5.2	Feature-Points based Calibration	47
5.3	Colour based Calibration and Panoramic Imaging	50
5.3.1	Cube Results	50
5.3.2	Cylinder Results	56
5.4	Image Blending Results	59
5.4.1	Cube	59
5.4.2	Cylinder	62
6	Conclusion	63
6.1	Conclusions	64
A	Appendix	69
A.1	Dataset	70
A.2	Panoramic Transformation	72
A.2.1	Cube Results	73
A.2.1.A	Variance Minimization Results	74
A.2.2	Cylinder Results	76
A.3	Image Blending Results	77
A.3.1	Cube	77
A.3.2	Cylinder	80

List of Figures

2.1	3D representation of the pin-hole camera model.	8
2.2	2D representation of the pin-hole camera model.	9
2.3	Conversion from metric coordinates to pixels.	9
2.4	Distinct world and camera reference frame.	10
2.5	Different types of radial distortion.[reprinted from [1]].	11
2.6	Traditional panoramic image generation process.	13
2.7	Speeded up robust features (SURF) feature detection and matching within two images.	13
2.8	PTZ panoramic image generation process.	15
2.9	The same composition of images are shown in 2.9(a) and 2.9(b), however 2.9(b) underwent a blending process.	15
2.10	Guided interpolation notations. f interpolates f^* in domain Ω , under the guidance vector field $\mathbf{v}$, which might be the gradient field of a source function g . [reprinted from [2]].	17
2.11	Discrete interpolation notations.	18
2.12	A selected region of a source image is pasted in a same shaped region of a target image. Two approaches are considered, the gradient field of the source image is used as the guidance field (image in the middle), and mixed gradients approach (image on the right).[reprinted from [2]].	19
3.1	Two different type of cameras. [Photos by Unsplash].	21
3.2	Camera pan(α) and tilt(β) coordinates.	22
3.3	Tuning of pan and tilt angles.	23
3.4	Matlab calibration algorithm based on a planar moving pattern.	26
3.5	Agapito et al. calibration procedure.	27
3.6	Sequence of images with intersecting Field of view (FOV)s taken with constant pan and tilt steps.	28
4.1	Mapping process depending on pan and tilt coordinates of each image.	31

4.2	Cube structure.	32
4.3	Critical coordinates and critical latitude.	33
4.4	Projection of 3D points onto a cylindrical surface with unit radius $r = 1$	36
4.5	Structural Similarity Index (SSIM) metric between the reference image 4.5(a) and modified versions of the original image.	39
4.6	Definition of Ω and $\delta\Omega$	44
5.1	Axis P5512 PTZ Camera [by Axis Communications AB on Axis website].	46
5.2	Dataset zoom level 20%.	47
5.3	Calibration approach.	48
5.4	Calibration results.	50
5.5	Results for a cube's face resolution of $N \times N$ where $N = 2000$ pixels. This results correspond to grayscale images.	51
5.6	Optimized focal length parameters. 5.6(a) and 5.6(b) illustrate the results obtained for the cube based representation.	53
5.7	Figures 5.7(a) and 5.7(c) illustrate the minimized variance of pasting results. Figures 5.7(b) and 5.7(d) illustrate the difference between minimized and not-minimized results.	54
5.8	Zoom level 20% (cube).	55
5.9	Zoom level 40% (cube).	55
5.10	Minimized variance, zoom level 20% (cube).	55
5.11	Minimized variance, zoom level 40% (cube).	55
5.12	Results for a cylinder's resolution of $N \times N$ where $N = 2000$ pixels. This results correspond to grayscale images.	57
5.13	Automatic calibration and estimated odometry, zoom level 10% (cylinder).	58
5.14	Automatic calibration and estimated odometry, zoom level 30% (cylinder).	58
5.15	Automatic calibration and estimated odometry, zoom level 50% (cylinder).	58
5.16	Mosaic quality measures regarding the multiple blending approaches (cube).	59
5.17	Blending results, zoom level 20% (cube).	60
5.18	Blending results, zoom level 40% (cube).	60
5.19	Mosaic quality measures regarding the multiple blending approaches (cylinder).	61
5.20	Blending results, zoom level 50% (cylinder).	62
A.1	Dataset zoom level 1%.	70
A.2	Dataset zoom level 10%.	70
A.3	Dataset zoom level 30%.	71
A.4	Dataset zoom level 40%.	71

A.5 Dataset zoom level 50%.	72
A.6 Dataset zoom level 60%.	72
A.7 Zoom level 1% (cube).	73
A.8 Zoom level 10% (cube).	73
A.9 Zoom level 30% (cube).	73
A.10 Zoom level 50% (cube).	74
A.11 Zoom level 60% (cube).	74
A.12 Minimized variance, zoom level 1% (cube).	74
A.13 Minimized variance, zoom level 10% (cube).	75
A.14 Minimized variance, zoom level 30% (cube).	75
A.15 Minimized variance, zoom level 50% (cube).	75
A.16 Minimized variance, zoom level 60% (cube).	75
A.17 Automatic calibration and estimated odometry, zoom level 1% (cylinder).	76
A.18 Automatic calibration and estimated odometry, zoom level 20% (cylinder).	76
A.19 Automatic calibration and estimated odometry, zoom level 40% (cylinder).	76
A.20 Automatic calibration and estimated odometry, zoom level 60% (cylinder).	77
A.21 Blending results, zoom level 1% (cube).	77
A.22 Blending results, zoom level 10% (cube).	78
A.23 Blending results, zoom level 30% (cube).	78
A.24 Blending results, zoom level 50% (cube).	79
A.25 Blending results, zoom level 60% (cube).	79
A.26 Blending results, zoom level 1% (cylinder).	80
A.27 Blending results, zoom level 10% (cylinder).	80
A.28 Blending results, zoom level 20% (cylinder).	81
A.29 Blending results, zoom level 30% (cylinder).	81
A.30 Blending results, zoom level 40% (cylinder).	82
A.31 Blending results, zoom level 60% (cylinder).	82

List of Tables

4.1	Setting the correct cube face.	34
4.2	Finding correct cube face through back-projected coordinates.	34
4.3	Rotation matrices associated with each cube face.	35
5.1	Calibration results.	48
5.2	Calibration results(2).	49
5.3	Focal length parameters variations.	52
5.4	Focal length variations on the automatic calibration and estimated odometry results. . . .	56

List of Algorithms

2.1	Automatic Panorama Stitching [3]	14
2.2	Grassfire Transform [4]	16

Acronyms

DLT	Direct Linear Transformation
FOV	Field of view
IP	Internet Protocol
ISR	Institute for Systems and Robotics
PTZ	Pan-tilt-zoom
POE	Power-over-Ethernet
RANSAC	Random sample consensus
RGB	Red, Green and Blue
SIFT	Scale-invariant feature transform
SSIM	Structural Similarity Index
SURF	Speeded up robust features

1

Introduction

Contents

1.1	Related Work	3
1.2	Problem Formulation	4
1.3	Thesis Structure	5

In a world where human life and its dependency on technology grows day by day, computer vision has been providing multiple answers and solutions for existing problems. Public security and traffic management are examples of problems assisted nowadays by computer vision algorithms.

As an alternative to traditional fixed cameras, the use of Pan-tilt-zoom (PTZ) cameras has been increasing over the years in multiple areas such as TV broadcasting, sports analysis, surveillance systems, among others. The orientation of PTZ cameras can be remotely controlled as well as its level of zoom and Field of view (FOV). Moreover, these cameras are usually available at a fair price, making it a tempting resource compared to fixed cameras.

Having full control over PTZ camera odometry and zoom level empowers the possibility of having an overall perception of a scene at a very high resolution, and thus improving the knowledge one has over the whole scene. Although these cameras offer a better solution, their implementation in automated surveillance systems is still restricted due to the insufficiency of automatic schemes and therefore relying on manual labour.

Panoramic images have become significantly important in surveillance tasks as it provides a compact representation of the observed scene from the camera perspective. Having a full representation of the background is crucial in activities such as object tracking and background subtraction. In recent years, PTZ cameras revealed themselves as useful resources in panoramic image mosaicing by taking advantage of the camera ability to cover a whole scene with its pan, tilt and zoom coordinates.

Compared to traditional image stitching techniques formulated as a multi-image matching problem, PTZ panoramic image generation process makes use of pan, tilt and zoom coordinates as input parameters in the panorama construction process. The building process addresses four key topics: camera calibration, background subtraction, panoramic transformation and image blending, according to [5].

This work focus on the development of PTZ panoramic image generation at different levels of zoom, studying different calibration methodologies, panoramic transformations and image blending techniques.

1.1 Related Work

In recent years PTZ cameras have been an active case study in activities such as traffic monitoring [6] and surveillance [7]. Moreover, studies over cooperative mapping of multiple PTZ cameras and PTZ panoramic image generation process have been performed in [8] and [5, 9].

Most applications require camera calibration as camera parameter estimation is a fundamental step. There are several documented methodologies on camera calibration as they are usually separated into two categories: automatic or non-automatic. This last one applies to calibration algorithms depending on user assistance.

In [10], Bouguet proposed estimating camera parameters by changing the orientation of a chess-

board positioned in front of a camera. In [11], Aziz and Karara established a calibration algorithm named Direct Linear Transformation (DLT) in which the camera projection matrix is estimated through the resolution of a linear system on the matrix entries, relying on a group of 3D points and the matching 2D points. Both methodologies are classified as non-automatic calibration algorithms.

In [12] the authors introduced an automatic calibration algorithm for fixed cameras, and later Agapito et al. [13] presented an automatic calibration for rotating and zooming cameras. Both algorithms estimate camera parameters without non-linear optimizations and with reasonable results.

In [9], Sinha and Pollefeys suggest an automatic calibration method using non-linear optimization of re-projection errors. Their algorithm used a sequence of images taken with a PTZ camera from which its parameters were estimated. In [14], a similar strategy was followed but with a reduced computational burden. The possibility of using different types of image sequences in the calibration process, regarding the camera FOV, is also available.

PTZ cameras walk hand in hand with Panoramic image mosaicing. This process allows a scene to be fully represented in a condensed structure, as a result of combining several single views. Traditional image stitching methodologies [3] set the problem as multi-image matching problem, using invariant local features as Scale-invariant feature transform (SIFT) [15] to look for matches between images. On the other hand, the PTZ panoramic image generation process makes use of camera orientation and zoom level on the panorama construction. Among the most common structures used for panoramic transformation are the cube based representation [9, 14, 16], sphere [17] and cylinder [5, 17].

Image blending is an important step in the panoramic image generation process. It compensates exposure differences and reduces artifacts created from small misalignments, fusing the overlapping areas within different images according to [5]. In [17] a spatially-weighting of image pixels is considered whereas in [18], a frequency-adaptive width is used by adopting a band pass (Laplacian) pyramid. The transition widths are set as a function of the pyramid level.

This idea is extended in a multi-band blending approach suggested in [3], although very ineffective on large datasets. In [5], an highly efficient multi-band blending process is presented, in contrast to what happens in [3]. As an alternative forthright method, for cases where the scene texture is well modelled as a single band, [2, 19] present gradient based approaches.

1.2 Problem Formulation

This thesis focuses on the study of the PTZ panoramic image generation process at distinct zoom levels, addressing the multiple steps of the building process. For this purpose, a dataset over different zoom levels is acquired from which a sequence of mosaics are then built. Each mosaic is associated with a distinct zoom level.

Mosaicing with PTZ cameras relies on camera orientation and calibration. Thus three calibration algorithms are considered: two classified as automatic and one non-automatic. The automatic calibration algorithms also allow the estimation of the pan and tilt angles associated with each image, and therefore the direct comparison between the camera and estimated odometry. From these three, two calibration methodologies assisted the construction of the panoramas. Both were compared over pasting and visual perception metrics.

A minimization process over the mosaic construction accuracy metrics is also considered by iteratively adjusting camera focal length parameters, and consequently improve the visual perception of the final panorama. The calibration results are used to build panoramas on cube and cylinder based representations.

The final part of this thesis consists in improving the final panoramas over its visual perception. Hence, different image blending techniques are studied. An approach over the gradient domain is proposed based in state-of-the-art methodologies, in contrast to image pixel weighting techniques also mentioned in this work.

1.3 Thesis Structure

Chapter 1 introduces the topic to be studied, as well as a brief discussion over the state-of-the-art and proposed approach. Chapter 2 addresses the pin-hole camera model and introduces traditional panoramic image generation processes in addition to state-of-the-art image blending techniques. Chapter 3 presents PTZ cameras and a sequence of camera calibration methodologies. Chapter 4 describes the PTZ panoramic image generation process, starting with the panoramic transformation step. A group of mosaic construction accuracy metrics are also introduced, followed by a minimization process over the variance of pasting. The chapter ends with the studied image blending techniques. Chapter 5 illustrates the experimental results on the proposed and implemented methodologies, and, lastly Chapter 6 outlines all the work developed in this thesis in addition to future work.

2

Background

Contents

2.1	Camera Model	7
2.2	Panoramic Image Generation Process	13
2.3	Image Blending	15

This chapter firstly addresses the chosen camera model and its specification, followed by state of the art methodologies associated with panoramic image generation processes. The various steps that are part of such process are introduced, finishing with a brief overview of image blending (also called feathering) techniques.

2.1 Camera Model

Among the different possible representations of a camera, the pin-hole camera model is the one selected for this work, as a trade-off between geometry accuracy and low complexity modelling. The 3D projective space is mapped to the 2D projective plane, meaning that a 3D point $M = [X \ Y \ Z \ 1]^T$, using homogeneous coordinates, is mapped to its corresponding 2D image point $m = [u \ v \ 1]^T$. This transformation is established by a 3×4 rank-3 projection matrix P , denoted as camera projection matrix. Their relation is given by

$$m \sim PM \quad (2.1)$$

where $\sim$ means equal up to scale factor. P can be decomposed in

$$P = K[R \ t] \quad (2.2)$$

where K represents the camera intrinsic matrix and R and t build the camera extrinsic matrix. Having an accurate estimate of camera projection matrix, one can estimate the camera extrinsic and intrinsic parameters by decomposing P as suggested in [20].

2.1.1 Intrinsic Parameters

In order to fully understand the structure behind the pin-hole camera model, let figure 2.1 be its 3D representation. Here, let the 3D world be represented by capital letters and the 2D image plane be represented in blue and by lower case letters.

Let $M = [X_M \ Y_M \ Z_M]^T$ be the 3D point associated with the 2D image point $m = [x_m \ y_m]^T$, let C represent the optical center, which is set as the origin of the 3D reference frame, and let c be the principal point, the origin of the 2D image frame. Z is set as the principal axis as it intersects both optical center and principal point.

The mapping of a 3D point to its corresponding 2D point is accomplished through the drawing of an imaginary line that goes from the 3D point to C . When this line intersects the image plane, one has the matching 2D point as can be seen in figure 2.1 for M and m . Considering the 2D representation of the pin-hole model illustrated in figure 2.2, it can be written

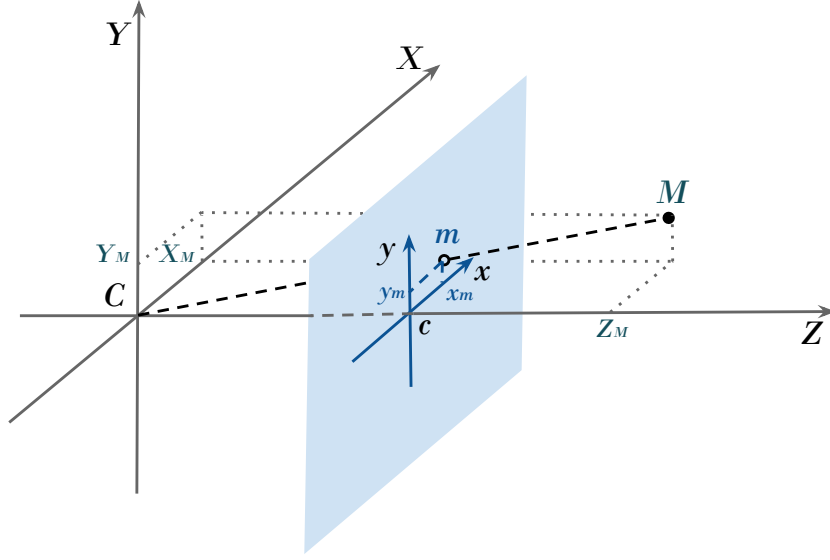


Figure 2.1: 3D representation of the pin-hole camera model.

$$\begin{cases} \frac{x_m}{f} = \frac{X_M}{Z_M} \\ \frac{y_m}{f} = \frac{Y_M}{Z_M} \end{cases} \quad (2.3)$$

where f is the distance between the principal point and the optical center, named as focal length. Generalizing to any point, equation 2.3 reads

$$\begin{cases} x = \frac{X}{Z} f \\ y = \frac{Y}{Z} f. \end{cases} \quad (2.4)$$

Image coordinates x and y are represented in metric units and therefore must be converted to pixels. Considering s_x and s_y as the number of pixels per meter in both horizontal and vertical directions respectively, and u_0 and v_0 as the coordinates of the principal point in pixels, the point can be converted to pixel coordinates through

$$\begin{cases} u = s_x x + u_0 \\ v = s_y y + v_0. \end{cases} \quad (2.5)$$

The image reference frame is not always coincident with the camera principal point, as can be seen in figure 2.3, thus the needed of u_0 and v_0 working as offsets. Substituting equation 2.4 in equation 2.5, and then re-writing this last one in matrix notation using homogeneous coordinates, it leads to

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 & 0 \\ 0 & k_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.6)$$

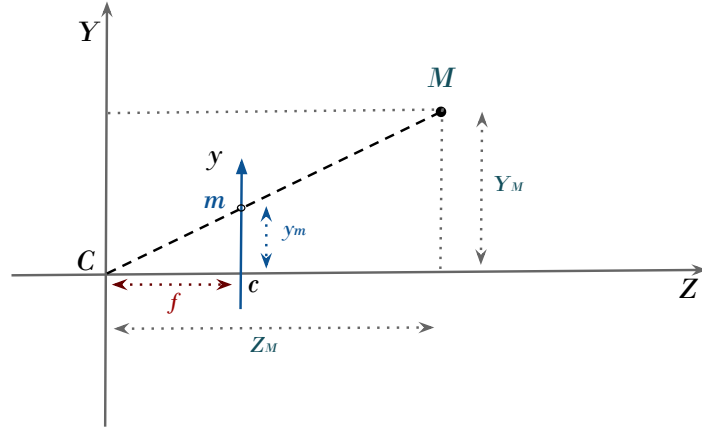


Figure 2.2: 2D representation of the pin-hole camera model.

where λ is a scale factor associated with depth and, $k_u = fs_x$ and $k_v = fs_y$ are camera horizontal and vertical focal length in pixels respectively. These last parameters work as scale factors relating pixels to distances. Considering equation 2.6, the intrinsic matrix is represent as

$$K = \begin{bmatrix} k_u & s & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

where K is 3×3 upper triangular matrix, and s is an additional parameter known as skew. This parameter is a coefficient between the u and v axis often considered as $s = 0$. The intrinsic matrix is fixed for a particular camera zoom level, only varying if the zoom level changes.

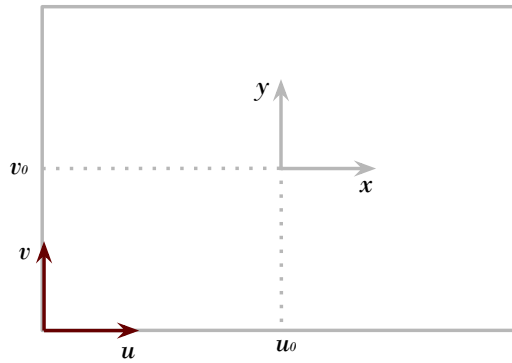


Figure 2.3: Conversion from metric coordinates to pixels.

2.1.2 Extrinsic Parameters

Considering the 3×4 matrix represented at right hand side of equation 2.6, this matrix can be decomposed in two variables R and t . Together they form the camera extrinsic parameters. R is a 3×3 rotation matrix and it describes the orientation of the reference coordinate system with respect to the camera reference frame and t is a column vector, $t = [X_t \ Y_t \ Z_t]^T$, representing the reference coordinate system position in the camera reference frame.

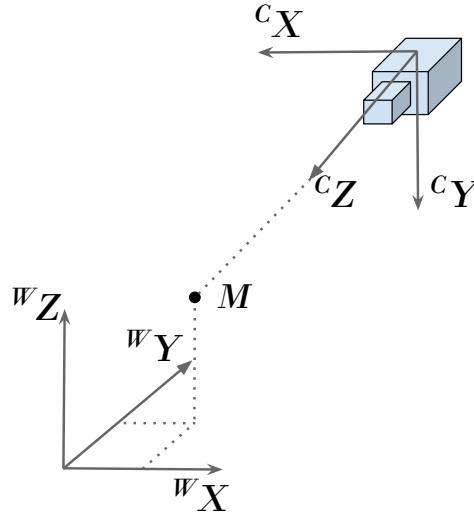


Figure 2.4: Distinct world and camera reference frame.

This type of representation is not very common, as representing both position and orientation of the camera with respect to the reference coordinate system is far more usual. Let R_C be the camera orientation with respect to the reference coordinate system and C the position of the camera in the reference coordinate system. Their relationship with R and t can be described as

$$\begin{bmatrix} R & t \\ \vec{0} & 1 \end{bmatrix} = \begin{bmatrix} R_C & C \\ \vec{0} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} R_C^T & -R_C^T C \\ \vec{0} & 1 \end{bmatrix} \quad (2.8)$$

where $\vec{0}$ represents a row vector of zeros with size 1×3 . Taking into account that the inverse of a rotation matrix is its transpose and the fact that inverting a transformation matrix negates the translation vector, the relationship can be described by

$$R = R_C^T \quad (2.9)$$

$$t = -RC. \quad (2.10)$$

Usually, the reference coordinate system is set coincident with the camera reference frame at its homing position ($t = [0 \ 0 \ 0]^T$), differently from what happens in figure 2.4. In any case, R is represented

as a combination of three possible rotations around its three fixed axis. The simple rotations can be written as

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} R_y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} R_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

where R_x is a rotation around the x axis, R_y around y axis and R_z around the z axis.

2.1.3 Radial Distortion

The pin-hole model is a very accurate model to represent a camera however, it does not take into account radial distortion leading to an effective camera model for most cameras. Here, straight lines in 3D world are rendered as curves lines on the camera sensor, giving a rounded effect to the final image. This distortion decreases when the camera focal length increases as demonstrated in [9].

There are different types of manifestations of radial distortion, but it usually breaks down to two types, the barrel distortion and the pincushion distortion as can be seen in figure 2.5. In the barrel effect, represented in figure 2.5(a), the image bulges from the center looking like it was mapped around a sphere in contrast to what happens in the pincushion effect, where the image bends inward towards its center as illustrated in figure 2.5(c).

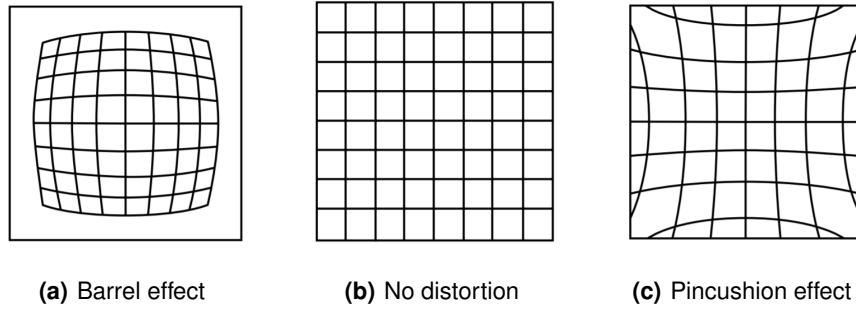


Figure 2.5: Different types of radial distortion.[reprinted from [1]].

Radial distortion is often very complex to model with high precision, as true lens distortion curves are usually not easy to determine, implying the use of look-up tables or high-order models to represent camera radial distortion, as noted by Fitzgibbon [21]. However taking into account the typical computer vision applications, an approximation to the cameras true distortion functions performs as well as the preciser ones. In that sense, two models are presented.

2.1.3.A Simplified model

Let $m_d = [x_d \ y_d]^T$ be a projection of 3D point into an image point, with normalized image coordinates (x_d and y_d are dimensionless). This point deviates from the real undistorted image point, $m_u = [x_u \ y_u]^T$ due to radial distortion. Their relationship can be described as

$$m_d = D(m_u, k_1, k_2) = \mathbf{L}(r)m_u = (1 + k_1r^2 + k_2r^4)m_u \quad (2.12)$$

where r is the radial distance, $r = \sqrt{x_u^2 + y_u^2}$, $\mathbf{L}(r)$ is a radially symmetric distortion factor and the two radial distortion coefficients, k_1 and k_2 .

