



**Universidade do Estado do Rio de Janeiro**  
Centro de Tecnologia e Ciências  
Instituto Politécnico

Gabriel Conceição Andrade

**An introduction to dynamical camera analysis with analytical  
mechanics and surface path motion**

Nova Friburgo

2024

Gabriel Conceição Andrade

**An introduction to dynamical camera analysis with analytical mechanics and  
surface path motion**



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Modelagem Computacional, da Universidade do Estado do Rio de Janeiro.

Orientador: Prof. Ricardo Fabbri

Orientador: Prof. Germano Monerat

Nova Friburgo

2024

CATALOGAÇÃO NA FONTE  
UERJ / REDE SIRIUS / BIBLIOTECA CTC/E

A553 Andrade, Gabriel Conceição.  
An introduction to dynamical camera analysis with analytical  
mechanics and surface path motion / Gabriel Conceição Andrade. -  
2024.  
73 f. : il.

Orientadores: Ricardo Fabbri e Germano Monerat.  
Dissertação (mestrado) – Universidade do Estado do Rio de  
Janeiro, Instituto Politécnico.

1. Visão computacional - Teses. 2. Mecânica analítica - Teses. 3.  
Sistemas dinâmicos – Teses. 4. Reconstrução de imagens – Teses. I.  
Fabbri, Ricardo. II. Monerat, Germano. III. Universidade do Estado  
do Rio de Janeiro. Instituto Politécnico. IV. Título.

CDU 004.922:531

Bibliotecária Cleide Sancho CRB7/5843

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta  
dissertação, desde que citada a fonte.

---

Assinatura

---

Data

Gabriel Conceição Andrade

**An introduction to dynamical camera analysis with analytical mechanics and  
surface path motion**

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Modelagem Computacional, da Universidade do Estado do Rio de Janeiro.

Aprovada em 10 de julho de 2024.

Banca Examinadora:

---

Prof. Ricardo Fabbri (Orientador)  
Instituto Politécnico – UERJ

---

Benjamin Kimia - Professor of Engineering  
Brown University

---

Julio César Guimarães Tedesco - Professor associado  
Instituto Politécnico – UERJ

Nova Friburgo

2024

## DEDICATION

I dedicate this text to my parents, grandparents and friends that I made along the way.

## **ACKNOWLEDGEMENTS**

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

Hegel remarks somewhere[\*] that all great world-historic facts and personages appear, so to speak, twice. He forgot to add: the first time as tragedy, the second time as farce.

*Karl Marx*

## ABSTRACT

ANDRADE, G.C. **An introduction to dynamical camera analysis with analytical mechanics and surface path motion.** 2024. 73 f. Dissertação (Mestrado em Modelagem Computacional) – Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Nova Friburgo, 2024.

Structure from motion (SfM) is a computer vision problem that aims to obtain three-dimensional scenes from a set of images captured from different viewpoints without prior knowledge of the camera configuration. SfM involves two primary steps: camera pose estimation and geometric recovery. In the initial phase, algorithms are employed to determine the position of a camera based on a single image and 3D scene, or using two or three images where the objective is to utilize one image to determine the positions of the other two cameras relative to it. The current investigation commenced by implementing a three-view-based reconstruction within a widely-used structure from motion system and examining the regions deemed problematic in the dynamical system of the model arising from fast polynomial homotopy continuation. Due to the high complexity of the model, locating these regions was deemed impractical at the time. The model itself was constructed using generic formulations adapted to the problem, necessitating an alternative approach. The work of Fabbri and Kimia provides foundational concepts of differential geometry for multiview reconstruction and basic notions on continuous time approaches for cameras, focusing on the velocity and acceleration of curves. Analyzing the camera structure, similarities with Lagrangian and Hamiltonian mechanics were identified. This work extends the concepts of Fabbri and Kimia by incorporating Lagrangian and Hamiltonian mechanics and introducing a model of camera path constrained by a surface. This constraint is utilized not only because it replicates practical applications but also simplifies both camera estimation and phase space analysis, which is employed to study the problematic regions and their implications for the formed image. Ultimately, a streamlined procedure for camera pose estimation for one or more cameras is proposed, along with conclusions and directions for future research.

Keywords: 3D reconstruction; computer vision; dynamical systems theory; analytical mechanics.

## RESUMO

ANDRADE, G.C. **Uma introdução à análise de câmera dinâmica com mecânica analítica e movimento sobre superfícies de caminho.** 2024. 73 f. Dissertação (Mestrado em Modelagem Computacional) – Instituto Politécnico, Universidade do Estado do Rio de Janeiro, Nova Friburgo, 2024.

Estrutura a partir de movimento (do inglês *Structure from motion* ou SfM) constitui um problema de visão computacional cujo objetivo é obter cenas tridimensionais a partir de um conjunto de imagens capturadas de diferentes pontos de vista, sem conhecimento prévio da configuração da câmera. O SfM compreende duas etapas fundamentais: a estimação da pose da câmera e a recuperação geométrica. Na primeira etapa, algoritmos buscam determinar a posição da câmera com base em uma única imagem e cena 3D, ou com duas ou três imagens, onde o objetivo é utilizar uma delas como referência para encontrar as outras. A pesquisa apresentada iniciou-se com a implementação de um método de reconstrução baseado em três câmeras, em um sistema de estrutura a partir de movimento amplamente utilizado, investigando as regiões consideradas problemáticas no sistema dinâmico do modelo proveniente da homotopia contínua polinomial rápida. Devido à alta complexidade do modelo, a identificação dessas regiões tornou-se impraticável e, à época, o modelo foi construído utilizando-se um modelo genérico adaptado ao problema, o que motivou a adoção de uma abordagem diferente. O trabalho de Fabbri e Kimia oferece noções de geometria diferencial para a reconstrução de múltiplas vistas e noções básicas em abordagem com câmera contínua, com foco na velocidade e aceleração de curvas. Ao analisar a estrutura da câmera, observa-se que ela se assemelha àquela utilizada em mecânica Lagrangiana e Hamiltoniana. Este trabalho expande os conceitos de Fabbri e Kimia, introduzindo mecânica Lagrangiana e Hamiltoniana, além de um modelo de caminho de câmera sob vínculo de superfície. Este vínculo é utilizado não apenas por reproduzir um modelo prático, mas também por simplificar tanto a estimação da câmera quanto a análise do espaço de fase, que é utilizado para estudar as regiões consideradas problemáticas e o que elas representam em relação à imagem formada. Ao final, propõe-se um procedimento para estimação de câmera para uma ou mais câmeras, além de serem apresentadas conclusões e sugestões para trabalhos futuros decorrentes desta pesquisa.

Palavras-chave: reconstrução 3D; visão vomputacional; teoria de sistemas dinâmicos; mecânica analítica.

## LIST OF FIGURES

Figure 1	- An general approach of single view methods. . . . .	14
Figure 2	- An general approach of bifocal method. . . . .	15
Figure 3	- An trifocal method example. . . . .	16
Figure 4	- Capitol High experimental frames. . . . .	17
Figure 5	- <i>Absolute Vodka</i> experimental images with no rotation. . . . .	17
Figure 6	- Notation for P3P lambdatwist. . . . .	23
Figure 7	- Five-point algorithm schematic illustration. . . . .	25
Figure 8	- A representation of the trifocal problem from oriented points codena- med Chicago . . . . .	28
Figure 9	- Plane real $\times$ imaginary homotopy. . . . .	30
Figure 10	- The 14 variables homotopy path generated by MiNuS Chicago problem. . . . .	31
Figure 11	- An ilustration of 2D image manifold to 2D local space. . . . .	39
Figure 12	- An ilustration of 3D scene manifold to 2D image. . . . .	40
Figure 13	- A representation of a particle in non-inertial frame. . . . .	43
Figure 14	- 3D camera distribution from synthcurves . . . . .	47
Figure 15	- Blender 3D barcelona chair sequence with its current view . . . . .	48
Figure 16	- Phase space $p_\theta \times \theta$ ( $p_0$ and $q_0$ respectively) . . . . .	53
Figure 17	- A coffee bag example. . . . .	54
Figure 18	- Coffee bottle example. . . . .	55

## LIST OF TABLES

Table 1	- Gauss elimination table. . . . .	26
Table 2	- Polynomial constraint matrix in $z$ . . . . .	27
Table 3	- Static camera estimation algorithms table. . . . .	56
Table 4	- Dynamic camera estimation algorithms table. . . . .	56

## LIST OF SYMBOLS

$\Gamma^w$	3D point world coordinates[Fabbri and Kimia IJCV2016]
$\Gamma^r$	Referential 3D point
$\Gamma_s^r$	Referential 3D point geometric derivative in camera space
$\Gamma_t^r$	Referential 3D point temporal derivative in camera space
$\Gamma$	3D point in camera space
$\Gamma_s$	3D point geometric derivative in camera space
$\Gamma_t$	3D point temporal derivative in camera space
$\gamma$	Projected 2D point camera space
$\gamma_s$	Projected 2D point camera space geometric derivative
$\gamma_t$	Projected 2D point camera space temporal derivative
$\rho$	Depth factor
$\rho_s$	Depth factor geometric derivative
$\rho_t$	Depth factor temporal derivative
$\mathbf{T}^r$	Referential 3D tangent
$\mathbf{T}_s^r$	Referential 3D tangent geometric derivative
$\mathbf{T}_t^r$	Referential 3D tangent temporal derivative
$\mathbf{T}$	3D tangent in camera space
$\mathbf{T}_s$	3D tangent geometric derivative in camera space
$\mathbf{T}_t$	3D tangent temporal derivative in camera space
$\mathbf{t}$	Projected 2D tangent camera space
$\mathbf{t}_s$	Projected 2D tangent geometric derivative camera space
$\mathbf{t}_t$	Projected 2D tangent temporal derivative camera space
$s_t$	Geometric parameter temporal derivative
$\mathbf{R}$	Rotation matrix
$\mathbf{R}_t$	Rotation matrix temporal derivative
$\mathcal{T}$	Translation matrix
$\mathcal{T}_t$	Translation matrix temporal derivative
$\theta$	Elevation angle
$\phi$	Azimuth angle
$r$	radius

## CONTENTS

	<b>INTRODUCTION</b> . . . . .	13
1	<b>PREVIOUS WORK</b> . . . . .	21
1.1	<b>One-view estimation</b> . . . . .	21
1.1.1	<u>Direct Linear Transform</u> . . . . .	21
1.1.2	<u>Perspective-three-point algorithm</u> . . . . .	22
1.2	<b>Bifocal geometry</b> . . . . .	25
1.2.1	<u>Five point algorithm</u> . . . . .	25
1.3	<b>Trifocal geometry</b> . . . . .	28
2	<b>BASIC CONCEPTS OF TIME-SPACE OPTICAL DYNAMICS</b>	33
2.1	<b>Differential geometry and dynamics</b> . . . . .	33
2.2	<b>Lagrangian and Hamiltonian mechanics</b> . . . . .	35
2.3	<b>Topology of image</b> . . . . .	37
2.3.1	<u>Image as Galilean group</u> . . . . .	39
2.4	<b>The point vs position problem</b> . . . . .	41
2.5	<b>Special case: Uncalibrated camera dynamics</b> . . . . .	41
2.6	<b>Translation from physics to SfM</b> . . . . .	42
3	<b>DYNAMICAL CAMERA MODEL UNDER SURFACE MOTION</b>	
	<b>PATH</b> . . . . .	46
3.1	<b>Model inspirations</b> . . . . .	46
3.2	<b>Modeling assumptions</b> . . . . .	46
3.3	<b>The Lagrangian of camera positioning</b> . . . . .	48
3.4	<b>Phase space analysis of camera position under spherical surface constraint</b> . . . . .	50
3.4.1	<u>Possible relationships between image and cameras in energy levels</u> . . . . .	53
3.4.1.1	Absolute single camera estimation problem . . . . .	54
3.4.1.2	Absolute multiple camera estimation problem . . . . .	54
3.4.1.3	Relative camera estimation problem . . . . .	55
3.5	<b>Dynamic camera pose estimation</b> . . . . .	56
3.5.1	<u>Absolute pose estimation</u> . . . . .	57
3.5.2	<u>Relative pose estimation</u> . . . . .	58
	<b>CONCLUSION AND FUTURE WORK</b> . . . . .	60
	<b>REFERENCES</b> . . . . .	63
	<b>APPENDIX A</b> – How compute stacked form from five point . . . . .	66
	<b>APPENDIX B</b> – How we compute $\Gamma$ , $\gamma$ geometric tangents, velocities and acceleration with $s = s(t)$ . . . . .	67
	<b>APPENDIX C</b> – Quick overview of Frenet frames . . . . .	70

<b>APPENDIX D</b> – Lagrangian and Hamiltonian metrics . . . . .	71
<b>APPENDIX E</b> – Hamiltonian form of axial geometric optics . . . . .	73

## INTRODUCTION

Structure from motion (SfM) is a computer vision problem that aims to obtain three-dimensional scenes from a set of images captured from different viewpoints without prior knowledge of the camera configuration. SfM involves two essential steps: camera pose estimation and geometric recovery.

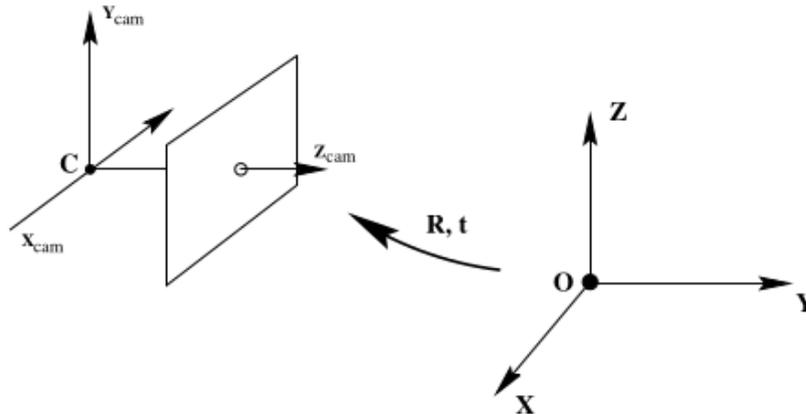
Geometric recovery aims to ascertain 3D points, curves, and surface properties, such as tangency, curvature, and torsion. Prominent works, including those by [Cipolla e Giblin \(1999\)](#) and [Fabbri e Kimia \(2016\)](#), develop concepts of differential geometry for curves and surfaces. Fabbri and Kimia demonstrate how these principles can be applied as an alternative to standard key-point applications, enabling the extraction of a complete curve sketch of the object [Fabbri e Kimia \(2010\)](#). This approach contrasts with traditional point-cloud geometric recovery methods by providing a more comprehensive representation of the object’s geometry.

Camera estimation, which precedes geometric recovery, endeavors to determine either the global position (referred to as absolute pose estimation) of a single or multiple cameras within a scene, or the relative positioning of a set of images with respect to a selected view (referred to as relative pose estimation). These processes are integral components of widely-utilized software, such as `OpenMVG` [Moulon et al. \(2016\)](#) and `Colmap` [Schönberger e Frahm \(2016\)](#), [Schönberger et al. \(2016\)](#). In these applications, an incremental (or sequential) pipeline structure is employed. Within this framework, relative pose estimation is used to establish an initial set of 3D points, and absolute pose estimation is subsequently applied to compute the positions of the remaining cameras sequentially, a process also known as resection.

### Drawbacks of absolute single view camera estimation

Single camera estimation holds significant relevance for computing the camera position by correlating a 3D scene with a single image. Various algorithms have been developed for this purpose. The classic **Direct Linear Transformation** (DLT) algorithm, as discussed by [Hartley e Zisserman \(2004\)](#), is one such method. The most widely utilized algorithm in standard multiview reconstruction software is the **three-point algorithm** (P3P) [Persson e Nordberg \(2018\)](#). More recently, the **two-point algorithm with tangent** (P2Pt) [Fabbri, Giblin e Kimia \(2020\)](#), which incorporates an additional geometric tangent constraint, has demonstrated notable improvements. However, this algorithm has only been validated experimentally and has yet to be extensively tested within these software applications.

Figure 1 – An general approach of single view methods.



Legend: Schematic representation of single view methods.

Source: [Hartley e Zisserman \(2004\)](#)

The general issues associated with these algorithms are as follows:

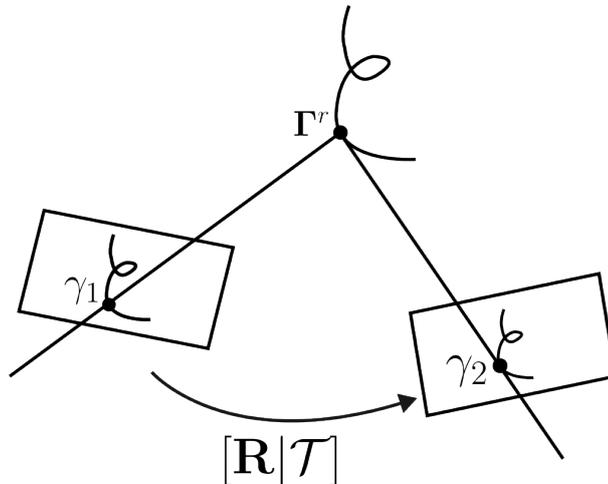
- Lack of image depth introduces an additional problem that can generate an ambiguity in scale, leading to issues in translation estimation [Helou, Shahpaski e Süstrunk \(2019\)](#). Many recent works, such as P3P, attempt to address this issue by determining the depth before computing the position of the camera.
- Many algorithms express their final equations with rotation representation unknowns that are not statistically optimal. For instance, using quaternions or Cayley parameters rather than axis-angle representation with a Gaussian error distribution.

## Drawbacks of two-view camera estimation

Two-view camera estimation, also referred to as bifocal estimation, constitutes the principal method for relative pose estimation. While single-view estimation is theoretically optimal due to its direct approach, practical application is infrequent because of the insufficient availability of 3D geometric information, thereby necessitating exclusive reliance on image data.

The fundamental principle involves computing the relative pose of one image with respect to another. This is typically achieved by solving the equation  $\gamma_2 \rho_2 = \mathbf{R} \gamma_1 \rho_1 + \mathcal{T}$ , where  $\mathbf{R}$  denotes rotation,  $\mathcal{T}$  represents translation, and  $\gamma$  and  $\rho$  correspond to the image point and depth factor, respectively. An epipolar constraint  $\gamma_2 \rho_2 \mathbf{E} \gamma_1 \rho_1 = 0$  is imposed, where  $\mathbf{E}$  is the essential matrix, defined as  $\mathbf{E} = \mathcal{T}_\times \mathbf{R}$ .

Figure 2 – An general approach of bifocal method.



Legend: Schematic representation of bifocal methods goal.

Source: The author, 2024.

The basic algorithm is the **eight-point algorithm** Hartley (1997), but the most used one is **five-point algorithm** Nistér (2004) both with similar structure, but the first one computes the fundamental matrix, which is more general considering unknown camera matrices and the second considers that they are known.

Approaches to reconstruction utilizing two-view estimation to generate the initial seed for reconstruction also exist. These methods often employ neural networks Xiao et al. (2023). However, they necessitate a significant quantity of input images to achieve precise reconstruction.

The general issues associated with these algorithms are as follows:

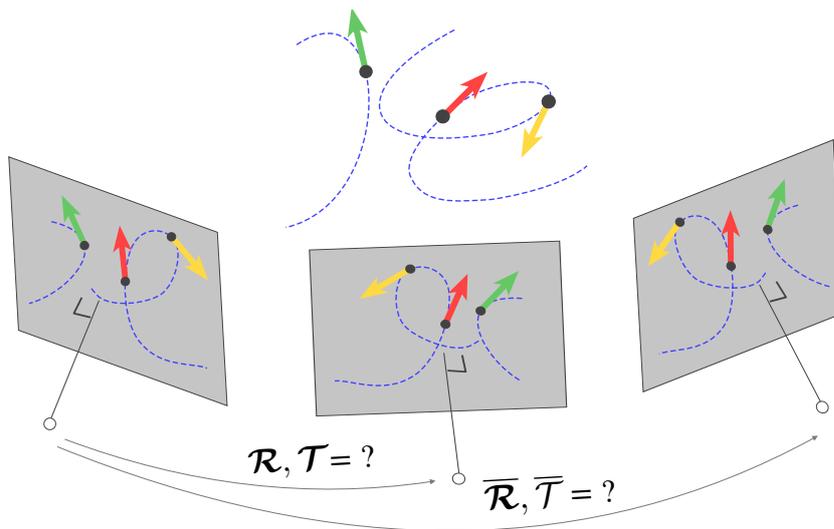
- Dependence on the geometric structure of each image: Images with ambiguous structures, such as rounded curves, can cause the algorithm to fail in reliably determining the camera position using RANSAC.
- Texture dependency: The number of detectable features by standard feature detection algorithms is reduced for objects with plain textures, leading to unreliable camera position estimation.
- Reflectivity and transparency: Reflective and transparent surfaces can introduce ambiguities in feature detection, causing failures in both detection algorithms and the estimator. These challenges are thoroughly described in ARkit (2018).
- Distant camera images: Two-view camera estimation often fails for images representing distant views due to a reduced number of common features across them.

## Drawbacks of three-view camera estimation

The three-view camera estimation, also referred to as trifocal estimation, is thoroughly detailed in chapters 15 and 16 of [Hartley e Zisserman \(2004\)](#). A more practical implementation of this method is realized through the `MInimal problem NUmerical continuation Solver` (MiNuS) which is available at <https://github.com/rfabbri/minus.git>.

