POLITECNICO DI TORINO

Master's Degree in Mathematical Engineering

Master's Degree Thesis

Computational Algebraic Topology in Computer Vision

Supervisors

Prof. FRANCESCO VACCARINO

Prof. CARLO MASONE

Candidate

GIOVANNI BARBARANI

November 2024

Summary

This thesis explores the application of algebraic topology to computer vision, specifically focusing on the challenge of scale-independent keypoint detection within image matching. Across three chapters, this work builds a theoretical and practical foundation, presenting a novel deep learning approach that leverages topological principles for keypoint detection.

The first chapter establishes essential background in algebraic topology, covering concepts such as CW-complexes, homology, and persistent homology, which allow for topological invariants to be computed via discrete structures. These concepts are particularly relevant to computer vision as they can model the structure of digital images represented by pixel grids, or cubical complexes. To provide practical tools for our work, the chapter also covers Morse theory and discrete Morse theory, which connect local extrema to topological features, and allow efficient computation of topological invariants. This theoretical framework lays the groundwork for keypoint detection methods that can generalize across different scales.

The second chapter delves into computer vision fundamentals, focusing on image matching and the detection of reproducible keypoints across different views of the same scene. We start with the basics of the pinhole camera model, projective geometry, and homographies, which are essential for understanding how images relate spatially. The chapter then introduces the current techniques of image matching and the keypoints-and-descriptors paradigm, which includes traditional methods such as SIFT and deep learning approaches like R2D2. The goal is to provide a comprehensive overview of the current methods and the limitations they face, particularly in achieving scale invariance—a gap this thesis aims to address.

The third chapter presents the main contribution of this thesis: a topology-based framework for keypoint detection. The proposed method is built on persistent homology and discrete Morse theory, employing these topological tools to detect keypoints as local maxima in a scale-agnostic manner. The approach includes a novel deep learning loss function that integrates topological invariants to ensure robust keypoint detection across different transformations. Empirical results are provided to validate the consistency and performance of our techniques.

In conclusion, this thesis bridges algebraic topology and computer vision, contributing a framework that addresses the challenge of scale-invariant keypoint detection. This work paves the way for further exploration of topology-based methods in computer vision applications, offering a promising direction for future research.

Acknowledgements

A few years ago, I decided to quit a job as a data science consultant to pursue a career in research. I fell in love with mathematics; I often found myself pretending to take long bathroom breaks just to do a few more exercises from Abstract Algebra by Dummit and Foote rather than from Munkres' Topology. At the time, I had a romantic view of mathematics, thinking there was something inherently divine in it. Now, I'm not sure if it's simply a language to communicate with mathematicians, not so different from any other language. But the love never faded, and there are some chats that are truly worth it.

I want to thank my coauthors and supervisors, Dr. Carlo Masone, for giving me the chance to work on interesting problems and for his patience, and Professor Francesco Vaccarino, whose mastery of the subject and knowledge are matched only by his ability to mentor and teach. I also want to thank all the other members of the teaching staff at the Polytechnic of Turin. I have never encountered another environment where people are so competent and committed to their mission, providing students with the best tools and opportunities possible.

I have learned much along this journey, for which I want to thank Ludmila and another person, who I'm not sure would like to be mentioned here. I will never be the same. Wherever you are, I hope you're doing well. My friend Lorenzo, without you I wouldn't be here nor anywhere else. Michele, I learned a lot from you; I hope I reciprocated at least half of it. Giuseppe, I'm sorry if I haven't been there all these years. I want to thank my family: my mother, Antonella, whom I could have loved more; my father, Vito, I am sure you will prove the Riemann Hypothesis one day, keep trying! And especially, my wise and kind sister Ginevra—may God let me die on the same day as you. I cannot imagine a world without you, and I want to be there for you until the end.

"You've almost convinced me I'm real."
Touch, Daft Punk.

Table of Contents

List of Tables

List of Figures

Chapter 1

Algebraic Topology

In this chapter, we provide the algebraic topology background necessary for developing our techniques and applications. The main focus is on cellular complexes, or CW-complexes, introduced in Section 2. Loosely speaking, these can be viewed as topological spaces constructed by assembling regular building blocks, such as a 3D shape composed of vertices, edges, and polygonal faces. Cellular complexes play a critical role in bridging continuous topological questions with discrete structures suitable for computation. A classic example is the Euler characteristic, which, in its simplest form, states that the number of vertices minus edges plus faces for any convex polyhedron equals 2. This characteristic is related to the topology of the sphere and applies to any complex homotopic to the sphere.