The radial distance is characterized by the distance from m_u to the center of distortion (x_c, y_c). Considering radial distortion at different zoom levels, represented as z , D^z takes as parameters ($x_c^z, y_c^z, k_1^z, k_2^z$). These parameters however, can be simplified to (k_1^z, k_2^z) assuming the principal point as the center of distortion. The model here described corresponds to a simplified two coefficient version of the one proposed by Heikkila [22].

2.1.3.B Division model

In this model, an undistorted point $m_u = [u_u \ v_u]^T$ (in pixels) is computed from the radially distorted image point $m_d = [u_d \ v_d]^T$ through

$$m_u = \frac{m_d}{(1 + \lambda \|m_d\|^2)} \quad (2.13)$$

where λ is the radial distortion parameter. Unlike the model proposed by Heikkila, this model only has one parameter characterizing radial distortion. This model can also be represented in homogeneous coordinates,

$$\begin{bmatrix} u_u \\ v_u \\ 1 \end{bmatrix} \doteq \begin{bmatrix} u_d \\ v_d \\ 1 + \lambda \|m_d\|^2 \end{bmatrix} \quad (2.14)$$

or considering $e_d = [0 \ 0 \ \|m_d\|^2]^T$, equation 2.14 can be rewritten as

$$m_u \doteq m_d + \lambda e_d. \quad (2.15)$$

2.2 Panoramic Image Generation Process

As studied in [3], image stitching is a multiple step procedure as illustrated in figure 2.6. The first step, feature detection, selects points of interest within the images and finds correspondences between them.

Traditional feature matching methodologies relied on Harris corners detection, whereas the techniques used today focus on invariant local features as SIFT [15] or Speeded up robust features (SURF) features [23]. Figure 2.7 illustrates an example of SURF feature detection and matching between two images taken with the same camera but with different orientations.

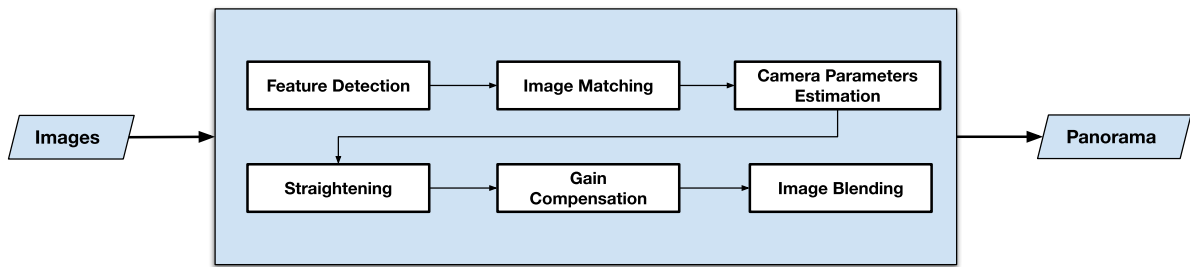


Figure 2.6: Traditional panoramic image generation process.

Following feature detection, all matching images must be found. This process is called image matching. Here, homography estimation is performed using Random sample consensus (RANSAC) [24], from which a set of observed points with outliers is used to perform robust parameter estimation fitting a selected mathematical model. For this particular case RANSAC takes as inputs the correspondences between images, and tries to estimate image transformation parameters. In [3] an additional step is taken where every match is verified using a probabilistic model.

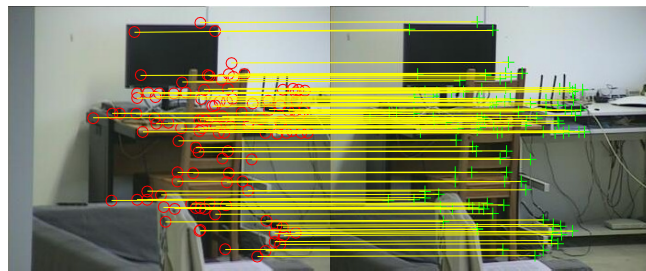


Figure 2.7: SURF feature detection and matching within two images.

The next step addresses camera parameter estimation. In [3] it is suggested a bundle adjustment approach where a set of initial camera parameters estimates go over an optimization process. Taking the initial estimates as inputs, it finds the set of parameters that most precisely predict the coordinates of the observed points in the group of images. This is an essential step as joining pairwise homographies

would neglect several constraints associated with the final panorama representation and would generate cumulative errors.

Images must also go through a straightening process as cameras often have an error associated with its orientation and position upon the acquisition process. Followed by camera parameter estimation and image straightening, luminance differences among the different images should also be addressed. This process is called gain compensation.

The last step focuses on the remaining visible overlapping regions between images. This usually happens due to registration errors, parallax effects, vignetting, in which image intensity decreases towards the periphery of an image, and various other causes. A full implementation of panoramic image stitching is suggested in [3], and thus stated in algorithm 2.1.

Algorithm 2.1: Automatic Panorama Stitching [3]

Input: I_N unordered images

Output: Panoramic Image(s)

begin

 Extract SIFT features from all I_N images;

 Find k nearest-neighbours for each feature using a k-d tree;

for all I_N images **do**

 Select m candidate matching images that have the most feature matches to this image;

 Find geometrically consistent feature matches using RANSAC to solve for the homography between pairs of images;

 Verify image matches using a probabilistic model;

 Find connected components of image matches;

for each connected component do

 Perform bundle adjustment;

 Render panorama using multi-band blending;

Different from panoramic image stitching, PTZ panoramic generation takes advantage of PTZ cameras physical properties to assist the development of such panoramas. In [5] it is taken a four step approach as illustrated in figure 2.8.

After performing camera calibration, the suggested methodology proposes background subtraction. It focuses on blurred areas and ghost artefacts generated by the moving objects within the scene during image acquisition. The images are then mapped to final panoramic representation according to its pan and tilt coordinates. This step is termed panoramic transformation. Among the most common representations are the cube, sphere, cylinder and plane.

At last, it deals with image blending similarly to the panorama development process mentioned above.

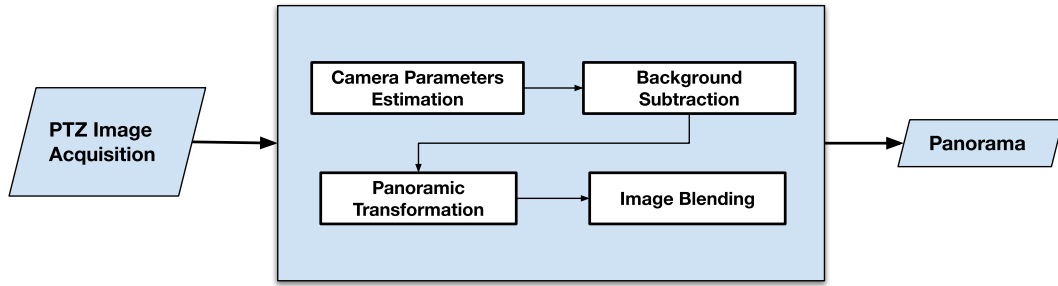


Figure 2.8: PTZ panoramic image generation process.

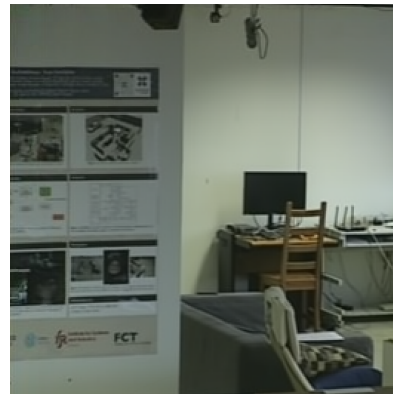
2.3 Image Blending

As previously stated, image blending is the last step in a panoramic image generation process. It aims at compensating for exposure differences and reducing generated artifacts in overlapping regions. These differences in images usually happen due to registration errors, radial distortion effects and other often unmodelled effects, which gain compensation can not handle.

Figure 2.9 illustrates two similar compositions of images. Figure 2.9(b) went over an image blending process contrarily to figure 2.9(a). Looking closely at figure 2.9(a) the overlapping regions are visible in the top left corner of the image, clearly showing the impact of the image blending step.



(a) No blending operation



(b) With blending operation

Figure 2.9: The same composition of images are shown in 2.9(a) and 2.9(b), however 2.9(b) underwent a blending process.

A possible implementation for this type of process is spatially-weighting of image pixels, where pixels at the center of the image have higher weights and pixels at the border have smaller weights. Traditional methodologies varies linearly from 0 at the boundaries to 1 at the center [5]. Algorithms such as grassfire transform (see algorithm 2.2) addresses this type of problem, as it computes the distance from a pixel

Algorithm 2.2: Grassfire Transform [4]

Input: Binary mask

Output: Grassfire transform

begin

 Initialize grassfire transform as a zero matrix with same size of the mask;

for each row in mask left to right **do**

for each column in mask top to bottom **do**

if pixel is in region **then**

 set transform pixel to 1 + minimum value of the north and west neighbours;

for each row in mask right to left **do**

for each column in mask bottom to top **do**

if pixel is in region **then**

 set transform pixel to min(value of the pixel, 1 + minimum value of the south and east neighbours);

to the image border.

A different type of image blending approach is presented in [18]. This method first decomposes images into a sequence of band-pass filtered component images and later, each group of images associated with a specific spatial frequency band are gathered into band-pass mosaic. As mentioned in [18], each component images are joined using a weighted average within a transition zone which is proportional in size to the wavelengths represented in the band. Lastly, the final image mosaic is obtained as a sum of all these band-pass mosaic images.

An extension of this idea is proposed in [5] and [3] with a multi-band blending technique for panoramic image generation, where each image goes through different blending functions depending on its frequency bands, similarly to the previous method. According to [5] the traditional method suggested in [3] is inefficient with large datasets. Thus, it proposes a fast multi-band blending approach with a lower computational complexity and small memory requirement, without sacrificing the blending accuracy.

Alternatively, one can work on the gradient domain as suggested in [2] and [19]. The first one performs seamless cloning, where a region of interest in a source image is seamlessly imported to a destination image, by solving a minimization problem over the region of interest. [19] takes a similar approach, however the optimization problem is stated and solved under different conditions. Both try to minimize the visibility of the seam between stitched images, as stated in [19].

2.3.1 Poisson Blending

Here, a brief introduction to the Poisson blending approach will be stated as most content was derived from [2]. As previously mentioned, this method undergoes a minimization problem over a specified

region of interest in a target image to seamless paste a common shape region of a source image.

The solution of this minimization problem involves solving a Poisson partial differential equation with Dirichlet boundary conditions, which specifies the Laplacian of an unknown function over the domain of interest, along with the unknown function values over the boundary of the domain.

From a minimization perspective, solving the Poisson equation results in the function whose gradient is the closest, in the L_2 -norm, to some prescribed guidance vector field under given boundary conditions. In that way the reconstructed function interpolates the boundary conditions inwards, while following the spatial variations of the guidance field as closely as possible.

Let S be the image domain, a closed subset of $\mathbb{R}^2$, and let Ω be a closed subset of S with boundary $\partial\Omega$. Let f be an unknown scalar function defined over the interior of Ω , f^* be a known scalar function defined over S minus the interior of Ω and let $\mathbf{v}$ be the guidance vector field, as illustrated in figure 2.10.

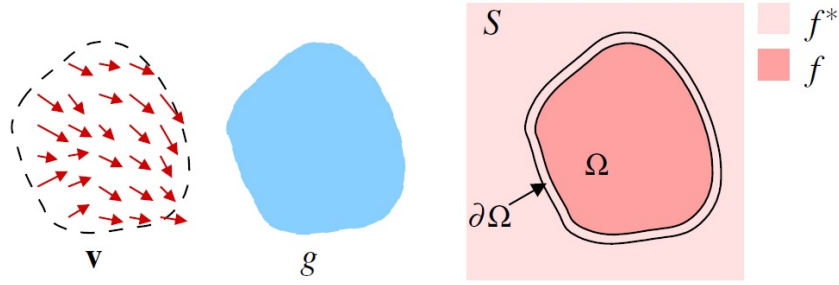


Figure 2.10: Guided interpolation notations. f interpolates f^* in domain Ω , under the guidance vector field $\mathbf{v}$, which might be the gradient field of a source function g . [reprinted from [2]].

The interpolant f of f^* over Ω is the membrane interpolant defined as the solution of the minimization problem

$$\min_f \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (2.16)$$

where $\nabla = [\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$ is the gradient operator. The solution of this minimization problem is given by the following Poisson equation with Dirichlet boundary conditions

$$\Delta f = \text{div} \mathbf{v} \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (2.17)$$

where $\text{div} \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ is the divergence of $\mathbf{v} = (u, v)$, and $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplacian operator.

The same problem can be discretized using the underlying discrete pixel grid. Let S and Ω now become groups of finite points defined on an infinite discrete grid, where S can include all pixels of an image or only a sub-group of them. For each pixel p in S , let N_p be the group of its four adjacent neighbors which belong to S , let $\langle p, q \rangle$ denote a pixel pair such that $q \in N_p$ and let f_p be the value of f at p , as can be seen in figure 2.11.

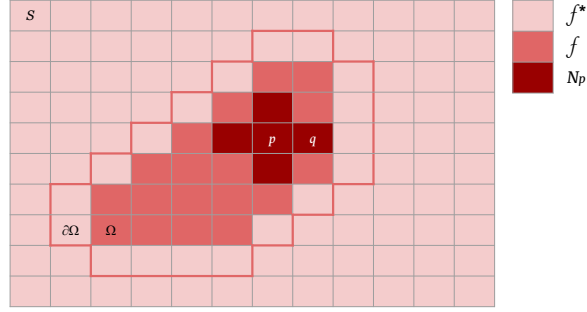


Figure 2.11: Discrete interpolation notations.

The finite difference discretization of equation 2.16 results in the following discrete, quadratic optimization problem

$$\min_{f|_{\Omega}} \sum_{\langle p, q \rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2, \text{ with } f_p = f_p^*, \text{ for all } p \in \partial\Omega \quad (2.18)$$

where v_{pq} is the projection of $\mathbf{v}(\frac{p+q}{2})$ on the oriented edge $[p, q]$. Considering the gradient field of a source image g as the chosen guidance field $\mathbf{v}$, the interpolation is performed under the guidance of

$$\mathbf{v} = \nabla g \quad (2.19)$$

and equation 2.17 is rewritten as,

$$\Delta f = \Delta g \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}. \quad (2.20)$$

From what concerns the discrete implementation, equation 2.19 translates into

$$\text{for all } \langle p, q \rangle, \quad v_{pq} = g_p - g_q. \quad (2.21)$$

The solution of the discretized problem represented in equation 2.18, results in the following linear equations

$$\text{for all } p \in \Omega, \quad |N_p|f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq} \quad (2.22)$$

where the left side of this equation addresses the spatial gradient of the unknown pixel values f_p in Ω within the target image, by taking the sum of differences between f_p and all its neighbors f_q . The first part of the right side addresses the pixels f_q^* at $\partial\Omega$ which are neighbors of the border pixels f_p in Ω . Pixels f_q^* are associated with the target image and therefore take its value. The last part is associated with the gradient field of the source image g , where each pixel value g_p is subtracted to its adjacent

neighbor g_q according to equation 2.21.

Usually $|N_p| = 4$ since most pixels p in Ω have four connected neighbors q . However when Ω contains pixels on the boarder of S , these pixels have a truncated neighbor such that $|N_p| < 4$. In these cases, pixels p in the interior of Ω will not have boundary terms and thus equation 2.22 results in

$$|N_p|f_p - \sum_{q \in N_p} f_q = \sum_{q \in N_p} v_{pq}. \quad (2.23)$$

Equation 2.22 forms a classical sparse, symmetric, positive-definite system. Setting it under the form of

$$Ax = b, \quad (2.24)$$

it can be solved separately for the different Red, Green and Blue (RGB) channels using iterative solvers as Gauss-Seidel iteration with successive overrelaxation, V-cycle multigrid, among others.



Figure 2.12: A selected region of a source image is pasted in a same shaped region of a target image. Two approaches are considered, the gradient field of the source image is used as the guidance field (image in the middle), and mixed gradients approach (image on the right).[reprinted from [2]].

Rather than taking the gradient field of the source image as the guidance field, [2] proposes a mixed gradient approach between the source image and the target image. Since in the first approach, no trace of f^* is kept inside Ω the second approach becomes very useful as combining properties of f^* with g allows the inclusion of objects with holes and preserves a textured background.

The Poisson methodology allows non-conservative guidance fields to be used. Thus, it is selected the stronger variation in f^* or in g , at each point of Ω , using the following guidance field

$$\text{for all } \langle p, q \rangle, v_{pq} = \begin{cases} f_p^* - f_q^* & \text{if } |f_p^* - f_q^*| > |g_p - g_q|, \\ g_p - g_q & \text{otherwise.} \end{cases} \quad (2.25)$$

Figure 2.12 illustrates the differences between both approaches. The mixed gradient method preserves the background of figure 2.12, whereas, adopting the first approach results in a blurred background in the region of pasting.

3

Pan-Tilt-Zoom Camera Modeling

Contents

3.1 Pan-Tilt-Zoom Camera	21
3.2 Camera Calibration	24

This chapter firstly introduces PTZ cameras regarding its physical properties. A general overview is taken over camera direct and inverse kinematics, as well as the transformations associated with zoom and rotation operations. The chapter ends with three calibration methods in order to effectively calibrate PTZ cameras and find its intrinsic and extrinsic parameters.

3.1 Pan-Tilt-Zoom Camera

PTZ cameras [see figure 3.1(a)] became very popular in the last few years in activities such as surveillance and traffic monitoring. Its ability to cover a whole scene at a very high resolution makes it a very tempting resource, compared to traditional fixed cameras, as illustrated in figure 3.1(b). PTZ cameras can change its zoom level and orientation, through its pan and tilt angles, contrarily to fixed cameras orientation restriction.



(a) PTZ camera



(b) Fixed camera

Figure 3.1: Two different type of cameras. [Photos by Unsplash].

This work uses PTZ cameras to study the process of panoramic image generation, another field where these cameras revealed themselves greatly useful. Having a good understanding of how these cameras operate is essential to explore their full potential. In that sense, understanding the camera direct and inverse kinematics becomes very useful for the sake of correctly manipulate these cameras, as well as understanding the different transformations related to the rotation and zoom actions.

3.1.1 Direct Kinematics

Let ${}^wM = [{}^wX_M \ {}^wY_M \ {}^wZ_M \ 1]^T$ be a 3D point in homogeneous coordinates represented in the world reference frame and let ${}^cM = [{}^cX_M \ {}^cY_M \ {}^cZ_M \ 1]^T$ be the same point represented in camera reference frame, as illustrated in figure 2.4. wM can be directly map to cM through

$${}^cM = TM^w \quad (3.1)$$

where T is a 4×4 transformation matrix, similarly to equation 2.8. It can be decomposed in a 3×3 rotation matrix R and a 3×1 column vector t , associated with the world reference frame orientation and position with respect to the camera frame, respectively.

Considering both frames coincident at camera homing position, t becomes a column vector of zeros. Therefore, equation 3.1 is simplified to

$${}^cM = RM^w \quad (3.2)$$

where wM and cM are no longer represented in homogeneous coordinates. R can be written as

$$R = R_C^{-1} = R_C^T \quad (3.3)$$

where R_C stands for camera orientation with respect to the world reference frame. Taking into account that inverting a rotation matrix it is the same as doing its transpose, one has a more intuitive type of representation with R_C than with R . In the particular case of PTZ cameras, R_C will be a function of the pan and tilt angles [$R_C = f(\alpha, \beta)$]. Thus it can be written as

$$R_C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\beta) & -\sin(\beta) \\ 0 & \sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ \sin(\beta) \sin(\alpha) & \cos(\beta) & -\sin(\beta) \cos(\alpha) \\ -\sin(\alpha) \cos(\beta) & \sin(\beta) & \cos(\beta) \cos(\alpha) \end{bmatrix} \quad (3.4)$$

where pan (α) is associated with a rotation over the y axis and tilt (β) is associated with a rotation over the x axis, according to figure 3.2. Both movements follow the right hand rule, as clock wise rotations are considered negative and counter clock wise rotations positive.

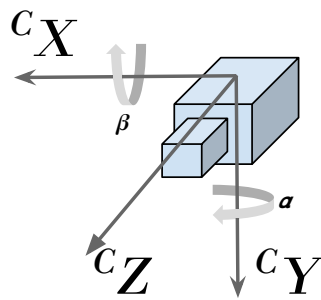


Figure 3.2: Camera pan(α) and tilt(β) coordinates.

3.1.2 Inverse Kinematics

Often in many applications, an object must be centred in an image. This is accomplished with the help of the inverse kinematics. It calculates the camera pan and tilts coordinates to aim the centre of the

camera to the desired world point.

Let ${}^wM = [{}^wX_M \ {}^wY_M \ {}^wZ_M]^T$ be the desired world point in which the camera must be centred, and let both camera and world reference frame be coincident at the neutral position. Firstly, the desired point wM , must be converted to the camera reference frame, cM , according to equation 3.2.

Considering the particular case of PTZ cameras, then the right pan and tilt coordinates must be found. Taking into account that PTZ cameras usually pan first and tilt later, the increments $\delta\alpha$ and $\delta\beta$ over the pan and tilt angles respectively, can be computed through

$$\delta\alpha = \begin{cases} \arctan\left(\frac{{}^cX_M}{{}^cZ_M}\right) & \text{if } {}^cZ_M > 0, \\ \arctan\left(\frac{{}^cX_M}{{}^cZ_M}\right) + \pi & \text{if } {}^cZ_M < 0 \end{cases} \quad \delta\beta = \begin{cases} -\arctan\left(\frac{{}^cY_M}{\sqrt{{}^cX_M^2 + {}^cZ_M^2}}\right) & \text{if } {}^cZ_M > 0, \\ \arctan\left(\frac{{}^cY_M}{\sqrt{{}^cX_M^2 + {}^cZ_M^2}}\right) & \text{if } {}^cZ_M < 0. \end{cases} \quad (3.5)$$

Equations 3.5 can be derived from figure 3.3 as it illustrates the centering of an arbitrary point ${}^cM = [{}^cX_M \ {}^cY_M \ {}^cZ_M]^T$. The relationship between cM and wM can be written as

$${}^wM = R(\alpha + \delta\alpha, \beta + \delta\beta)[0 \ 0 \ \|{}^cM\|]^T. \quad (3.6)$$

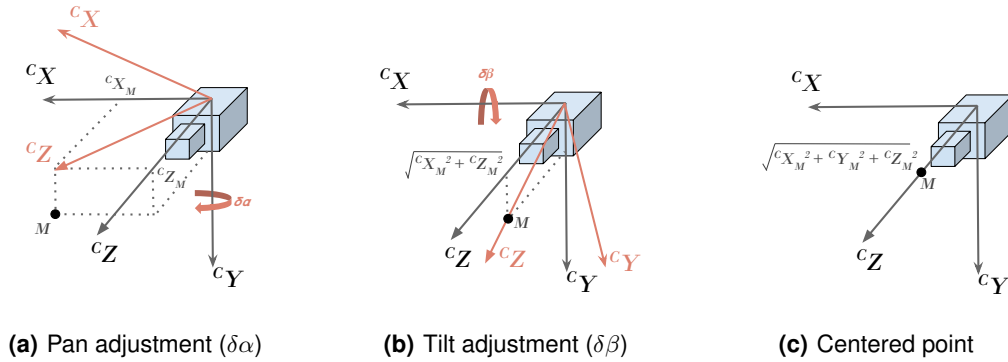


Figure 3.3: Tuning of pan and tilt angles.

3.1.3 Zoom and Rotation Transformations

PTZ cameras can either zoom or rotate, and each one of these movements can be described by homographies. Let m_1 and m_2 be the projections of the same 3D point M in two separate images taken at a different time instant, represented in equations 3.7 and 3.8 respectively.

$$m_1 = K_1[R_1 \ t]M \quad (3.7)$$

$$m_2 = K_2[R_2 \ t]M \quad (3.8)$$

Taking M as the relationship between both points, it can be written from equations 3.7 and 3.8,

$$m_2 = K_2 R_2 R_1^{-1} K_1^{-1} m_1, \quad (3.9)$$

considering that the reference frame is coincident with the camera frame at its homing position, $t = [0 \ 0 \ 0]^T$. Henceforth, two cases might be considered. Assuming that both images were taken at the same zoom level but with different pan and tilt configurations, K_1 will be the same as K_2 , as the intrinsic matrix remains the same for a constant zoom level. Thereby equation 3.9 can be rewritten as

$$m_2 = H_r m_1 = K R_r K^{-1} m_1 \quad (3.10)$$

where $R_r = R_2 R_1^{-1}$ is the relative camera rotation about its projection center within two scenarios, and H_r is the homography that describes the transformation between the two image points for this type of movement.