The primary objective of the trifocal method is to augment the previously mentioned software by offering an alternative solution in cases where bifocal estimation is inadequate. Additionally, three views are necessary for curve-based structure from motion. Although it may seem to simply involve an additional image in the system, the trifocal method employs a distinct computational approach compared to the previous relative pose estimators. For instance, MiNuS utilizes different classes of problems to compute the trifocal tensor in a unique manner.

Figure 3 – An trifocal method example.



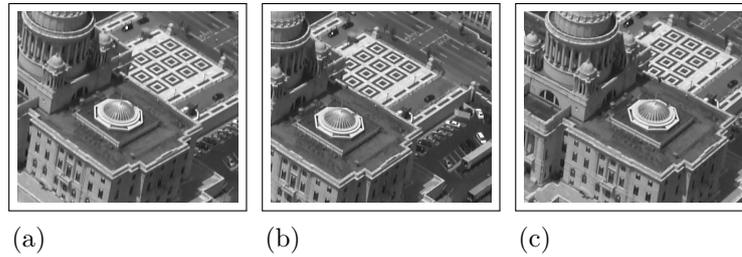
Legend: An illustration showing what is the final goal of trifocal implementations.

Source: [Fabbri e Kimia \(2016\)](#)

During the initial year of our research, we conducted experiments with MiNuS's standard differential equations and addressed formulation issues. Using the Capitol High Building test, we presented findings at the *XXVI Encontro Nacional de Modelagem Computacional e XIV Encontro de Ciência e Tecnologia dos Materiais* [Andrade et al. \(2023\)](#), indicating that out of 199 correspondences, the least favorable scenario yielded 68.84% inliers, while the most favorable achieved 79.65% inliers from the triplet of images 4.

Furthermore, the stability of trifocal estimation was demonstrated by establishing the minimal number of points necessary to compute camera positions using rotation-free

Figure 4 – Capitol High experimental frames.

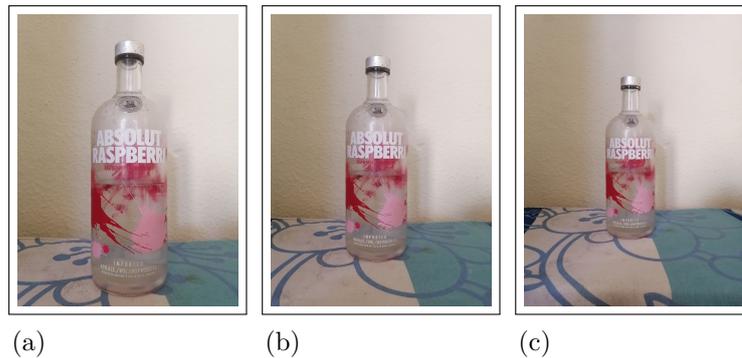


Legend: Selected frames for trifocal testing (a) frame\_00001, (b) frame\_00030 and (c) frame\_00066.

Source: [Fabbri e Kimia \(2010\)](#)

images from a curated bottle dataset [5](#).

Figure 5 – *Absolute Vodka* experimental images with no rotation.



Legend: The stability of trifocal was shown by getting at least the minimum of points to enable the trifocal model to recover the camera positions.

Source: The author, 2024.

The general issues associated with these algorithms are as follows:

- Testing with the official openMVG demo dataset, [Sceaux Castle](#), revealed that camera distance remains a significant issue. If at least one of the three images is sufficiently sparse, the trifocal tensor cannot be computed. This was determined by testing each possible triplet from the dataset.
- The trifocal method still encounters issues similar to those in two-view camera estimation concerning transparency, reflectivity, and texture of the material. In-

troducing rotation into the bottle dataset resulted in an inability to compute the camera positions.

- Some features generated by the solver, upon analysis of the correspondences, were found to be incorrect in the third image correspondence or at least within the threshold. This was observed in tests with the Capitol High frames 4, where each inlier considered by the model was analyzed.

### Advantages of dynamical camera estimation

Continuous camera approaches, also referred to as dynamical methods, are discussed in works such as [Ma et al. \(2004\)](#), providing valuable yet primarily limited insights to two-view camera estimation. [Fabbri e Kimia \(2016\)](#) further contribute to this field by introducing a version of motion in 3D camera space and 2D image projective space, focusing on point velocity and acceleration. This study advances their work by incorporating a novel approach using Lagrangian and Hamiltonian mechanics.

The application of functionals is prevalent in physics and differential geometry for computing the least path between two points or determining a curve that minimizes the geometric derivative of the curvature function, as elucidated in [Kimia, Frankel e Popescu \(2003\)](#). By employing a similar approach in the temporal domain and utilizing analytical mechanics, functionals can be effectively applied in continuous time camera estimation, a method referred to as the dynamical approach.

It is important to address a significant aspect derived from discussions with physicists: when importing ideas from mechanics, certain aspects pertinent to formalism, such as physical quantities, are considered in our context as parameterization values. These can be viewed, to some extent, as information-theoretic interpretations of the physical concepts.

### Main contribution

Our main contributions are outlined as follows:

- Expansion of existing theory through the integration of Lagrangian and Hamiltonian mechanics. This involves a foundational review based on [Fabbri e Kimia \(2016\)](#), introducing novel concepts such as image topology. It is illustrated that images can be interpreted as part of the Galileo group at a high level, along with dynamics of the camera calibration matrix.
- Review of physics literature, particularly in analytical mechanics and physical geo-

metry, to establish connections between Structure from Motion (SfM) and mechanics. Building upon concepts observed in works by [Ma et al. \(2004\)](#) and [Fabbri e Kimia \(2016\)](#), the discussion extends to include the Lagrangian formulation of 3D curve points in camera space and its applicability to projected 2D points. Additionally, Fermat's principle for optical cameras, as described by [Holm \(2008a\)](#), [Holm \(2008b\)](#), is introduced as a foundation for future work.

- Development of a practical model incorporating a surface constraint—specifically a spherical surface—where cameras move freely or under a simulated potential field constraint. Phase space analysis is conducted, establishing an association between orbital energy levels and "states of the image."
- Proposal of a practical camera estimation procedure based on this theory. Leveraging the camera surface path model, the minimum number of required points is minimized, employing tangents as an additional constraint for rotational estimation. The proposal outlines a unified procedure applicable to both absolute and relative pose estimation.

## Dissertation organization

In Chapter 1, the default algorithms used by the community for estimating camera positioning, such as DLT, P3P for absolute positioning, the five-point algorithm for bifocal estimation, and the trifocal method using MiNuS as a framework, are presented.

In Chapter 2, fundamental concepts and innovative ideas are introduced. Initially, Fabbri and Kimia's theories on point and tangent velocities are presented, which are subsequently extended to incorporate time-dependent geometric parameters. The importance of these parameters in camera estimation within conventional frameworks is discussed. Following this, the principles of Lagrangian and Hamiltonian mechanics are explored, with a detailed examination of Hamiltonian systems of differential equations. A distinction is made between geometric and optical considerations, highlighting the cameras' fundamental relationship with mechanics and geometry. Additionally, brief discussions on image topology concepts are provided, followed by an overview analysis of the dynamics of projection matrices and camera matrices.

In Chapter 3, the model assumptions, sources of inspiration, literature review connecting physics with Structure from Motion (SfM), historical development overview, and the model itself are presented. This includes a detailed analysis of phase space and hypotheses regarding the relationship between energy orbitals and their effects on images. Finally, the presented theory proposes a procedure for camera pose estimation applicable to both absolute and relative pose estimation.

In Chapter [3.5.2](#), a summary of the study's achievements is presented alongside future research directions.

## 1 PREVIOUS WORK

In this section, detailed discussions are presented on the algorithms utilized for camera estimation. The primary objective is to provide a thorough description of these algorithms as they are employed in the estimation process. A concise review of previous work is necessary to highlight the advancements achieved in this study.

### 1.1 One-view estimation

The classical approach used for both camera estimation and error estimation is the Direct Linear Transform method (DLT), detailed in [Hartley e Zisserman \(2004\)](#). A contemporary family of algorithms known as Perspective-n-Points (PnP) is considered state-of-the-art for estimating single camera positions. Among these, the perspective-three-point algorithm (P3P) is the current standard, while the more cutting-edge perspective-two-point algorithm with a tangent (P2Pt) [Fabbri, Giblin e Kimia \(2020\)](#) has garnered significant attention.

The implementation of the P2Pt algorithm in practical software frameworks such as `OpenMVG` was developed as a library by Ariel Nogueira Kovaljski during his undergraduate final project, under the supervision of Prof. Fabbri. This implementation forms part of a larger package that includes the trifocal method (implemented by the author under similar conditions) and the P2Pt resection method. Given that P2Pt, along with trifocal method, is currently in the final stages of implementation (including testing and preparation for publication), this discussion will focus on DLT and P3P, which are extensively employed within the SfM community.

#### 1.1.1 Direct Linear Transform

The Direct Linear Transform (DLT) algorithm is defined as follows:

**Definition 1.1.1** *Given at least four points  $\Gamma$  and  $\gamma'$  being both 3D scene space and 2D image space, compute the projection matrix  $\mathbf{P}$  that  $\Gamma \leftrightarrow \gamma'$ .*

If the camera calibration matrix is known, the camera pose is computed as  $[\mathbf{R}|\mathcal{T}] = \mathcal{K}^{-1}\mathbf{P}$ . Otherwise, both the rotation and translation are derived directly from the QR decomposition of  $\mathbf{P}$ . The relationship between the 2D image and 3D representation is expressed by  $\gamma'_i = \mathbf{P}\Gamma'_i$ , where  $\gamma'_i = (x_i, y_i, z_i)^\top$  and  $\Gamma_i = (Z_i, Y_i, Z_i, 1)^\top$  denote homogenized

coordinates. The construction of the cross product  $\gamma'_i \times \mathbf{P}\Gamma'_i$  leads to equation (1).

$$\mathbf{A}_i \mathbf{p} = \begin{bmatrix} 0 & -z_i \Gamma'_i & y_i \Gamma'_i \\ z_i \Gamma'_i & 0 & -x_i \Gamma'_i \\ -y_i \Gamma'_i & x_i \Gamma'_i & 0 \end{bmatrix} \begin{bmatrix} P^1 \\ P^2 \\ P^3 \end{bmatrix} \quad (1)$$

Given a scale factor  $z_i$  where  $x_i$  and  $y_i$  are independent, construct  $\mathbf{M}$  using only the first two rows of  $\mathbf{A}_i$  from equation (2):

$$\mathbf{M}_i \mathbf{p} = \begin{bmatrix} 0 & -z_i \Gamma'_i & y_i \Gamma'_i \\ z_i \Gamma'_i & 0 & -x_i \Gamma'_i \end{bmatrix} \begin{bmatrix} P^1 \\ P^2 \\ P^3 \end{bmatrix} \quad (2)$$

Next, compute the Singular Value Decomposition (SVD) of the stacked matrix  $\mathbf{H} = (\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_4)^\top$  to obtain the vector corresponding to the minimum singular value, which is found in the last column of  $\mathbf{D}$  from  $\mathbf{H} = \mathbf{SAD}^\top$ .

DLT typically exhibits the following issues:

- The high computational complexity of DLT primarily arises from the Singular Value Decomposition (SVD), which can significantly slow down the algorithm's performance. When the calibration matrix is unknown, the process of obtaining rotation and translation becomes even more complex, often requiring QR decomposition of the projection matrix. These factors contribute to DLT's computational demands and can affect its efficiency in practical applications.
- The geometric ambiguity inherent in images can severely limit the number of discernible features available for camera estimation. This limitation often results in either poor performance of the estimation algorithms or, in some cases, the inability to estimate the camera pose altogether.
- A noisy 3D scene can lead to inaccurate selection of points, causing incorrect correspondences between images and the scene.
- Occluding curves can lead to insufficient features for accurately estimating camera positions using DLT.

### 1.1.2 Perspective-three-point algorithm

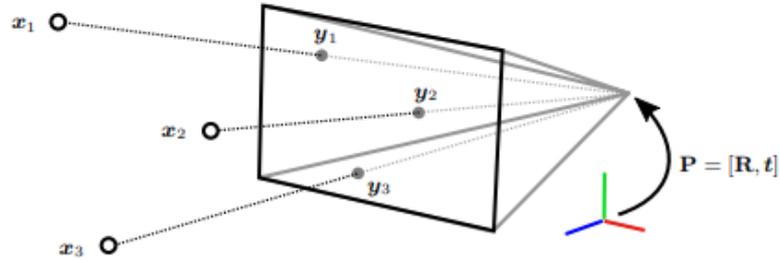
The current perspective-three-point algorithm, referred to as the **Lambda twist Pers-son e Nordberg (2018)**, is integrated into *openMVG* as the conventional resection method. The problem is formulated as follows:

**Definition 1.1.2** Given a calibrated pinhole camera, three 3D points  $\Gamma_i = (x_i, y_i, z_i)^\top$ , and corresponding homogeneous image coordinates  $y_i \sim (u_i, v_i, 1)$  such that  $|y_i| = 1$ , then:

$$\rho_i y_i = \mathbf{R} \gamma_i + \mathcal{T}, \quad i = 1, 2, 3 \quad (3)$$

where the rotation  $\mathbf{R} \in SO(3)$  and translation  $\mathcal{T} \in \mathbb{R}^3$ . Depending on the configuration of the points, P3P has up to four solutions.

Figure 6 – Notation for P3P lambdatwist.



Source: [Persson e Nordberg \(2018\)](#)

The initial phase of the method involves establishing invariant constraints by computing the distance between two points,  $\gamma_i$  and  $\gamma_j$ , as defined in (3). Furthermore, given that  $|\gamma_i| = 1$ , the values  $a_{ij}$  and  $b_{ij}$  are determined for  $i, j \in 12, 13, 23$  as shown in (4).

$$\begin{cases} |\lambda_i y_i - \lambda_j y_j|^2 = |x_i - x_j|^2 = a_{ij} \Rightarrow \lambda_i^2 + \lambda_j^2 - 2b_{ij} = a_{ij} \\ b_{ij} = y_i^\top y_j \end{cases} \quad (4)$$

From (4), five quadratic equations are derived: three inhomogeneous and two homogeneous. The inhomogeneous equations are represented in matrix form as (4), while the homogeneous equations are derived from these (5).

$$\begin{cases} \Lambda^\top \mathbf{M}_{12} \Lambda = a_{12}, \quad \Lambda^\top \mathbf{M}_{13} \Lambda = a_{13}, \quad \Lambda^\top \mathbf{M}_{23} \Lambda = a_{23} \\ \mathbf{D}_1 = \mathbf{M}_{12} \cdot a_{23} - \mathbf{M}_{23} \cdot a_{12} = (\mathbf{d}_{11}, \mathbf{d}_{12}, \mathbf{d}_{13}), \\ \mathbf{D}_2 = \mathbf{M}_{13} \cdot a_{23} - \mathbf{M}_{23} \cdot a_{13} = (\mathbf{d}_{21}, \mathbf{d}_{22}, \mathbf{d}_{23}) \\ \Lambda^\top \mathbf{D}_i \Lambda = 0, \quad i = 1, 2 \end{cases} \quad (5)$$

The matrices  $\mathbf{M}_{12}$ ,  $\mathbf{M}_{13}$ , and  $\mathbf{M}_{23}$  are defined as shown in (6).

$$\mathbf{M}_{12} = \begin{bmatrix} 1 & -b_{12} & 0 \\ -b_{12} & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{M}_{13} = \begin{bmatrix} 1 & 0 & -b_{13} \\ 0 & 0 & 0 \\ -b_{13} & 0 & 1 \end{bmatrix}, \mathbf{M}_{23} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -b_{23} \\ 0 & -b_{23} & 1 \end{bmatrix} \quad (6)$$

Next, an additional constraint  $\Lambda^\top \mathbf{D}_0 \Lambda = 0$  is constructed, where  $\mathbf{D}_0 = \mathbf{D}_1 + \alpha \mathbf{D}_2$ . Following this,  $\det(\mathbf{D}_0) = 0$  is computed to derive the cubic equation with coefficients as presented in (7).

$$\mathbf{c} = \begin{bmatrix} c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} \det(\mathbf{D}_2) \\ \mathbf{d}_{21}^\top(\mathbf{d}_{12} \times \mathbf{d}_{13}) + \mathbf{d}_{22}^\top(\mathbf{d}_{13} \times \mathbf{d}_{11}) + \mathbf{d}_{23}^\top(\mathbf{d}_{11} \times \mathbf{d}_{12}) \\ \mathbf{d}_{11}^\top(\mathbf{d}_{22} \times \mathbf{d}_{23}) + \mathbf{d}_{12}^\top(\mathbf{d}_{23} \times \mathbf{d}_{21}) + \mathbf{d}_{13}^\top(\mathbf{d}_{21} \times \mathbf{d}_{22}) \\ \det(\mathbf{D}_1) \end{bmatrix} \quad (7)$$

$\mathbf{D}_0$  undergoes diagonalization to simplify its form through eigenvalue decomposition, yielding  $\mathbf{D}_0 = \mathbf{E} \mathbf{S} \mathbf{E}^\top$ , where  $\mathbf{E}$  spans  $\mathbb{R}^3$  and  $\mathbf{S} = [\sigma_1 \mathbf{e}_1, \sigma_2 \mathbf{e}_2, 0 \mathbf{e}_3]$  with  $\sigma_1 > 0$  and  $\sigma_2 \geq 0$ . The zero in the third eigenvalue arises from the condition  $\det(\mathbf{D}_0) = 0$ , which necessitates at least one eigenvalue being zero.

Subsequently, the problem is formulated as  $\mathbf{p}^\top \mathbf{S} \mathbf{p} = 0$  with  $\mathbf{p} = \mathbf{E}^\top \Lambda$ . For each solution  $s$ ,  $\Lambda = (\lambda_1, \lambda_2, \lambda_3)^\top$  is computed. Geometric consistency is verified by reparameterizing  $\lambda_i$  in terms of  $\tau$  and solving the quadratic homogeneous equations, considering only real positive values as valid solutions. For each feasible solution,  $\Lambda_k^\top \mathbf{M}_{23} \Lambda_k = a_{23}$  is solved, discarding any solutions where  $\lambda_{1k} < 0$ . Finally,  $\mathbf{R}$  and  $\mathcal{T}$  are recovered using (8).

$$\begin{cases} \mathbf{R} = ((\lambda_1 \gamma_1 - \lambda_2 \gamma_2) \times (\lambda_2 \gamma_2 - \lambda_3 \gamma_3)) ((\boldsymbol{\Gamma}_1 - \boldsymbol{\Gamma}_2) \times (\boldsymbol{\Gamma}_2 - \boldsymbol{\Gamma}_3))^{-1} \\ \mathcal{T} = \lambda_i \gamma_i - \mathbf{R} \boldsymbol{\Gamma}_i \end{cases} \quad (8)$$

The challenges associated with this version of P3P, as documented in the paper and during meetings conducted in early 2020, are as follows:

- Both the paper and one of the authors reported that  $c_3$  or  $c_0$  can be zero, or computationally close to zero, leading to instability.
- As of the time of the meeting, there was no practical implementation of P3P in `openMVG` utilizing the quaternion form detailed in the appendix section.

## 1.2 Bifocal geometry

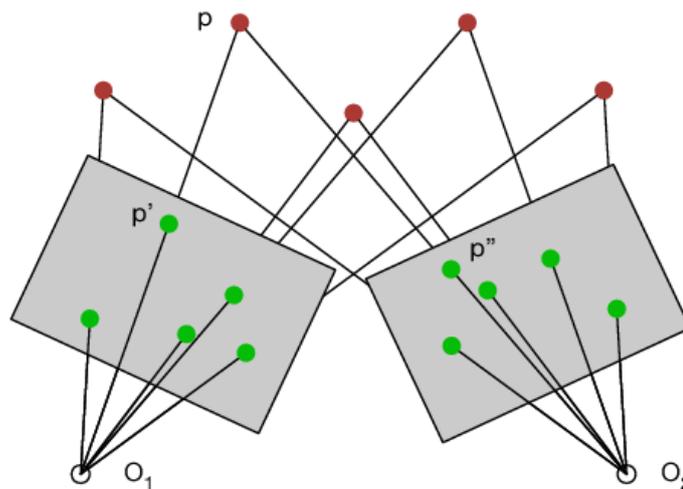
The two-view camera estimation, alternatively known as bifocal geometry or relative pose estimation, aims to compute rotation and translation matrices using information from one image relative to another. Traditionally, this is achieved through the eight-point algorithm, while the contemporary approach utilizes the five-point algorithm integrated with AContrario RANSAC (ACRANSAC) [Moulon, Monasse e Marlet \(2012\)](#), [Moisan, Moulon e Monasse \(2012\)](#), renowned for its robust model estimation capabilities. ACRANSAC enhances the standard methodology by dynamically adjusting the threshold during the model estimation process. Presented here are the foundational principles of the five-point algorithm applicable to both standard and ACRANSAC variants of RANSAC.

### 1.2.1 Five point algorithm

The five-point algorithm represents a fundamental method in Structure from Motion (SfM) for computing the relative pose between two views. This problem is typically formulated as follows:

**Definition 1.2.1** *Given at five points  $\gamma$  and  $\gamma'$  from two images, compute the essential matrix  $\mathbf{E}$  that relates  $\gamma \leftrightarrow \gamma'$ .*

Figure 7 – Five-point algorithm schematic illustration.