More generally, many topological invariants of a space can be captured and computed by means of a discrete triangulation. Particularly relevant to our work are homology modules and persistent homology modules, which are discussed in Sections 1.2 and 1.3, respectively. Cellular complexes are also inherently interesting because many problems are naturally formulated in terms of discrete structures. For example, in computer vision, a digital image captured by a sensor can be represented as a two-dimensional array—a grid of pixel values—that can be modeled as a cubical complex. For an in-depth treatment of homology theory and persistent homology, we refer to *(Ghrist 2014)* and *(Edelsbrunner and Harer 2022).*

Finally, in Sections 1.4 and 1.5, we cover Morse theory and discrete Morse theory, which relate these topological invariants to the local extrema of a given function. These theories provide both computational tools and algorithms for computing the invariants, as well as a theoretical connection between local extrema and topology that finds applications in our computer vision work. For further reading on these topics, we refer to *(Milnor 1963)* and *(Forman 2002).* Our discussion aims to provide a sufficient and practical understanding of all relevant details for our applications, even without prior knowledge of the subjects. However, some basic notions of topology are required, which are reviewed in Appendix A.

1.1 Complexes

In general, a CW-complex (or cellular complex) is a space that can be decomposed into simple pieces, each homeomorphic to a ball $\mathcal{B}^m$. We begin this section by introducing the notion of CW-complex and regular CW-complex, following an inductive construction as in *(Hatcher 2005).* However, we avoid the treatment of abstract complexes, and directly consider the geometric realization of a complex $\mathcal{K}$ embedded in Euclidean space $\mathbb{R}^N$, to simplify the discussion and notation.

We will soon specialize in two specific types of complexes: simplicial complexes and cubical complexes. The former has numerous applications in the literature, even if it plays a marginal role in our work; nonetheless, we believe it is useful to present it to the reader. Cubical complexes, on the other hand, are primarily used in computer vision and are central to our development.

Definition 1.1.1. *(n-cell) An n-cell of dimension $n \geq 1$ is a topological space homeomorphic to the open ball $\mathcal{B}^n$. For $n = 0$, a 0-cell is a point. Given an n-cell σ, we denote $\texttt{dim}\,\sigma = n$ as a function that returns the cell's dimension.*

We usually indicate a cell with a Greek letter σ or σ^n to highlight its dimension; however, to simplify proofs, we often identify a cell or its boundary with a corresponding homeomorphic space, e.g., $\mathcal{B}^n$, omitting the underlying chain of morphisms where necessary.

Definition 1.1.2. *(CW-complex) A compact set $\mathcal{K} \subseteq \mathbb{R}^N$ is a (finite) CW-complex if it belongs to a chain of sets $\mathcal{K}^0 \subseteq \mathcal{K}^1 \subseteq \cdots \subseteq \mathcal{K}^n$ constructed from a finite collection of cells $\mathcal{C}_\mathcal{K} = \{\sigma_i\}$ in the following way:*

1. *$\mathcal{K}_0$ is a collection of points in bijection with the 0-cells in $\mathcal{C}_\mathcal{K}$.*
2. *For every m-cell σ_i, identified for convenience with $\mathcal{B}^m$, there is a continuous function from the closure of the cell $\varphi_i : \overline{\mathcal{B}^m} \to \mathbb{R}^N$, called the attaching map, such that: (i) when restricted to the cell boundary, the image $\varphi_i(\mathcal{S}^{m-1})$ is included in $\mathcal{K}^{m-1}$, and for every k-cell σ_j of lower dimension, $k < m$, if $\varphi_j(\mathcal{B}^k) \cap \varphi_i(\mathcal{S}^{m-1}) \neq \varnothing$ then $\varphi_j(\mathcal{B}^k) \subseteq \varphi_i(\mathcal{S}^{m-1})$. (ii) When restricted to the interior, $\varphi_i : \mathcal{B}^m \to \mathbb{R}^N$ is an embedding, and $\varphi_i(\mathcal{B}^m) \cap \mathcal{K}^{m-1} = \varnothing$. (iii) For every other m-cell σ_j, the images of their interiors do not intersect, $\varphi_i(\mathcal{B}^m) \cap \varphi_j(\mathcal{B}^m) = \varnothing$. We define $\mathcal{K}_m = \mathcal{K}_{m-1} \bigcup_i \varphi_i(\mathcal{B}^m)$, where i ranges over all m-cells.*
3. *If n is the highest dimension of a cell in $\mathcal{C}_\mathcal{K}$ and $\mathcal{K} = \mathcal{K}^n$, then $\mathcal{K}$ is a CW-complex of dimension n, and $\mathcal{C}_\mathcal{K}$ represents its cells.*