On the other hand, if both images were taken with a fixed camera orientation, $R_1 = R_2$, and principal point, but at different zoom levels, equation 3.9 is simplified to

$$m_2 = H_z m_1 = K_2 K_1^{-1} m_1 \quad (3.11)$$

as the product of $R_2 R_1^{-1}$ results in the identity matrix. H_z describes the homography in charge of the transformation between the two image points for a zooming camera.

3.2 Camera Calibration

Finding the camera intrinsic and extrinsic parameters is a fundamental step in most computer vision applications given the need to match world points with the corresponding image points. PTZ panoramic image generation process, the main topic studied in this work, requires camera calibration to successfully build the final panoramic view with all the acquired images.

There are different types of camera calibration algorithms as they are usually divided in automatic or non-automatic methods. Non-automatic stands for algorithms where human assistance is required, such as waiving a chessboard in front of a camera, in contrast to automatic algorithms which are usually fully autonomous.

This work studied three calibration methods, one non-automatic and two automatic, as they represent the first layer of the PTZ panoramic image generation process following image acquisition. It was assumed that PTZ cameras could be represented by the pin-hole model described in section 2.1, where

the extrinsic matrix is represented by the pan and tilt coordinates, as illustrated in equations 3.3 and 3.4. At its homing position, both the camera and the world reference frames are considered coincident.

3.2.1 Matlab Calibration Toolbox

Inspired in the work of Jean Ives Bouguet [10], the calibration toolbox for Matlab [25] estimates the intrinsic and extrinsic parameters of a camera, as well as its radial distortion coefficients, using a planar calibration pattern (chessboard) as the key element of the approach.

Having its foundations on Zhang work [26], J.Bouguet calibration takes advantage of the planar surface of the chessboard and thus characterizes it as $z = 0$. Placing the reference frame at one of the corners, all 3D points characterizing the chessboard can be generated knowing the size of each square.

Setting the board at $z = 0$, any 3D point within the chessboard can be described as $M = [X_M \ Y_M \ 0 \ 1]^T$ in homogeneous coordinates. Looking at the camera model in section 2.1, the relationship between the 2D points (m) and the 3D points (M) is expressed as

$$\begin{aligned} m &\sim K[R \ t]M \\ &\sim K[r_1 \ r_2 \ r_3 \ t][X_M \ Y_M \ 0 \ 1]^T \\ &\sim K[r_1 \ r_2 \ t][X_M \ Y_M \ 1]^T \end{aligned} \quad (3.12)$$

where r_i represents the column i of the rotation matrix. r_3 can be removed, considering its null contribution due to $Z_M = 0$. This simplification leads to an homography transformation between m and M described as

$$H = [h_1 \ h_2 \ h_3] \sim K[r_1 \ r_2 \ t]. \quad (3.13)$$

From 3.13 it is written

$$r_1 = K^{-1}h_1 \quad (3.14)$$

$$r_2 = K^{-1}h_2 \quad (3.15)$$

from which it is derived two equations on the columns of H , h_1 and h_2 , considering the properties of rotation matrices (r_1 and r_2 are orthonormal meaning that $r_1^T \cdot r_2 = 0$ and that $\|r_1\| = \|r_2\| = 1$),

$$h_1^T K^{-T} K^{-1} h_2 = 0 \quad (3.16)$$

$$h_1^T K^{-T} K^{-1} h_1 = h_2^T K^{-T} K^{-1} h_2. \quad (3.17)$$

Taking equations 3.16, 3.17 and one homography H , it is possible to write two constraints on the camera intrinsic parameters. Considering that K has six parameters, it is necessary at least three homographies to compute K . Both equations 3.16 and 3.17 do not depend on t , allowing different board poses within image acquisitions. Multiple homographies can be then generated, where each H_j is associated with a particular image j in the dataset.

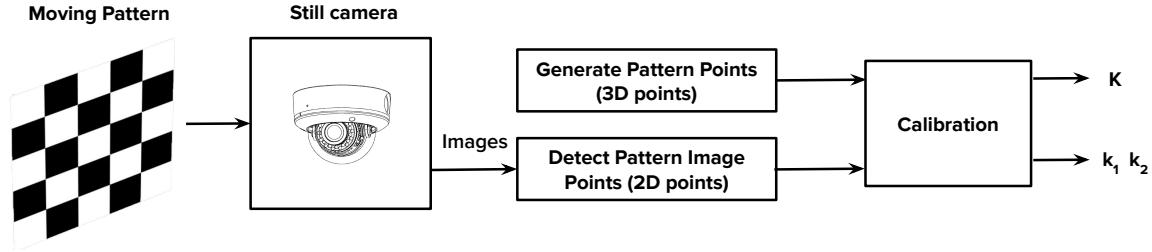


Figure 3.4: Matlab calibration algorithm based on a planar moving pattern.

Matlab calibration toolbox procedure illustrated in figure 3.4 is much similar to J.Bouguet toolbox, but with its differences. Matlab automatically detects the 2D points associated with the chessboard pattern, contrary to what happens in J.Bouguet toolbox where the user must click on the corners of the chessboard.

Matlab automatically generates the 3D points assuming $z = 0$ for the chessboard and then calibrates in two separate steps. First solves for the intrinsic and extrinsic parameters in closed form assuming zero radial distortion, and then estimates all parameters including radial distortion coefficients using nonlinear least-squares minimization (Levenberg-Marquardt algorithm). As initial estimates for the intrinsic and extrinsic matrices, it uses the results acquired in the first step, and then zero for the radial distortion coefficients as mention in [25].

This method is a simple method, however it is not automatic as it needs an individual to constantly change the position of the board with respect to the camera. In addition to that, the pattern must occupy a large part of the image to generate accurate results. These two restrictions reveal themselves as true disadvantages compared to nowadays automatic calibration algorithms.

3.2.2 Image Based Calibration

The calibration procedure illustrated in figure 3.5 and mentioned hereafter, was developed by Agapito et al. [27]. It focuses on a self-calibration methodology for rotating cameras relying on image texture.

Let M be a 3D world point represented in multiple images as $m_i = P_i M$. The relationship between the images points associated with M can be written as

$$m_i = K_i R_i R_j^{-1} K_j^{-1} m_j = K_i R_{ij} K_j^{-1} m_j, \quad (3.18)$$

where $R_{ij} = R_i R_j^{-1}$. Looking at equation 3.18 it can be derived

$$H_{ij} = K_i R_{ij} K_j^{-1}, \quad (3.19)$$

where H_{ij} represents an homography transformation similar to the one described in section 3.1.3. Using SIFT features to find matches $\langle m_i, m_j \rangle$ between pairs of images $\langle i, j \rangle$, H_{ij} can be automatically computed in a least square sense.

Considering the rotation matrix properties ($R_{ij} = R_{ij}^{-T}$) and equation 3.19, one writes

$$K_i^{-T} K_i^{-1} = H_{ij}^{-T} K_j^{-T} K_j^{-1} H_{ij}^{-1} \quad (3.20)$$

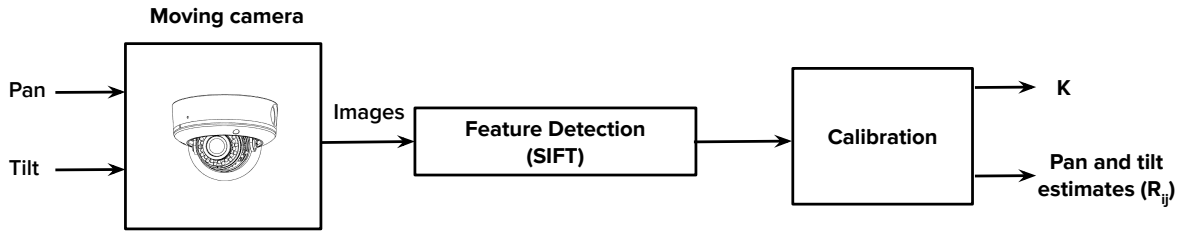


Figure 3.5: Agapito et al. calibration procedure.

which can be simplified considering that all images were taken at a fixed zoom level ($K_i = K_j$). Therefore, equation 3.20 reads

$$w = H_{ij}^{-T} w H_{ij}^{-1}, \quad (3.21)$$

where $w = K^{-T} K^{-1}$ is a symmetric matrix and thus it can be parameterized as a single vector $\vartheta = [w_1 \ w_2 \ w_3 \ w_4 \ w_5 \ w_6]$. With one homography are written six equations from equation 3.21, which can be rewritten under the form of a linear system $A\vartheta = 0$.

Having six equations and six unknowns, the solution can be found using singular value decomposition where $\|A\vartheta\|$ is minimized under the constraint $\|\vartheta\| = 1$. With ϑ , one can define the matrix w and consequently compute K . Multiple constraints can be considered on w to help the estimation of K , e.g., a null skew parameter or known principal point coordinates (u_0, v_0) . Hence, let $s = 0$ and

$$K^{-1} = \begin{bmatrix} \frac{1}{k_u} & \frac{-s}{k_u k_v} & \frac{s v_0 - k_v u_0}{k_u k_v} \\ 0 & \frac{1}{k_v} & \frac{-v_0}{k_v} \\ 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

directly derived from equation 2.7 (K is an upper triangular matrix). w reads

$$w = K^{-T} K^{-1} = \begin{bmatrix} \frac{1}{k_u^2} & 0 & \frac{-u_0}{k_u^2} \\ 0 & \frac{1}{k_v^2} & \frac{-v_0}{k_v^2} \\ \frac{-u_0}{k_u^2} & \frac{-v_0}{k_v^2} & \frac{u_0^2}{k_u^2} + \frac{v_0^2}{k_v^2} + 1 \end{bmatrix} \quad (3.23)$$

from which the camera intrinsic parameters can be calculated. Having computed K , R_{ij} is directly estimated by

$$R_{ij} = K^{-1} H_{ij} K. \quad (3.24)$$

This step becomes very useful in this work, since it provides an estimation of the pan and tilt coordinates of the PTZ camera for each image. Furthermore, it allows a direct comparison between camera odometry and estimated results. This method, unlike the previous one described in section 3.2.1, does not estimate radial distortion coefficients.

3.2.3 Automatic Calibration Method

Inspired on the work of Diogo Leite et al. [14], this methodology for PTZ camera calibration relies on non-linear optimization of re-projection errors. The proposed method stated in [14] is a two step calibration procedure. First both intrinsic and radial distortion coefficients are iteratively estimated at minimum zoom level and then computed for an increasing zoom sequence. The first step starts by capturing images with intersecting FOVs, with constant pan and tilt steps as illustrated in figure 3.6.

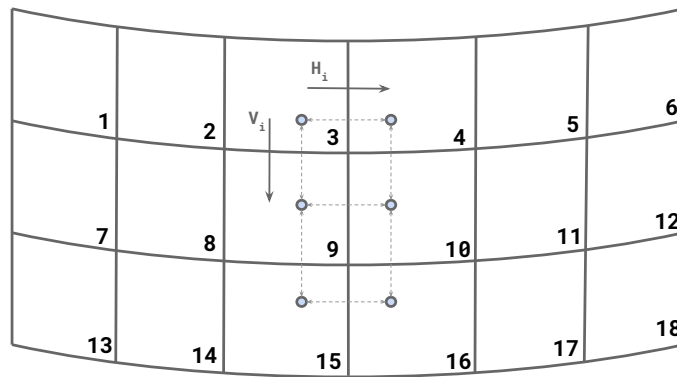


Figure 3.6: Sequence of images with intersecting FOVs taken with constant pan and tilt steps.

All images then go through key feature detection and matching using SIFT algorithm, and later excluded all matches not forming closed loops. RANSAC based homography estimation and non-linear optimization [20] is used to compute homographies between every adjacent vertical images V_i , and be-

tween every adjacent horizontal images H_i (see figure 3.6). From composition of both H_i and V_i , T_i homographies are calculated mapping all I_i images to a chosen reference image I_r .

Considering eventual residual errors accumulated in the homographies computation process, a non-linear optimization of the re-projection errors is applied using Levenberg-Marquardt algorithm. Initializing this minimization problem with the K matrix obtained from Agapito method described in the previous section 3.2.2 and null radial distortion coefficients, $k_1 = 0$ and $k_2 = 0$, the cost function reads

$$(K^{z_0^*}, \mathbf{D}^{z_0^*}) = \arg \min_{K^{z_0}, \mathbf{D}^{z_0}} \sum_{j=1}^m \sum_{i=1}^n \left\| u_i^j - h(K^{z_0} \mathbf{D}^{z_0} (R_i X^j, k_1, k_2)) \right\|^2 \quad (3.25)$$

where $\mathbf{D}$ is a homogenization of D defined on equation 2.12, i.e. transforming a 2-vector into a 3-vector output by adding a third unitary coordinate. h is a dehomogenization function, i.e. $h([a \ b \ c]^T) = [a/c \ b/c]^T$, u_i^j are the selected features for the given image I_i , R_i are the rotation matrices directly obtained from the pan and tilt coordinates as stated in section 3.1.1 and m , n and X^j are the feature count, image count and global feature list of back-projected points respectively, for the given images.

This work uses the first step of the calibration procedure developed in [14] and generalizes it to all levels of zoom. Are acquired different spherical grids of images at different zoom levels where later each grid is submitted to the process previously described. The cost function represented in equation 3.25 is used for each zoom level independently and so it reads,

$$(K^{z_k^*}, \mathbf{D}^{z_k^*}) = \arg \min_{K^{z_k}, \mathbf{D}^{z_k}} \sum_{j=1}^m \sum_{i=1}^n \left\| u_i^j - h(K^{z_k} \mathbf{D}^{z_k} (R_i X^j, k_1, k_2)) \right\|^2 \quad (3.26)$$

where K^{z_k} is the initial estimate for a particular zoom level k obtained with Agapito method.

The second step of the calibration procedure described in [14] uses a sequence of images taken at a fixed pan and tilt coordinates, and the estimates obtained in the first step, $K^{z_0^*}$ and $\mathbf{D}^{z_0^*}$, to compute the remaining $K^{z_k^*}$ and $\mathbf{D}^{z_k^*}$.

The computation process is similar to the first step except for the homography computation. This step becomes unnecessary, considering that the cost function for each zoom level is initialized with the results obtained from the previous zoom level. Worth mentioning that the first level of zoom is only calibrated in the first step.

Although stated, this second step is not considered in this work since each zoom level is calibrated separately using the methodology described in the first step.

4

Pan-Tilt-Zoom Camera Panoramic Image Generation Process

Contents

4.1 Panoramic Transformation	31
4.2 Panorama Construction Assessment Metrics	36
4.3 Optimize Mosaic Alignment	40
4.4 Seamless Mosaicing by Image Blending	41

Unlike the traditional panoramic image generation processes described in section 2.2, the PTZ panoramic image generation process relies on the camera pan and tilt coordinates to build the final panorama. This process is usually divided in four steps, camera calibration, background subtraction, panoramic transformation and image blending as stated in [5].

This chapter addresses the third and fourth steps. The first step, camera calibration, was already described in the previous chapter. The second step, background subtraction, required in [5] due to the moving cars in the scene, is not considered in our case. We assume the indoors environment is static while acquiring the images to build the panorama.

Different evaluation metrics are considered to understand better the results provided by the different calibration algorithms, as well as by both camera odometry and estimated pan and tilt coordinates. An optimization process over the evaluation metrics is also suggested to improve the final result. Regarding the image blending step, different methods are explored. It is further proposed an image blending method based on a gradient approach.

4.1 Panoramic Transformation

A wide scenario can be densely scanned by a small FOV PTZ camera through the acquisition of images where the full range of pan and tilt angles are explored. Gathering all images into a single representation is quite useful to synthesize views of the scenario, which can be later used in applications such as background subtraction, among others.

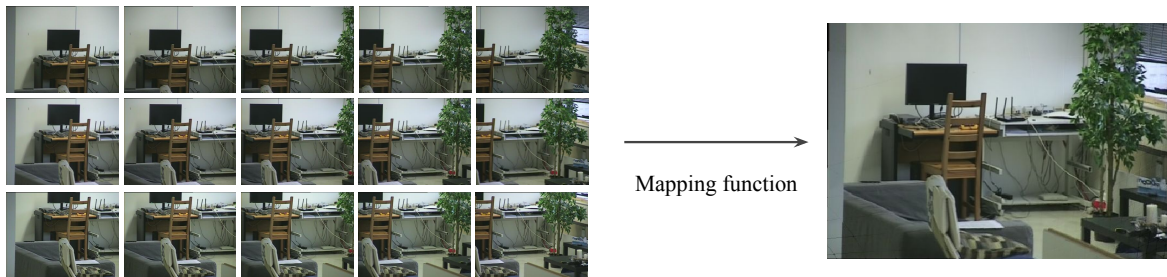


Figure 4.1: Mapping process depending on pan and tilt coordinates of each image.

In the PTZ panoramic image generation process, this transformation takes place in the panoramic transformation step. Here, the images are mapped into a single representation relying on the pan and tilt coordinates of the camera as represented in figure 4.1.

Camera calibration is also a necessary step considering that most mapping procedures depend on the camera intrinsic parameters, e.g. camera focal length. This step is usually performed before the panoramic transformation, as detailed in the previous chapter.

Among the most common representations are the cube based representation studied in the works of [9, 14, 16], the sphere [5, 17], the cylinder [5, 17], and not so common representations as the circle and overlooking panoramas stated in [5]. This work addresses the cube and cylinder based representations.

4.1.1 Cube based representation

The cube based representation, unlike other types of representations such as the cylinder or planar structures, can handle a complete spherical FOV, $360^\circ \times 360^\circ$, with limited memory requirements. Using only six images to completely represent the background and considering its simple mapping to reobtain new viewing directions, make it one of the most efficient types of representations.

Many well known panoramic images generation applications, such as Google Street View, rely on this structure due to its appealing characteristics.

Let the camera reference frame at its homing position be coincident with both the world and cube frame, as represented in figure 4.2. After both the acquisition process, where multiple images are taken at different pan and tilt configurations and camera calibration steps, the mapping procedure into the cube is usually divided into two separate actions.

First, every image point is back-projected and then established which cube face does it belongs to, and lastly, the back-projected image point is now re-projected to the correct cube face.

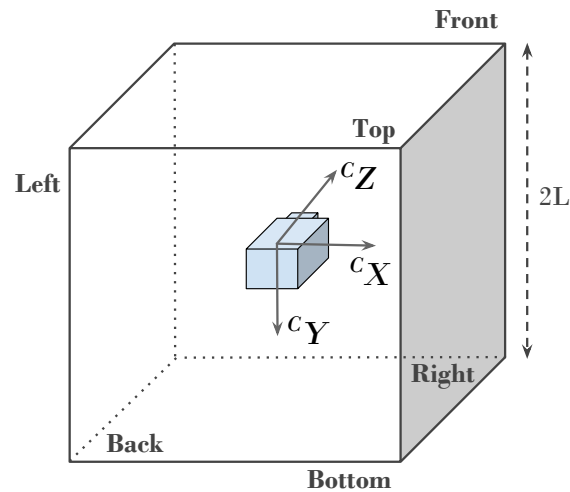


Figure 4.2: Cube structure.

4.1.1.A Back-Projection and Identification of the Correct Cube Face

With both camera intrinsic parameters, K , and camera orientation, R , every 2D point can be directly back-projected to a 3D point through

$$[X_M \ Y_M \ Z_M]^T = (KR)^{-1}m. \quad (4.1)$$

Considering that the given 3D point does not have null X_M or Z_M coordinates, it can be scaled to be on the lateral faces of a cube (which are enlarged to infinity in the cY direction) with a $2L$ edge. This transformation can be written as

$$[X_L \ Y_L \ Z_L]^T = [X_M \ Y_M \ Z_M]^T \frac{L}{\max(|X_M|, |Z_M|)}. \quad (4.2)$$

Let θ denote the azimuth of the back-projection ray starting at $[0 \ 0 \ 0]^T$ and passing through $[X_M \ Y_M \ Z_M]^T$, i.e. $\theta = \text{atan2}(X_M, Z_M)$. Then one may define the critical elevation, $\varphi_L(\theta)$ as

$$\varphi_L(\theta) = \arctan\left(\frac{L}{\sqrt{X_L^2 + Z_L^2}}\right) \quad (4.3)$$

which indicate the transition from the lateral face of the cube to the bottom or top faces, for back-projecting each image point (see figure 4.3). $\varphi_L(\theta)$ can also be computed directly from the back-projected point as

$$\varphi_L(\theta) = \arctan\left(\frac{\max(|X_M|, |Z_M|)}{\sqrt{X_M^2 + Z_M^2}}\right). \quad (4.4)$$

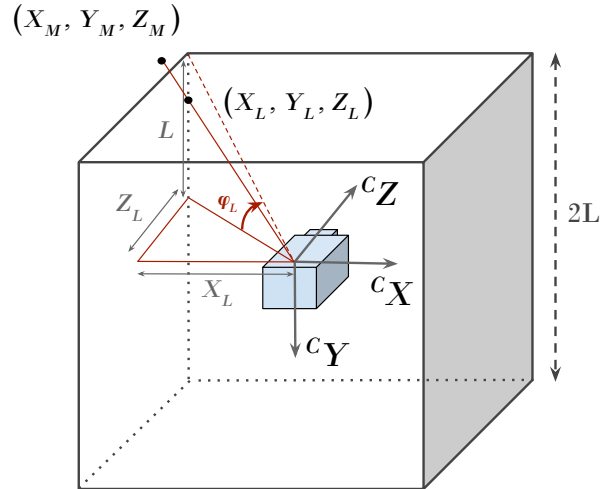


Figure 4.3: Critical coordinates and critical latitude.

Computing the elevation of the back-projection as

$$\varphi = \arctan\left(\frac{-Y_M}{\sqrt{X_M^2 + Z_M^2}}\right) \quad (4.5)$$

allows identifying the cube face. For instance, $|\theta| \leq 45^\circ$ and $|\varphi| < \varphi_L(\theta)$ indicates the front face. The complete set of rules is in the table 4.1.

Cube Face	Condition
Top	$\varphi \geq \varphi_L(\theta)$
Bottom	$\varphi \leq -\varphi_L(\theta)$
Front	$ \varphi < \varphi_L(\theta) \wedge \theta \leq 45^\circ$
Back	$ \varphi < \varphi_L(\theta) \wedge \theta \geq 135^\circ$
Right	$ \varphi < \varphi_L(\theta) \wedge 45^\circ < \theta < 135^\circ$
Left	$ \varphi < \varphi_L(\theta) \wedge -135^\circ < \theta < -45^\circ$

Table 4.1: Setting the correct cube face.

An alternative way to identify which cube face does an image point belong to, is through the analysis of the dominant coordinate of each image point back-projection. For example, when $|Z_M|$ is larger than $|Y_M|$ and $|X_M|$ then it is a back-projection to the cube front face. Computing $\chi = \max(|X_M|, |Y_M|, |Z_M|)$, the rules are listed in table 4.2.

Cube Face	Condition
Top	$\chi \equiv -Y_M$
Bottom	$\chi \equiv Y_M$
Front	$\chi \equiv Z_M$
Back	$\chi \equiv -Z_M$
Right	$\chi \equiv X_M$
Left	$\chi \equiv -X_M$

Table 4.2: Finding correct cube face through back-projected coordinates.

4.1.1.B Re-projection to Cube Face

The cube based representation can be thought as a sequence of six fixed cameras, where each camera is associated with a single cube face. All cameras are positioned in order to share the same origin as the reference frame, $t = [0 \ 0 \ 0]^T$, however with different orientations, R . In terms of specifications, each camera has an equal number of pixels in width and length and cover a FOV of $90^\circ \times 90^\circ$.

Once established the correct cube faces, as described in the previous section, one can map the back-projected image points through a projection matrix

$$P_{WF} = K_F [R_{WF} \ 0_{3 \times 1}]. \quad (4.6)$$

K_F consists in an intrinsic parameters matrix characterizing the resolution (size) of the cube faces and R_{WF} are rotation matrices setting the optical axis orthogonal to the cube faces. Considering a cube face with a resolution of $N \times N$ pixels, K_F is represented as

$$K_F = \begin{bmatrix} (N+1)/2 & 0 & (N-1)/2 \\ 0 & (N+1)/2 & (N-1)/2 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.7)$$

which stands for a perspective camera with an image coordinate system where the top-left pixel corresponds to $(1, 1)$, and a $90^\circ \times 90^\circ$ FOV. R_{WF} is responsible for rotating the 3D points from world reference frame to the correct cube face.