Source: [Rabozzi et al. \(2018\)](#)

The algorithm proposed by [Nistér \(2004\)](#) assumes that the cameras are calibrated,

thereby reducing the epipolar constraint to its simplified form (9).

$$\begin{cases} \tilde{\gamma}^\top \tilde{E} = 0 \\ \tilde{\gamma} = (\gamma_1 \gamma'_1, \gamma_1 \gamma'_2, \gamma_1 \gamma'_3, \gamma_2 \gamma'_1, \gamma_2 \gamma'_2, \gamma_2 \gamma'_3, \gamma_3 \gamma'_1, \gamma_3 \gamma'_2, \gamma_3 \gamma'_3)^\top \\ \tilde{E} = (e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33})^\top \end{cases} \quad (9)$$

and the additional constraints (10).

$$\begin{cases} \det(\mathbf{E}) = 0 \\ \mathbf{E}\mathbf{E}^\top \mathbf{E} - \frac{1}{2} \text{Tr}(\mathbf{E}\mathbf{E}^\top) \mathbf{E} = 0 \end{cases} \quad (10)$$

First, the null-space of the matrix representing all five points is obtained by performing a QR decomposition of a  $9 \times 9$  matrix, as described in equation (11) (see A).

$$[\tilde{\gamma}_1 \ \tilde{\gamma}_2 \ \tilde{\gamma}_3 \ \tilde{\gamma}_4 \ \tilde{\gamma}_5 | \mathbf{I}_{4 \times 9}]^\top, \quad (11)$$

Next, extract the four rightmost vectors from the decomposition to populate the  $3 \times 3$  matrices  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{Z}$ , and  $\mathbf{W}$ . These matrices are subsequently utilized within the essential matrix framework, represented as equation (12).

$$\mathbf{E} = x\mathbf{X} + y\mathbf{Y} + z\mathbf{Z} + w\mathbf{W}. \quad (12)$$

Here,  $x$ ,  $y$ ,  $z$ , and  $w$  represent scalars that, when normalized by a common factor, result in  $w = 1$ . The subsequent step involves deriving a system of 10 equations by expanding both conditions in (10). Partial Gaussian elimination is then applied, following the authors' recommendation, terminating four rows earlier within the matrix  $\mathbf{A}$ . Here, . denotes a

Table 1 – Gauss elimination table.

<b>A</b>	$x^3$	$y^3$	$x^2y$	$xy^2$	$x^2z$	$x^2$	$y^2z$	$y^2$	$xyz$	$xy$	x	y	1
$\langle a \rangle$	1	.	.	.	.	.	.	.	.	.	[2]	[2]	[3]
$\langle b \rangle$		1	.	.	.	.	.	.	.	.	[2]	[2]	[3]
$\langle c \rangle$			1	.	.	.	.	.	.	.	[2]	[2]	[3]
$\langle d \rangle$				1	.	.	.	.	.	.	[2]	[2]	[3]
$\langle e \rangle$					1	.	.	.	.	.	[2]	[2]	[3]
$\langle f \rangle$						1	.	.	.	.	[2]	[2]	[3]
$\langle g \rangle$							1	.	.	.	[2]	[2]	[3]
$\langle h \rangle$								1	.	.	[2]	[2]	[3]
$\langle i \rangle$									1	.	[2]	[2]	[3]
$\langle j \rangle$										1	[2]	[2]	[3]

Source: [Nistér \(2004\)](#).

scalar value, and  $[I]$  represents a polynomial of degree  $I$  in the variable  $z$ . Additionally, it

is defined as shown in (13).

$$\begin{cases} \langle k \rangle = \langle e \rangle - z \langle f \rangle \\ \langle l \rangle = \langle g \rangle - z \langle h \rangle \\ \langle m \rangle = \langle i \rangle - z \langle j \rangle \end{cases} \quad (13)$$

they are arranged into matrix  $\mathbf{B}$  containing polynomials in  $z$ . with  $(x, y, 1)^\top$  being the

Table 2 – Polynomial constraint matrix in  $z$ .

$\mathbf{B}$	x	y	1
$\langle k \rangle$	[3]	[3]	[4]
$\langle l \rangle$	[3]	[3]	[4]
$\langle m \rangle$	[3]	[3]	[4]

Source: [Nistér \(2004\)](#).

nullvector of  $\mathbf{B}$ ,  $\det(\mathbf{B}) = 0$  which a tenth degree polynomial, thus compute its roots and, for each real root, find  $x$  and  $y$  in system represented by matrix  $\mathbf{B}$ . Finally compute  $\mathbf{E}$  and recover  $\mathbf{R}$  and  $\mathcal{T}$  that is represented by  $\mathbf{E} = \mathbf{R}\mathcal{T}_\times$  using and SVD  $\mathbf{E} = \mathbf{U}\pm\mathbf{V}^\top$ , then  $\mathcal{T} = (u_{13}, u_{23}, u_{33})$  and  $\mathbf{R} = \mathbf{UDV}^\top$ , with  $\mathbf{D}$  defined as

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (14)$$

The five-point algorithm typically exhibits the following issues:

- Requires prior information on internal parameters. Without this prerequisite knowledge, the method fails due to insufficient information.
- The computational cost can be significant depending on the model estimation approach employed. Despite its widespread use by the SfM community and the availability of methods such as ACransac that mitigate the cost, the algorithm can reach the maximum number of iterations without computing the essential matrix, depending on the views selected.
- Necessitates additional steps post-calibration, such as bundle adjustment and cheirality test refinement. While these steps are also utilized in the trifocal method, they exhibit less sensitivity to errors. Internal tests conducted during the implementation of the trifocal method from 2020 to 2022 demonstrated that the five-point algorithm exhibits lower precision compared to the trifocal method. This discrepancy is attributed to the fact that fewer tests were required to confirm that a point was an

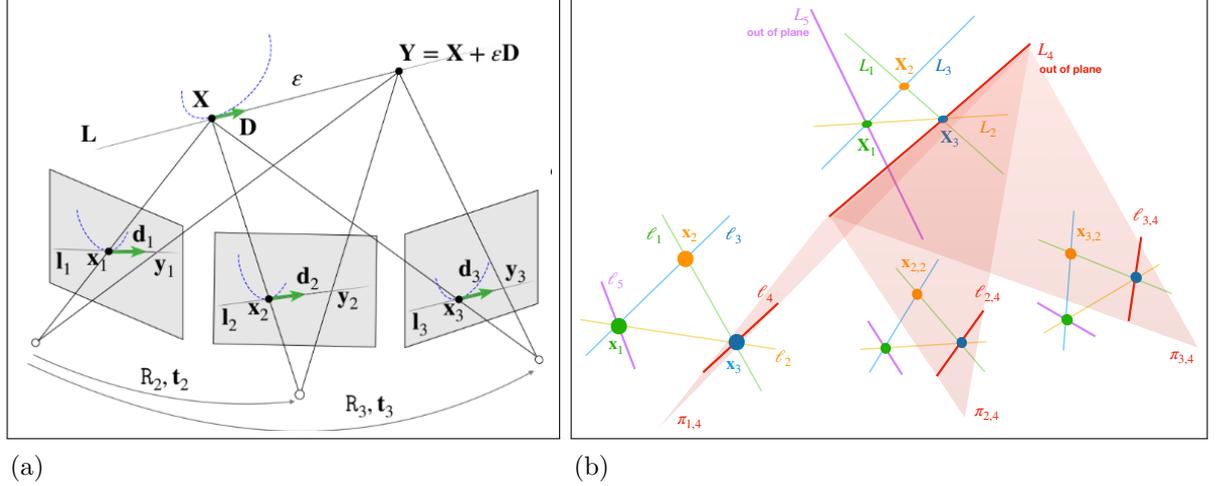
inlier in the trifocal method, potentially introducing noise, even if undetectable to the human eye, in camera positioning.

### 1.3 Trifocal geometry

The initial year of the research was dedicated to advancing the implementation of MiNuS within `openMVG`, using the *Chicago problem* as the standard approach. MiNuS [Fabbri et al. \(2023\)](#) is a C++ framework developed to compute minimal problems, particularly the trifocal tensor, with the two primary problems being *Chicago* and *Cleveland*. The *Chicago* problem was selected as the standard due to its relative simplicity compared to the *Cleveland* problem, and it is defined as follows:

**Definition 1.3.1** *Given three corresponding points  $x_{1v}, x_{2v}, x_{3v}$  and two lines  $l_{1v}, l_{2v}$  where in views  $v = 1, 2, 3$ , such that  $l_{pv}$  meets  $x_{pv}$  for  $p = 1, 2, v = 1, 2, 3$ , compute relative pose  $\mathbf{R}_2, \mathcal{T}_2, \mathbf{R}_3, \mathcal{T}_3$*

Figure 8 – A representation of the trifocal problem from oriented points codenamed Chicago



Legend: (a) The general trifocal approach, (b) A diagram of Chicago problem itself.

Source: [Fabbri et al. \(2023\)](#)

These corresponding points can be obtained through oriented point-feature detection, edge detection, or moving point trajectories. The fundamental equations utilized for the trifocal method are the parametric forms of the point and tangent equations (15).

$$\begin{cases} \rho_{pv} \gamma_{pv} = \mathbf{R}_v \rho_{p1} \gamma_{p1} + \mathcal{T}_v & p = 1, 2 \\ (\rho_{pv} \mathbf{t}_{pv} + \mu_{pv} \gamma_{pv}) = \mathbf{R}_v (\rho_{p1} \mathbf{t}_{p1} + \mu_{p1} \gamma_{p1}) & v = 1, 2, 3 \end{cases}, \quad (15)$$

The two vector equations yield a system consisting of 30 equations with 9 depth parameters and 12 line parameters. Upon applying constraints, this system is reduced to 14 equations with 14 variables. These variables include four associated with the Cayley homogenized rotation (16) and three related to the translation parameters for  $\mathbf{R}_2$ ,  $\mathcal{T}_2$ ,  $\mathbf{R}_3$ , and  $\mathcal{T}_3$ . Despite this reduction, the number of parameters increases to 56.

$$\mathbf{R} = \begin{bmatrix} w_v & -z_v & y_v \\ z_v & w_v & -x_v \\ y_v & x_v & w_v \end{bmatrix} \begin{bmatrix} w_v & -z_v & y_v \\ z_v & w_v & -x_v \\ y_v & x_v & w_v \end{bmatrix}^{-1}. \quad (16)$$

In total, there are 27 independent line parameters, 12 dependent line parameters, and three random variables:  $\mathbf{v}_1$  and  $\mathbf{v}_2$  in  $\mathbb{C}^5$ , and  $\mathbf{v}_3$  in  $\mathbb{C}^7$ . These variables originate from the linear inhomogeneous chart equations (17).

$$(\mathbf{r}_1 \ 1)\mathbf{v}_1 = 0, (\mathbf{r}_2 \ 1)\mathbf{v}_2 = 0, (\mathcal{T}_2^\top \ \mathcal{T}_3^\top \ 1)\mathbf{v}_3 = 0. \quad (17)$$

Let  $\mathbf{r}_1 = (w_2, x_2, y_2, z_2)$  and  $\mathbf{r}_2 = (w_3, x_3, y_3, z_3)$  be selected for simplification. It is posited that incorporating a more nonlinear set of internal constraints for these variables could improve algorithmic performance by potentially enhancing stability.

These equations are employed in homotopy continuation, an algebraic technique that, in essence, relates two functions (or systems of functions) through the homotopic function (18).

$$H(z, t) = f(z)(1 - t) + g(z)t, \quad z \in \mathbb{C}^n, \quad t \in [0, 1]. \quad (18)$$

This provides input to Davidenko's partial differential equation (19) in its matrix form.

$$Jac(H)_z \frac{dz}{dt} + \frac{\partial H}{\partial t} = 0, \quad (19)$$

that results in the system (20)

$$\frac{dz}{dt} = -(Jac(H)_z)^{-1} \frac{\partial H}{\partial t}. \quad (20)$$

The homotopy function employed in the trifocal model is referred to as the *global homotopic function*, defined as  $H(\mathcal{R}, s) = f(\mathcal{R}, (1 - s)\mathcal{A}_0 + s\mathcal{A}^*)$ . Here,  $\mathcal{R}$  represents the variables,  $\mathcal{A}_0$  denotes the parameters of the initial system which have been precomputed and hardcoded into the algorithm, and  $\mathcal{A}^*$  represents the parameters of the target system. These parameters are encapsulated in a function  $\alpha(s)$  such that  $f(\mathcal{R}, (1 - s)\mathcal{A}_0 + s\mathcal{A}^*) =$

$f(\mathcal{R}, \alpha(s))$  (21).

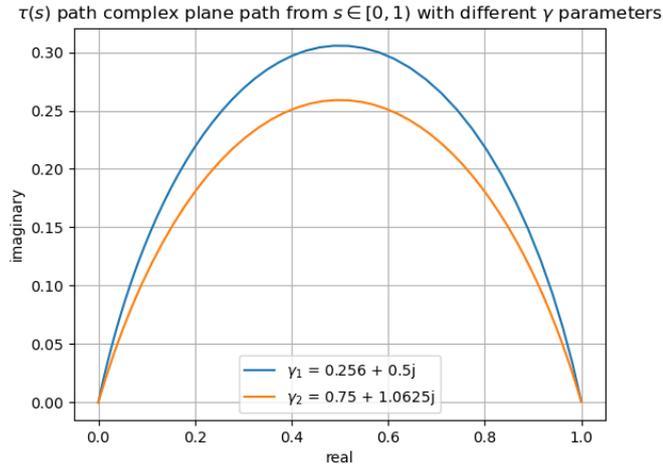
$$\alpha(s) = (1 - s)\mathcal{A}_0 + s\mathcal{A}^*, \quad (21)$$

In order to maintain genericity and path a technique called the *gamma trick* is used that consists in converting the homotopy's path from real to complex space using a random  $\gamma \in \mathbb{C}$  (different from the parametric point) and building a function  $\tau(s) \in \mathbb{C}, s \in [0, 1)$  that corresponds to it (22).

$$\begin{cases} F(\mathcal{R}, \alpha(s)) = F(\mathcal{R}, (1 - \tau(s))\mathcal{A}_0 + \tau(s)\mathcal{A}^*) = (1 - \tau(s))F(\mathcal{R}, \mathcal{A}_0) + \tau(s)F(\mathcal{R}, \mathcal{A}^*) \\ \tau(s) = \frac{\gamma s}{1 + (\gamma - 1)s}, \quad s \in [0, 1) \end{cases}, \quad (22)$$

The  $\tau(s)$  function follows a circular arc instead of the straight line described by  $\alpha(s)$  in the homotopy function. This approach allows for path variation with each execution of the algorithm, ensuring that the multiple paths generated by the homotopy function are smooth and do not intersect each other. It serves as a sub-optimal yet broadly applicable method to mitigate failure, although it may still lead to a slowdown of the homotopy continuation solver 9. To obtain the initial system, the monodromy method is employed.

Figure 9 – Plane real  $\times$  imaginary homotopy.



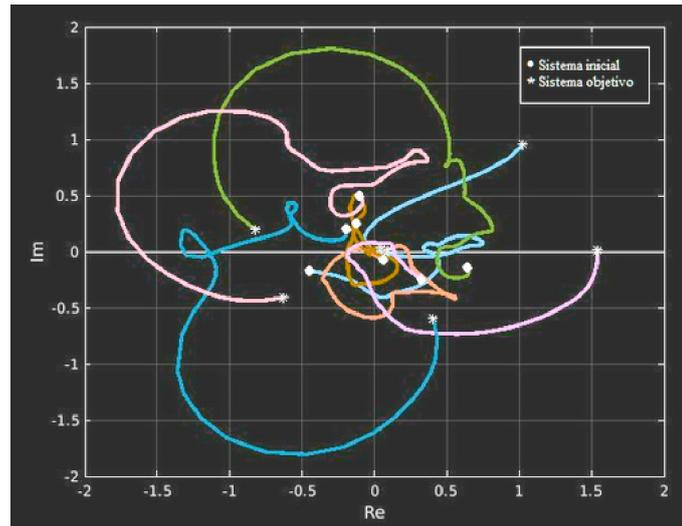
Legend: Homotopy path with gamma trick, depending which  $\gamma$  is chosen, has an increased execution time of MiNuS.

Source: The author, 2024.

This procedure involves selecting one of the 14 variables to be free, with the remaining

variables expressed as functions of the chosen variable. Consequently, the Chicago problem transforms into a 20-degree polynomial equation with 312 complex solutions. These solutions are utilized to seed the variables, which may result in singularities, blow-ups, or well-conditioned solutions for the continuation exemplified in 10. The initial task in-

Figure 10 – The 14 variables homotopy path generated by MiNuS Chicago problem.



Legend: This was done as part of Juliana Barcellos, member of Labvis team, work.

Source: [Ventura, Fabbri e Moura Neto \(2022\)](#).

involved investigating the regions corresponding to singularities that are responsible for sub-optimal estimation and incorporating "avoidance conditions" into the solver. This investigation utilized dynamical systems theory to analyze the relationship between the fixed (or critical) points of the differential system of equations (20) and the singularities, which could represent the complete presence or absence of the latter. It was anticipated that this approach might be ineffective, computationally expensive, and time-consuming due to the following reasons:

- The equations are highly coupled to the extent that not only symbolic computation libraries such as `Sympy`, but also software tools like `Maple`, `WxMaxima`, and `Matlab` were unable to symbolically solve both the differential equation system and its fixed points. Attempting to obtain the solution for (20) manually was also deemed unfeasible due to its low practical payoff after careful consideration. Additionally, the coupling problem poses a significant challenge, as it can lead to rank-deficient solutions. Such solutions, while algebraically valid, may not correspond to critical points of the system [Prado \(1994\)](#).

- The continuation method and the *gamma trick* do not inherently represent the model itself but rather provide a generic approach for performing trifocal camera estimation. While this approach may be functional, it does not elucidate the limitations and internal structure of the problem, nor does it address the experimental issues outlined in the introduction.

## 2 BASIC CONCEPTS OF TIME-SPACE OPTICAL DYNAMICS

The concept of developing a dynamical camera model emerged while addressing the complexities associated with trifocal geometry. As detailed in 1.3, the equations involved in the current trifocal model are highly complex, rendering stability analysis of the differential equation system intractable using both symbolic computation and manual methods due to significant complexity and variable coupling. In the course of this research, the exploration of alternative solutions led to an examination of a pertinent study [Fabbri e Kimia \(2016\)](#). This investigation revealed the potential for formulating a direct model for camera positioning over time, utilizing principles from dynamical systems theory and analytical mechanics. This approach represents a viable strategy for addressing the challenges previously encountered.

The units of measurement used in this work are defined as follows: the geometric parameter  $s$  is measured in pixels when referring to image space and in meters when referring to 3D scene space. The time parameter  $t$  is defined as the runtime of future algorithms developed based on this work.

### 2.1 Differential geometry and dynamics

Differential geometry and dynamics underlie the differential geometry approach introduced by [Fabbri e Kimia \(2016\)](#) (for details regarding the geometry involving curvature and torsion, see C). The 3D projected point equation  $\Gamma = [x, y, z]^\top$ , as specified in (23), has been adapted in this context from the original notation  $\mathbf{\Gamma}^w$  to  $\mathbf{\Gamma}^r$ . Here,  $r$  represents the reference frame, which may correspond to either the three-dimensional world coordinates or the three-dimensional parametric representation of the image used for estimation. This adaptation is intended to enhance clarity in communication with physicists, although both notations remain functionally equivalent.

$$\mathbf{\Gamma}(s, t) = \mathbf{R}(t)\mathbf{\Gamma}^r(s, t) + \mathcal{T}(t). \quad (23)$$

Here,  $s$  denotes the geometric parameter and  $t$  represents the temporal parameter. Additionally, a point in the image  $\gamma = [\xi, \eta, 1]^\top$  is related to  $\Gamma$  by projecting it onto the plane  $z = 1$  and incorporating the depth factor  $\rho = z = \Gamma_z \mathbf{e}_z$ , as defined by (24).

$$\mathbf{\Gamma}(s, t) = \rho(s, t)\gamma(s, t) = [\rho\xi, \rho\eta, \rho]^\top. \quad (24)$$

The image is derived by applying the inverse distortion correction to the projected plane, utilizing the calibration matrix  $\mathcal{K}$  as specified in (25).

$$\gamma_{im}(s, t) = \mathcal{K}\gamma(s, t). \quad (25)$$

The velocities of the 3D and 2D points are derived by differentiating the initial equations with respect to the temporal parameter  $t$ , as specified in the point equation (26).

$$\begin{cases} \mathbf{\Gamma}_t = \mathbf{R}_t \mathbf{\Gamma}^r + \mathbf{R} \mathbf{\Gamma}_t^r + \mathcal{T}_t \\ \gamma_t = \frac{1}{\rho} (\mathbf{R}_t \mathbf{\Gamma}^r + \mathbf{R} \mathbf{\Gamma}_t^r + \mathcal{T}_t) - \frac{\rho_t}{\rho^2} (\mathbf{R} \mathbf{\Gamma}^r + \mathcal{T}) \end{cases} \quad (26)$$

For camera estimation, when the geometric parameter  $s$  is independent of time, it is sufficient to consider only the tangent. By differentiating the equation (23) with respect to the geometric parameter  $s$ , the 3D tangent vector  $\mathbf{T}$  is derived as follows:

$$\mathbf{T}(s, t) = \frac{\partial \mathbf{\Gamma}}{\partial s} = \frac{\partial}{\partial s} (\mathbf{R} \mathbf{\Gamma}^w + \mathcal{T}) = \mathbf{R} \frac{\partial \mathbf{\Gamma}^r}{\partial s} = \mathbf{R} \mathbf{T}^r \quad (27)$$

Further differentiation of  $\mathbf{T}$  with respect to the arc parameter  $s$  yields the expression  $K\mathbf{N}$ , where:

$$K\mathbf{N} = \mathbf{R} K^r \mathbf{N}^r \quad (28)$$

The binormal vector  $\mathbf{B}$  is then given by:

$$\mathbf{B} = \mathbf{T} \times \mathbf{N} = \mathbf{R} \mathbf{T}^r \times \mathbf{R} K^r \mathbf{N}^r = \mathbf{R} K^r (\mathbf{T}^r \times \mathbf{N}^r) = \mathbf{R} K^r \mathbf{B}^r \quad (29)$$

Since all Frenet frame components depend solely on the rotation matrix, it is sufficient to include only the tangent vector as an additional constraint in the camera pose estimation process.