The types of rotations are set in Table 4.3, as they are dependent on the cube face. $I_{3 \times 3}$ stands for the identity matrix, and the shape of the rotations matrices R_x and R_y are illustrated in equation 2.11.

Cube Face	R_{WF}
Top	$R_x(-90^\circ)$
Bottom	$R_x(90^\circ)$
Front	$I_{3 \times 3}$
Back	$R_y(180^\circ)$
Right	$R_y(-90^\circ)$
Left	$R_y(90^\circ)$

Table 4.3: Rotation matrices associated with each cube face.

In brief, an image point m_i is re-projected to a point on a cube face m_{Fi} through

$$m_{Fi} \sim K_F R_{WF} R^{-1} K^{-1} m_i. \quad (4.8)$$

4.1.2 Cylinder based representation

Although it can not represent the complete spherical FOV, namely the north and south pole, it still is a common type of representation among panoramic image generation applications.

Let the camera reference frame at its homing position be coincident with both the world and cylinder reference frame, as illustrated in figure 4.4. To successfully map an image into a cylinder based representation, first, the 2D points must be back-projected into 3D points. This is achieved with the help of the camera intrinsic and extrinsic parameters, as performed in the cube based representation. This step is represented in equation 4.1. Then, the 3D points must be projected onto a cylindrical surface with unitary radius given by

$$[X_L \ Y_L \ Z_L]^T = [X_M \ Y_M \ Z_M]^T \frac{1}{\sqrt{(X_M^2 + Z_M^2)}} \quad (4.9)$$

as illustrated in figure 4.4.

Any point in the cylindrical surface can be parameterized with a height h and an angle θ , known as 3D cylindrical coordinates (h, θ) . The transformation between coordinates is given by

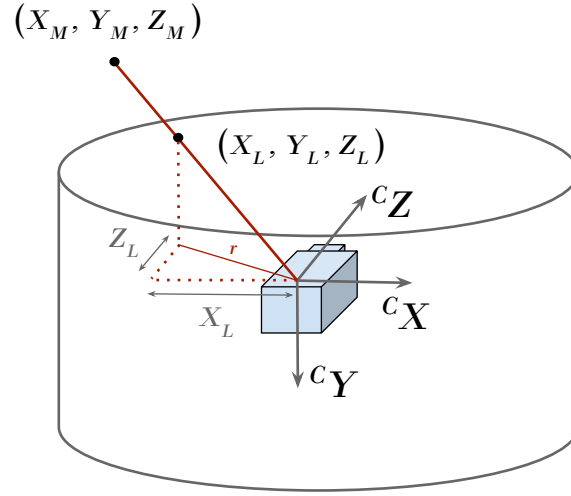


Figure 4.4: Projection of 3D points onto a cylindrical surface with unit radius $r = 1$.

$$[\sin \theta \ h \ \cos \theta]^T = [X_L \ Y_L \ Z_L]^T \quad (4.10)$$

from which, it is derived

$$\begin{cases} \theta = \text{atan2}(X_L, Z_L) \\ h = Y_L. \end{cases} \quad (4.11)$$

The final mapped coordinates in the final panorama are given by

$$\begin{cases} u = f_L \theta + u_L = f_L \text{atan2}(X_L, Z_L) + u_L \\ v = f_L h + v_L = f_L Y_L + v_L \end{cases} \quad (4.12)$$

where f_L is a scaling factor in pixels that directly influences the panorama resolution. This value must be carefully adjusted according to the size of the panorama. u_L and v_L are simply offsets associated with the panorama coordinate system, similarly to the principal point coordinates in the intrinsic matrix.

4.2 Panorama Construction Assessment Metrics

As previously stated, the panoramic transformation step in the PTZ panoramic image generation process relies on camera calibration and the camera pan and tilt coordinates. Different results on camera calibration and odometry data can lead to misalignment between images and an incorrect pasting. Therefore, assessing approaches should be considered to gauge the results provided by different methodologies.

This section introduces three evaluation metrics considered in this work. They assess the accuracy of pixel pasting into the cube and cylinder based representations and the quality of the final image.

4.2.1 Variance and Standard Deviation of Pasting

In the panoramic transformation procedure, multiple images are pasted in the corresponding representations according to its pan and tilt coordinates and camera intrinsic parameters. This pixel to pixel transformation leads to superimposing regions of images, i.e. two images can share a group of pixels in the transformation structure since most of them have intersecting FOVs.

In the presence of a very accurate pasting procedure, each pixel, although shared multiple times, would have the right pixel intensity since all images would be perfectly pasted and aligned. However, due to small errors in the calibration process, odometry data and radiometric differences, the pasting deviates from the expected results.

The variance and standard deviation of pasting evaluate how accurately the pasting procedure was performed by computing the deviation each pixel has from its average intensity value.

Let μ_i be the average pixel intensity of pixel i in the final mosaic, where i is the linear indexing format of pixel (u, v) [pixel $(u, v) = (1, 1)$ corresponds to $i = 1$, $(u, v) = (2, 1)$ to $i = 2$, and so on]. μ_i is given by

$$\mu_i = \frac{1}{K_i} \sum_{j=1}^{K_i} x_{i_j} \quad (4.13)$$

where K_i is the total number of images superimposed under pixel i , and x_{i_j} are the intensity values associated with these images resampled in panoramic coordinates.

μ_i has been computed using an iterative approach where each x_{i_j} is stored and sum to the next $x_{i_{j+1}}$, as the images are pasted in the final panorama. The total number of images pasted in pixel i is also iteratively obtained as the pasting process takes place. Both values are kept in auxiliary cumulative matrices, having the size of the mosaic, which are later used in a pixel to pixel division for non-zero values of K_i . Worth mentioning that equation 4.13 is only valid when $K_i > 0$, i.e. when at least one image has been pasted over i .

The sum of squared intensities $x_{i_j}^2$ is also computed within the pasting process, using a similar approach as the one described for x_{i_j} . The variance of pixel i can be then calculated as

$$\sigma_i^2 = \frac{1}{K_i} \sum_{j=1}^{K_i} (x_{i_j} - \mu_i)^2 = \left(\frac{1}{K_i} \sum_{j=1}^{K_i} x_{i_j}^2 \right) - \mu_i^2 \quad (4.14)$$

where the auxiliary cumulative matrix containing the sums of all $x_{i_j}^2$ is divided by the auxiliary matrix associated with K_i , in a pixel to pixel division for non-zero elements of K_i . These results are later subtracted to the corresponding μ_i^2 .

Given the variance of pasting σ_i^2 of each pixel i , an average value of all variances is consider in order to get an estimate of the variance of pasting associated with a particular mosaic. This metric is given by

$$M_{\sigma^2} = \frac{1}{M} \sum_{i=1}^T \sigma_i^2 \quad (4.15)$$

where M and T are the total number of non-zero K_i elements and the total number of pixels of the final panorama, respectively. This metric can either be calculated separately for the RGB channels or calculated for greyscale images. In the RGB case every auxiliary cumulative variable, apart from K_i , will be a three dimensional variable.

The standard deviation of pasting is also consider as an assessment metric. Considering the variance σ_i^2 , the standard deviation of each pixel i can be computed as

$$\rho_i = \sqrt{\sigma_i^2}. \quad (4.16)$$

An average value of all ρ_i is then computed, following the approach adopted in the average variance computation. This metric is therefore given by

$$M_\rho = \frac{1}{M} \sum_{i=1}^T \rho_i \quad (4.17)$$

The metrics M_{σ^2} and M_ρ are termed variance and standard deviation of pasting respectively. Both reflect the efficiency of the pasting process considering that high values of the variance and standard deviation of pasting are associated with inaccurate pasting processes.

4.2.2 Assessment based on the Structural Similarity Index

Presented in the work of [28, 29], the Structural Similarity Index (SSIM) metric is an image quality metric taken between two images, I_1 and I_2 , where the I_1 is chosen to be the reference image. This metric addresses the visual impact of image luminance, contrast and structure. The SSIM is written as

$$SSIM(I_1, I_2) = [l(I_1, I_2)]^\alpha [c(I_1, I_2)]^\beta [s(I_1, I_2)]^\gamma \quad (4.18)$$

where $l(I_1, I_2)$ is related to the luminance term, $c(I_1, I_2)$ to the contrast term and $s(I_1, I_2)$ with the structural term. Individually, they are given by

$$l(I_1, I_2) = \frac{2\mu_{I_1}\mu_{I_2} + C_1}{\mu_{I_1}^2 + \mu_{I_2}^2 + C_1} \quad (4.19)$$

$$c(I_1, I_2) = \frac{2\sigma_{I_1}\sigma_{I_2} + C_2}{\sigma_{I_1}^2 + \sigma_{I_2}^2 + C_2} \quad (4.20)$$

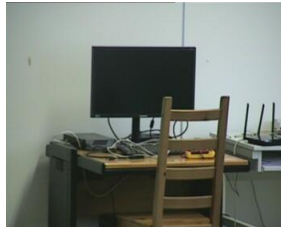
$$s(I_1, I_2) = \frac{\sigma_{I_1 I_2} + C_3}{\sigma_{I_1} \sigma_{I_2} + C_3} \quad (4.21)$$

where μ_{I_1} , μ_{I_2} , σ_{I_1} , σ_{I_2} and $\sigma_{I_1 I_2}$ are the local means, standard deviation and cross-covariance for images I_1 and I_2 . Taking $\alpha = \beta = \gamma = 1$, $C_1 = (0.01L)^2$, $C_2 = (0.03L)^2$ and $C_3 = \frac{C_2}{2}$, where L is the specified dynamic range value ($L = 255$ for images of data type *uint8*), equation 4.18 reads

$$SSIM(I_1, I_2) = \frac{(2\mu_{I_1}\mu_{I_2} + C_1)(2\sigma_{I_1 I_2} + C_2)}{(\mu_{I_1}^2 + \mu_{I_2}^2 + C_1)(\sigma_{I_1}^2 + \sigma_{I_2}^2 + C_2)}. \quad (4.22)$$

C_1 , C_2 and C_3 are regularization constants for the luminance, contrast and structural terms respectively, used to avoid instability for image areas where the standard deviation or local mean are close to zero.

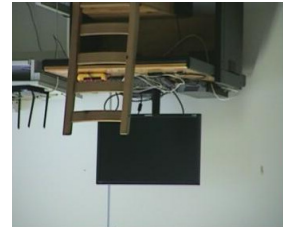
The SSIM function [29] returns a value between 0 and 1 which can also be represented in $[0\% - 100\%]$ interval. Figure 4.5 illustrates the SSIM value between the reference image 4.5(a) and other modified versions of this image.



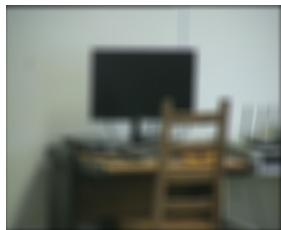
(a) Reference image.
SSIM = 100%



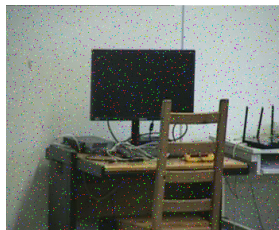
(b) Increased brightness.
SSIM = 87.55%



(c) Inverted image.
SSIM = 21.07%



(d) Blurred image.
SSIM = 76.28%



(e) Noisy image.
SSIM = 49.05%



(f) Incremented pan.
SSIM = 53.79%

Figure 4.5: SSIM metric between the reference image 4.5(a) and modified versions of the original image.

The extraction of images from the final panorama according to its original pan and tilt coordinates, i.e. the inverse process of the panoramic transformation procedure, allows a direct assessment to be made with the original images. These extracted images, however, are usually compositions of several images due to the intersecting FOVs. Thus, the assessing process should focus on the following strategy.

A mosaic is classified as a low-quality panorama if the original and extracted images illustrate significant discrepancies. On the other hand, if the images visually match then the mosaic is rated as a high-quality panorama. For this case, the pasting process is considered accurate.

Following this approach, the SSIM metric is used as an assessment metric between the original and the extracted images. An average value over the SSIM metric of all pairs of images that are part of the final panorama is considered to evaluate the efficiency and quality of the final result. This visual perception metric is termed mosaic quality.

4.3 Optimize Mosaic Alignment

As previously mentioned, the pasting procedure into the cube and cylinder based representations is highly dependent on the camera intrinsic parameters, namely the focal length parameters. Small changes in these values, can lead to different results on the final output of the PTZ panoramic image generation process. In section 4.2.1, the variance of pasting was introduced as a performance metric that allows the measurement of the accuracy of pasting.

This section presents an optimization approach where the camera focal length parameters are iteratively adjusted within the panoramic transformation step in order to minimize the average variance of pasting represented in equation 4.15. Minimizing the variance will, theoretically, improve the pasting accuracy and the quality of the final panorama. Moreover, this procedure can be interpreted as a colour based calibration. Given an initial estimate over the camera focal length parameters, these variables are optimized according to the optimal mosaic alignment.

This optimization was performed under Matlab optimization function *fminsearch* [30], and its cost function reads

$$(k_u^*, k_v^*) = \arg \min_{k_u, k_v} \frac{1}{M} \sum_{i=1}^T \left(\sum_{j=1}^{K_i} \frac{x_{ij}^2(k_u; k_v)}{K_i} - \mu_i^2(k_u; k_v) \right) \quad (4.23)$$

where $x_{ij}^2(k_u; k_v)$ stands for the high dependability of the pixel to pixel transformation on the camera focal length parameters k_u and k_v , during the panoramic transformation step.

Different calibration algorithms provide different calibration matrices for each level of zoom. Therefore, this optimization was considered for every case. Chapter 5 illustrates the results obtained with this minimization approach.

4.4 Seamless Mosaicing by Image Blending

Already introduced in section 2.3, the image blending process is usually the last step in any panoramic image generation application. It is responsible for compensating exposure differences and reducing generated artifacts in the overlapping regions of the final panorama.

This section starts with two basic approaches such as the average and median of pixel intensities, then moving onto the linear and hard blending techniques. The section ends with a blending approach where the final mosaic is built in the gradient domain and then reconstructed by solving a Poisson equation.

4.4.1 Average and Median

Among the more basic approaches are the average and median pixel intensity computation. A pixel within the cube or cylinder based representations can be overlapped by multiple images according to the image positioning in the final panorama. Hence, an intuitive idea of a blending approach is to consider the average pixel intensity value. This computation is given by

$$P(u, v) = \sum_{j=1}^L \frac{I_j(u, v)}{K(u, v)} = \frac{1}{K(u, v)} \sum_{j=1}^L I_j(u, v) \quad (4.24)$$

where $P(u, v)$ is the average pixel intensity of pixel (u, v) in the panoramic image P , $I_j(u, v)$ is the pixel intensity value of (u, v) set by the resampled image I_j in panoramic coordinates, $K(u, v)$ is the number of overlapping images in pixel (u, v) and L is the total number of images that are part of the panorama. Equation 4.24 is an alternative representation of equation 4.13 but with row and column indices instead of a single index.

In practise, the average is computed in two parts. First, the sum of I_j and K variables for each (u, v) pixels are stored in two independent auxiliary matrices with the same size of the final panorama. Then both matrices undergo a pixel to pixel division for non-zero elements.

Another simple approach, is the median pixel intensity computation. Here, every non-zero $I_j(u, v)$ is saved in a vector $P(u, v)[n]$ with size $K(u, v)$ from smallest to greatest under the form of n_i where $1 \leq i \leq K(u, v)$, as represented in equation 4.25.

$$P(u, v)[n] = [n_1 \ n_2 \ \dots \ n_{K(u, v)}] \quad s.t. \quad n_1 \leq n_2 \leq \dots \leq n_{K(u, v)} \quad (4.25)$$

The median pixel intensity of pixel $P(u, v)$ depends on $K(u, v)$. If $K(u, v)$ is an odd number, the median is the middle value of $P(u, v)[n]$. If $K(u, v)$ is even then the median is the average of both values in the center. This computation is given by

$$P(u, v) = \frac{P(u, v) \lceil \frac{K(u, v)+1}{2} \rceil + P(u, v) \lfloor \frac{K(u, v)+1}{2} \rfloor}{2} \quad (4.26)$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ stand for the ceiling and floor functions of index $\frac{K(u, v)+1}{2}$ in vector $P(u, v)[n]$, respectively. Both approaches can be considered separately for the three RGB channels or computed for greyscale images.

The median, unlike the average approach where all values are stored under a sum, is computationally more expensive since all values attributed to pixel (u, v) are saved under vector form. This leads to bigger memory requirements and time consuming computations. The results provided by both methodologies are later presented in chapter 5.

4.4.2 Linear and Hard Blending

Both linear and hard blending approaches rely on spatial-weighting of image pixels i.e, each pixel is associated with a weight according to its position in the image with respect to the image borders.

There are multiple documented methods on spatially-weighting techniques, from which is part the grassfire transform described in section 2.3 (see algorithm 2.2). As output this function returns a weight matrix, also called weight image, with the same image size of the original image, where pixel weighting varies linearly from 1 at the center of the image to a near 0 value (to avoid instabilities at pixel intensity computation) at the boundaries.

The linear blending approach uses the grassfire transform to perform a weighted pixel average intensity computation. In this method, all pixel intensities are multiplied by its corresponding weight and then summed accordingly to the number of overlapping images in each pixel. These values are later divided by the sum of weights for each pixel. The linear blending computation can be described as

$$P(u, v) = \frac{\sum_{j=1}^L I_j(u, v) w_j(u, v)}{\sum_{j=1}^L w_j(u, v)} \quad (4.27)$$

where $P(u, v)$, $I_j(u, v)$, $w_j(u, v)$ and L are the final pixel intensity value of pixel (u, v) within the panoramic image P , the pixel intensity value of the resampled image I_j in panoramic coordinates, the pixel weight of $I_j(u, v)$ set by the weight image w_j also resampled in panoramic coordinates and the total number of images, respectively.

Similarly to the standard average approach described in the previous section, the linear blending method is a two step computation process. First, the sum of products between every $I_j(u, v)$ and $w_j(u, v)$ are saved in an auxiliary matrix, as well as the sum of all weights $w_j(u, v)$. Then, both matrices are divided in a pixel to pixel division for non-zero elements.

In the hard blending method every $P(u, v)$ is determined by the the maximum weight of pixel (u, v) in all weight images $w_j(u, v)$, differently from what happens in the linear blending approach.

Let one consider that pixel $P(2, 1)$ in the final panorama was overlapped by three images $I_1(2, 1)$, $I_2(2, 1)$ and $I_3(2, 1)$ resampled in panoramic coordinates. Each $I_j(u, v)$ has an associated weight $w_j(u, v)$ determined by the weight image w_j , which in this case is $w_1(2, 1)$, $w_2(2, 1)$ and $w_3(2, 1)$ respectively. The hard blending method determines that the final value of $P(u, v)$ is equal to the $I_j(u, v)$ who has the highest $w_j(u, v)$. Considering that $w_2(2, 1) > w_1(2, 1) > w_3(2, 1)$ then $P(2, 1) = I_2(2, 1)$.

This example reflects how the hard blending technique sets the final pixel intensity. Overall, its computation is given by

$$P(u, v) = \sum_{j=1}^L I_j(u, v) W_j(u, v) \quad (4.28)$$

$$W_j(u, v) = \begin{cases} 1 & \text{if } w_j(u, v) = \arg \max_k w_k(u, v), \\ 0 & \text{otherwise,} \end{cases} \quad (4.29)$$

where W_j is a binary weight image for each I_j . This technique is computationally more expensive compared to the linear blending approach since all weights and image indices must be stored and replaced iteratively in a first step, so that then the final panoramic image can be constructed with the right $I_j(u, v)$ values.

The linear and hard blending methodologies can be used separately for each colour channel individually or for greyscale images. The results provided by both methods are later presented in chapter 5.

4.4.3 Blending based on a Gradient Approach

The image blending approach described hereafter is a gradient method from which the final panoramic image is reconstructed by solving a Poisson equation similarly to the Poisson blending methodology described in section 2.3.1.

The developed approach starts by computing the discrete image gradient for all $I_j(u, v)$ represented in panoramic coordinates. g_{u_j} and g_{v_j} stand for the image gradient in the u and v directions of image I_j resampled in panoramic coordinate. They can be computed using the discrete gradient computation described in equation 2.21. With the resulting image gradients, all g_{u_j} and g_{v_j} must be combined using a weight average approach in a similar way to the linear blending method described in the previous section. This computation is given by

$$G_u(u, v) = \frac{\sum_{j=1}^L g_{u_j}(u, v) w_j(u, v)}{\sum_{j=1}^L w_j(u, v)} \quad G_v(u, v) = \frac{\sum_{j=1}^L g_{v_j}(u, v) w_j(u, v)}{\sum_{j=1}^L w_j(u, v)} \quad (4.30)$$

where $G_u(u, v)$ and $G_v(u, v)$ represent the combined gradient values of the final panoramic image in both u and v directions respectively. w_j is obtained with the grassfire transform as previously described

in section 4.4.2. Taking $G = (G_u, G_v)$, the divergence of G is given by

$$\text{div } G = \frac{\delta G_u}{\delta u} + \frac{\delta G_v}{\delta v} \quad (4.31)$$

from which $\frac{\delta G_u}{\delta u}$ and $\frac{\delta G_v}{\delta v}$ can be computed in a discrete form in a similar way as the gradient computation (see equation 2.21).

To successfully solve equation 2.22, one must determine the Ω and $\delta\Omega$ regions. Considering that the main goal is to recover an image from its combined gradient, the whole image was set as Ω .

At this point, $\delta\Omega$ although undefined, is still required to solve such equation. Hence, $\delta\Omega$ was extracted from the result obtained in the linear blending approach due to the expected similarity between both approaches final image. In fact, any effective image blending procedure could be chosen to assist this approach. The image blending approach was selected due to fast and accurate results showed during experimental tests.

A variable b determined the number of boundary pixels from the linear blending result that were used for $\delta\Omega$, as the same amount of boundary pixels that were removed in Ω (combined gradient). This operation is illustrated in figure 4.6.

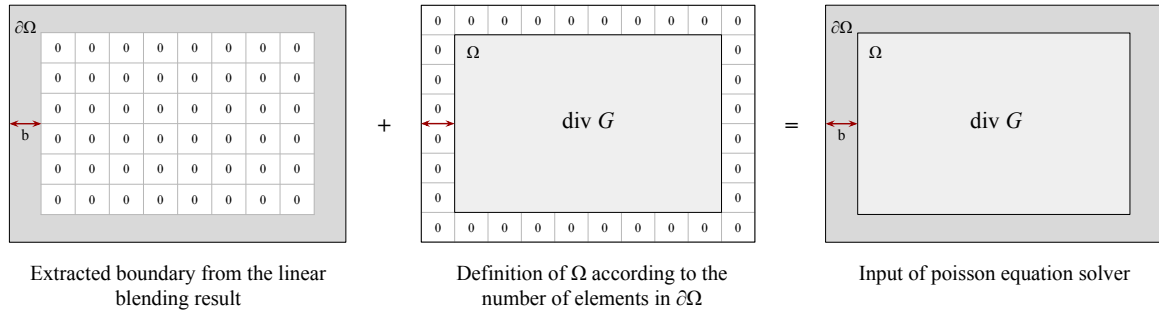


Figure 4.6: Definition of Ω and $\delta\Omega$.

From both Ω and $\delta\Omega$, equation 2.22 is written in the form of equation 2.24, in which A was obtained using matlab *delsq* function [31]. The linear system was solved under matlab *mldivide* function [32], where a sparse LU decomposition algorithm is considered due to the shape of A . This method has a closed formed solution unlike the iterative approaches suggested in [2].

Reshaping the solution x into an image shape, one gets the final panoramic image. Equation 2.22 was solved separately for each colour channel, as the final panoramic image was obtained as composition of these three.

5

Experimental Results

Contents

5.1 Dataset	46
5.2 Feature-Points based Calibration	47
5.3 Colour based Calibration and Panoramic Imaging	50
5.4 Image Blending Results	59

This chapter introduces the results obtained with the different methodologies described in the previous chapters. It starts by describing the dataset from which the multiple PTZ mosaic images were built, as well as a brief description of the camera utilized on this work.

Following the steps that are part of the PTZ panoramic images generation process described in this thesis, we illustrate the results obtained with the different calibration procedures and how they translate into visible differences over the panoramic transformation step. This section is followed by the results provided by the minimized variance of pasting approach, for both cube and cylinder representations. The chapter ends with the image blending results.