The 3D and 2D tangent velocities, expressed in terms of the 3D tangent velocities, are described by equations (30).

$$\begin{cases} \mathbf{T}_t = \mathbf{R}_t \mathbf{T}^r + \mathbf{R} \mathbf{T}_t^r \\ \mathbf{t}_t = \frac{1}{\rho} (\mathbf{R}_t \mathbf{T}^r + \mathbf{R} \mathbf{T}_t^r) + \frac{1}{\rho^2} [\rho_t \mathbf{R} \mathbf{T}^r - (\rho_{st} + \frac{\rho_s \rho_t}{\rho}) (\mathbf{R} \mathbf{\Gamma}^r + \mathcal{T}) - \rho_s (\mathbf{R}_t \mathbf{\Gamma}^r + \mathbf{R} \mathbf{\Gamma}_t^r + \mathcal{T}_t)]. \end{cases} \quad (30)$$

The 2D point and tangent velocities are obtained by differentiating  $\gamma = \frac{\Gamma}{\rho} = \frac{1}{\rho} (\mathbf{R} \mathbf{\Gamma}^r + \mathcal{T})$  first with respect to time and subsequently with respect to space.

$$\gamma_t = \frac{1}{\rho} (\mathbf{R}_t \mathbf{\Gamma}^r + \mathbf{R} \mathbf{\Gamma}_t^r + \mathcal{T}_t) - \frac{\rho_t}{\rho^2} (\mathbf{R} \mathbf{\Gamma}^r + \mathcal{T}) \quad (31)$$

$$\mathbf{t}_t = -\frac{\rho_s}{\rho^2}(\mathbf{R}_t\mathbf{\Gamma}^r + \mathbf{R}\mathbf{\Gamma}_t^r + \mathcal{T}_t) + \frac{1}{\rho}(\mathbf{R}_t\mathbf{T}^r + \mathbf{R}\mathbf{T}_t^r) - \frac{1}{\rho^2}(\mathbf{R}\mathbf{\Gamma}^r + \mathcal{T})\left(\rho_{st} - \frac{\rho_s\rho_t}{\rho}\right) - \frac{\rho_t}{\rho^2}\mathbf{R}\mathbf{\Gamma}^r.$$

An aspect not addressed in their work, but considered in this study for further investigation, is the fact that the spatial parameter can be a function of time,  $s = s(t)$ . Consequently, the time derivatives require modification, as detailed for the points in (32).

$$\begin{cases} \mathbf{\Gamma}_t + \mathbf{\Gamma}_s s_t = \mathbf{R}_t\mathbf{\Gamma}^r + \mathbf{R}(\mathbf{\Gamma}_t^r + \mathbf{\Gamma}_s^r s_t) + \mathcal{T}_t \\ \gamma_t + \gamma_s s_t = \frac{1}{\rho}[\mathbf{R}_t\mathbf{\Gamma}^r + \mathbf{R}(\mathbf{\Gamma}_t^r + \mathbf{\Gamma}_s^r s_t) + \mathcal{T}_t - \frac{1}{\rho}(\mathbf{R}\mathbf{\Gamma}^r + \mathcal{T})(\rho_t + \rho_s s_t)], \end{cases} \quad (32)$$

and for tangent (33)

$$\begin{cases} \mathbf{\Gamma}_{ts} + \mathbf{\Gamma}_{tt}\frac{1}{s_t} - \frac{1}{s_t^2}\mathbf{\Gamma}_t s_{tt} = \mathbf{R}_t(\mathbf{\Gamma}_t^r\frac{1}{s_t} + \mathbf{\Gamma}_s^r) + \mathbf{R}(\mathbf{\Gamma}_{ts}^r + \mathbf{\Gamma}_{tt}\frac{1}{s_t} - \frac{1}{s_t^2}\mathbf{\Gamma}_t^r s_{tt}) \\ \gamma_{ts} + \gamma_{tt}\frac{1}{s_t} - \frac{1}{s_t^2}\gamma_t s_{tt} = \frac{1}{\rho}[\mathbf{R}_t(\mathbf{\Gamma}_t^r\frac{1}{s_t} + \mathbf{\Gamma}_s^r) + \mathbf{R}(\mathbf{\Gamma}_{ts}^r + \mathbf{\Gamma}_{tt}\frac{1}{s_t} - \frac{1}{s_t^2}\mathbf{\Gamma}_t^r s_{tt})] \\ + \frac{1}{\rho^2}[(\rho_t + \rho_s\frac{1}{s_t})\mathbf{R}(\mathbf{\Gamma}_t^r\frac{1}{s_t} + \mathbf{\Gamma}_s^r) \\ - [(\rho_s + \rho_t\frac{1}{s_t}) + \frac{1}{\rho}(\rho_s + \rho_t\frac{1}{s_t})(\rho_t + \rho_s s_t)](\mathbf{R}\mathbf{\Gamma}^r + \mathcal{T}) \\ - (\rho_s + \rho_t\frac{1}{s_t})(\mathbf{R}_t\mathbf{\Gamma}^r + \mathbf{R}(\mathbf{\Gamma}_t^r + \mathbf{\Gamma}_s^r s_t) + \mathcal{T}_t)] \end{cases} \quad (33)$$

## 2.2 Lagrangian and Hamiltonian mechanics

Given that the parameters of camera location are solely time-dependent, the utilization of Lagrangian and Hamiltonian mechanics is warranted. This study adopts a contemporary approach, in contrast to a classical one, notwithstanding the latter's importance.

As articulated by [Landau e Lifshitz \(1976\)](#) and further developed in [Holm \(2008a\)](#), [Holm \(2008b\)](#), the Lagrangian is derived from the principle of least action. This principle asserts that the action, defined by (34), remains stationary under small variations. Fundamentally, this implies that the Lagrangian function delineates a minimal path taken by the action integral, which can be expressed in terms of either the temporal or geometric parameter. Given the emphasis on the temporal parameter, the action integral (34) is defined with respect to  $t$ .

$$\mathcal{S} = \int_{t_0}^{t_1} L(q, \dot{q}) dt, \quad (34)$$

Where  $L$  denotes the Lagrangian function, and let  $q$  and  $\dot{q}$  represent the generalized coordinates and velocities, respectively, where  $\dot{q} = dq/dt$ . According to a modern approach involving topology [Frankel \(2011\)](#), given a configuration space  $\mathcal{M}^n$ , the Lagrangian func-

tion encapsulates not only the dynamics of the configuration space in its local coordinates  $(q, \dot{q})$ , but also serves as both a Finsler metric and a pseudo-Riemannian metric (see D).

When the action integral is assumed to be stationary, it is stated that a variation  $\delta$  between the functions defining this action does not affect its path behavior, represented by  $\delta\mathcal{S} = 0$ . Applying the differential  $\delta$  to both sides of (34) yields (35).

$$\delta\mathcal{S} = \int_{t_0}^{t_1} \left[ \frac{\partial L}{\partial q} - \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) \right] \delta q dt = 0. \quad (35)$$

$$\frac{\partial L}{\partial q} - \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) = 0 \quad (36)$$

The Hamiltonian function, as defined in (37), serves to analyze the phase space dynamics of the phenomena under study. In a modern framework Frankel (2011), the Hamiltonian function is employed within symplectic geometry, which finds applications in fields such as relativity.

$$H(q, p) = p \cdot \dot{q} - L(q, \dot{q}), \quad (37)$$

Where the generalized momenta  $p$  is defined as  $p = \partial L / \partial \dot{q}$ , obtained through a process known as the Legendre transform. This transform aims to convert  $\dot{q}$ , which depends on coordinates, into an independent variable (see D). This transformation results in a  $2n$ -dimensional phase space derived from the configuration space  $\mathcal{M}^n$ . By applying the Poisson brackets, the Hamiltonian system of first-order differential equations (38) is obtained.

$$\left\{ \begin{array}{l} \frac{dq_1}{dt} = \frac{\partial H}{\partial p_1} \\ \frac{dp_1}{dt} = -\frac{\partial H}{\partial q_1} \\ \vdots \\ \frac{dq_n}{dt} = \frac{\partial H}{\partial p_n} \\ \frac{dp_n}{dt} = -\frac{\partial H}{\partial q_n} \end{array} \right. \quad (38)$$

To construct an accurate Lagrangian model for Structure from Motion (SfM) problem classes, it is essential to determine not only the specific structure under study (e.g.: curve motion, curve geometric information, camera motion, or any combination thereof) but also the nature of the parameters involved, namely the geometric parameter  $s$  or the temporal parameter  $t$ . Thus, the following hypotheses are proposed.

- The geometric parameter  $s$  is conceptually linked to Fermat's principle and the

Eikonal equation, as discussed by [Holm \(2008a\)](#). This relationship stems from the notion of the ray path, which characterizes how a point in a three-dimensional scene is projected onto the camera sensor. The projection process is influenced by factors such as the reflectivity of materials and the refractive index of both the medium in the scene and the camera lens. In the case of non-rigid curves, this hypothesis is extended to encompass a more general framework that considers the "dynamics" of the curve parameter  $s$  over time, thus framing the problem as an optical-dynamical issue.

- The time parameter  $t$  pertains to the dynamics of the scene or the camera itself. In this context, the Lagrangian and Hamiltonian formulations are associated with kinetic and potential energies. An overview by [Ma et al. \(2004\)](#) introduces the concept of rigid body dynamics in camera motion, while [Fabbri e Kimia \(2016\)](#) explores the concept of curve motion in three-dimensional and two-dimensional spaces. The hypothesis asserts that, regardless of the specific modeling approach employed, the system must fundamentally comply with the laws of motion, thereby necessitating a classical mechanical framework.

Given that this work focuses on examining dynamical camera motion, it is imperative that the problem adheres to the fundamental laws of motion as previously discussed. Accordingly, within the framework of classical mechanics, the Lagrangian and Hamiltonian functions are characterized by their relationships with kinetic and potential energies, as detailed in (39).

$$\begin{cases} L(q, \dot{q}) = K(\dot{q}) - U(q) = \frac{1}{2}m\dot{q}^2 - U(q) \\ H(q, p) = K(p) + U(q) = \frac{1}{2m}p^2 + U(q) \end{cases} \quad (39)$$

By separating camera localization from geometry recovery and employing this Lagrangian functional, the aim is to achieve more reliable modeling in real-world scenarios. This approach not only facilitates a deeper understanding of the practical issues in estimation but also addresses the root causes of these issues. Commonly used algorithms for camera estimation often focus solely on mathematical limitations without considering their practical implications. For instance, certain positions may be problematic for the camera, raising questions about the underlying causes and their significance in real-world experiments.

### 2.3 Topology of image

Prior to the development of this dissertation, during the initial phases of this research, a discussion was prompted by a master's course on Differentiable Manifolds. As

outlined in Frankel (2011), a manifold is a topological space that locally resembles  $\mathbb{R}^n$  in accordance with certain structures, which may be topological, differentiable, or algebraic. The following statement is proposed for consideration:

The image space can be regarded as a differentiable manifold of dimension two, with the pixel coordinates  $\gamma_{im}$  serving as its local coordinates. If the 3D scene manifold and the image domain are considered as the embedding space of the 2D spatio-temporal manifold, they can be directly related through a function. While the image is typically flat, it can exhibit various structures depending on the level of modeling, such as topological structure, affine structure, or projective structure. Additionally, viewing an image as a manifold helps in understanding the curved nature of projective space, which represents the space of un-oriented light rays passing through the camera center. This assertion is based on the premise that detection algorithms can be mathematically expressed as local coordinate functions mapping the image manifold  $\mathcal{M}^2$  into the  $\mathbb{R}^2$  image space. In a specific manner,  $\gamma_{im}$  functions as a mapping from  $\mathcal{M}$  into  $\mathbb{R}^2$ . Given a fixed value in the time parameter, this can be simplified to  $\gamma_{im}(s, t_0) = \gamma_{im}(s)$ .

$$\begin{aligned} \gamma_{im}: \mathcal{M}^2 &\rightarrow \mathbb{R}^2 \\ m(s) &\mapsto \gamma_{im}(m(s)) = (x, y)^\top. \end{aligned} \quad (40)$$

The local derivatives, corresponding to the geometrical Frenet frames  $\mathbf{t}$ ,  $\mathbf{n}$ , are comprehensively detailed in Fabbri e Kimia (2016).

$$\mathbf{t} = \frac{1}{\left\| \frac{d\gamma}{ds} \right\|} \frac{d\gamma}{ds}, \quad \mathbf{n} = \mathbf{t}^\perp = \frac{1}{\left\| \frac{d\mathbf{t}}{ds} \right\|} \frac{d\mathbf{t}}{ds}. \quad (41)$$

Image 11 illustrates this concept.

A similar concept can be applied to 3D space. By assuming that the 3D manifold's local coordinates are based on a fixed rigid 3D referential curve, we can simplify the representation to  $\Gamma^r(s, t_0) = \Gamma^r(s)$ .

$$\begin{aligned} \Gamma^r: \mathcal{N}^3 &\rightarrow \mathbb{R}^3 \\ n(s) &\mapsto \Gamma^r(n(s)) = (X, Y, Z)^\top. \end{aligned} \quad (42)$$

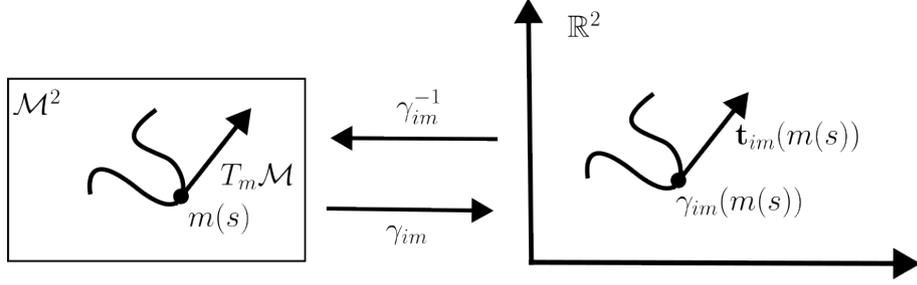
$$(43)$$

The local derivatives, corresponding to the geometrical Frenet frames  $\mathbf{T}$ ,  $\mathbf{N}$ ,  $\mathbf{B}$ , are comprehensively detailed in Fabbri e Kimia (2016).

$$\mathbf{T}^r = \frac{1}{\left\| \frac{d\Gamma^r}{ds} \right\|} \frac{d\Gamma^r}{ds}, \quad \mathbf{N} = \mathbf{T}^\perp = \frac{1}{\left\| \frac{d\mathbf{T}^r}{ds} \right\|} \frac{d\mathbf{T}^r}{ds}, \quad \mathbf{B}^r = \mathbf{T}^r \times \mathbf{N}^r. \quad (44)$$

To represent the projection in a coordinate-free manner, it is necessary to compute a

Figure 11 – An illustration of 2D image manifold to 2D local space.



Legend: A representation, in this work notation, of the transition of symbolic system of coordinates to numeric.

Source: The author, 2024

function from  $\mathcal{N}$  to  $\mathcal{M}$ . Let the function  $f$  be defined as follows:

$$f: \mathcal{N}^3 \rightarrow \mathcal{M}^2 \quad (45)$$

$$n(s) \mapsto f(n(s)).$$

It is established that both local coordinates can be related by combining equations (23) and (25), resulting in  $\gamma_{im} = \mathcal{K}_p^{\frac{1}{\rho}}(\mathbf{R}\Gamma^r + \mathcal{T})$ . This leads to the conclusion that  $f(n(s)) = \gamma_{im}^{-1} \circ \Gamma^r(n(s))$ . The image 12 illustrates how the 3D scene manifold and the 2D image manifold are interconnected.

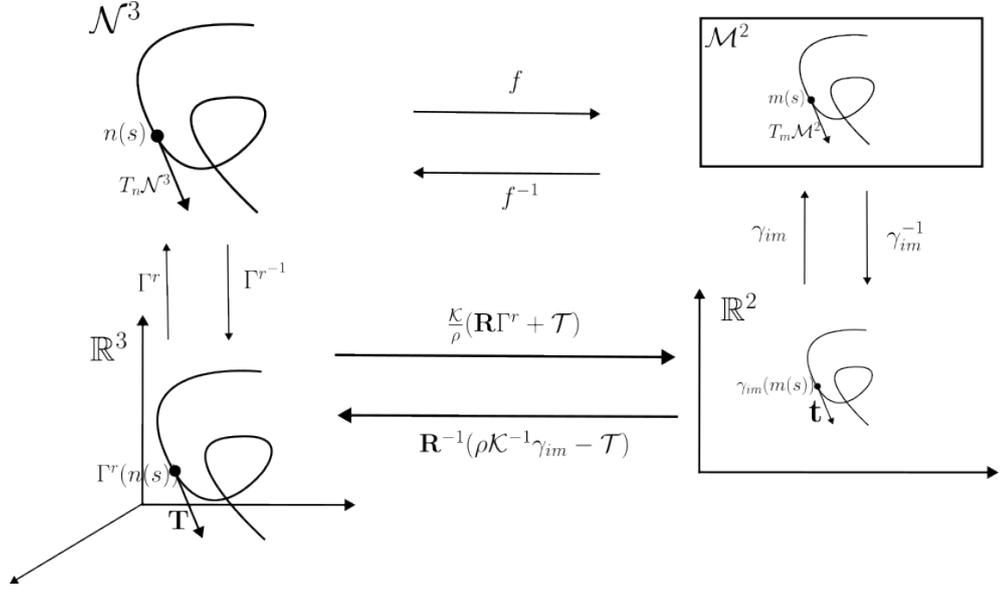
This approach provides a foundation for a deeper understanding of the geometric parameters of a 2D image curve, as it allows for an abstract and symbolic relationship between the image and the world. This is particularly useful when dealing with numerous coordinate transformations, which, when performed continuously, may exhibit complex behavior independent of the geometric problem itself.

### 2.3.1 Image as Galilean group

The proposed approach considers the image as part of the Galilean group. In relativistic mechanics, the Galilean group encompasses all Galilean transformations. Essentially, a Galilean transformation refers to a change in the reference frames over time.

The underlying concept is that each image corresponds to a distinct 2D projective space frame, which is linked to its specific camera position. This relationship can be framed within the Galilean group  $\text{Gal}(3)$  under specific conditions. Although a com-

Figure 12 – An illustration of 3D scene manifold to 2D image.



Legend: An Illustrative representation of how 2D image topology and 3D scene space are related.

Source: The author, 2024

prehensive proof demonstrating that images belong to the Galilean group  $\text{Gal}(3)$  involves detailed verification of group properties, this discussion outlines the manner in which such a relationship can be conceptualized. By definition, the Galileo group  $\text{Gal}(3)$ , as described in [Nadjafikhah e Forough \(2007\)](#), is defined in matrix form (46)

$$\mathbf{G} \cdot \mathbf{r} = \begin{bmatrix} \mathbf{R} & \mathbf{v} & \mathcal{T} \\ 0 & 1 & s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ t \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}\mathbf{x} + \mathbf{v}t + \mathcal{T} \\ t + s \\ 1 \end{bmatrix}, \quad (46)$$

where  $\mathbf{R}$  denotes the 3D rotation matrix,  $\mathcal{T}$  represents the translation vector,  $\mathbf{v}$  is the relative motion velocity of the frame in  $\mathbb{R}^3$ , and  $s \in \mathbb{R}$  signifies the time difference between frames. If the relative motion velocity  $\mathbf{v} = 0$ , the perspective projection formula (23) can be derived as follows. By selecting the first coordinate vector of the Galilean group  $\text{Gal}(3)$  (defined as  $\pi_1(\mathbf{G} \cdot \mathbf{r})$ ) and assuming  $\mathbf{x} = \Gamma^r$ , the image coordinates are computed as:

$$\mathbf{G} \cdot \bar{\mathbf{r}} = \begin{bmatrix} \mathbf{R} & 0 & \mathcal{T} \\ 0 & 1 & s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Gamma^r \\ t \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}\mathbf{x} + \mathcal{T} \\ t + s \\ 1 \end{bmatrix} \rightarrow \gamma_{im} = \mathcal{K}\pi_1(\mathbf{G} \cdot \bar{\mathbf{r}}). \quad (47)$$

## 2.4 The point vs position problem

Given two images of an object that initially appear similar, how can one ascertain that both images represent the same object, with one being a rotated version of the other? There are two primary approaches to address this question:

- One approach involves mentally reconstructing the object and manually rotating and translating it to match the projections seen in the images. The main challenge with this method is that it may result in the loss of crucial information necessary to determine if the images depict the same object, such as hidden curves or features that are not visible from certain angles.
- Another approach involves searching for correspondences between points and edges in the two images, and then inferring whether the images represent the same object. This method does not require prior knowledge of the exact rotation and translation but instead relies on identifying matching features to deduce the transformations between the images.