5.1 Dataset

In order to study the multiple steps that are part of the PTZ panoramic image generation process, a sequence of images were taken with a fixed range of pan, tilt and zoom coordinates. The images followed a grid structure as illustrated in figure 3.6, with the pan ranging from 20° to 24° and tilt from -5° to -7° with steps of 1° for both angles, performing a total of fifteen images per zoom. The zoom ranged from 1% to 60% with steps of 10%, approximately.

These images were taken under a controlled environment where the background remains still from image to image. Part of the dataset is illustrated in figure 5.2 as the remaining images are shown in appendix A.1.



Figure 5.1: Axis P5512 PTZ Camera [by Axis Communications AB on Axis website].

The sequence of images were acquired in the Institute for Systems and Robotics (ISR), room 8.18 of north tower in Instituto Superior Técnico, using an AXIS P5512 PTZ Camera illustrated in figure 5.1.

This camera is an Internet Protocol (IP) camera, with a pan range of -180° to 180° and tilt range of -90° to 90° , a total zoom of 48x (12x optical and 4x digital), automatic focus adjustment and with a minimum and maximum focal length of 3.8 mm and 46 mm respectively. It is also a Power-over-Ethernet (POE) device, meaning that the camera is powered through the Ethernet cable. For more specifications, the camera datasheet is available on [33].

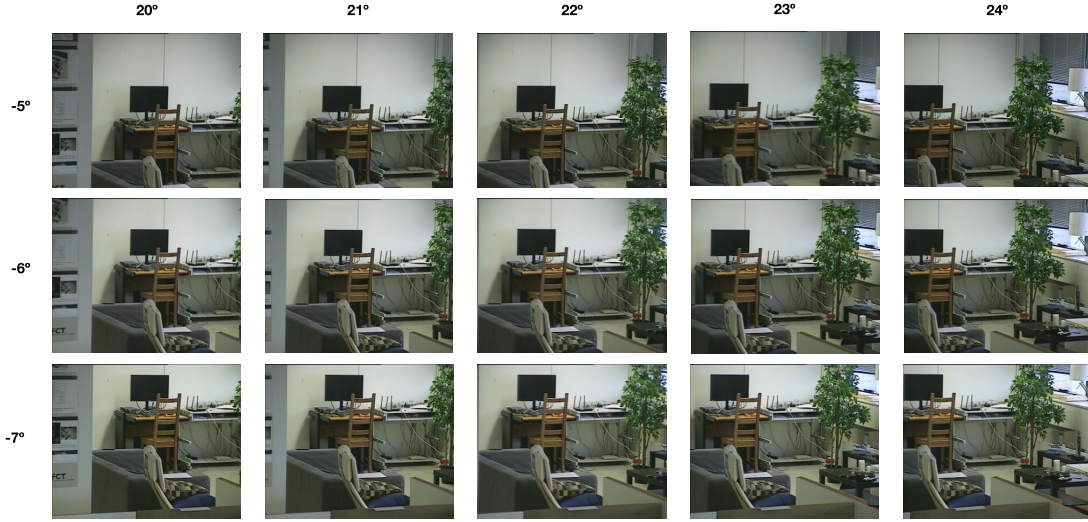


Figure 5.2: Dataset zoom level 20%.

5.2 Feature-Points based Calibration

As previously mentioned, camera calibration is usually the first step in the PTZ panoramic image generation process. Section 3.2 introduced the calibration methodologies studied in this work, and therefore used to calibrate Axis P5512 PTZ Camera. Those methodologies are:

- Matlab calibration toolbox, described in section 3.2.1, where the camera parameters are estimated by changing the orientation of a chessboard;
- Image based calibration, described in section 3.2.2, which is an automatic calibration method for rotating cameras relying on image texture;
- Automatic calibration method, described in section 3.2.3, where PTZ cameras are calibrated through non-linear optimization of re-projection errors;

The given camera, Axis P5512, has been previously calibrated in the work of Luís Alves et al. [34] with Jean-Yves Bouguet camera calibration toolbox for Matlab. At this study, the calibration was performed for zoom levels of 1%, 20%, 40% and 60% and an interpolation method was used to find the intrinsic parameters for the remaining zoom levels.

The interpolation approach has been justified with the need for a large dataset of images of a calibration pattern at different orientations for each zoom level. Calibration would be impractical considering all zoom levels provided by the camera and the non-automatic calibration algorithm.

The radial distortion coefficients were also estimated in [34] however not considered in this work as these parameters have shown an unforeseen behavior. Although one of its coefficients, k_1 , remained

close to zero as zoom varied, k_2 abruptly deviated from zero. The k_2 abrupt deviation is far from being expected. Large zoom-in levels imply that radial distortion is visually not noticeable. The non-zero estimates of k_1 and k_2 are mostly associated to noise in the calibration data.

The dataset here used also confirms this idea that the radial distortion is barely noticed for large levels of zoom in. Due to these reasons, it was decided to not taking into account the radial distortion coefficients. The results of this work are illustrated in both table 5.1 and figure 5.4.

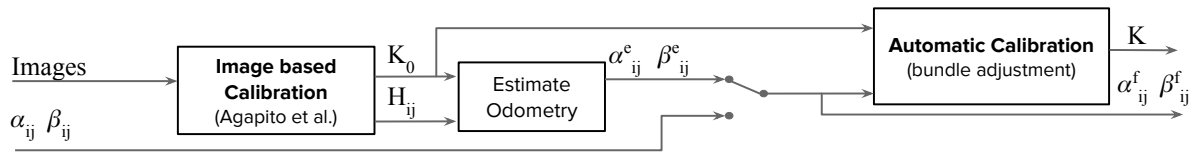


Figure 5.3: Calibration approach.

Considering the work of Sinha et al. [9], where two models of PTZ cameras have been successfully calibrated, the obtained focal length parameters k_u and k_v illustrate a similar behavior as the calibration done in [9]. On the other hand, the results illustrate a different behaviour for the principal point coordinates u_0 and v_0 . The existence of 2D noise, such as image compression or imprecise corner detection or even 3D noise associated with calibration pattern, can be correlated to such unpredicted results.

Section 3.2.3 introduced an automatic calibration technique based on the image calibration method described in section 3.2.2. Taking as input a grid of images acquired at a fixed range of pan and tilt orientations, the algorithm calculates an initial estimate of the camera intrinsic parameters through SIFT feature detection and matching between images. From this initial estimate, it is also possible to estimate camera odometry with the given homography transformations between images H_{ij} .

Zoom level	Toolbox calibration				Image based calibration			
					initial estimate			
	k_u	k_v	u_0	v_0	k_u	k_v	u_0	v_0
1%	460,6	510,2	189,7	137,8	696,2	582,3	178,5	121,9
10%	1154,1	1292,6	198,7	142,3	955,1	1000,5	198,1	105,6
20%	1924,6	2162,1	208,6	147,2	1225,3	1050,3	277,1	43,2
30%	2658,8	2976,6	214,7	158,3	1201,1	1176,9	205,8	64,2
40%	3393,1	3791,2	220,8	169,4	966,4	1026,3	214,5	80,3
50%	4387,9	4782,2	235,6	167,4	752,3	1093,5	175,6	154,5
60%	5382,8	5773,1	250,4	165,4	1253,6	1312,6	293,1	-395,0

Table 5.1: Calibration results.

Zoom level	Automatic calibration + Camera odometry				Automatic calibration + Estimated odometry			
	k_u	k_v	u_0	v_0	k_u	k_v	u_0	v_0
1%	447,2	478,3	175,9	142,9	682,0	567,4	220,1	85,6
10%	1223,7	1321,1	160,3	130,8	995,5	996,2	198,8	115,8
20%	2068,8	2232,4	71,2	148,1	1265,2	1224,4	257,1	51,1
30%	2912,0	3130,4	183,2	170,5	1060,8	1175,3	182,0	137,3
40%	3507,1	3569,4	166,5	640,1	939,6	1264,9	168,1	189,3
50%	3530,6	3186,0	179,6	695,1	753,3	1082,5	167,2	215,0
60%	3874,4	2604,4	209,5	-787,1	1263,7	813,7	323,9	-324,2

Table 5.2: Calibration results(2).

Selecting camera odometry or estimated odometry together with the initial estimate, the camera intrinsic parameters go over a bundle adjustment process with the aim of minimizing the re-projection errors. As output, this method returns the final intrinsic matrix, as well as the camera pan and tilt coordinates used to run the optimization process. This procedure is demonstrated in figure 5.3.

The results are illustrated in tables 5.1 and 5.2. Figures 5.4(a) and 5.4(b) display the focal length parameters k_u and k_v respectively, and figures 5.4(c) and 5.4(d) the principal point coordinates u_0 and v_0 , regarding the multiple results obtained with the different calibration methodologies.

As can be seen in figures 5.4(a) and 5.4(b), the initial estimate (green line) provided a very unexpected behaviour in both camera's focal length parameters k_u , k_v and principal point coordinates u_0 , v_0 . A similar behaviour is observed in the automatic calibration results with estimated odometry (orange line). Such results can be associated with a decreasing image resolution in Axis P5512 for an increasing level of digital zoom.

As previously mentioned in the camera specifications section, Axis P5512 not only works with optical zoom but also with digital zoom. This last feature reduces image resolution and hence is a possible justification regarding Axis P5512 behaviour. Moreover, this constraint directly affects SIFT feature detection and matching, and thus translate in calibration and odometry estimation errors.

Although affected by the constant decrease in image resolution, the automatic calibration method with camera odometry (red line) demonstrated a steady increase on the camera focal length parameters until a level of zoom of 40%.

The toolbox calibration results with camera odometry (blue line), and automatic calibration results with camera and estimated odometry are tested as inputs of the panoramic transformation step in the next section.

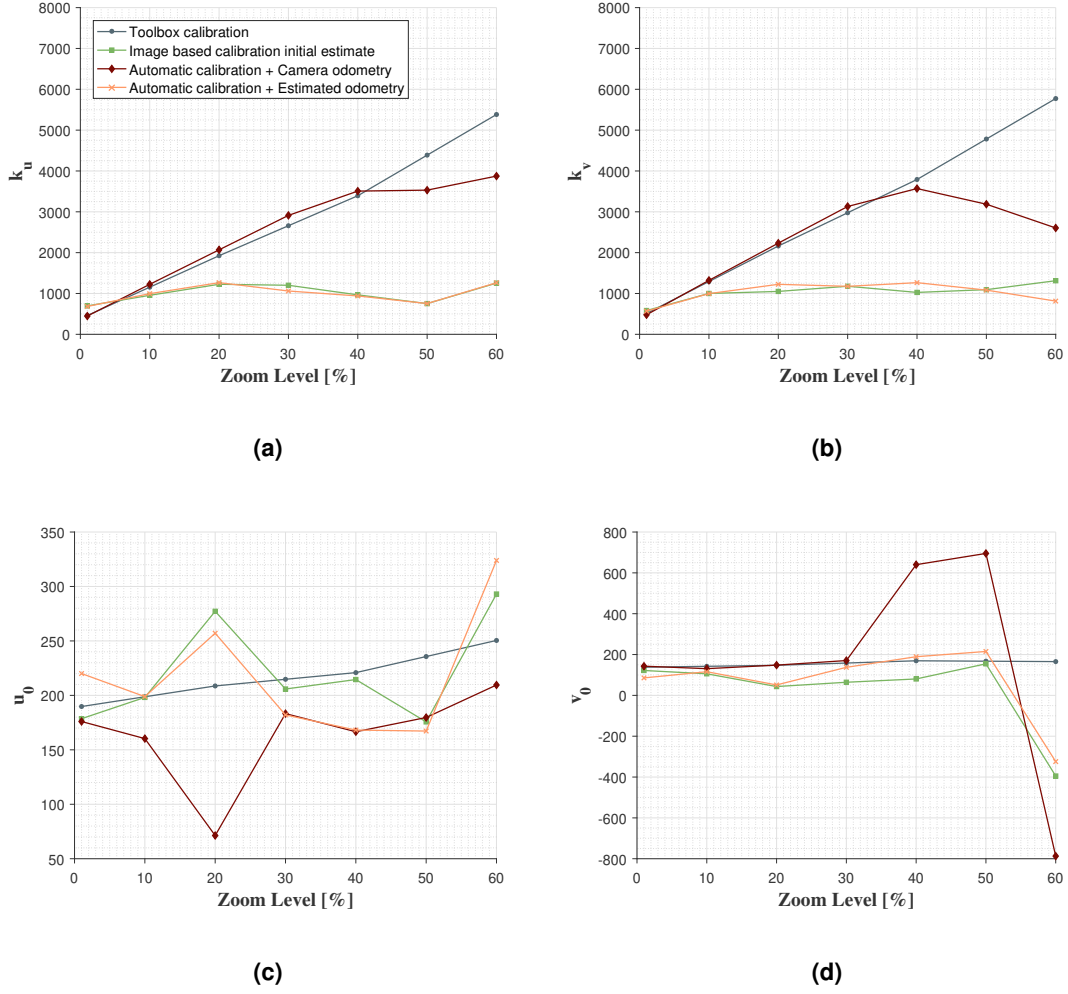


Figure 5.4: Calibration results.

5.3 Colour based Calibration and Panoramic Imaging

This section illustrates the panoramic transformation results for both cube and cylinder based representations described in section 4.1. The calibration results acquired in the previous section are used as inputs, together with camera odometry and estimated odometry.

5.3.1 Cube Results

The cube based representation presented in section 4.1.1 is a six face structure where each face has a resolution of $N \times N$ pixels. In this work, N was set as $N = 2000$ equally for all zoom levels in order to preserve image resolution.

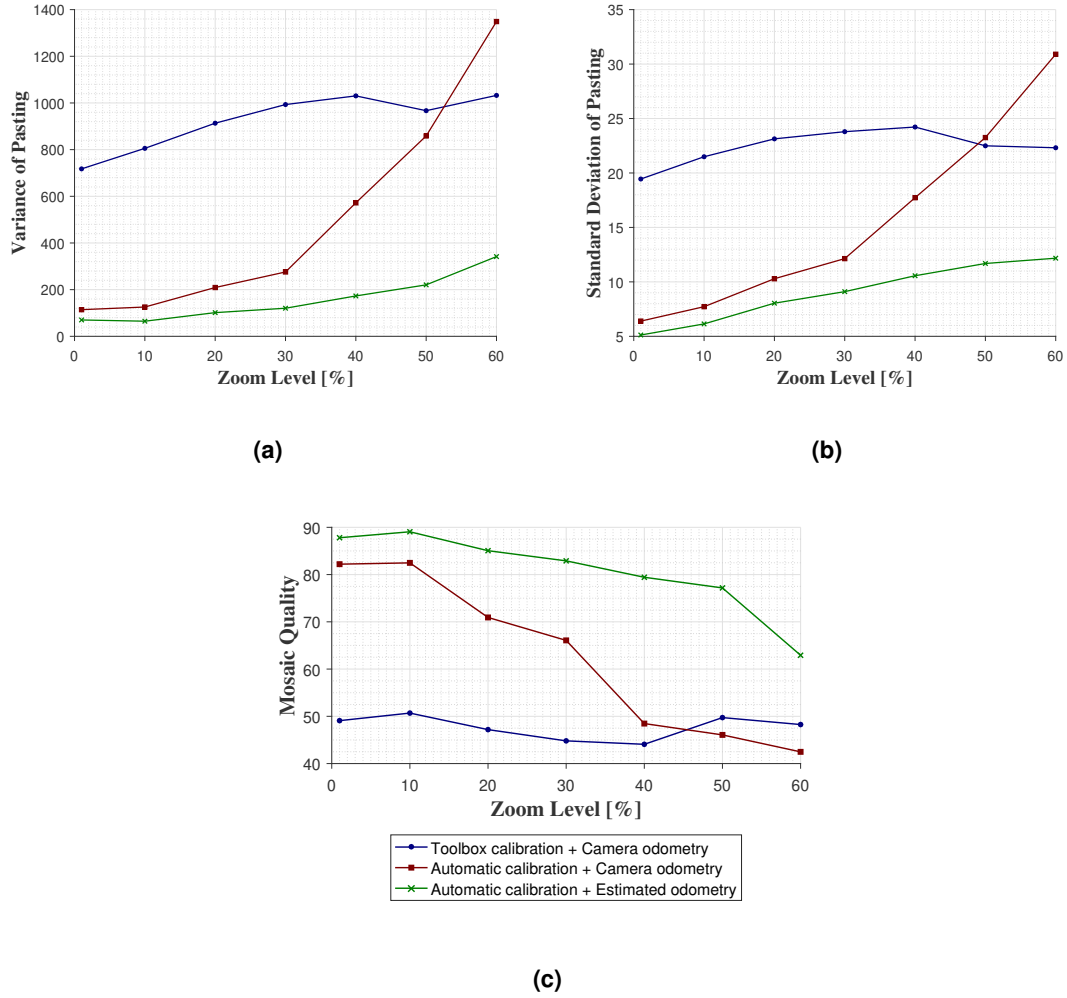


Figure 5.5: Results for a cube's face resolution of $N \times N$ where $N = 2000$ pixels. This results correspond to grayscale images.

Considering the multiple calibration results given in the previous section, the panoramic transformation step for the cube based representation was studied for (i) the toolbox calibration results with camera odometry, and (ii) automatic calibration considering camera odometry or estimated odometry.

The results are shown in figure 5.5, where the variance and standard deviation of pasting, and mosaic quality metric were considered as performance measures. As mentioned in section 4.2.2, the mosaic quality focus on evaluating the degree of similarity and quality between the dataset and the new images extracted from the panoramic representations. As the panoramic transformation is highly dependent on the camera intrinsic parameters, the camera odometry and the resolution factor of the given representation, the trends illustrated in figure 5.5(c) represent the influence these parameters have on the final mosaic.

Figures 5.8, 5.9 and appendix A.2.1 illustrate the given mosaics for each zoom level. The toolbox

calibration results together with camera odometry correspond to the images on the left, and automatic calibration result with camera and estimated odometry are associated with the images at the center and on the right, respectively.

Given the pan and tilt coordinates of most images and camera's focal length parameters, it was only necessary for the cube's front face to build the complete mosaics. Figures 5.8, 5.9 and appendix A.2.1 illustrate the cube's front face for each panoramic transformation input.

Looking closely at figure 5.5, the results for all three performance measures indicate the best performance is obtained with the automatic calibration encompassing odometry estimation. In fact, the lowest variance and standard deviation of pasting and, higher mosaic quality (SSIM based metric) are observed in the obtained mosaics.

Zoom level	Toolbox calibration + Camera odometry		Automatic calibration + Camera odometry		Automatic calibration + Estimated odometry	
	δk_u [%]	δk_v [%]	δk_u [%]	δk_v [%]	δk_u [%]	δk_v [%]
1%	-53,458	-9,184	-3,565	-2,122	-0,422	-0,659
10%	-48,328	2,911	0,015	-0,276	-0,138	-0,020
20%	-47,136	2,671	-0,456	-0,096	0,101	0,194
30%	-45,487	5,402	0,458	-0,384	0,233	0,118
40%	-44,374	7,669	7,042	12,033	0,030	-0,138
50%	-49,296	1,186	25,662	48,616	0,388	0,351
60%	-41,267	22,864	67,015	168,264	1,017	2,458

Table 5.3: Focal length parameters variations.

The mosaics based on the toolbox calibration and camera odometry show much distortion. The automatic calibration together with camera odometry provides more accurate results but still more distorted than using estimated odometry. This behaviour goes in a different direction than what was observed in the calibration results, suggesting that there may have been some calibrations errors in the toolbox calibration. Moreover, it reinforces the idea presented in [34] about the lesser effectiveness of this calibration method in zooming cameras.

Another contributing factor for the illustrated behaviour may be the imprecision of camera odometry for increasing zoom levels. The estimated odometry is also prone to errors in the image based calibration method due to the decreasing image resolution with increasing zoom level in Axis P5512, as illustrated in figure A.11 (image on the right side).

Section 4.3 introduced a variance minimization approach where the camera focal length parameters are iteratively adjusted, within the panoramic transformation step, in order to minimize the variance of pasting.

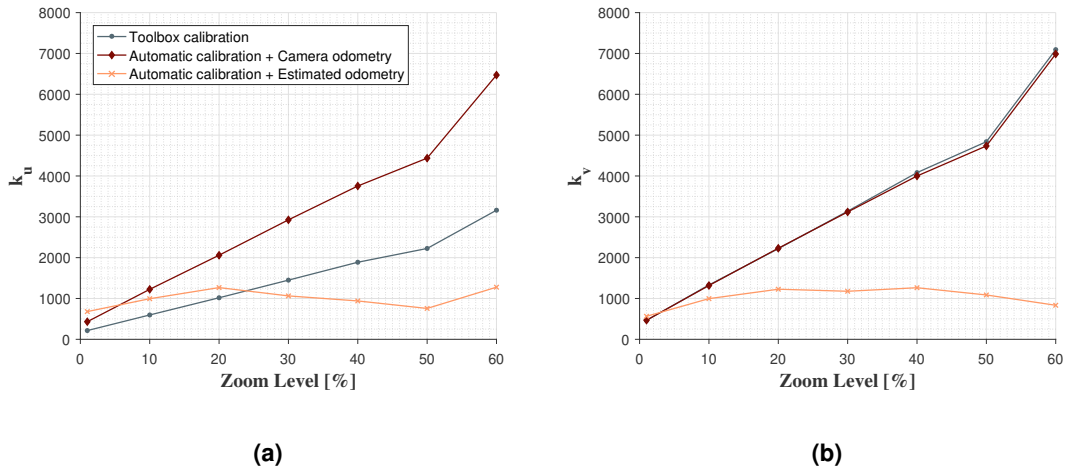


Figure 5.6: Optimized focal length parameters. 5.6(a) and 5.6(b) illustrate the results obtained for the cube based representation.

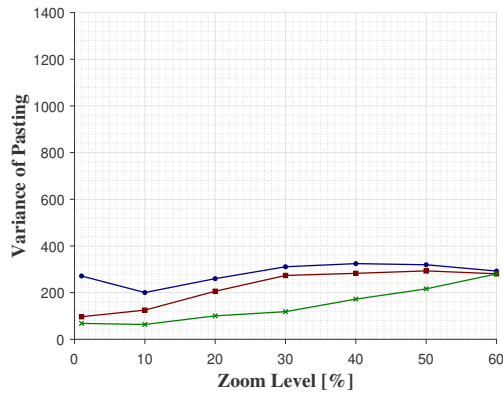
Table 5.3 shows the variation of the focal length parameters for each calibration method described in section 5.2, and figures 5.6(a) and 5.6(b) illustrate the optimized focal length parameters. These results correspond to the minimum variance of pasting results that can be achieved with the remaining fixed variables, camera and estimated odometry, for the cube based representation.

Figure 5.7 shows the variance and mosaic quality results for the optimized focal length parameters as well as the difference between minimized and non-minimized results. The standard deviation results are omitted given the similar impact the optimization had on this metric and the variance. Figures 5.10, 5.11 and appendix A.2.1.A on the other hand, illustrate the corresponding mosaics for each panoramic transformation input parameters.

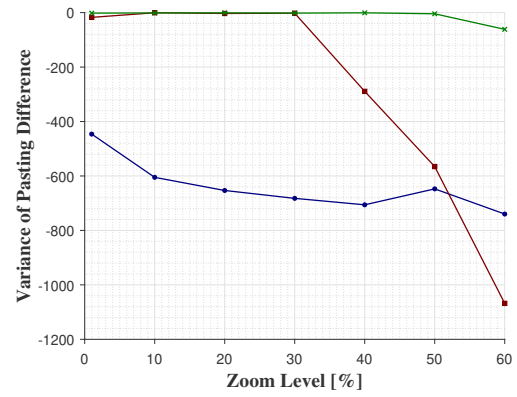
Similarly to appendix A.2.1 and, figures 5.8 and 5.9, toolbox calibration and camera odometry results are demonstrated on the left images, and automatic calibration results together with camera and estimated odometry are associated with the images at the center and on the right, respectively.

As illustrated in figure 5.7, the toolbox and automatic calibration with camera odometry results were effectively adjusted, resulting in a lower variance and standard deviation of pasting and higher mosaic quality. This results are explicitly demonstrated in figures 5.7(b) and 5.7(d).