At a broad level, points and curves are often considered fundamental to object reconstruction. However, in practice, these perspectives address different aspects of Structure from Motion (SfM). The multiview triangulation problem underscores that both viewpoints are applicable, depending on the specific objectives and challenges within SfM. The distinction is as follows:

For multiview triangulation, where camera positions are known, points and curves are crucial for verifying that two images represent the same object from different viewpoints. In contrast, for camera position estimation, the primary focus is on the rotations and translations. In this case, the points and curves projected onto the image are a result of the camera's position and orientation.

## 2.5 Special case: Uncalibrated camera dynamics

The dynamics considered in this study are predominantly based on the assumption of a constant calibration matrix, whether utilizing a single device or, when switching devices, by knowing the internal parameters of each. The impact of an uncalibrated camera on the dynamical approach has not been thoroughly explored at a fundamental level. This opens avenues for future research, particularly regarding the dynamics of the calibration matrix with practical applications, such as zooming effects. The core assertion of this research is that the camera calibration matrix can exhibit its own dynamics.

The principle is analogous to the human eye, where the lens can adjust its shape over time due to its muscular structure. For cameras with unknown internal parame-

ters, the projection matrix  $\mathbf{P} = \mathcal{K}[\mathbf{R}|\mathbf{T}]$  translates 3D points into 2D image projections. Standard estimation algorithms generally assume  $\mathbf{P}$  is fixed. In the proposed model,  $\mathbf{P}$  is considered a function of time,  $\mathbf{P} = \mathbf{P}(t)$ , to account for changes due to the motion of the physical device, while assuming internal parameters remain constant. If  $\mathcal{K}$  is also time-dependent,  $\mathcal{K} = \mathcal{K}(t)$ , the projection matrix's velocity can be described by equation (48).

$$\frac{dP}{dt} = \frac{d\mathcal{K}}{dt}[\mathbf{R}|\mathbf{T}] + \mathcal{K}\frac{d}{dt}([\mathbf{R}|\mathbf{T}]) \quad (48)$$

If the parameters of  $\mathcal{K}$  are known, its differential is zero, reverting to the conventional scenario where internal parameters are fixed. In the absence of camera movement, the dynamics of the internal parameters can be examined by analyzing  $d\mathcal{K}/dt$ , as described in equation (49).

$$\frac{d\mathcal{K}}{dt} = \begin{bmatrix} \dot{f}_x & \dot{\beta} & \dot{c}_x \\ 0 & \dot{f}_y & \dot{c}_y \\ 0 & 0 & 0 \end{bmatrix} \quad (49)$$

The foundational equations proposed for initial investigation involve the focal length parameters  $\dot{f}_x$ ,  $\dot{f}_y$ , and  $\dot{\beta}$  as detailed in (50).

$$\begin{cases} \dot{f}_x = \dot{f} \frac{1}{p_x} \\ \dot{f}_y = \dot{f} \frac{1}{p_y} \\ \dot{\beta} = \dot{f}_x \tan(\alpha) \end{cases} \quad (50)$$

A thorough investigation into this matter is deferred to future research due to the substantial mathematical and physical complexity involved, which has yet to be fully addressed. Currently, a literature review is being conducted, and recent advancements in this area have been reported in [Cin et al. \(2024\)](#).

## 2.6 Translation from physics to SfM

One of the primary challenges in this work is translating the concepts of analytical mechanics into the Structure from Motion (SfM) framework. Given that camera motion and image positioning are interpreted as a dynamical system exhibiting characteristics akin to physical problems, the objective is to develop a model that effectively integrates these mechanical principles into the SfM context.

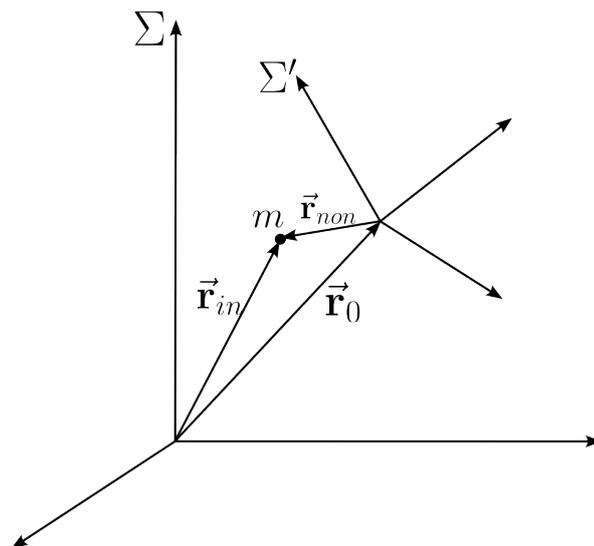
The initial approach involves a comprehensive review of the literature to identify

the most appropriate physical model for camera estimation. The foundational reference was Landau e Lifshitz (1976), a classical source for the preliminary model. Upon reviewing Holm (2008a), it was noted that the first chapter covers optical phenomena, providing valuable insights into how the geometric parameter of a curve can be linked to the optical properties of image formation, including Fermat's Principle and the Eikonal equation. Additionally, sources such as Holm (2008b), Ma et al. (2004) clarify that the camera estimation problem aligns closely with the rigid body problem, which is the main concern of this work.

The analysis of works such as Lemos (2018), Landau e Lifshitz (1976) on rigid body problems has revealed two distinct categories: the camera body motion problem and the curve motion problem. In the context of camera motion, two aspects are considered: the motion of the camera body in the inertial frame (from the scene's perspective) and the motion in the body frame (the camera's own space), which involves the rotation of the camera. Further expansion includes the rotation motion of the inertial frame, a topic reserved for future exploration.

On the side of curve motion, the problem is primarily one of coordinate transformation, specifically transforming the curve motion from inertial frame coordinates to camera frame coordinates. This transformation is analogous to canonical transformations in physics, as represented by figure 13. In this notation, the 3D point within the

Figure 13 – A representation of a particle in non-inertial frame.



Legend: A representation of particle in non-inertial frame.  $\Sigma$  is the inertial frame and  $\Sigma'$  is the non-inertial frame.

Source: The author, 2024.

camera space is represented by the particle with "dimensionless" mass. As stated in , this

mass serves as a parameterization value, utilized as an information-theoretic analogue to physical concepts. The vector  $\mathbf{r}_0$  designates the position of the center of the camera in non-inertial frame coordinates,  $\mathbf{r}_{non}$  indicates the position of the particle in non-inertial frame coordinates, and  $\mathbf{r}_{in}$  represents the position of the particle in the inertial frame.

$$\mathbf{r}_{non} = \mathbf{R}(\mathbf{r}_{in} - \mathbf{r}_0), \quad (51)$$

with  $\mathbf{R}$  representing the rotation matrix corresponding to the vectorial basis of the non-inertial system of coordinates  $\Sigma'$ . To compute the velocity of the particle, we take the time derivative of the position vector given in (51). This will yield the velocity vector in terms of the time rates of change of the respective components.

$$\mathbf{v}_{non} = \mathbf{R}(\mathbf{v}_{in} - \mathbf{v}_0) + \mathbf{R}_t(\mathbf{r}_{in} - \mathbf{r}_0), \quad (52)$$

with  $\mathbf{v}_{in}$  denoting the velocity of the particle in the inertial frame coordinates, and  $\mathbf{v}_{non}$  represent the velocity of the particle in the non-inertial frame coordinates. The term  $\omega \times (\mathbf{r}_{in} - \mathbf{r}_0)$ , where  $\omega$  is the angular velocity of the frame, represents the rotational velocity of the non-inertial frame. Converting the notation from figure 13 to align with the notation used in this work, based on [Fabbri e Kimia \(2016\)](#), the center of the camera space  $\mathbf{r}_0$  corresponds to  $\mathbf{c}$ , which is defined as  $\mathbf{c} = -\mathbf{R}^\top \mathcal{T}$ . The 3D point in camera coordinates is denoted as  $\mathbf{r}_{non} = \mathbf{\Gamma}$ , while the 3D point in scene coordinates is denoted as  $\mathbf{r}_{in} = \mathbf{\Gamma}^r$ . Thus, equation (51) is rewritten as follows:

$$\mathbf{r}_{non} = \mathbf{R}(\mathbf{r}_{in} - \mathbf{r}_0) = \mathbf{R}\mathbf{r}_{in} - \mathbf{R}\mathbf{r}_0 = \mathbf{R}\mathbf{\Gamma}^r - \mathbf{R}\mathbf{c} = \mathbf{R}\mathbf{\Gamma}^r + \mathcal{T} = \mathbf{\Gamma}. \quad (53)$$

Converting (52) to its 3D curve point version

$$\begin{aligned} \mathbf{v}_{non} &= \mathbf{R}(\mathbf{v}_{in} - \mathbf{v}_0) + \mathbf{R}_t(\mathbf{r}_{in} - \mathbf{r}_0) = \mathbf{R}\mathbf{v}_{in} - \mathbf{R}\mathbf{v}_0 + \mathbf{R}_t\mathbf{r}_{in} - \mathbf{R}_t\mathbf{r}_0 \\ &= \mathbf{R}\mathbf{\Gamma}_t^r - \mathbf{R}\mathbf{c}_t + \mathbf{R}_t\mathbf{\Gamma}^r - \mathbf{R}_t\mathbf{c} = \mathbf{R}\mathbf{\Gamma}_t^r + \mathbf{R}_t\mathbf{\Gamma}^r - (\mathbf{R}\mathbf{c}_t + \mathbf{R}_t\mathbf{c}) \\ &= \mathbf{R}\mathbf{\Gamma}_t^r + \mathbf{R}_t\mathbf{\Gamma}^r + \mathcal{T}_t = \mathbf{\Gamma}_t \end{aligned} \quad (54)$$

The corresponding Lagrangian, which characterizes the motion of a 3D curve within the camera space, is derived from the Lagrangian of non-inertial motion as described by [Landau e Lifshitz \(1976\)](#). The mechanical formulation is delineated below:

$$L = \frac{m}{2}\mathbf{v}_{in}^2 + m\mathbf{v}_{in} \cdot (\omega \times \mathbf{r}_{in}) + \frac{m}{2}(\omega \times \mathbf{r}_{in})^2 - m\mathbf{a}_0\mathbf{r} - U, \quad \mathbf{a}_0 = \frac{d\mathbf{v}_0}{dt} \quad (55)$$

Finally, the Lagrangian for the motion of a 3D camera curve is derived using equation (55),

resulting in the following expression:

$$L = \frac{m}{2}(\mathbf{R}\Gamma^r)^2 + \mathbf{R}\Gamma^r \cdot (\mathbf{\Omega}\mathbf{R}\Gamma^r) + \frac{m}{2}(\mathbf{\Omega}\mathbf{R}\Gamma^r)^2 - \mathcal{T}_{tt}\mathbf{R}\Gamma^r - U \quad (56)$$

In this formulation,  $U$  represents the potential energy applied to the system, potentially incorporating additional constraints. The 2D projected version can be derived by calculating  $\gamma_t$  and adapting our previous equations accordingly. This marks the initial attempt to apply Lagrangian mechanics to the motion of curves, with potential applications in visual odometry.

This research also establishes a relationship between the geometric parameter  $s$  of 3D and 2D curves and optical phenomena at a physical level, proposing further investigation to enhance the theory of multiview differential geometry. Fermat's principle, as introduced in Holm (2008a) and presented in (57), describes the action of a stationary ray path  $\mathbf{r}(s)$ .

$$\mathcal{S} = \int_A^B \eta(\mathbf{r}(s))ds = \int_A^B \eta(\mathbf{r}(s))ds = \int_A^B L(\mathbf{r}, \mathbf{r}_s)ds, \quad L(\mathbf{r}, \mathbf{r}_s) = \eta(\mathbf{r}(s))\sqrt{\mathbf{r}_s \cdot \mathbf{r}_s} \quad (57)$$

The axial geometric interpretation of Fermat's principle posits that the light path within optical systems aligns with the line of sight, which is coextensive with the  $z$ -axis of the camera. Consequently, this alignment also corresponds with the image plane projection. The differential element  $ds$  is thereby substituted with  $dz$ , yielding the formulation presented in (58).

$$\mathcal{S} = \int_A^B L(x, x_z, y, y_z, z)dz, \quad L(x, x_z, y, y_z, z) = \eta(x, y, z)\sqrt{x_z^2 + y_z^2 + 1}. \quad (58)$$

Assuming an  $\mathbf{q} = (x, y)$ , (58) product is rewritten as

$$\mathcal{S} = \int_A^B L(\mathbf{q}, \mathbf{q}_z)dz, \quad L(\mathbf{q}, \mathbf{q}_z) = \eta(\mathbf{q}, z)\sqrt{|\mathbf{q}_z|^2 + 1} \quad (59)$$

This assumption leads to the Hamiltonian form, as detailed in E, expressed as follows:

$$H(q, p) = -\sqrt{\eta(q, z)^2 - |p|^2} \quad (60)$$

Here,  $\eta$  represents the refractive index,  $q$  denotes the position coordinates, and  $p$  denotes the conjugate momentum coordinates in the optical system's phase space.

### 3 DYNAMICAL CAMERA MODEL UNDER SURFACE MOTION PATH

The static camera estimation algorithms previously discussed frequently incorporate various mathematical techniques to mitigate issues that may arise. These issues include incorrect polynomial root solutions, matrix rank deficiencies (e.g., the p3p algorithm cannot accurately determine the camera position if the matrices are rank-deficient, or if they are mathematically singular with a determinant of zero, or computationally ill-conditioned, which is characterized by a high condition number—a measure related to the normalized determinant). Additionally, problems may occur in Gaussian elimination, QR decomposition, and Singular Value Decomposition (SVD) computations. Neural network-based approaches, such as those proposed in Xiao et al. (2023), necessitate a substantial amount of input data and are sensitive to noise. While these methods primarily address overall reconstruction challenges from a geometric perspective, they often lack a comprehensive explanation specifically tailored to camera estimation problems.

The key distinction of this research, in contrast to previous methodologies, is the application of principles from physics. The adoption of a continuous-time camera model facilitates the use of an extensive range of source materials for both modeling and analysis. This approach enables the integration of tools and techniques from analytical mechanics, as detailed in 2. Furthermore, phase space analysis was performed using software developed in 2019, known as Hsystem Analysis Andrade et al. (2019). The forthcoming public release of openHSA (open Hamiltonian System Analysis) is currently pending, due to ongoing code review and comprehensive testing by collaborators involved in the project.

#### 3.1 Model inspirations

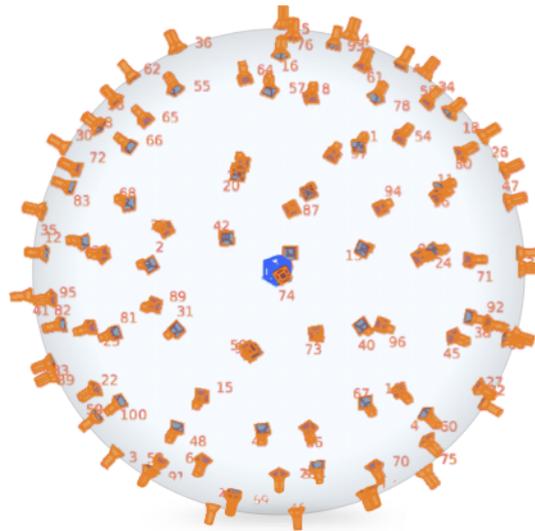
The surface path camera motion model is based on the experimental synthcurves dataset, which is accessible at <https://github.com/labv1z/synthcurves-multiview-3d-dataset> Computer... (2012), Fabbri, Giblin e Kimia (2012), Fabbri e Kimia (2016). In this dataset, cameras are positioned on a spherical surface with the object of interest centrally located, as depicted in image 14.

#### 3.2 Modeling assumptions

The fundamental assumptions underlying this work are outlined to establish a practical framework:

- **Known and Consistent Camera Internal Parameters:** The camera’s internal

Figure 14 – 3D camera distribution from synthcurves



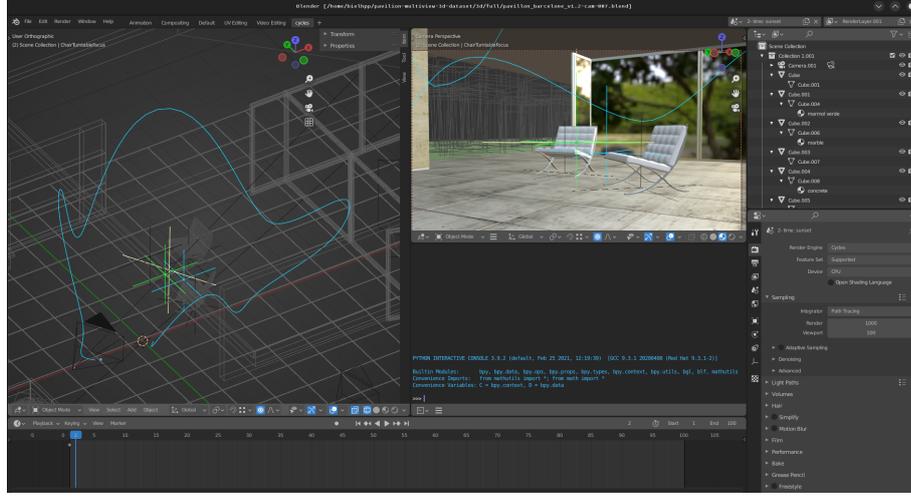
Legend: The orange ones are the cameras and the blue structure in the center is the 3D scene object

Source: [Computer. . . \(2012\)](#), [Fabbri, Giblin e Kimia \(2012\)](#), [Fabbri e Kimia \(2016\)](#).

parameters are assumed to be known and consistent across all images in the dataset. This assumption facilitates future practical applications by enabling the use of less common features in multiview camera relative pose estimation.

- **Implicit Surface or Curve Path Distribution of Cameras:** It is posited that the cameras are arranged along an implicit surface or curve path. The surface camera path is considered the global model, encompassing all potential curve paths that a camera may follow. The curve path, a specific instance of this model, assumes that cameras are arranged along a closed or semi-closed path, as illustrated in [15](#). This approach effectively reduces the model's degrees of freedom from six (comprising three rotational and three translational components) to at most four (one rotational and up to three translational components).
- **Using 3D stationary curves:** The study employs 3D stationary curves ( $\Gamma_t^r = 0$ ), simplifying both 3D and 2D projected curve motion to be solely dependent on camera movement. This is significant in the initial investigation as it suggests that image projection is a consequence of camera motion. Practical experiments, such as the "dark chamber," which represents the primitive pinhole camera model, can be conducted to validate this hypothesis.

Figure 15 – Blender 3D barcelona chair sequence with its current view



Legend: Example of camera curve path of the experimental dataset represented by the blue line on the left panel of Blender.

Source: [Usmezbas, Fabbri e Kimia \(2016\)](#), [Computer... \(2016\)](#) and screenshot by the author, 2024.

### 3.3 The Lagrangian of camera positioning

Despite demonstrating that the complete model can be classified as a rigid body problem, it is important to outline the progression of the modeling process over time to highlight its refinement. Initially, the hypothesis was to model the camera as a free particle traversing a spherical surface. This assumption of a surface camera path implies that the camera's field of view is consistently directed towards the scene, thereby allowing the rotational aspect of the model to be disregarded.

This simplified representation, employed to study the phenomenon under investigation, provided a clearer understanding of the overall absolute camera movement. However, it was hypothesized that the dynamics might differ when considering the relative pose estimation problem. The initial hypothesis suggests that by overlapping images, a temporal trajectory of a point could be formed. Additionally, applying the same logic in camera space could generate the action described in equation (61).

$$\mathcal{S} = \int_a^b |d\Gamma|. \quad (61)$$

To analyze the motion of the curve over time, the total time derivative was computed

using the chain rule, resulting in the following expression:

$$\mathcal{S} = \int_0^t \left| \frac{d\Gamma}{dt} \right| dt, \quad (62)$$

The conclusion that the Lagrangian is  $L = |d\Gamma/dt|$  addresses the motion of a point and, by extension, a curve over time. However, this approach does not capture the motion of the camera itself. **The camera, as a physical object in motion, must adhere to the laws of motion. Instead of using  $\Gamma$  directly as the Lagrangian functional, it should be considered within the context of the camera's movement.**

To describe the camera motion accurately, the Lagrangian should be formulated to reflect the dynamics of the camera as a physical entity. Applying the mechanical form of the Lagrangian equation with  $(\Gamma, \Gamma_t)$  as the generalized coordinates and velocities results in:

$$L(\Gamma, \Gamma_t) = K(\Gamma_t) - U(\Gamma) \quad (63)$$

Considering that  $\Gamma$  incorporates both rotational and translational components, the Lagrangian can be expressed as  $L = L_{rot} + L_{tra}$ . This decomposition is based on the assumption that the point in the reference frame  $\Gamma^r$  does not influence the camera motion, thereby allowing the Lagrangian to be formulated as the sum of its rotational and translational components, as described in equation (64).

$$L = L_{rot} + L_{tra} = (K_{rot}(\Gamma_t) - U_{rot}(\Gamma)) + (K_{tra}(\Gamma_t) - U_{tra}(\Gamma)) \quad (64)$$

The initial model, which constitutes the main focus of this initial exploration, characterizes a free particle within spherical coordinates, under the assumption that the camera's rotational component does not influence the phase space analysis and that the translational potential  $U_{tra}(\Gamma)$  is zero. Accordingly, the Lagrangian for this model is given by equation (65).

$$L(\mathcal{T}_t) = \frac{m}{2} \mathcal{T}_t^2 \quad (65)$$

A more generalized formulation accounts for free rotation about all three axes of the camera, indicating that the camera's orientation is no longer constrained to focus on the scene. Consequently, the Lagrangian function becomes more complex, as represented in equation (66). This expanded model includes both rotational and translational components, thereby extending the analysis to encompass both rotational and translational phase spaces.

$$L = \frac{1}{2} \omega \mathcal{I} \omega + \frac{m}{2} \mathcal{T}_t^2 \quad (66)$$

In this formulation, both the mass  $m$  and the moment of inertia tensor  $\mathcal{I}$  are parameters representing information-theoretic interpretations of their respective physical concepts.