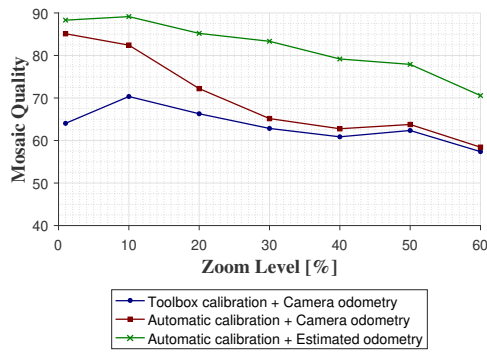
As the variance is expected to decrease with the optimization approach, figure 5.7(b) vividly reflects the impact the variance minimization approach had on the variance by illustrating the difference between minimized and non-minimized results. The bigger the difference, the bigger the impact of the variance minimization approach. The same difference is demonstrated in figure 5.7(d) regarding the mosaic quality metric. The optimization is expected to increase the mosaic quality results comparatively to the non-optimized results.



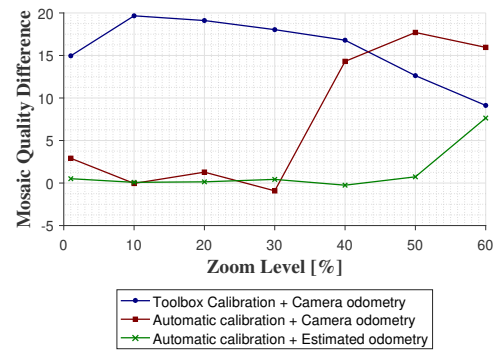
(a)



(b)



(c)



(d)

Figure 5.7: Figures 5.7(a) and 5.7(c) illustrate the minimized variance of pasting results. Figures 5.7(b) and 5.7(d) illustrate the difference between minimized and not-minimized results.

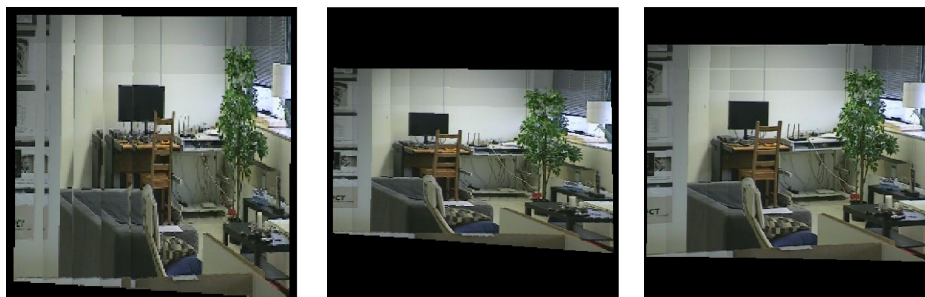


Figure 5.8: Zoom level 20% (cube).

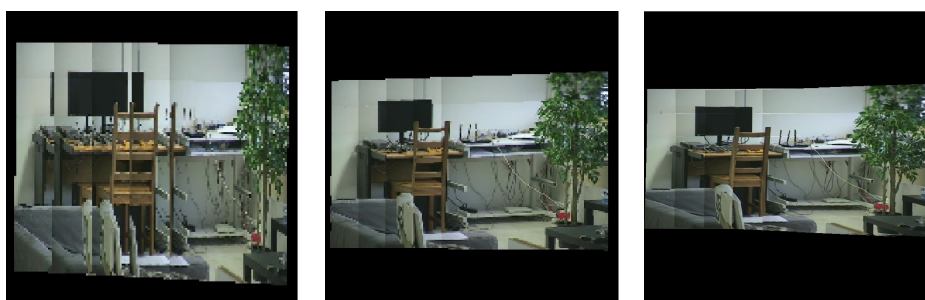


Figure 5.9: Zoom level 40% (cube).

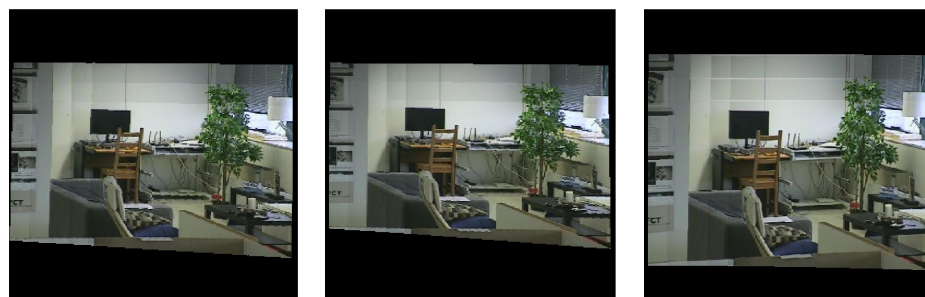


Figure 5.10: Minimized variance, zoom level 20% (cube).

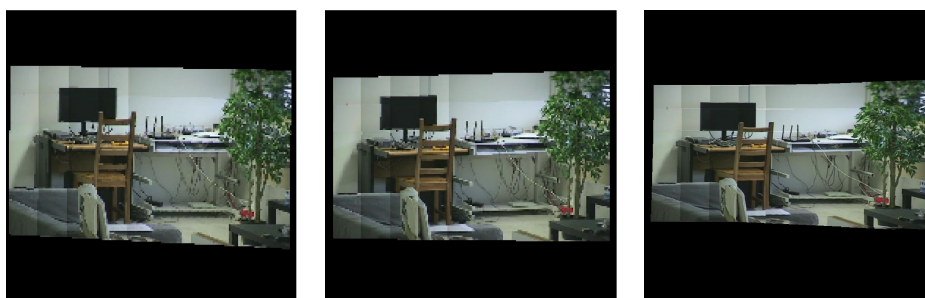


Figure 5.11: Minimized variance, zoom level 40% (cube).

Considering the images illustrated in figures 5.10, 5.11 and appendix A.2.1.A, they are less distorted than the ones presented in figures 5.8, 5.9 and appendix A.2.1. The automatic calibration results with estimated odometry however, were barely affected by the minimization approach, therefore suggesting that these results were already close to its best case scenario.

Although this approach translated into better results for the toolbox and automatic calibration results with camera odometry, the automatic calibration results with estimated odometry still presents the best overall results.

5.3.2 Cylinder Results

Alternatively to the cube based representation, the cylinder was also consider as panoramic transformation representation. Described in section 4.1.2, it was assumed a cylinder resolution of $N \times N$ pixels with $N = 2000$ and scale factor of $f_L = 500$, fixed for all zoom levels. Under these constraints, the cylinder went under a similar case study of the one described in the previous sections regarding the cube structure.

The results were identical. The automatic calibration results together with estimated odometry showed the best performance.

Although the cube and the cylinder based representations are two different types of panoramic structures, the results were particularly consistent with the cube. The cylinder however, presented a slightly lower image resolution associated with its lower resolution factor. Due to the similar behaviour illustrated in both cube and cylinder results, this section only illustrates the best overall performance associated with the automatic calibration results with estimated odometry (see figure 5.12).

Zoom level	Automatic calibration + Estimated odometry	
	δk_u [%]	δk_v [%]
1%	-0,133	-0,382
10%	0,249	-0,044
20%	0,078	0,412
30%	0,281	-0,084
40%	0,070	-0,070
50%	0,410	-0,089
60%	0,965	2,596

Table 5.4: Focal length variations on the automatic calibration and estimated odometry results.

Similarly to the cube based representation, the variance minimization approach in the cylinder was especially effective in both toolbox and automatic calibration results with camera odometry. Even though

it is not illustrated, the optimized focal length parameters followed a similar behaviour as the one displayed in figures 5.6(a) and 5.6(b), where the focal length parameters kept increasing as the zoom level increased. The panoramic evaluation metrics were also successfully improved, resulting in a lower variance and standard deviation of pasting and higher mosaic quality.

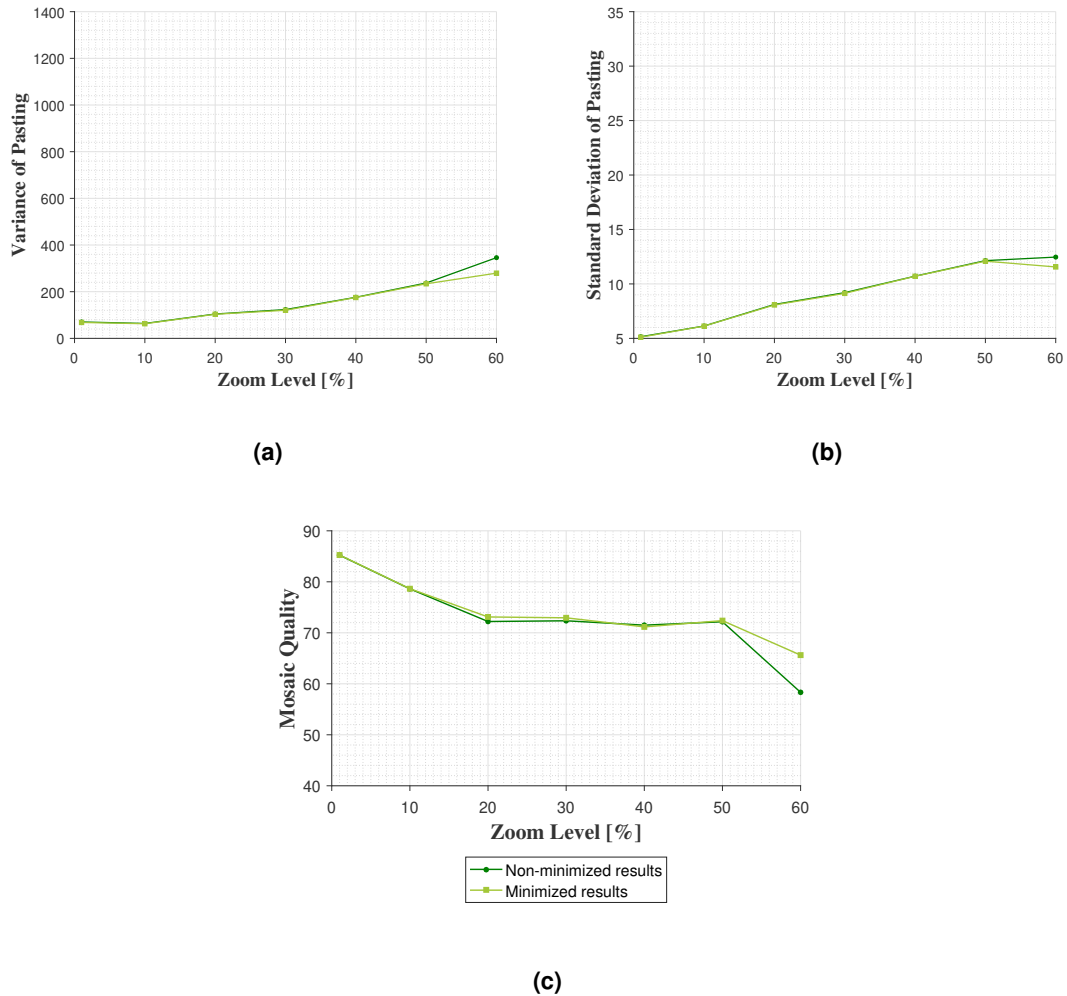


Figure 5.12: Results for a cylinder's resolution of $N \times N$ where $N = 2000$ pixels. This results correspond to grayscale images.

The automatic calibration results with estimated odometry on the other hand, were somewhat improved by this minimization approach. These results are illustrated in table 5.4 and figure 5.12, as the focal length parameters and the panoramic evaluation metrics were barely affected by the optimization approach. This behaviour has also been observed in the cube based representation. Figures 5.13, 5.14, 5.15 and appendix A.2.2 illustrate the images for both not-minimized and minimized variance of pasting approaches regarding the automatic calibration results with estimated odometry.



Figure 5.13: Automatic calibration and estimated odometry, zoom level 10% (cylinder).



Figure 5.14: Automatic calibration and estimated odometry, zoom level 30% (cylinder).

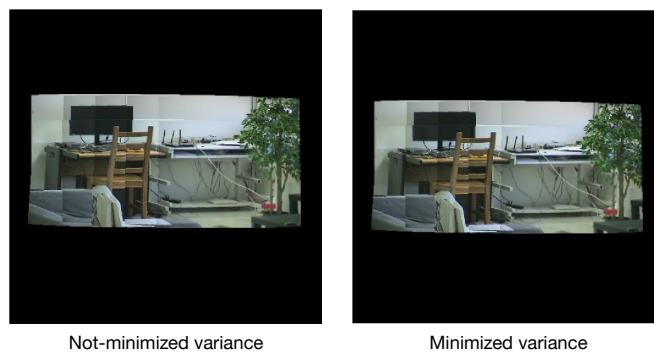


Figure 5.15: Automatic calibration and estimated odometry, zoom level 50% (cylinder).

5.4 Image Blending Results

This section presents the image blending results. Each blending method, previously described in section 4.4, was applied to the optimized automatic calibration results together with estimated odometry, considering its best overall performance for both cube and cylinder based representations in the previous sections.

5.4.1 Cube

The image blending step is responsible for reducing generated artifacts and compensating exposure differences in image's overlapping regions of the final panorama. Section 4.4 introduced the average and median techniques, the linear and hard blending methods and the proposed gradient based approach. This last method, however depends on the linear blending results and a variable b determining the boundary pixels to be extracted from these results.

In this study, b was set at $b = 8$ pixels, as most images were successfully recovered from its combined gradient. Figures 5.16, 5.17, 5.18 and appendix A.3.1 illustrate the optimized results for the automatic calibration results with estimated odometry, as previously stated. The image blending step does not affect the positioning of the images within the final mosaic. Therefore, the variance and standard deviation of pasting were not studied as these variables are unaffected by this last step.

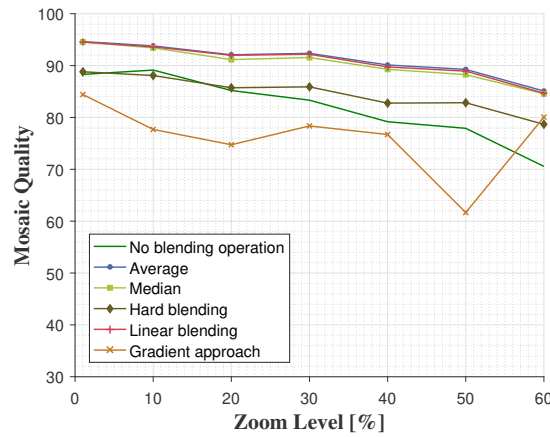


Figure 5.16: Mosaic quality measures regarding the multiple blending approaches (cube).

Looking at figure 5.16, a particular trend among the different methodologies is illustrated as the mosaic quality tends to steadily decrease for an increasing zoom level. As previously mentioned, the panoramic transformation is highly dependent on the camera intrinsic parameters, the camera odometry and the panoramic representation resolution factor.

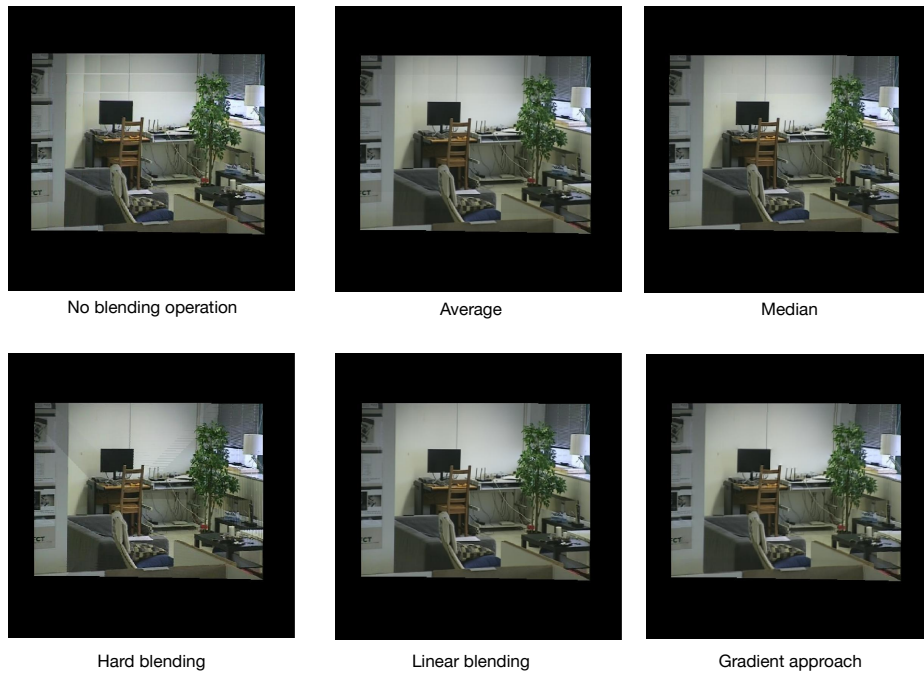


Figure 5.17: Blending results, zoom level 20% (cube).

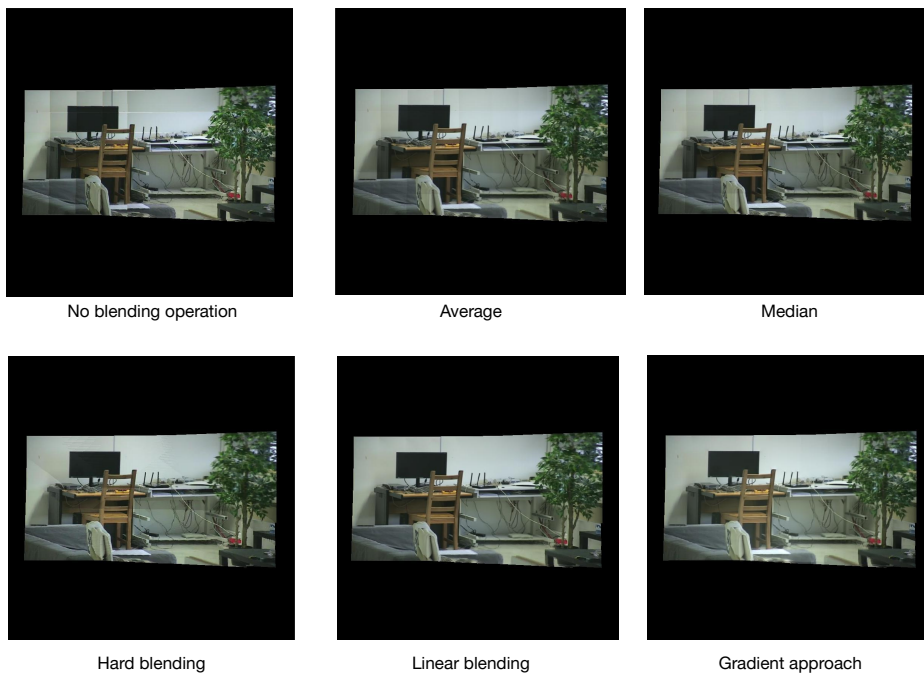


Figure 5.18: Blending results, zoom level 40% (cube).

These parameters later reflect on the outcome of the mosaic quality metric as it evaluates the degree of similarity and quality between the original images and the ones extracted from the panoramic representations. Although the intrinsic parameters and camera odometry determine the position of the images within the panoramic representation, it is believed that the representation resolution factor is one of elements associated with such trend.

In fact, as the panoramic transformation and the inverse process take place, the images go over transformation processes in order to fit a determined panorama resolution or image size, respectively. These processes might enlarge or dwindle the images, as it depends on the zoom level. Considering the panorama fixed resolution for all zoom levels, the images might be losing some definition aside from suffering small deformations with these transformation processes. As an end result, these constraints might lead to a steady decrease in the mosaic quality results.

Focusing on the image blending methodologies, the average, median and linear blending approaches successfully improved the mosaic quality metric over all zoom levels as suggested by figure 5.16. On the other hand, the hard blending approach barely affected the mosaic quality metric in the first levels of zoom, and the gradient approach even worsened the original panoramic transformation results.

The existence of undefined pixels intensities within the selected boundary $\delta\Omega$ is thought as one of the possible justifications regarding the outcome associated with the gradient based approach. These undefined intensities are spread along the image, resulting in final intensity values which are different from the original images.

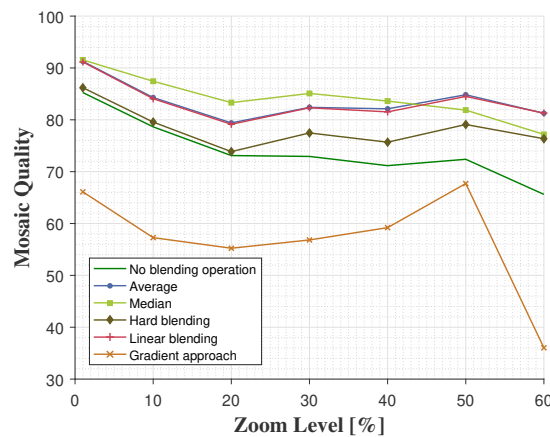


Figure 5.19: Mosaic quality measures regarding the multiple blending approaches (cylinder).

Figures 5.17, 5.18 and appendix A.3.1 however, illustrate a different behaviour as the linear and gradient blending approaches show the best overall results. Taking a better look at the average and median results, the overlapping regions are still visible showing clearly the frontiers of the images making the mosaic.

Apart from figure A.24 which was not effectively recovered from its combined gradient due to the selected boundary ($\delta\Omega$), mosaic quality metric incorrectly evaluates this blending method. Furthermore, the average and median are among the best mosaic quality results although they do not totally blend the overlapping regions. It is therefore suggested that the mosaic quality can be an inconclusive metric in the assessment of image blending methodologies, when significant artifacts are present in the mosaic.

5.4.2 Cylinder

A similar behaviour has been observed on the cylinder image blending results. Aside from a common decrease in the mosaic quality measures associated with the cylinder lower image resolution, the average, median and linear blending methods demonstrated the best results of the test as shown in figure 5.19. Contrarily, the gradient based approach has shown an inferior performance.

Considering the cylinder representations illustrated in figure 5.20 and appendix A.3.2 on the other hand, the gradient approach effectively recovered most images from its combined gradient, in contrast to the average and median techniques which were not effective on covering images overlapping regions. Aside from figure A.31 which illustrates a particular case where the gradient approach was unsuccessful due to an odometry estimation error, the mosaic quality incorrectly evaluates this blending method.

Given these results, the idea associated with the imprecision of the mosaic quality metric on the assessment of image blending techniques is once again reinforced.

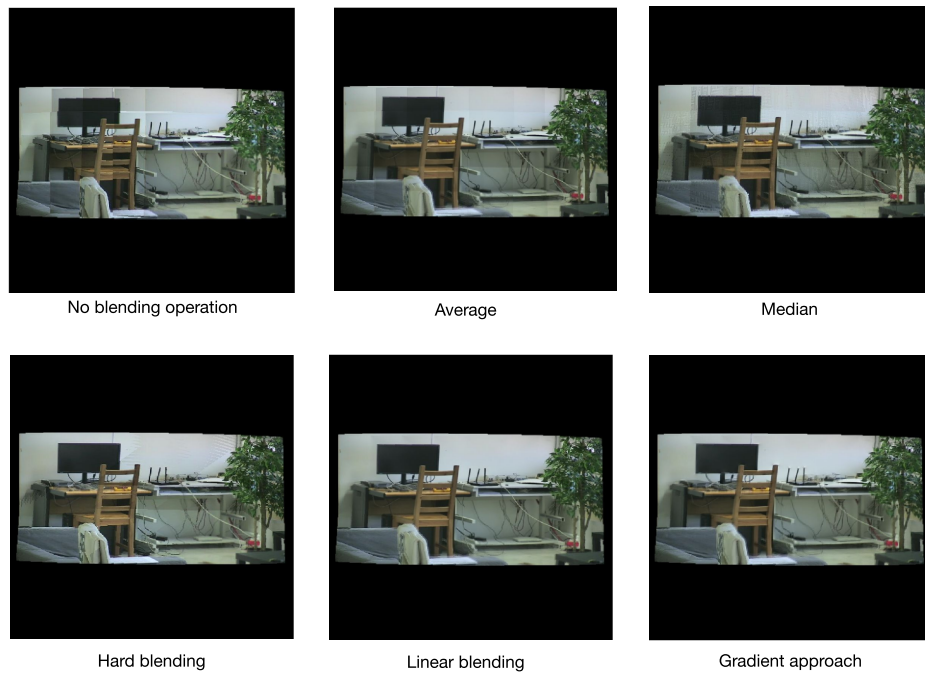


Figure 5.20: Blending results, zoom level 50% (cylinder).

6

Conclusion

Contents

6.1 Conclusions	64
---------------------------	----

6.1 Conclusions

The work described within this thesis focused on studying the PTZ panoramic image generation process. Usually divided into four separate parts, the work addressed PTZ camera calibration, panoramic transformation and image blending methodologies.

Starting with the acquisition of a sequence of images taken at a fixed range of pan and tilt coordinates for multiple zoom levels, the given PTZ camera was calibrated with an automatic calibration method relying on an image based calibration technique. This texture based approach provided an initial estimate on camera intrinsic parameters as well as on camera odometry, which were later used as inputs on the automatic calibration bundle adjustment process.

Along with the camera calibration results obtained in [34] with Jean-Yves Bouguet camera calibration toolbox, the automatic calibration results with camera odometry illustrated a continuous increase in camera focal length parameters for an increasing level of zoom. On the other hand, the automatic calibration results together with estimated odometry illustrated a variable behaviour possibly associated with camera decreasing image resolution for an increasing level of digital zoom. This constraint results in calibration and odometry estimation errors, considering its impact on the detection and matching of features.

The multiple calibration results were later tested as input parameters of the panoramic transformation step for the cube and cylinder based representations. Contrarily to the previous results, the automatic calibration results with estimated odometry presented the lowest variance and standard deviation of pasting as well as the highest mosaic quality results. The final mosaics were also less distorted and more appealing than the ones presented by the other inputs.

With the variance minimization approach, the automatic and toolbox calibration results with camera odometry were effectively adjusted, leading to a major improvement on all performance measures and final mosaics. The automatic calibration results with estimated odometry were barely affected. This behaviour suggests that these results were already close to its best case scenario.

On the image blending step, the average and median techniques, the linear and hard blending methods, and the proposed gradient based approach were tested. The average, median and linear blending techniques improved the mosaic quality measures, although the image overlapping regions were still visible on the average and median results.

The gradient based approach successfully recovered most images from its combined gradient compensating exposure differences in image overlapping regions. However, it presented the lowest mosaic quality measures. Given these results, the mosaic quality metric was somewhat ineffective on evaluating the image blending results and therefore can be recorded as an inconclusive performance measure regarding this topic.

Considering future work, the gradient based approach presented in this work can be optimized. In

this method, the selected boundary assumed a rectangular shape in which existed, for some cases, pixels with undefined intensities. Although it has not affected the quality of the final mosaics, this boundary can be more carefully selected, i.e. avoiding undefined pixels, and thus assume an arbitrary shape.

The memory requirements of PTZ panoramic image generation process for multiple zoom levels should also be taken into account. As most panoramic images demand high resolution images, having an efficient way of saving a sequence of multi zoom level panoramas preserving image quality would be especially useful in any panoramic image generation process.

Lastly, PTZ camera odometry should also be addressed. In the presence of a textureless scene, the odometry estimation as presented in this work becomes a difficult task. Therefore, different strategies should be considered to estimate camera odometry and consequently integrate these measures for a better odometry reading.

Bibliography

- [1] G. Vass and T. Perlaki, "Applying and removing lens distortion in post production," in *Proceedings of the 2nd Hungarian Conference on Computer Graphics and Geometry*, 2003, pp. 9–16.
- [2] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on graphics (TOG)*, vol. 22, no. 3, pp. 313–318, 2003.
- [3] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International journal of computer vision*, vol. 74, no. 1, pp. 59–73, 2007.
- [4] H. Blum *et al.*, "A transformation for extracting new descriptors of shape," *Models for the perception of speech and visual form*, vol. 19, no. 5, pp. 362–380, 1967.
- [5] H. Yong, J. Huang, W. Xiang, X. Hua, and L. Zhang, "Panoramic background image generation for ptz cameras," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3162–3176, 2019.
- [6] K.-T. Song and J.-C. Tai, "Dynamic calibration of pan–tilt–zoom cameras for traffic monitoring," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 5, pp. 1091–1103, 2006.
- [7] Y. Xu and D. Song, "Systems and algorithms for autonomous and scalable crowd surveillance using robotic ptz cameras assisted by a wide-angle camera," *Autonomous Robots*, vol. 29, no. 1, pp. 53–66, 2010.
- [8] C.-C. Chen, Y. Yao, A. Drira, A. Koschan, and M. Abidi, "Cooperative mapping of multiple ptz cameras in automated surveillance systems," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1078–1084.
- [9] S. N. Sinha and M. Pollefeys, "Pan–tilt–zoom camera calibration and high-resolution mosaic generation," *Computer Vision and Image Understanding*, vol. 103, no. 3, pp. 170–183, 2006.
- [10] J.-Y. Bouguet, "Camera calibration toolbox for matlab; 2010," *Software available at: http://www.vision.caltech.edu/bouguetj/calib_doc*, vol. 80, 2011.

- [11] Y. Abdel-Aziz and H. Karara, "Direct linear transformation into object space coordinates in close-range photogrammetry, in proc. symp. close-range photogrammetry," *Urbana-Champaign*, pp. 1–18, 1971.
- [12] R. I. Hartley, "Self-calibration of stationary cameras," *International journal of computer vision*, vol. 22, no. 1, pp. 5–23, 1997.
- [13] L. de Agapito, E. Hayman, and I. D. Reid, "Self-calibration of a rotating camera with varying intrinsic parameters," in *BMVC*. Citeseer, 1998, pp. 1–10.
- [14] D. Leite and J. Gaspar, "Target tracking with pan-tilt-zoom cameras," Master's thesis, Universidade de Lisboa, Nov 2011.
- [15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [16] A. B. Ricardo Galego and J. Gaspar, "Geometric and radiometric calibration for pan-tilt surveillance cameras," Master's thesis, Universidade de Lisboa, Jun 2011.
- [17] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [18] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM transactions on Graphics*, vol. 2, no. 4, pp. 217–236, 1983.
- [19] A. Levin, A. Zomet, S. Peleg, and Y. Weiss, "Seamless image stitching in the gradient domain," in *European Conference on Computer Vision*. Springer, 2004, pp. 377–389.
- [20] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [21] A. W. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1. IEEE, 2001, pp. I–I.
- [22] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE computer society conference on computer vision and pattern recognition*. IEEE, 1997, pp. 1106–1112.
- [23] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

- [24] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [25] T. M. Inc. (2020) Single camera calibration app. (visited on 2019-9-30). [Online]. Available: <https://www.mathworks.com/help/vision/ug/single-camera-calibrator-app.html>
- [26] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1. Ieee, 1999, pp. 666–673.
- [27] L. De Agapito, R. I. Hartley, and E. Hayman, "Linear self-calibration of a rotating and zooming camera," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 1. IEEE, 1999, pp. 15–21.
- [28] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [29] T. M. Inc. (2020) Structural similarity index (ssim) for measuring image quality. (visited on 2019-10-15). [Online]. Available: <https://www.mathworks.com/help/images/ref/ssim.html>
- [30] T. M. Inc. (2020) Find minimum of unconstrained multivariable function using derivative-free method. (visited on 2019-9-27). [Online]. Available: <https://www.mathworks.com/help/matlab/ref/fminsearch.html>
- [31] T. M. Inc. (2020) Finite difference laplacian. (visited on 2020-1-12). [Online]. Available: <https://www.mathworks.com/help/matlab/math/finite-difference-laplacian.html>
- [32] T. M. Inc. (2020) Solve systems of linear equations $ax = b$ for x . (visited on 2020-1-12). [Online]. Available: <https://www.mathworks.com/help/matlab/ref/mldivide.html>
- [33] *User Manual, AXIS P5512 PTZ Dome Network Camera*, Axis Communications AB, 10 2013, ver. M4.4.
- [34] L. Alves and J. Gaspar, "Background representation and subtraction for pan-tilt-zoom cameras," Master's thesis, Universidade de Lisboa, Nov 2018.



Appendix

A.1 Dataset



Figure A.1: Dataset zoom level 1%.

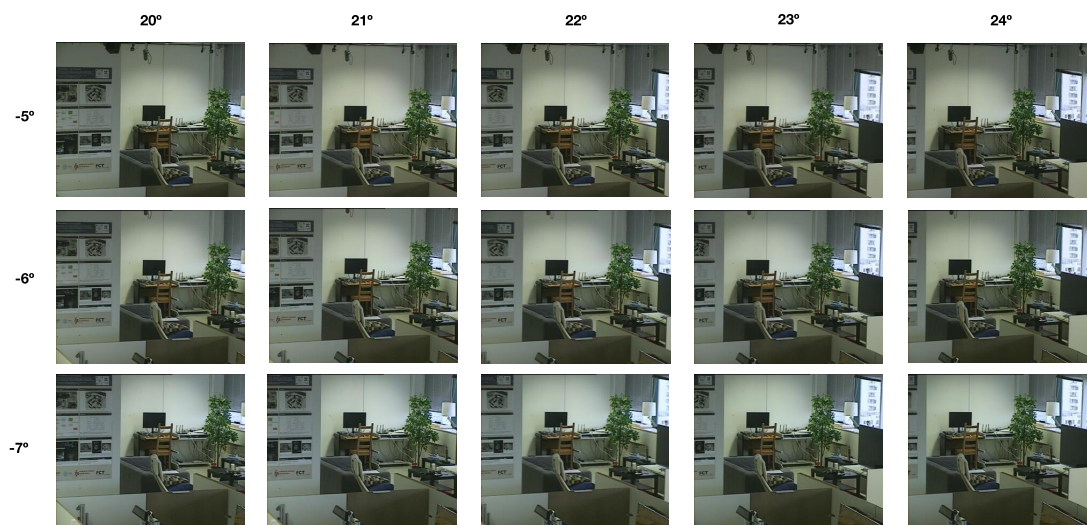


Figure A.2: Dataset zoom level 10%.

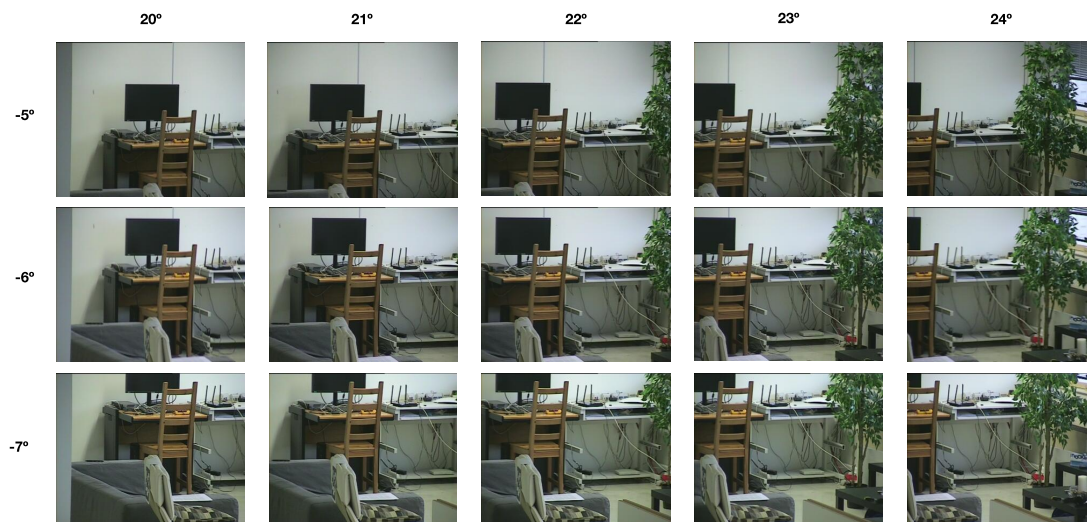


Figure A.3: Dataset zoom level 30%.

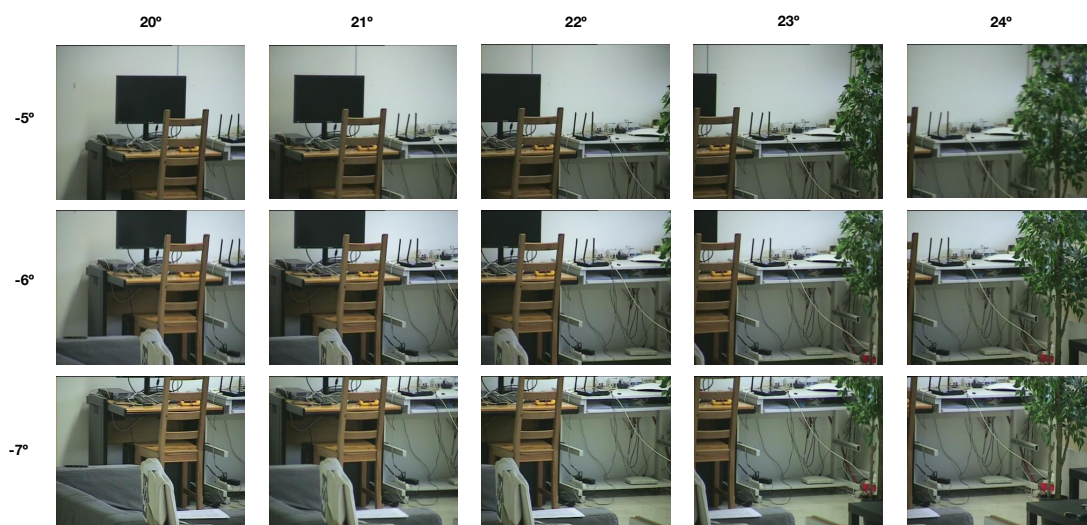


Figure A.4: Dataset zoom level 40%.

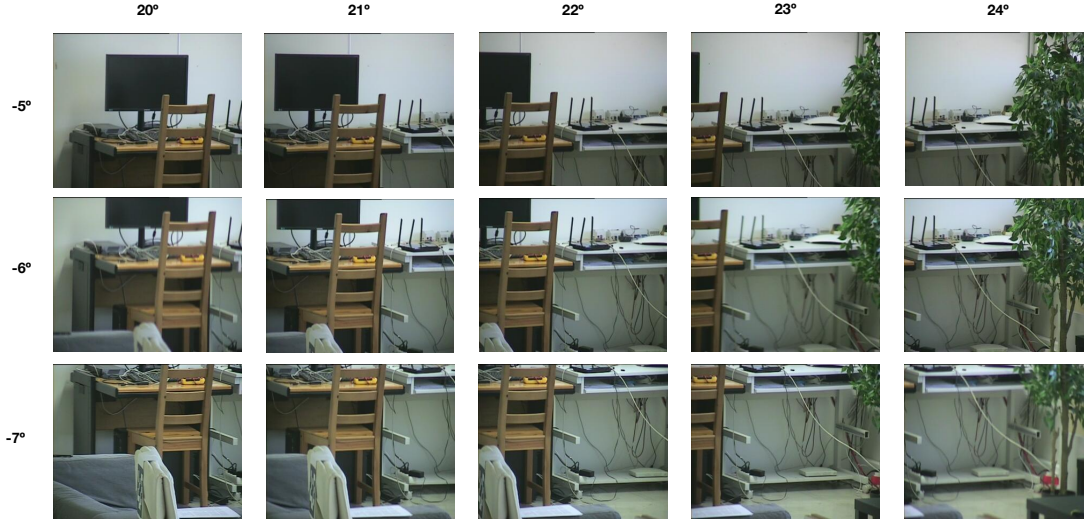


Figure A.5: Dataset zoom level 50%.

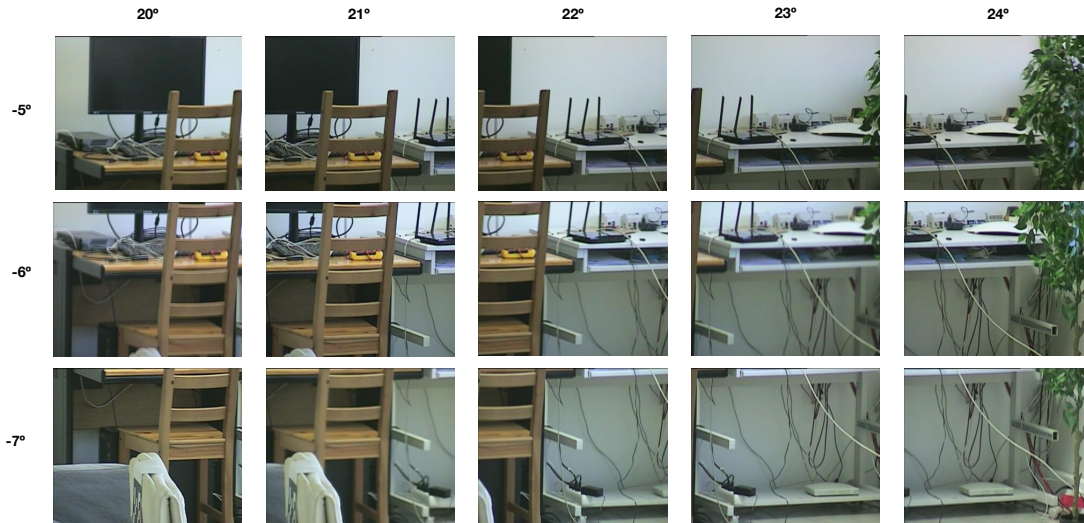


Figure A.6: Dataset zoom level 60%.

A.2 Panoramic Transformation

The cube results shown hereafter correspond to the toolbox calibration with camera odometry (images on the left), automatic calibration with camera odometry (images on the middle) and automatic calibration with estimated odometry (images on the right). In the cylinder results are only illustrated the automatic calibration results with estimated odometry for both not-minimized and minimized variance of pasting, as mentioned in section 5.3.2.

A.2.1 Cube Results

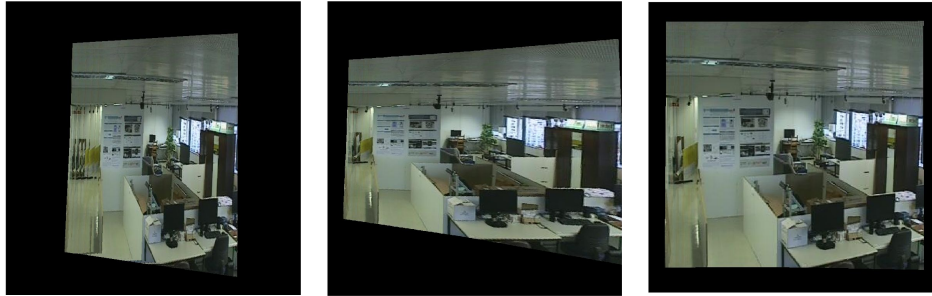


Figure A.7: Zoom level 1% (cube).

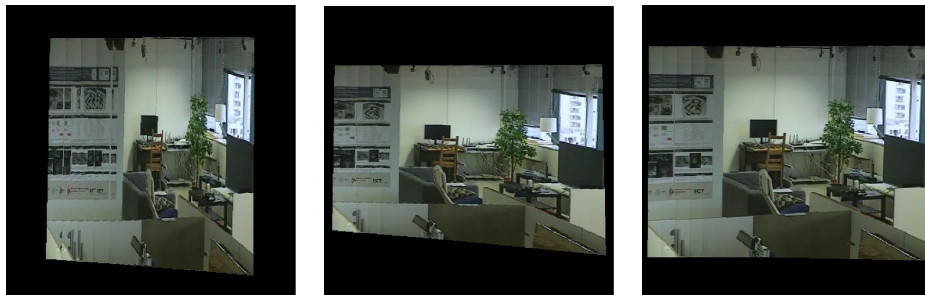


Figure A.8: Zoom level 10% (cube).

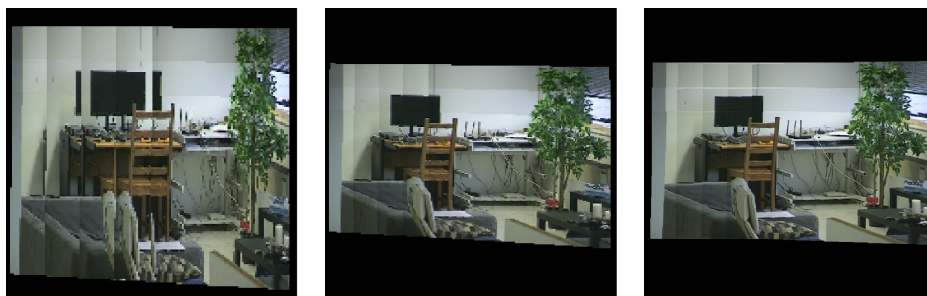


Figure A.9: Zoom level 30% (cube).

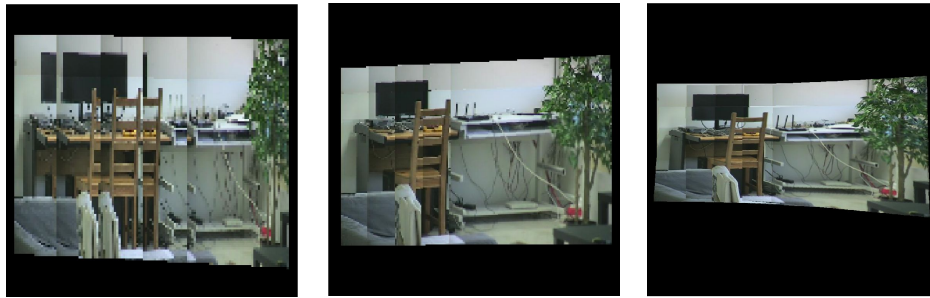


Figure A.10: Zoom level 50% (cube).

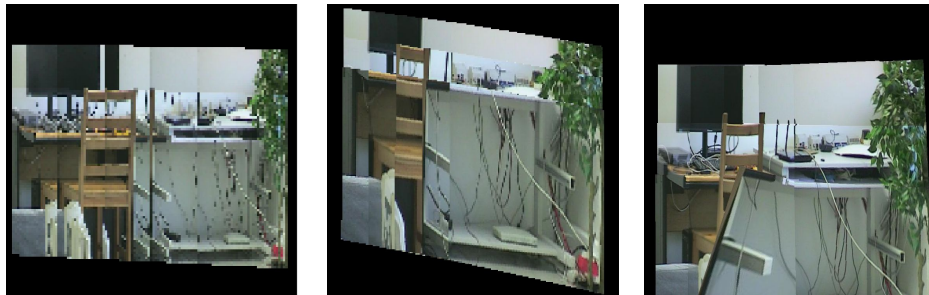


Figure A.11: Zoom level 60% (cube).

A.2.1.A Variance Minimization Results

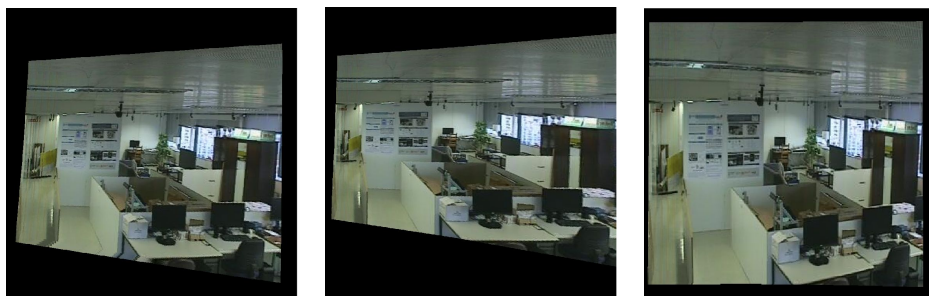


Figure A.12: Minimized variance, zoom level 1% (cube).

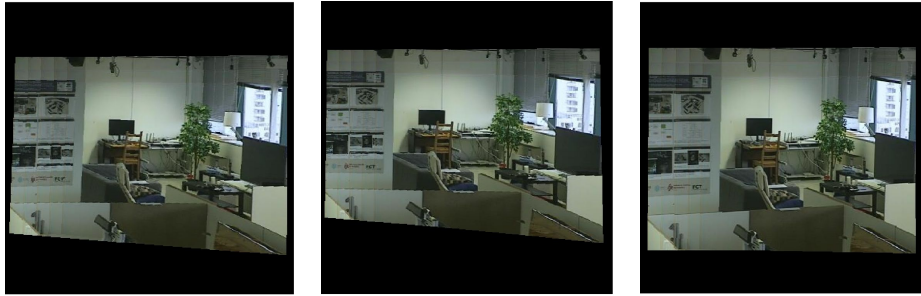


Figure A.13: Minimized variance, zoom level 10% (cube).

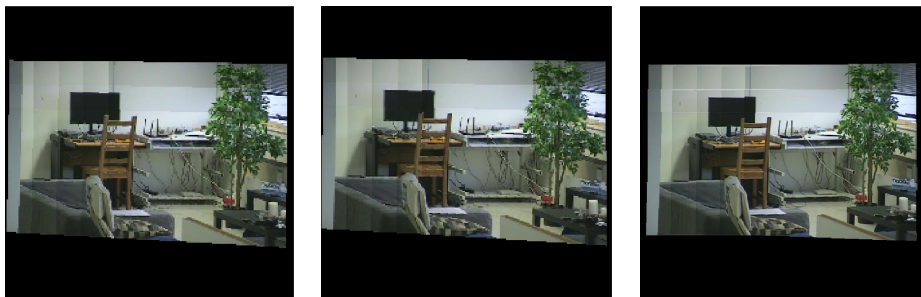


Figure A.14: Minimized variance, zoom level 30% (cube).