### 3.4 Phase space analysis of camera position under spherical surface constraint

An advancement of this approach, compared to previous methodologies, lies not only in the continuous-time formulation of camera movement but also in the introduction of phase space analysis. This analytical tool provides a comprehensive framework for examining the dynamics of camera systems across various camera path surfaces. The absence of rotational influence in the phase space analysis is due to the use of surface path camera motion with a focus on the scene. This approach results in a model where only the  $z$ -axis of the camera remains unconstrained, while the other axes are fixed and dependent on the  $z$ -axis rotation. Consequently, the rotational component can be excluded from the phase space analysis, leaving only the translational component to contribute to the analysis.

The proposed hypothesis suggests that the energy level orbits derived from the physical model can be translated not only in terms of spatial positioning but also in their influence on the captured image. To establish this, the translation matrix in spherical coordinates is presented, as specified in equation (67).

$$\mathcal{T}(t) = r \cdot \begin{bmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{bmatrix}. \quad (67)$$

In this framework, the angles  $\theta$  and  $\phi$  are treated as time-dependent functions, while  $r$  denotes a fixed radius. Previous works, such as those by [Landau e Lifshitz \(1976\)](#) and [Lemos \(2018\)](#), have provided formulations for the Lagrangian and Hamiltonian functions in spherical coordinates using time-dependent radius function. However, analysis indicates that these formulations are not directly applicable to the practical model under consideration, as the critical radial momentum is zero, meaning only the angular components  $\theta$  and  $\phi$  vary over time. The choice of rotation representation is flexible; however, the model assumes the camera's rotational axes are defined by the unit vectors corresponding to the spherical coordinate system  $(\hat{\theta}, \hat{\phi}, \hat{r})$ . Given that the model allows free rotation around the  $z$ -axis, careful consideration is necessary in selecting the appropriate vectors  $(\hat{\theta}, \hat{\phi}, \hat{r})$ , since a point can lie on multiple intersecting paths. The rotational matrix  $\mathbf{R}(t) = \text{span}(\hat{\phi}, \hat{\theta}, -\hat{r})$  is delineated in (68), with a practical approach for implementation

proposed in 3.5.

$$\mathbf{R}(t) = \begin{bmatrix} -\sin \phi & \cos \theta \cos \phi & -\sin \theta \cos \phi \\ \cos \phi & \cos \theta \sin \phi & -\sin \theta \sin \phi \\ 0 & -\sin \theta & -\cos \theta \end{bmatrix}, \quad (68)$$

Since the camera's rotational component does not influence the phase space analysis and the camera motion is considered to be under free motion, the computation proceeds from equation (65). The first step involves computing the time differential of  $\mathcal{T}_t$ , yielding the following result:

$$\mathcal{T}_t(t) = r \cdot \begin{bmatrix} \cos \theta \cos \phi \dot{\theta}_t - \sin \theta \sin \phi \dot{\phi}_t \\ \cos \theta \sin \phi \dot{\theta}_t + \sin \theta \cos \phi \dot{\phi}_t \\ -\sin \theta \dot{\theta}_t \end{bmatrix}. \quad (69)$$

Next, the inner product  $\mathcal{T}_t^2 = \langle \mathcal{T}_t, \mathcal{T}_t \rangle$  is computed, resulting in the expression:

$$\begin{aligned} \mathcal{T}_t^2 &= r^2[(\cos \theta \cos \phi \dot{\theta}_t - \sin \theta \sin \phi \dot{\phi}_t)^2 + (\cos \theta \sin \phi \dot{\theta}_t + \sin \theta \cos \phi \dot{\phi}_t)^2 + (-\sin \theta \dot{\theta}_t)^2] \\ &= r^2[(\cos \theta \cos \phi \dot{\theta}_t)^2 - 2 \cos \theta \cos \phi \dot{\theta}_t \sin \theta \sin \phi \dot{\phi}_t + (\sin \theta \sin \phi \dot{\phi}_t)^2 + (\cos \theta \sin \phi \dot{\theta}_t)^2 + \\ &\quad 2 \cos \theta \sin \phi \dot{\theta}_t \sin \theta \cos \phi \dot{\phi}_t + (\sin \theta \cos \phi \dot{\phi}_t)^2 + (-\sin \theta \dot{\theta}_t)^2] \\ &= r^2[(\cos \theta \cos \phi \dot{\theta}_t)^2 + (\sin \theta \sin \phi \dot{\phi}_t)^2 + (\cos \theta \sin \phi \dot{\theta}_t)^2 + (\sin \theta \cos \phi \dot{\phi}_t)^2 + (-\sin \theta \dot{\theta}_t)^2] \\ &= r^2\{[(\sin^2 \phi + \cos^2 \phi) \cos^2 \theta + \sin^2 \theta] \dot{\theta}_t^2 + (\sin^2 \phi + \cos^2 \phi) \sin^2 \theta \dot{\phi}_t^2\} \\ &= r^2(\dot{\theta}_t^2 + \sin^2 \theta \dot{\phi}_t^2). \end{aligned} \quad (70)$$

By substituting the computed  $\mathcal{T}_t^2$  into equation (65) and assuming the parametric mass  $m = 1$ , the Lagrangian function is obtained as:

$$L(\theta, \dot{\theta}_t, \dot{\phi}_t) = \frac{r^2}{2}(\dot{\theta}_t^2 + \sin^2 \theta \dot{\phi}_t^2). \quad (71)$$

Applying the Euler-Lagrange equation yields a system of second-order differential equations.

$$\begin{cases} \frac{\partial L}{\partial \theta} - \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_t} \right) = 0 \rightarrow r^2(-\dot{\theta}_t + \sin \theta \cos \theta \dot{\phi}_t^2) = 0 \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\phi}_t} \right) = 0 \rightarrow \frac{\partial L}{\partial \dot{\phi}_t} = r^2 \sin^2 \theta \dot{\phi}_t = c_0 \rightarrow \dot{\phi}_t = \frac{c_0}{r^2 \sin^2 \theta} \end{cases}. \quad (72)$$

This indicates that the momentum is conserved in the  $\phi$  direction ( $p_\phi = c_0$ ), which reduces the system to a single degree of freedom. To conduct the phase space analysis,

the Hamiltonian, derived from the corresponding Lagrangian, must be calculated, followed by solving Hamilton's equations. Initially,  $\theta_t$  is expressed in terms of  $p_\theta$ .

$$p_\theta = r^2 \theta_t \rightarrow \theta_t = \frac{1}{r^2} p_\theta. \quad (73)$$

Compute the Legendre transform  $H(q, p) = \langle p, q_t \rangle - L(q, q_t)$  with  $p = p_\theta$  and  $q = \theta$ , as given in equation (74).

$$H(\theta, p_\theta) = \frac{1}{2r^2} (p_\theta^2 - \csc^2 \theta c_0^2). \quad (74)$$

Next, construct the Hamilton's system of equations

$$\begin{cases} \frac{d\theta}{dt} = \frac{1}{r^2} p_\theta \\ \frac{dp_\theta}{dt} = \frac{c_0^2}{r^2} \csc^2 \theta \cot \theta \end{cases}. \quad (75)$$

The critical points are obtained by solving the simultaneous equations  $d\theta/dt = 0$  and  $dp_\theta/dt = 0$ . This yields the following results:

$$\begin{cases} \frac{d\theta}{dt} = \frac{1}{r^2} p_\theta = 0 \rightarrow p_\theta = 0 \\ \frac{dp_\theta}{dt} = \frac{c_0^2}{r^2} \csc^2 \theta \cot \theta = \frac{c_0^2}{r^2} \csc^3 \theta \cos \theta = 0 \rightarrow \theta = \pm \frac{\pi}{2} \end{cases}. \quad (76)$$

The critical points  $P_{1,2} = (\pm\pi/2, 0)$  indicate that the camera experiences no movement at the north and south poles of the sphere, while maintaining the freedom to move in any azimuthal direction. The corresponding energies at these critical points are  $E_{north} = E_{south} = -c_0^2/2r^2$ . The Jacobian matrix is then computed by performing a Taylor expansion of (75) to the first derivative, resulting in (77).

$$\mathbf{J} = \begin{bmatrix} 0 & \frac{1}{r^2} \\ \csc^2 \theta (\csc^2 \theta + 2 \cot^2 \theta) \frac{c_0^2}{r^2} & 0 \end{bmatrix}. \quad (77)$$

Both critical points have same Jacobian matrix (78)

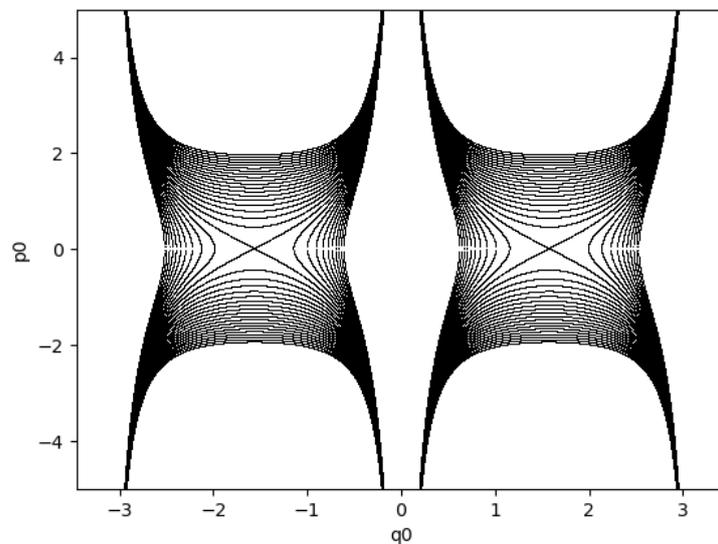
$$\mathbf{J} = \begin{bmatrix} 0 & \frac{1}{r^2} \\ \frac{c_0^2}{r^2} & 0 \end{bmatrix}. \quad (78)$$

Next, compute the eigenvalues of the Jacobian matrix, which are used to analyze the local behavior near the poles.

$$\det(\mathbf{J} - \lambda \mathbf{Id}) = \begin{vmatrix} -\lambda & \frac{1}{r^2} \\ \frac{c_0^2}{r^2} & -\lambda \end{vmatrix} = \lambda^2 - \frac{c_0^2}{r^4} = 0 \rightarrow \lambda_{1,2} = \pm \frac{c_0}{r^2}, \quad (79)$$

The critical points are characterized as hyperbolic saddle points, signifying their inherent instability. As a result, even minor perturbations in energy will lead the system away from these points without returning. This instability poses challenges for the estimation process, potentially hindering its efficiency or, in extreme cases, compromising its feasibility, as detailed in subsequent sections. Figure 16 depicts the phase space  $p_\theta \times \theta$ .

Figure 16 – Phase space  $p_\theta \times \theta$  ( $p_0$  and  $q_0$  respectively)



Source: From author software openHSA, 2024.

### 3.4.1 Possible relationships between image and cameras in energy levels

With the physical framework for the camera problem established, we propose a relationship between energy levels, as represented by phase space orbits, and image states. This relationship addresses both absolute and relative pose estimation problems. Specifically, for a given orbit, various states of the image correspond to positions on a circumference where  $\theta = \theta_0$  and  $\phi \in [0, 2\pi]$ . Given that the rotational component has been removed from the phase space analysis, it is sufficient to state that in the context of a rotation-free scenario across different cameras under a spherical motion model, the cameras maintain fixed angles  $\theta_0$  and  $\phi_0$  while varying their radial distance. This is equivalent to asserting that the camera is either approaching or receding from the scene while maintaining the same angular orientation.

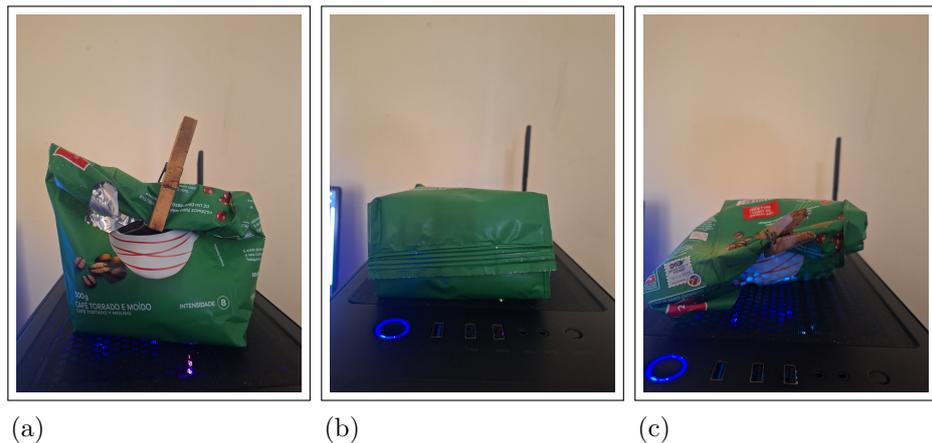
While certain challenges are directly associated with physical problems, the image projection resulting from the camera position and the geometric properties of the 3D scene

can introduce additional complexities. The following subsections present hypotheses on the relationship between the phase space, represented by the set of camera positions that may lead to a specific energy orbital, and the geometric information of the image projection.

#### 3.4.1.1 Absolute single camera estimation problem

The image formed by a camera at a critical point position is characterized by degeneration. Degeneration occurs when the majority of curves are occluded in the resultant image, rendering the image unable to provide sufficient information for the error function in practical algorithms. This phenomenon is exemplified in image 17, illustrating the hypothesis.

Figure 17 – A coffee bag example.



Legend: The top and bottom of coffee bag occludes partially or completely the curves represented in the front of it. (a) Front side, (b) Bottom side and (c) Top side.

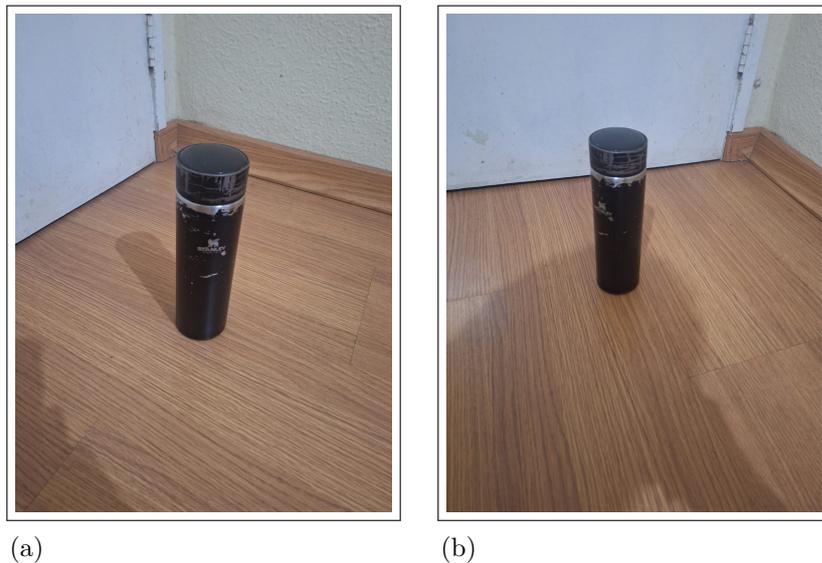
Source: The author, 2024.

#### 3.4.1.2 Absolute multiple camera estimation problem

Images of a geometrically ambiguous 3D scene can cause practical algorithms to inaccurately determine the camera's position on the same energy level orbit. This issue is particularly pertinent in incremental pipeline applications, such as those implemented in software like openMVG. After constructing the initial reconstruction, these systems typically use single camera pose estimation to identify the remaining images. If a geo-

metrically ambiguous image corresponds to a camera located on the same energy orbital but at a different position, practical algorithms might incorrectly interpret this image as originating from the same position as the initial camera determined by the resection method. This hypothesis is illustrated in image 18.

Figure 18 – Coffee bottle example.



Legend: Geometrically speaking, two images with ambiguity, the same energy level can make the dynamical camera will find one camera and will understand the other ones as the first found.

Source: The author, 2024

#### 3.4.1.3 Relative camera estimation problem

In camera relative pose estimation, two main hypotheses may arise in practical applications with a dynamic camera approach:

- Geometric ambiguity in one or more images can be interpreted as the camera either following a closed path, oscillating between positions, or remaining stationary. Such ambiguities may occur if matching algorithms are unable to accurately distinguish between repeated observations of the same point, which can result in incorrect camera estimations.
- In practical applications using a dynamic approach, image projections from cameras positioned at critical points can be understood as being rotated by  $90^\circ$  relative to each other. This can be derived from the parametric form of the projection equation

$\rho\gamma = \mathbf{R}\rho^r\gamma^r + \mathcal{T}$ . If  $\rho^r$  is arbitrary and both  $\mathbf{R}$  and  $\mathcal{T}$  correspond to critical point regions, then the condition  $\langle \rho\gamma, \rho^r\gamma^r \rangle = 0$  holds, indicating that the projections are orthogonal.

### 3.5 Dynamic camera pose estimation

The static camera position estimation algorithms discussed in 1 are based exclusively on the projective geometry properties of the image using algebraic formulations, with the assumption that the calibration matrix is known. Consequently, the formulations that account for the case where the camera is not calibrated are not considered at this stage. The table below summarizes the number of points required by each algorithm outlined in 1.

Table 3 – Static camera estimation algorithms table.

Type	Algorithm	Required
Single Camera	<b>DLT</b>	At least 4 points across 3D scene and image
	<b>P3P</b>	Exactly 3 points across 3D scene and image
Bifocal	<b>Five-point algorithm</b>	Exactly 5 points across two images
Trifocal	<b>Chicago Problem</b>	3 points and 2 lines across three images

Source: The author, 2024.

The dynamical approach posits that image projections are determined by the camera's position at a specific instant in time. For single-camera position estimation, it is proposed that using two 3D oriented points and two oriented image correspondences is sufficient to determine the camera location. This approach is inspired by the P2Pt method [Fabbri, Giblin e Kimia \(2020\)](#), which employs tangents as additional constraints to deal with multiple rotation representations. For bifocal and trifocal scenarios, the hypothesis suggests that three oriented points, similar to those used in trifocal cases, could be effective, though it needs to be validated for bifocal configurations. This is summarized in the table below: Given that images may exhibit local or global geometric ambiguity,

Table 4 – Dynamic camera estimation algorithms table.

Type	Required
Single Camera	At least two 3D points and tangents $(\Gamma, \mathbf{T})$ and two set of image points and tangent $(\gamma, \mathbf{t})$ correspondences
Bifocal	At least three points and tangents $(\gamma, \mathbf{t})$ across two images
Trifocal	At least three points and tangents $(\gamma, \mathbf{t})$ across three images

Source: The author,2024.

it is proposed that utilizing multiple correspondences can mitigate these issues. Moreo-

ver, employing additional correspondences enhances the robustness of practical estimation solvers. Given the potential for multiple rotation representations and the surface camera motion path, it is possible to constrain the camera rotation such that only the camera  $z$ -axis is known. This constraint permits the use of Rodrigues' rotation formula as the rotation representation, leading to the following result:

$$\mathbf{R} = \mathbf{Id} + \sin(\nu)\mathbf{A}|_{\mathbf{p}} + (1 - \cos(\nu))\mathbf{A}^2|_{\mathbf{p}} \quad (80)$$

Where  $\mathbf{p} = (\theta_0, \phi_0)^\top$  and  $\mathbf{A}$  is the skew-matrix form of the camera  $z$ -axis that is described in (81)

$$\mathbf{A} = \begin{bmatrix} 0 & \cos \theta & -\sin \theta \sin \phi \\ -\cos \theta & 0 & \sin \theta \cos \phi \\ \sin \theta \sin \phi & -\sin \theta \cos \phi & 0 \end{bmatrix}. \quad (81)$$

To compute the translational motion in a surface path camera model, it is necessary to derive the Lagrangian system of second-order differential equations. This process yields the following:

$$\begin{cases} \theta_{tt} - 2 \sin \theta \cos \theta \phi_t^2 = 0 \rightarrow \theta_{tt} - \sin 2\theta \phi_t^2 = 0 \\ \sin \theta^2 \phi_{tt} = 0 \end{cases}. \quad (82)$$

Now build a system of ODEs

$$\begin{cases} \theta_t = p_\theta \\ \phi_t = \frac{c_0}{\sin \theta^2} \\ p_{\theta_t} = \sin 2\theta \phi_t^2 \end{cases}. \quad (83)$$

Assuming that the simplified Lagrangian system of equations is more suitable than the Hamiltonian approach for directly addressing camera dynamics space and avoiding potential simplifications from reducing degrees of freedom, the resulting simplified Lagrangian system yields first-order differential equations of the form  $dq/dt = \mathbf{F}(q, t)$ . Although a functional algorithm is not yet available, a step-by-step procedure for computing both absolute and relative pose estimation will be detailed in the following sections.

### 3.5.1 Absolute pose estimation

The proposed steps for absolute pose estimation are as follows:

1. Input: A 3D scene object, the calibration matrix and an Image. Output: Rotation and translation matrices
2. Choose a random  $(\theta_0, \dot{\theta}_0)$  and  $(\phi_0, \dot{\phi}_0)$  to begin the system.
3. Compute with Runge Kutta the step  $i + 1$  from the Euler-Lagrange system of equations.
4. For each  $\eta \in [0, 2\pi)$  compute the rotation matrix with Axis-angle Rodrigues notation.
  - (a) Project both the 3D point in image with  $\gamma_{proj} = \mathcal{K}_{\rho}^{\frac{1}{\rho}}(\mathbf{R}\mathbf{\Gamma} + \mathcal{T})$ ,  $\rho = \mathbf{\Gamma}_z \mathbf{e}_3$  and 3D tangent  $\mathbf{t}_{proj} = \mathcal{K}_{\rho}^{\frac{1}{\rho}}(\mathbf{R}\mathbf{T}) - \frac{\rho_s}{\rho^2}(\mathbf{R}\mathbf{\Gamma} + \mathcal{T})$ ,  $\rho = \mathbf{\Gamma}_z \mathbf{e}_3$ ,  $\rho_s = \mathbf{T}_z \mathbf{e}_3$
  - (b) Compute the distance  $d(\gamma, \gamma_{proj})$  and angle  $\angle(\mathbf{t}, \mathbf{t}_{proj})$ .
  - (c) If  $d(\gamma, \gamma_{proj}) < \delta$  and  $\angle(\mathbf{t}, \mathbf{t}_{proj}) < \epsilon$ , finish the process.
5. If the camera was not found, update the angles and velocities and go to step 3.
6. If in some iteration  $\theta = \pm \frac{\pi}{2}$ , go to step 2.

### 3.5.2 Relative pose estimation

A notable aspect of the model for relative camera pose estimation is its capacity to reduce an  $N \geq 2$  camera pose estimation problem to multiple "absolute pose estimation" problems. This is accomplished by analyzing each camera pose  $\gamma_{im1}$  at  $t = t_1$ ,  $\gamma_{im2}$  at  $t = t_2$ , and so on up to  $\gamma_{imN}$  at  $t = t_N$  as separate absolute pose estimation tasks.

The relative pose estimation process is similar to the absolute pose estimation, but with three key differences:

1. **Parametric Form:** Employ the parametric form  $\mathbf{\Gamma} = \rho\gamma$  to represent camera poses.
2. **Initial Depth:** Use an arbitrary initial depth  $\rho_0$ , with  $\rho_0 = 1$  recommended for simplification.
3. **Calibration Matrix:** The calibration matrix is not required, as it is assumed that all images are from the same calibrated camera. The proposed steps for relative pose estimation are as follows
  1. Input: N images  $im_1, \dots, im_N$  correspondences. Output: Rotation(s) and translation(s) matrices  $[R_k | \mathcal{T}_k]$ ,  $k = 1, \dots, N$
  2. Choose a random  $(\theta_0, \dot{\theta}_0)$  and  $(\phi_0, \dot{\phi}_0)$  to begin the system.

3. Compute with Runge Kutta the step  $i + 1$  from the Euler-Lagrange system of equations.
4. For each  $\eta \in [0, 2\pi)$  compute the rotation matrix with Axis-angle Rodrigues notation.
  - (a) Reproject both the point and tangent from the first image to the image with  $(\rho\gamma)_{reproj} = \mathbf{R}_k\rho_0\gamma_0 + \mathcal{T}_k$  and  $(\rho_s\gamma + \rho\mathbf{t})_{reproj} = \mathbf{R}_k(\rho_{s0}\gamma_0 + \rho_0\mathbf{t}_0)$
  - (b) Compute the distance  $d(\gamma, \gamma_{reproj})$  and angle  $\angle(\mathbf{t}, \mathbf{t}_{reproj})$ .
  - (c) If  $d(\gamma, \gamma_{reproj}) < \delta$  and  $\angle(\mathbf{t}, \mathbf{t}_{reproj}) < \epsilon$ , finish the process.
5. If the camera was found, store  $[\mathbf{R}_k|\mathcal{T}_k]$ , update the angles and velocities and go to step 3. Otherwise update the angles and velocities and go to step 3.
6. If in some iteration  $\theta = \pm\frac{\pi}{2}$ , go to step 2.

## CONCLUSION AND FUTURE WORK

This dissertation presents an extension of the dynamical camera theory, building upon the temporal parameter theory discussed in [Fabbri e Kimia \(2016\)](#), and addressing previously unexplored aspects. The work introduces a novel relationship between mechanics and camera pose estimation through the application of Lagrangian and Hamiltonian mechanics. During the first year of research, the focus was on advancing the trifocal method and addressing implementation challenges. The robustness of the trifocal method was demonstrated in [Andrade et al. \(2023\)](#), achieving 79.65% inliers with the best model obtained via RANSAC, and showing stability in deriving a model with camera rotation-free properties.

Despite its robustness, the trifocal method's high complexity necessitated substantial computational effort to identify singularity regions within the model. The use of generic formulations, such as homotopy continuation adapted for computing the trifocal tensor, did not fully address the potential issues inherent in the method. To address these challenges, an initial formulation of a continuous camera approach was developed. This approach, representing a type of continuation in real space, offers a more precise representation of the phenomena and facilitates easier study and optimization. As a result, the project `Hsystems-analysis`, developed by the author, has been improved and will soon be available as `Open Hamiltonian System Analysis (openHSA)`.

On the theoretical side, additional concepts were introduced regarding geometric curves that depend on time, such as the velocity of 3D and 2D points and their respective tangents. The appendix includes details on acceleration for both cases. Propositions were also made to provide further insight into the topological aspects of images. This includes an explanation of why an image can be considered both a differential manifold and a member of the Galilean group, and how these perspectives can be applied to study functionals relevant to each problem.

The difference between point and curve correspondences for identifying rotation and translation across images was also discussed. This involved matching curves and points by modeling a potential 3D object and rotating it to align with each image. This exploration, while seemingly peripheral, contributes to the development of various functionals for different applications, such as those recently developed by the LEMS group for multiview triangulation (currently in the process of publication).

Subsequently, the model is presented with its fundamental constraints. These constraints include the assumption that the reference 3D point is part of a stationary rigid curve, and the formulation of a surface camera path constraint. This constraint facilitates the temporal distribution of cameras, resulting in a model where the primary issues are determined by translation, while rotation is unrestricted only around the  $z$ -axis

of the camera. Rotation around the  $x$  and  $y$  axes of the camera is fixed according to the surface constraint. Hypotheses were presented regarding the interpretation of energy orbitals in phase space analysis and their relation to image states. It was concluded that challenges in camera pose estimation may stem not only from the physical model but also from the geometric structure of the image, which can lead to incorrect location estimation.

A procedure for determining both absolute and relative pose estimation, based on the theory presented, was also proposed.

## Future work

In the theoretical investigation of camera positioning problems using physics, the initial step involves assuming that the curve is not stationary. This assumption introduces a potential on the camera side, arising from the 3D camera space point. Even without a potential constraint, the standalone rotation of the 3D point contributes to the kinetic energy.

The subsequent step involves analyzing a scenario where the camera is fixed while the scene is moving. This results in both the camera and the scene undergoing movement according to the following:

- **First Configuration:** This configuration facilitates the examination of the dynamics of the calibration matrix. By analyzing the camera lens equation, it is proposed that variations in parameters such as lens radius or refractive index over time might be significant. The hypothesis is that, despite being undetectable by the human eye, photon pressure could affect the lens's internal calibration by altering the refractive index. Future work will focus on applying Fermat's Principle with time-dependent refractive index and lens radius, and assessing these effects. Additionally, the potential impact of aperture velocity on internal parameters may be investigated if necessary.
- **Moving Scene and Camera Interaction:** When both the camera and scene are in motion, interactions between points in camera space and scene space can occur. This investigation will focus on understanding how a moving scene affects camera position algorithms and overall reconstruction accuracy.

Subsequently, the investigation will focus on geometric parameters in the context of physics. Initial studies into the dynamics of Frenet frames indicated that curvature and torsion significantly influence phase space direction within the **TNB** space. Following this, cuspidal theory will be explored, as preliminary results suggest that critical regions are linked to occluding curves or epipolar tangencies. The objective is to further examine the relationship between critical points on curves and their associated occlusions.

We have proposed to future research the application of the Lagrangian of axial geometry and Fermat’s principle as presented by [Holm \(2008a\)](#) to study geometric parameters using optical Lagrangian and Hamiltonian formulations. This approach may facilitate the exploration of the phase space of camera optics and enable connections with established differential geometry concepts. Further investigation of the literature is needed to determine whether this idea has been previously explored and to build upon those findings.

The latest review of literature revealed that what initially appeared to be a Hamilton-Jacobi formulation was, with the assistance of Prof. Germano, identified as a Lagrangian density functional. This distinction, supported by materials from field theory, suggests potential applications of field theory in multiview reconstruction and camera estimation theory.

Additionally, a recent study by Pascual [Pascual-Escudero et al. \(2021\)](#) provides a singularity analysis of the P4P problem and proposes that a similar approach could be applied to trifocal and P2Pt problems. A detailed review of this paper is required to determine its applicability and potential improvements for our work.

On the practical side, the next steps are as follows:

1. **Complete Integration Tests:** Finalize the integration of the trifocal and P2Pt methods within the openMVG framework.
2. **Further Developments:**
3. **Bifurcation Analysis:** Since direct identification of topological regions in homotopic differential equations via phase space analysis has been challenging, bifurcation theory will be applied to investigate stability loss in these systems. Bifurcations occur when a dynamical system begins to lose integrability, indicating that the homotopic differential equation is approaching instability.
4. **Dynamical Camera Model:** Develop and experiment with the dynamical camera model as outlined in the presented methodology. Report the results to validate and refine the approach.
5. **Expansion and Publication:** Publish and expand the Open Hamiltonian System Analysis (openHSA) package, enhancing its capabilities for both qualitative and quantitative analysis in physics. This will also involve creating an internal framework to study qualitative aspects of multiview reconstruction, curves, and camera-related problems.

## REFERENCES

- ANDRADE, G. C. et al. O pacote hsystems-analisys. *XXII ENCONTRO NACIONAL DE MODELAGEM COMPUTACIONAL e X ENCONTRO DE CIÊNCIA E TECNOLOGIA DE MATERIAIS*, v. 1, p. 1667–1686, 2019. [46](#)
- ANDRADE, G. C. et al. ExtensÃo de pipelines em reconstruÇÃo 3d usando geometria trinocular. *Anais do Encontro Nacional de Modelagem Computacional, Encontro de CiÃncia e Tecnologia de Materiais*, p. 1–10, 2023. [16](#), [60](#)
- ARKIT, A. Understanding arkit tracking and detection. WWDC. 2018. [15](#)
- CARMO, M. P. do. *Differential Geometry of Curves and Surfaces*. New Jersey: Prentice-Hall, 1976. [70](#)
- CIN, A. P. D. et al. Minimal perspective autocalibration. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2024. p. 5064–5073. [42](#)
- CIPOLLA, R.; GIBLIN, P. *Visual Motion of Curves and Surfaces*. [S.l.]: Cambridge University Press, 1999. [13](#)
- COMPUTER Vision - ECCV 2012, 12th European Conference on Computer Vision, Firenze, Italy, October 7-13, 2012, Proceedings, (Lecture Notes in Computer Science). [S.l.]: Springer, 2012. [46](#), [47](#)
- COMPUTER Vision - ECCV 2016, 14th European Conference on Computer Vision, Amsterdam, Netherlands, October 8-16, 2016, Proceedings, (Lecture Notes in Computer Science). [S.l.]: Springer, 2016. [48](#)
- FABBRI, R. et al. Trifocal relative pose from lines at points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 45, n. 6, p. 7870–7884, 2023. [28](#)
- FABBRI, R.; GIBLIN, P.; KIMIA, B. Camera pose estimation using first-order curve differential geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2020. [13](#), [21](#), [56](#)
- FABBRI, R.; GIBLIN, P. J.; KIMIA, B. B. Camera pose estimation using first-order curve differential geometry. In: *Proceedings of the IEEE European Conference in Computer Vision*. [S.l.]: Springer, 2012. (Lecture Notes in Computer Science). [46](#), [47](#)
- FABBRI, R.; KIMIA, B. B. 3D curve sketch: Flexible curve-based stereo reconstruction and calibration. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2010. [13](#), [17](#)
- FABBRI, R.; KIMIA, B. B. Multiview differential geometry of curves. *International Journal of Computer Vision*, Springer, v. 117, p. 1–23, 2016. [13](#), [16](#), [18](#), [19](#), [33](#), [37](#), [38](#), [44](#), [46](#), [47](#), [60](#)
- FRANKEL, T. *The Geometry of Physics: An Introduction*. [S.l.]: Cambridge University Press, 2011. ISBN 9781139505611. [35](#), [36](#), [38](#), [71](#), [72](#)

- HARTLEY, R. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 6, p. 580–593, 1997. [15](#)
- HARTLEY, R. I.; ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. Second. [S.l.]: Cambridge University Press, ISBN: 0521540518, 2004. [13](#), [14](#), [16](#), [21](#)
- HELOU, M.; SHAHPASKI, M.; SÜSSTRUNK, S. Solving the depth ambiguity in single-perspective images. *OSA Continuum*, v. 2, p. 2901, 10 2019. [14](#)
- HOLM, D. D. *Geometric mechanics: Part I: Dynamics and Symmetry*. 2. ed. London : Hackensack, NJ: Imperial College Press, 2008. v. 1. (G - Reference, Information and Interdisciplinary Subjects Series, v. 1). OCLC: ocn225820372. ISBN 9781848161559. [19](#), [35](#), [37](#), [43](#), [45](#), [62](#)
- HOLM, D. D. *Geometric Mechanics: Part II: Rotating, Translating and Rolling*. [S.l.]: IMPERIAL COLLEGE PRESS, 2008. ISBN 9781848161559 9781848161573. [19](#), [35](#), [43](#)
- KIMIA, B. B.; FRANKEL, I.; POPESCU, A.-M. Euler spiral for shape completion. *International Journal of Computer Vision*, v. 54, p. 159–182, 2003. [18](#)
- LANDAU, L. D.; LIFSHITZ, E. M. *Mechanics*. Third. Oxford: Butterworth-Heinemann Limited, 1976. v. 1. (Course of Theoretical Physics, v. 1). [35](#), [43](#), [44](#), [50](#)
- LEMOS, N. A. *Analytical Mechanics*. [S.l.]: Cambridge University Press, 2018. [43](#), [50](#)
- MA, Y. et al. *An invitation to 3D vision*. [S.l.]: Springer, 2004. [18](#), [19](#), [37](#), [43](#)
- MOISAN, L.; MOULON, P.; MONASSE, P. Automatic homographic registration of a pair of images, with a contrario elimination of outliers. *Image Processing On Line*, v. 2, p. 56–73, 2012. [25](#)
- MOULON, P.; MONASSE, P.; MARLET, R. Adaptive structure from motion with a contrario model estimation. In: *Proceedings of the Asian Computer Vision Conference (ACCV 2012)*. [S.l.]: Springer Berlin Heidelberg, 2012. p. 257–270. [25](#)
- MOULON, P. et al. OpenMVG: Open multiple view geometry. In: SPRINGER. *International Workshop on Reproducible Research in Pattern Recognition*. [S.l.], 2016. p. 60–74. [13](#)
- NADJAFIKHAH, M.; FOROUGH, A. Galilean geometry of motions. *Applied Sciences*, v. 11, 07 2007. [40](#)
- NISTÉR, D. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, USA, v. 26, n. 6, p. 756–777, jun 2004. ISSN 0162-8828. [15](#), [25](#), [26](#), [27](#)
- PASCUAL-ESCUADERO, B. et al. Complete singularity analysis for the perspective-four-point problem. *International Journal of Computer Vision*, Springer, v. 129, n. 4, p. 1217–1237, 2021. [62](#)
- PERSSON, M.; NORDBERG, K. Lambda twist: An accurate fast robust perspective three point (p3p) solver. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. [13](#), [22](#), [23](#)

- PRADO, N. F.-F. C. P. C. do. *Caos: uma Introdução*. [S.l.]: Editora Blucher, 1994. 31
- RABOZZI, M. et al. Five-point algorithm: An efficient cloud-based fpga implementation. *2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, p. 1–8, 2018. 25
- SCHÖNBERGER, J. L.; FRAHM, J.-M. Structure-from-motion revisited. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. 13
- SCHÖNBERGER, J. L. et al. Pixelwise view selection for unstructured multi-view stereo. In: *European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2016. 13
- USUMEZBAS, A.; FABBRI, R.; KIMIA, B. B. From multiview image curves to 3D drawings. In: *Proceedings of the European Conference in Computer Vision*. [S.l.: s.n.], 2016. 48
- VENTURA, J. S. B. C.; FABBRI, R.; MOURA NETO, F. D. Visual data science for the optimization of numerical trifocal geometry algorithms. *Anais do Encontro Nacional de Modelagem Computacional, Encontro de Ciência e Tecnologia de Materiais, Conferência Sul em Modelagem Computacional e Seminário e Workshop em Engenharia Oceânica*, p. 1–10, 2022. 31
- XIAO, Y. et al. Level-s<sup>2</sup>fm: Structure from motion on neural level set of implicit surfaces. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2023. 15, 46

## APPENDIX A – How compute stacked form from five point

Here we are going to show how is computed the stacked form of five point algorithm since the original paper from Niéster does not show how and seems quite confusing. Assume  $\mathbf{x}'\mathbf{E}\mathbf{x}$  with  $\mathbf{x} = (x, y, w)^\top$ ,  $\mathbf{x}' = (x', y', w')^\top$  and  $\mathbf{E}$  a three by three essential matrix.

$$\tilde{\mathbf{x}}'\mathbf{E}\mathbf{x} = [x', y', w'] \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \quad (84)$$

$$xx'e_{11} + xy'e_{21} + xw'e_{31} + yx'e_{12} + yy'e_{22} + yw'e_{32} + yx'e_{13} + wy'e_{23} + ww'e_{33} = \tilde{\mathbf{x}}'\tilde{\mathbf{E}}$$

**APPENDIX B** – How we compute  $\Gamma$ ,  $\gamma$  geometric tangents, velocities and acceleration with  $s = s(t)$

Let's take  $\Gamma(s, t) = \Gamma(s(t), t)$  and compute total differential  $d\Gamma$

$$d\Gamma = \frac{\partial\Gamma}{\partial s} ds + \frac{\partial\Gamma}{\partial t} dt \quad (85)$$

The 3D point velocity is obtained by computing total time derivative:

$$\frac{d\Gamma}{dt} = \frac{\partial\Gamma}{\partial s} \frac{ds}{dt} + \frac{\partial\Gamma}{\partial t} \quad (86)$$

The 3d point acceleration is computing taking second total time derivative

$$\frac{d^2\Gamma}{dt^2} = \frac{d}{dt} \left( \frac{d\Gamma}{dt} \right) = \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial s} \frac{ds}{dt} + \frac{\partial\Gamma}{\partial t} \right) = \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial s} \frac{ds}{dt} \right) + \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial t} \right) = \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial s} \right) \frac{ds}{dt} + \frac{\partial\Gamma}{\partial t} \frac{d^2s}{dt^2} + \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial t} \right) \quad (87)$$

The 3D geometric tangent is obtained in

$$\frac{d\Gamma}{ds} = \frac{\partial\Gamma}{\partial s} + \frac{\partial\Gamma}{\partial t} \frac{dt}{ds} \quad (88)$$

The 3D geometric tangent velocity is obtained

$$\frac{d}{dt} \left( \frac{d\Gamma}{ds} \right) = \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial s} + \frac{\partial\Gamma}{\partial t} \frac{dt}{ds} \right) = \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial s} \right) + \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial t} \frac{dt}{ds} \right) = \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial s} \right) + \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial t} \right) \frac{dt}{ds} - \left( \frac{dt}{ds} \right)^2 \frac{\partial\Gamma}{\partial t} \frac{d^2s}{dt^2} \quad (89)$$

and 3D tangent acceleration is given by

$$\begin{aligned} \frac{d}{dt} \left( \frac{d}{dt} \left( \frac{d\Gamma}{ds} \right) \right) &= \frac{d}{dt} \left( \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial s} \right) + \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial t} \right) \frac{dt}{ds} - \left( \frac{dt}{ds} \right)^2 \frac{\partial\Gamma}{\partial t} \frac{d^2s}{dt^2} \right) \\ &= \frac{d^2}{dt^2} \left( \frac{\partial\Gamma}{\partial s} \right) + \frac{d^2}{dt^2} \left( \frac{\partial\Gamma}{\partial t} \right) \frac{dt}{ds} - 2 \frac{d}{dt} \left( \frac{\partial\Gamma}{\partial t} \right) \left( \frac{dt}{ds} \right)^2 \frac{d^2s}{dt^2} - \frac{\partial\Gamma}{\partial t} \left( \frac{dt}{ds} \right)^2 \frac{d^3s}{dt^3} \\ &\quad + \frac{\partial\Gamma}{\partial t} \left( \frac{dt}{ds} \right)^3 \left( \frac{d^2s}{dt^2} \right)^2 \end{aligned} \quad (90)$$