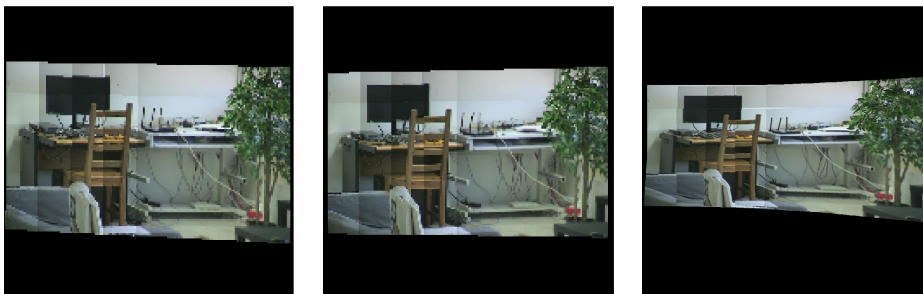


Figure A.15: Minimized variance, zoom level 50% (cube).

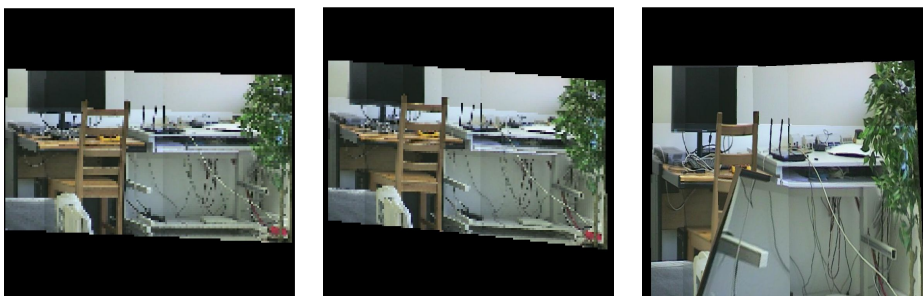


Figure A.16: Minimized variance, zoom level 60% (cube).

A.2.2 Cylinder Results



Figure A.17: Automatic calibration and estimated odometry, zoom level 1% (cylinder).



Figure A.18: Automatic calibration and estimated odometry, zoom level 20% (cylinder).



Figure A.19: Automatic calibration and estimated odometry, zoom level 40% (cylinder).

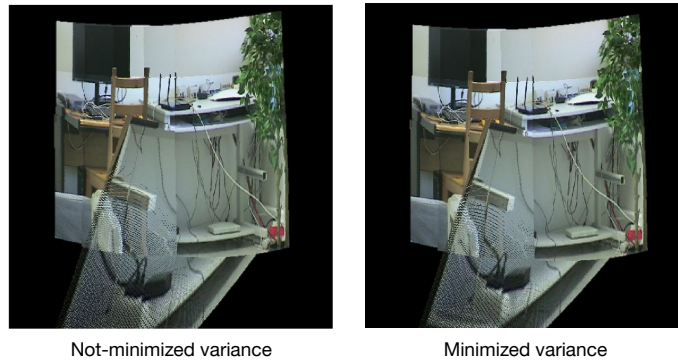


Figure A.20: Automatic calibration and estimated odometry, zoom level 60% (cylinder).

A.3 Image Blending Results

The results shown hereafter correspond to the different blending approaches mention in section 5.4. They were applied to the optimized automatic calibration results with estimated odometry.

A.3.1 Cube



Figure A.21: Blending results, zoom level 1% (cube).

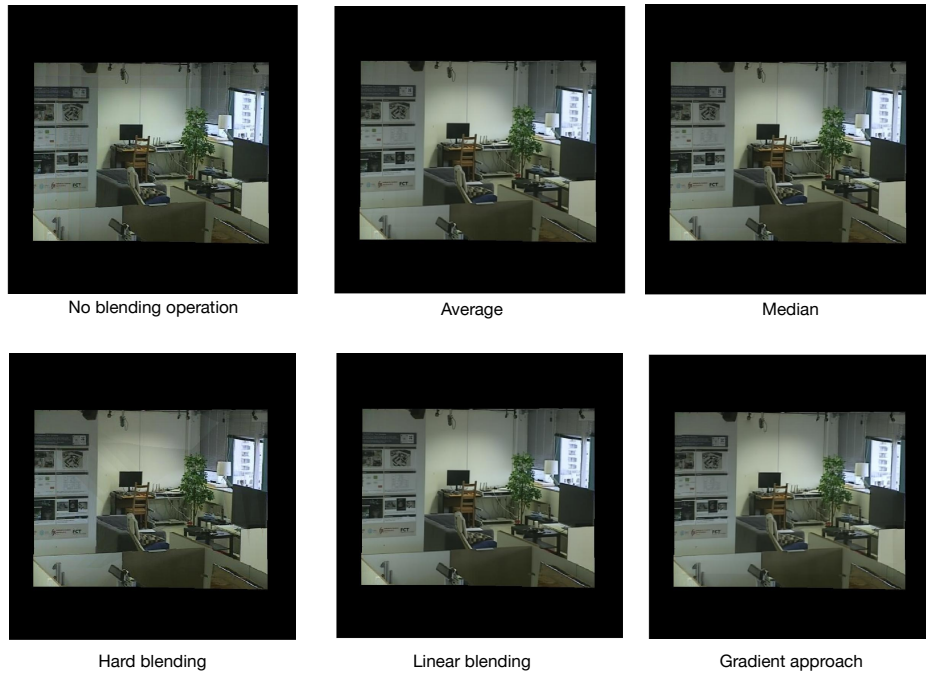


Figure A.22: Blending results, zoom level 10% (cube).

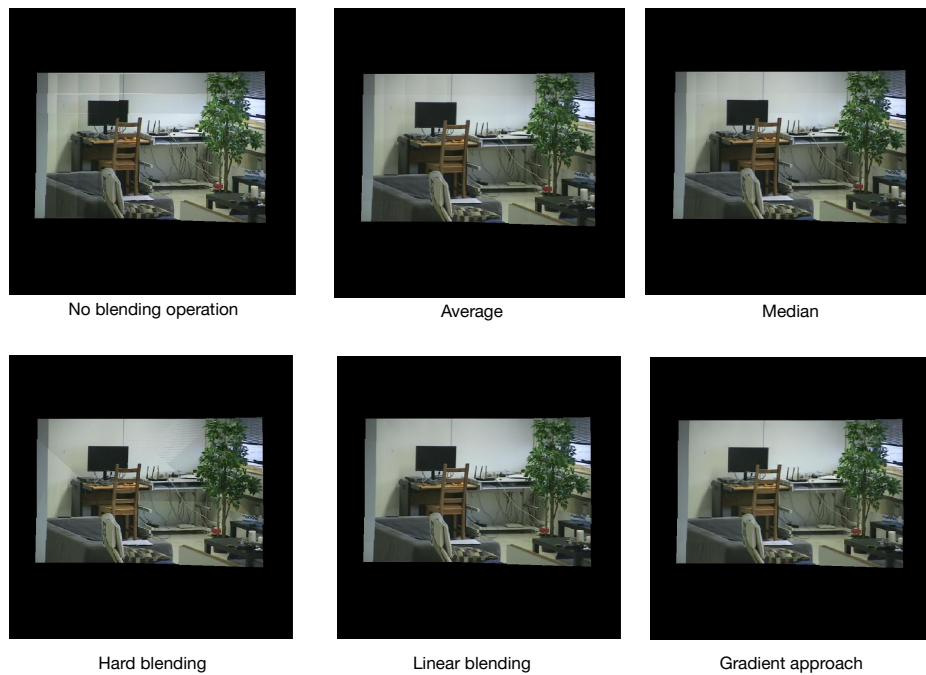


Figure A.23: Blending results, zoom level 30% (cube).

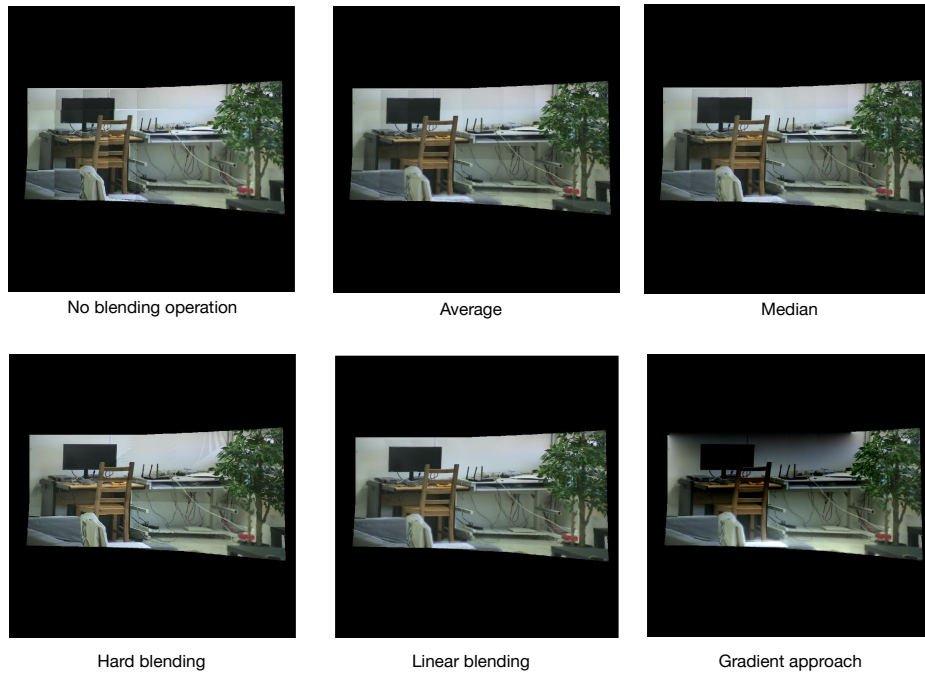


Figure A.24: Blending results, zoom level 50% (cube).

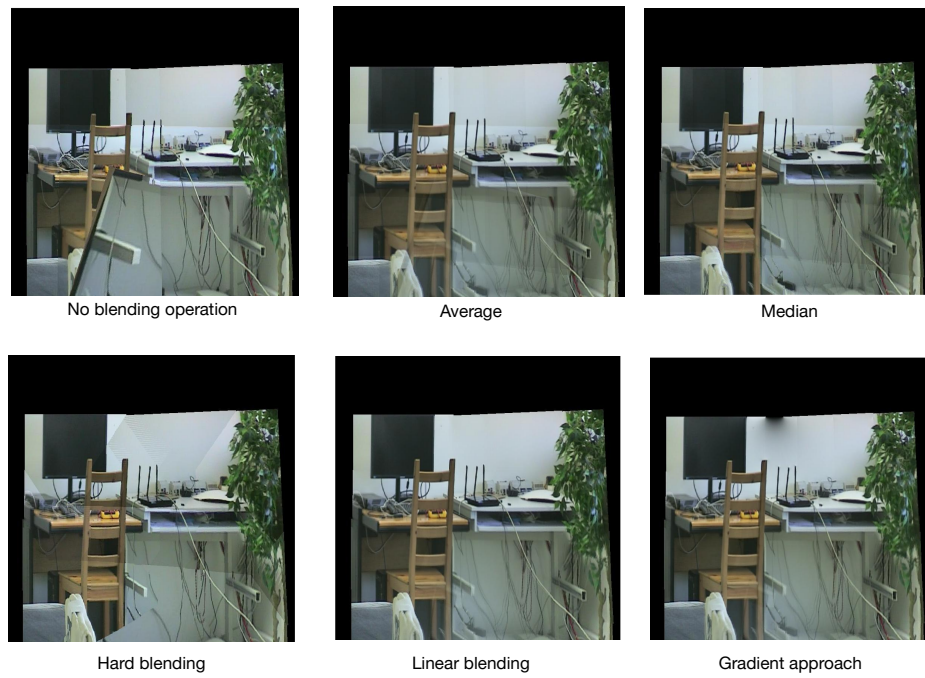


Figure A.25: Blending results, zoom level 60% (cube).

A.3.2 Cylinder

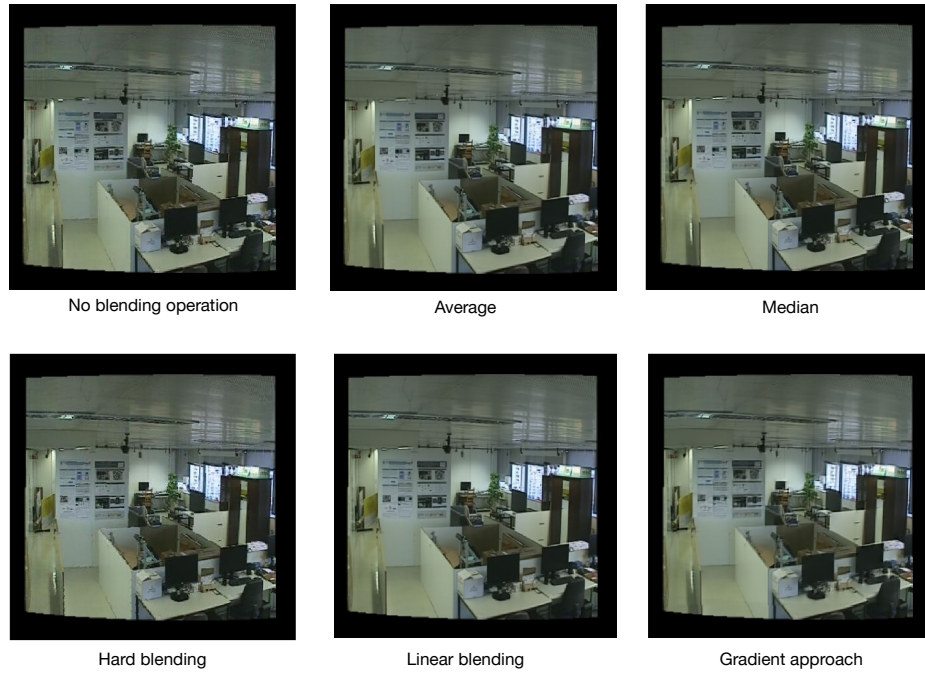


Figure A.26: Blending results, zoom level 1% (cylinder).

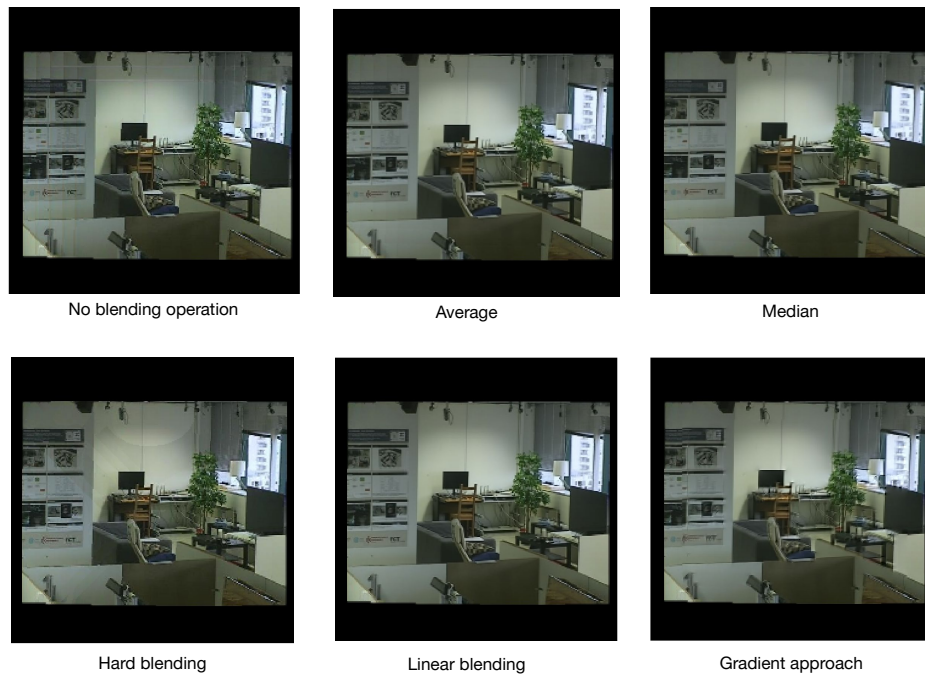


Figure A.27: Blending results, zoom level 10% (cylinder).

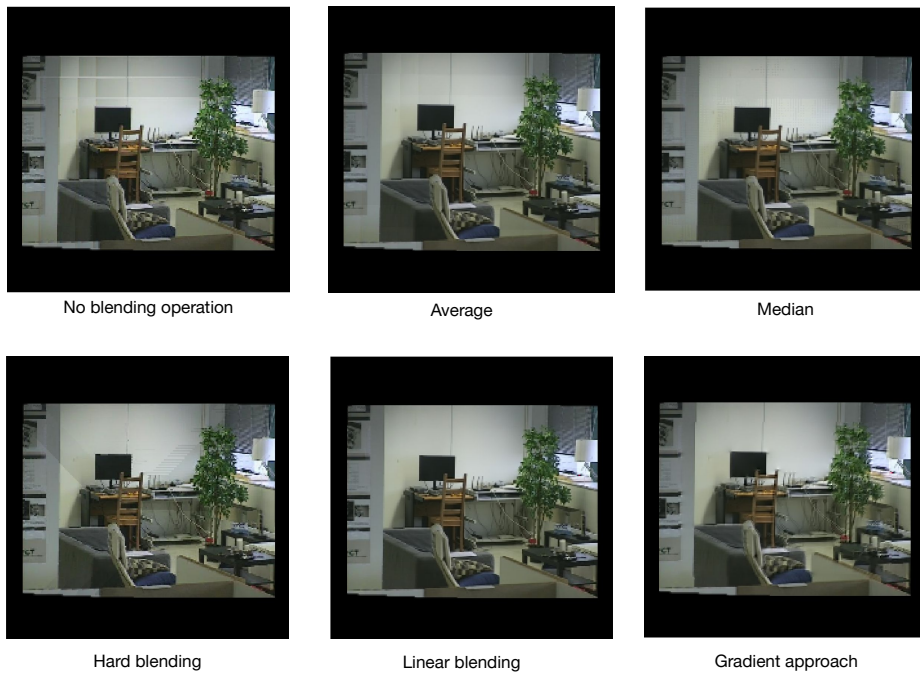


Figure A.28: Blending results, zoom level 20% (cylinder).

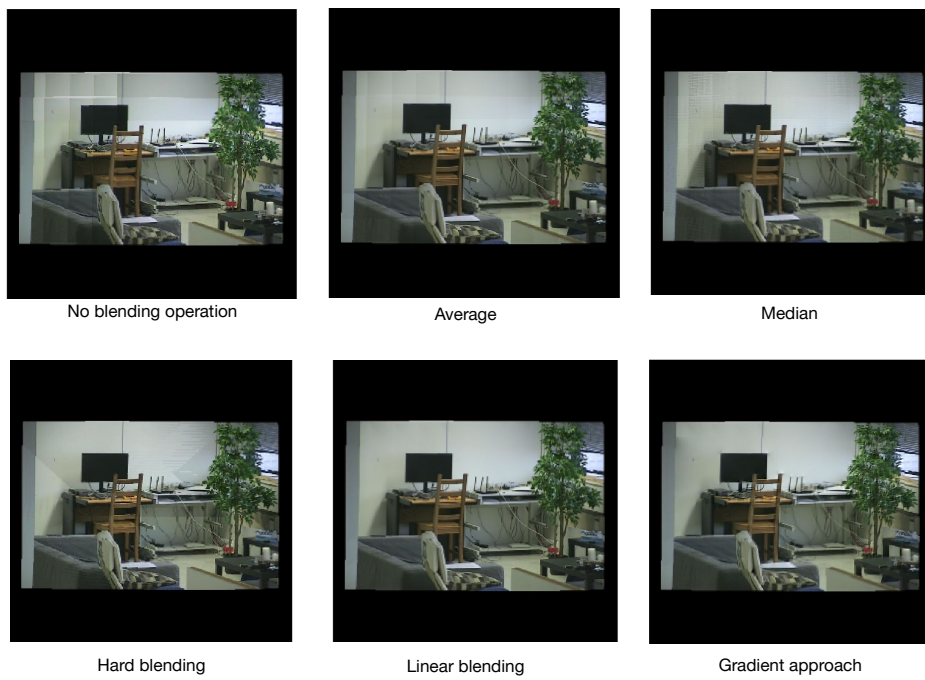


Figure A.29: Blending results, zoom level 30% (cylinder).

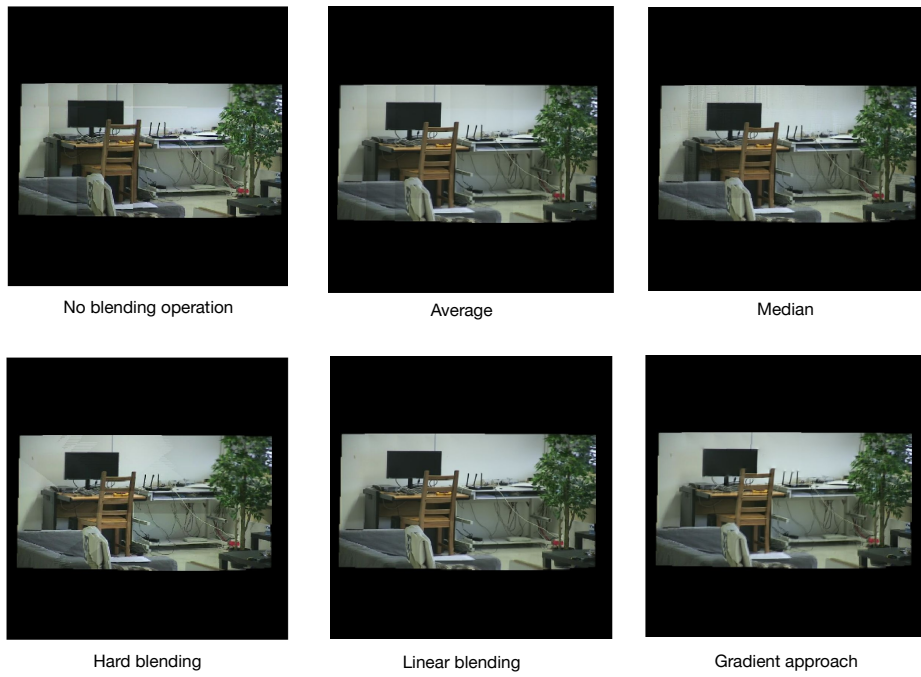


Figure A.30: Blending results, zoom level 40% (cylinder).

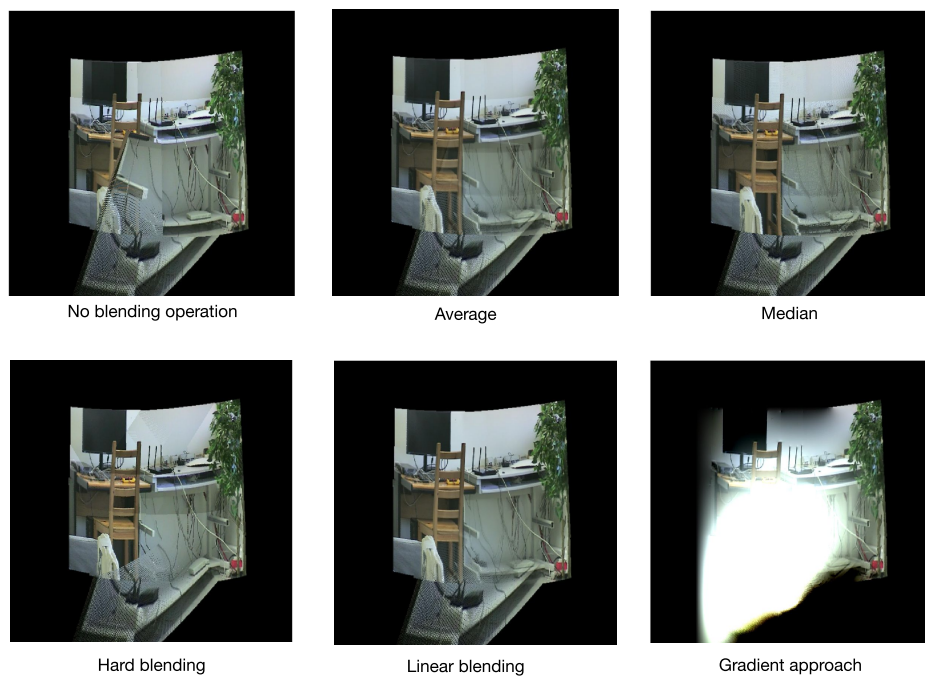


Figure A.31: Blending results, zoom level 60% (cylinder).