Now computing 2D versions. compute differential element  $d\gamma$

$$d\gamma = \frac{\partial\gamma}{\partial s} ds + \frac{\partial\gamma}{\partial t} dt \quad (91)$$

now compute the 2d point velocity

$$\frac{d\gamma}{dt} = \frac{\partial\gamma}{\partial s} \frac{ds}{dt} + \frac{\partial\gamma}{\partial t} \quad (92)$$

and 2D point acceleration

$$\frac{d^2\gamma}{dt^2} = \frac{d}{dt} \left( \frac{d\gamma}{dt} \right) = \frac{d}{dt} \left( \frac{\partial\gamma}{\partial s} \frac{ds}{dt} + \frac{\partial\gamma}{\partial t} \right) = \frac{d}{dt} \left( \frac{\partial\gamma}{\partial s} \frac{ds}{dt} \right) + \frac{d}{dt} \left( \frac{\partial\gamma}{\partial t} \right) = \frac{d}{dt} \left( \frac{\partial\gamma}{\partial s} \right) \frac{ds}{dt} + \frac{\partial\gamma}{\partial t} \frac{d^2s}{dt^2} + \frac{d}{dt} \left( \frac{\partial\gamma}{\partial t} \right) \quad (93)$$

This can be computed in relation to 3D by assuming  $\gamma = \frac{\Gamma}{\rho}$

$$\frac{d\gamma}{dt} = \frac{1}{\rho} \left( \frac{d\Gamma}{dt} - \frac{1}{\rho} \frac{d\rho}{dt} \Gamma \right) \quad (94)$$

with  $\frac{d\Gamma}{dt}$  described earlier and  $\frac{d\rho}{dt}$

$$\frac{d\rho}{dt} = \frac{\partial\rho}{\partial s} \frac{ds}{dt} + \frac{\partial\rho}{\partial t} \quad (95)$$

and its acceleration is given by

$$\frac{d^2\gamma}{dt^2} = \frac{d}{dt} \left( \frac{1}{\rho} \left( \frac{d\Gamma}{dt} - \frac{1}{\rho} \frac{d\rho}{dt} \Gamma \right) \right) = \frac{1}{\rho} \left[ -\frac{d\rho}{dt} \frac{1}{\rho} \left( \frac{d\Gamma}{dt} - \frac{1}{\rho} \frac{d\rho}{dt} \Gamma \right) + \frac{d^2\Gamma}{dt^2} + \frac{2}{\rho^2} \left( \frac{d\rho}{dt} \right)^2 \Gamma - \frac{1}{\rho} \frac{d^2\rho}{dt^2} \Gamma - \frac{1}{\rho} \frac{d\rho}{dt} \frac{d\Gamma}{dt} \right] \quad (96)$$

With  $\frac{d^2\rho}{dt^2}$  defined as

$$\frac{d^2\rho}{dt^2} = \frac{d}{dt} \left( \frac{\partial\rho}{\partial s} \right) \frac{ds}{dt} + \frac{\partial\rho}{\partial t} \frac{d^2s}{dt^2} + \frac{d}{dt} \left( \frac{\partial\rho}{\partial t} \right) \quad (97)$$

the 2D tangent is written as

$$\frac{d\gamma}{dt} = \frac{\partial\gamma}{\partial s} + \frac{\partial\gamma}{\partial t} \frac{dt}{ds} \quad (98)$$

The 2D tangential velocity is given by

$$\frac{d}{dt} \left( \frac{d\gamma}{ds} \right) = \frac{d}{dt} \left( \frac{\partial\gamma}{\partial s} + \frac{\partial\gamma}{\partial t} \frac{dt}{ds} \right) = \frac{d}{dt} \left( \frac{\partial\gamma}{\partial s} \right) + \frac{d}{dt} \left( \frac{\partial\gamma}{\partial t} \frac{dt}{ds} \right) = \frac{d}{dt} \left( \frac{\partial\gamma}{\partial s} \right) + \frac{d}{dt} \left( \frac{\partial\gamma}{\partial t} \right) \frac{dt}{ds} - \left( \frac{dt}{ds} \right)^2 \frac{\partial\gamma}{\partial t} \frac{d^2s}{dt^2} \quad (99)$$

and 2D tangent acceleration is given by

$$\begin{aligned}
\frac{d}{dt} \left( \frac{d}{dt} \left( \frac{d\gamma}{ds} \right) \right) &= \frac{d}{dt} \left( \frac{d}{dt} \left( \frac{\partial\gamma}{\partial s} \right) + \frac{d}{dt} \left( \frac{\partial\gamma}{\partial t} \right) \frac{dt}{ds} - \left( \frac{dt}{ds} \right)^2 \frac{\partial\gamma}{\partial t} \frac{d^2s}{dt^2} \right) \\
&= \frac{d^2}{dt^2} \left( \frac{\partial\gamma}{\partial s} \right) + \frac{d^2}{dt^2} \left( \frac{\partial\gamma}{\partial t} \right) \frac{dt}{ds} - 2 \frac{d}{dt} \left( \frac{\partial\gamma}{\partial t} \right) \left( \frac{dt}{ds} \right)^2 \frac{d^2s}{dt^2} - \frac{\partial\gamma}{\partial t} \left( \frac{dt}{ds} \right)^2 \frac{d^3s}{dt^3} \\
&\quad + \frac{\partial\gamma}{\partial t} \left( \frac{dt}{ds} \right)^3 \left( \frac{d^2s}{dt^2} \right)^2
\end{aligned} \tag{100}$$

The  $\gamma = \frac{\Gamma}{\rho}$  of the tangent is

$$\frac{d\gamma}{ds} = \frac{1}{\rho} \left( \frac{d\Gamma}{ds} - \frac{1}{\rho} \frac{d\rho}{ds} \Gamma \right) \tag{101}$$

with  $\frac{d\Gamma}{ds}$  described earlier and  $\frac{d\rho}{ds}$

$$\frac{d\rho}{ds} = \frac{\partial\rho}{\partial s} + \frac{\partial\rho}{\partial t} \frac{dt}{ds} \tag{102}$$

Its velocity is computed as

$$\begin{aligned}
\frac{d}{dt} \left( \frac{d\gamma}{ds} \right) &= \frac{d}{dt} \left( \frac{1}{\rho} \left( \frac{d\Gamma}{ds} - \frac{1}{\rho} \frac{d\rho}{ds} \Gamma \right) \right) \\
&= \frac{1}{\rho} \left[ - \frac{1}{\rho} \frac{d\rho}{dt} \left( \frac{d\Gamma}{ds} - \frac{1}{\rho} \frac{d\rho}{ds} \Gamma \right) + \frac{d^2\Gamma}{dt ds} - \frac{d\Gamma}{dt} \left( \frac{d\rho}{ds} \right) \frac{1}{\rho} - \Gamma \left( \frac{d^2\rho}{dt ds} \right) \frac{1}{\rho} \right. \\
&\quad \left. + \Gamma \frac{d\rho}{ds} \frac{1}{\rho^2} \frac{d\rho}{dt} \right]
\end{aligned} \tag{103}$$

And finally the acceleration is given by

$$\begin{aligned}
\frac{d^2}{dt^2} \left( \frac{d\gamma}{ds} \right) &= \frac{d}{dt} \left( \frac{1}{\rho} \left[ - \frac{1}{\rho} \frac{d\rho}{dt} \left( \frac{d\Gamma}{ds} - \frac{1}{\rho} \frac{d\rho}{ds} \Gamma \right) + \frac{d^2\Gamma}{dt ds} - \frac{d\Gamma}{dt} \left( \frac{d\rho}{ds} \right) \frac{1}{\rho} - \Gamma \left( \frac{d^2\rho}{dt ds} \right) \frac{1}{\rho} + \Gamma \frac{d\rho}{ds} \frac{1}{\rho^2} \frac{d\rho}{dt} \right] \right) \\
&= \frac{1}{\rho} \left\{ \frac{1}{\rho} \frac{d\rho}{dt} \left[ \frac{1}{\rho} \frac{d\rho}{dt} \left( \frac{d\Gamma}{ds} - \frac{1}{\rho} \frac{d\rho}{ds} \Gamma \right) - \frac{d^2\Gamma}{dt ds} + \frac{d\Gamma}{dt} \left( \frac{d\rho}{ds} \right) \frac{1}{\rho} + \Gamma \left( \frac{d^2\rho}{dt ds} \right) \frac{1}{\rho} - \Gamma \frac{d\rho}{ds} \frac{1}{\rho^2} \frac{d\rho}{dt} \right] \right. \\
&\quad + \frac{d^3\Gamma}{dt^2 ds} - \left[ \frac{d^2\Gamma}{dt^2} \left( \frac{d\rho}{ds} \right) \frac{1}{\rho} + \frac{d\Gamma}{dt} \left( \frac{d^2\rho}{dt ds} \right) \frac{1}{\rho} - \frac{d\Gamma}{dt} \frac{d\rho}{ds} \frac{d\rho}{dt} \frac{1}{\rho^2} \right] \\
&\quad - \left[ \frac{d\Gamma}{dt} \left( \frac{d^2\rho}{dt ds} \right) \frac{1}{\rho} + \Gamma \left( \frac{d^3\rho}{dt^2 ds} \right) \frac{1}{\rho} - \Gamma \frac{d^2\rho}{dt ds} \frac{d\rho}{dt} \frac{1}{\rho^2} \right] \\
&\quad \left. - \left[ \frac{1}{\rho} \left[ - \frac{1}{\rho} \frac{d\rho}{dt} \left( \frac{d\Gamma}{ds} - \frac{1}{\rho} \frac{d\rho}{ds} \Gamma \right) + \frac{d^2\Gamma}{dt ds} - \frac{d\Gamma}{dt} \left( \frac{d\rho}{ds} \right) \frac{1}{\rho} - \Gamma \left( \frac{d^2\rho}{dt ds} \right) \frac{1}{\rho} + \Gamma \frac{d\rho}{ds} \frac{1}{\rho^2} \frac{d\rho}{dt} \right] \right\}
\end{aligned} \tag{104}$$

## APPENDIX C – Quick overview of Frenet frames

Fabbri and Kimia IJCV2016 provide in page 6 a quick overview in differential geometry. Here we are going to adapt from theirs to ours.

The Frenet frame of a point in a 3D curve is composed by three vectors which are the tangent, the normal and binormal. Assuming the time invariance for simplicity and, from the authors, the local Frenet frame of a 3D point  $\Gamma$  defined by its unit vectors  $\mathbf{T}$ ,  $\mathbf{N}$  and  $\mathbf{B}$  are defined as

$$\mathbf{T} = \frac{1}{G}\Gamma_s, \mathbf{N} = \mathbf{T}^\perp = \frac{1}{G \cdot K}\mathbf{T}_s, \mathbf{B} = \mathbf{T} \times \mathbf{N}, \quad (105)$$

with the tangential parametric “velocity”  $G = \|\Gamma_s\|$  the 3D curvature  $K = \frac{\|\mathbf{T}_s\|}{G}$  and torsion  $\tau = -\frac{\mathbf{B}_s \cdot \mathbf{N}}{G}$ . Since classical books such as [Carmo \(1976\)](#) provide in-depth detail in Frenet frame “dynamics”, we state them here it, in matrix form, as

$$\begin{bmatrix} \mathbf{T}_s \\ \mathbf{N}_s \\ \mathbf{B}_s \end{bmatrix} = \begin{bmatrix} 0 & K & 0 \\ -K & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} \mathbf{T} \\ \mathbf{N} \\ \mathbf{B} \end{bmatrix} \quad (106)$$

The local frame for 2D  $\gamma$  is composed only by its 2D tangent  $\mathbf{t}$  and normal  $\mathbf{n}$

$$\mathbf{t} = \frac{1}{g}\gamma_s, \mathbf{n} = \mathbf{t}^\perp = \frac{1}{g \cdot \kappa}\mathbf{t}_s, g = \|\gamma_s\|, \kappa = \frac{\|\mathbf{t}_s\|}{g}. \quad (107)$$

## APPENDIX D – Lagrangian and Hamiltonian metrics

This work uses notions and definition of manifolds from Frankel (2011). The tangent space at a point  $x$  is defined as the space containing all tangent vectors passing through  $x$ . If a function  $f : \mathbb{R}^n \mapsto \mathbb{R}^m$  is a smooth map where  $y^i = f^i(x)$ ,  $i = 1, \dots, m$ , then a tangent vector  $\mathbf{v}$  of a smooth curve  $x(t)$  in  $\mathbb{R}^n$  at  $x_0 = x(0)$  defined as  $\dot{x}(0) = \mathbf{v} := dx/dt(0)$  has tangent vector  $\mathbf{w}$  at  $y_0$  defined by the chain rule

$$w^i = y^i(0) = \sum_{k=1}^n \frac{\partial y^i}{\partial x^k}(x_0) \mathbf{v}^k. \quad (108)$$

This expression leads to a linear transformation, differential form of  $f$  at  $x_0$  that maps  $\mathbf{v} \mapsto \mathbf{w}$

$$f_* : \mathbb{R}_{x_0}^n \mapsto \mathbb{R}_{y_0}^m, f_*(\mathbf{v}) = \mathbf{w}. \quad (109)$$

Where the matrix that defines this transformation is called *Jacobian matrix* which its elements are  $\partial y^i / \partial x^k(x_0)$ .

The tangent bundle  $\mathbf{T}\mathcal{M}^n$  in a differential manifold  $\mathcal{M}^n$  is defined as the union of all tangent spaces  $\mathbf{T}\mathcal{M}_m^n$  at all points  $m \in \mathcal{M}^n$ . A point in the tangent bundle consists in  $2n$ -tuple  $(m, v)$  where  $m \in \mathcal{M}^n$  (with local coordinate map  $q \in \mathbb{R}^n \rightarrow q(m) = (q_1(m), q_2(m), \dots, q_n(m))^\top$  that can be used applying the chain rule) and  $v$  is the tangent space  $\mathbf{T}\mathcal{M}_m^n$  (with local tangent map defined as  $\mathbf{w} : \mathbf{T}\mathcal{M}_m^n \mapsto \mathbb{R}^n$ ,  $\mathbf{w} = \sum_{k=1}^n (\partial q^i / \partial m^k)|_{m_0} (dm^k / dt)$ ).

The cotangent bundle  $\mathbf{T}^*\mathcal{M}^n$  in a differential manifold  $\mathcal{M}^n$  is defined as the union of all cotangent spaces  $\mathbf{T}^*\mathcal{M}_m^n$  at all points  $m \in \mathcal{M}^n$ . A cotangent space is defined as the dual space of the tangent space  $\mathbf{T}\mathcal{M}^n$ , in other words, contains all linear functions in  $\mathbf{T}\mathcal{M}^n$ . Let a function  $g : \mathcal{M}^n \mapsto \mathbb{R}$  with differential of  $f$  at  $m$ , defined as  $df : \mathcal{M}_m^n \mapsto \mathbb{R}$ ,  $df(\mathbf{v}) = \mathbf{v}_m(f)$ . Rewriting to local coordinates,  $\mathbf{e}_j = \partial / \partial q_j|_m$  defining a basis for  $\mathcal{M}_m^n$  with

$$df \left( \sum v_j \frac{\partial}{\partial q_j} \right) = \sum v_j(m) \frac{\partial f}{\partial q_j}(m). \quad (110)$$

If  $f$  is a function of the local mapping of the components of  $\mathbf{v}$ , the differential of the  $i$ -th component is given by

$$dq_i \left( \frac{\partial}{\partial q_j} \right) = \frac{\partial q_i}{\partial q_j} = \delta_{i,j}, \quad (111)$$

with  $\delta_{i,j}$  being the Kronecker delta. Rewriting (112), is obtained

$$dq_i \left( \sum v_j \frac{\partial}{\partial q_j} \right) = \sum v_j dq_i \left( \frac{\partial}{\partial q_j} \right) = v_j. \quad (112)$$

Consequently,  $dq_i$ ,  $i = 1, \dots, n$  represents the linear functionals that forms the basis of the cotangent space  $\mathcal{M}_m^{n*}$ .

The Lagrangian function, in terms of classical mechanics, is defined as a scalar function in a tangent bundle in configuration space with local generalized coordinates  $q$  and local generalized velocities  $\dot{q} = dq/dt$ ,

$$L : \mathbf{T}\mathcal{M}^n \rightarrow \mathbb{R} \quad (113)$$

$$(q, \dot{q}) \mapsto L(q, \dot{q}) = K(q, \dot{q}) - U(q),$$

where  $K(q, \dot{q})$  is the kinetic energy and  $U(q)$  is the potential energy. Typically the potential energy is considered independent of  $\dot{q}$  and the kinetic energy is oftenly a positive definite symmetric quadratic form of the velocities

$$K(q, \dot{q}) = \frac{1}{2} \sum_{ij} g_{ij}(q) \dot{q}^i \dot{q}^j, \quad (114)$$

with  $g_{ij}(q)$  defining a "mass matrix" according to Frankel (2011). The construction of the kinetic energy constitutes a Finsler metric since the Lagrangian function is positive definite, is homogeous of first degree in  $\dot{q}$  (a proof can be performed by applying the Lagrangian in Euler-Lagrange equation) and has the property of triangular inequality at  $\dot{q}$  ( $L(q, \dot{q}_1 + \dot{q}_2) \leq L(q, \dot{q}_1) + L(q, \dot{q}_2)$ ). The generalized momenta is defined as,

$$p : \mathbf{T}\mathcal{M}^n \rightarrow \mathbf{T}^*\mathcal{M}^n \quad (115)$$

$$(q, \dot{q}) \mapsto p(q, \dot{q}) = \frac{\partial L}{\partial \dot{q}_i} = \sum_{ij} g_{ij}(q) \dot{q}^j.$$

This work finally states the reasons why the Hamiltonian function is part of the symplectic geometry from the Differential equations course at Msc. program. They are the following:

- It has a closed and non-degenerate 2-form  $\omega(X, Y)$  where  $X, Y \in \mathbf{T}\mathcal{M}$  (this means that  $d\omega = 0$  and  $\forall m \in \mathcal{M}, v \in \mathbf{T}\mathcal{M} \rightarrow \exists w \in \mathbf{T}\mathcal{M}, \omega(v, w) \neq 0$ )
- $Dim(\mathcal{M}) = 2n$

**APPENDIX E** – Hamiltonian form of axial geometric optics

Beginning with the axial geometric Lagrangian function

$$L(\mathbf{q}, \mathbf{q}_z) = \eta(\mathbf{q}, z) \sqrt{|\mathbf{q}_z|^2 + 1}. \quad (116)$$

Computing  $p = \partial L / \partial \dot{q}$ , results in

$$p = \eta(\mathbf{q}, z) \frac{\mathbf{q}_z}{\sqrt{|\mathbf{q}_z|^2 + 1}}. \quad (117)$$

Isolating  $\dot{q}$ , results in

$$\begin{aligned} p^2 &= \eta^2(\mathbf{q}, z) \frac{\mathbf{q}_z^2}{|\mathbf{q}_z|^2 + 1} \rightarrow \\ p^2(|\mathbf{q}_z|^2 + 1) &= \eta^2(\mathbf{q}, z) \mathbf{q}_z^2 \rightarrow \\ \mathbf{q}_z^2(\eta^2(\mathbf{q}, z) - |p|^2) &= p^2 \rightarrow \\ \mathbf{q}_z &= \frac{p}{\sqrt{\eta^2(\mathbf{q}, z) - |p|^2}}. \end{aligned} \quad (118)$$

Computing the Lagrangian function with new variable  $p$

$$L(\mathbf{q}, \mathbf{q}_z) = \eta(\mathbf{q}, z) \sqrt{\frac{p^2}{\eta^2(\mathbf{q}, z) - |p|^2} + 1} = \eta(\mathbf{q}, z) \sqrt{\frac{(p^2 + \eta^2(\mathbf{q}, z) - |p|^2)}{\eta^2(\mathbf{q}, z) - |p|^2}} = \frac{\eta^2(\mathbf{q}, z)}{\sqrt{\eta^2(\mathbf{q}, z) - |p|^2}}. \quad (119)$$

Applying this result in the Legendre transformation  $H = p \cdot \dot{q} - L$ , results in

$$H = p \cdot \dot{q} - L = \frac{p^2}{\sqrt{\eta^2(\mathbf{q}, z) - |p|^2}} - \frac{\eta^2(\mathbf{q}, z)}{\sqrt{\eta^2(\mathbf{q}, z) - |p|^2}} = -\frac{\eta^2(\mathbf{q}, z) - |p|^2}{\sqrt{\eta^2(\mathbf{q}, z) - |p|^2}} = -\sqrt{\eta^2(\mathbf{q}, z) - |p|^2} \quad (120)$$