Definition 1.1.3. *(Subcomplex) A subcomplex $\mathcal{K}'$ of complex $\mathcal{K}$ is a set $\mathcal{K}' \subseteq \mathcal{K}$ that is a complex constructed from a subset of cells $\mathcal{C}_{\mathcal{K}'} \subseteq \mathcal{C}_\mathcal{K}$.*

Example 1.1.1. *(Skeleton) For every m, the set $\mathcal{K}^m$ from Definition 1.1.2 is a subcomplex of $\mathcal{K}$. In particular, $\mathcal{K}^m$ is called the m-skeleton of $\mathcal{K}$ and is the subcomplex constructed from all cells of dimension at most m.*

Definition 1.1.4. *(Regular CW-complex) A regular CW-complex is a complex $\mathcal{K}$ for which every attaching map in Definition 1.1.2 is an embedding when restricted to the cell boundary. Notice that, in this case, each attaching map φ_i is an embedding of the entire domain, thus defining a homeomorphism of $\overline{\mathcal{B}^m}$ into the image $\varphi_i(\overline{\mathcal{B}^m})$.*

Theorem 1.1.5. *For $n \geq 1$, in a regular CW-complex, the boundary $\varphi_i(\mathcal{S}^{n-1})$ of an n-cell is the closure of the union of a collection of at least two $(n-1)$-cells $\bigcup_j \varphi_j(\mathcal{B}^{n-1})$ (or equivalently the union of the closures).*

Proof: For $n = 1$, the boundary of a 1-cell is a pair of points, thus 0-cells that must necessarily be different in a regular complex. Let n be greater than 1. From the definition of the attaching map, it is clear that $\varphi_i(\mathcal{S}^{n-1})$ can be expressed as a disjoint union of cells in $\mathcal{K}^{n-1}$. The set of all the $(n-1)$-dimensional cells contained in its image $A = \bigcup_j \varphi_j(\mathcal{B}^{n-1})$ is (relatively) open in $\mathcal{K}^{n-1}$. To see this, consider that an $(n-1)$-cell does not appear in the boundary of any cell in $\mathcal{K}^{n-1}$, so the complement of A is a necessarily closed subcomplex. If A is not dense in $\varphi_i(\mathcal{S}^{n-1})$, then the complement of its closure $B = \varphi_i(\mathcal{S}^{n-1}) - \overline{A}$ contains a non-empty open set U of $\mathcal{K}^{n-1}$. B can be written as a disjoint union of cells that are in $\varphi_i(\mathcal{S}^{n-1})$ but not in $\overline{A}$, explicitly as $B = \bigcup_k \varphi_k(\mathcal{B}^{m_k})$ with $m_k \leq n-2$ for every k. There must be at least one k for which $\varphi_k^{-1}(U)$ is a non-empty open set (if not, U would be empty). On this set, $\varphi_i^{-1} \circ \varphi_k$ is a homeomorphism from an open set of a ball $\mathcal{B}^{m_k}$ and an open set of a sphere $\mathcal{S}^{n-1}$, with $m_k \leq n-2$, which is not possible (see "invariance of domain" in Appendix A). We conclude that A must be dense. Thus, $\varphi_i(\mathcal{S}^{n-1})$ must equal the closure of at least the image of an $(n-1)$-cell, but if it contains only one, then $\varphi_i(\mathcal{S}^{n-1}) = \overline{\varphi_j(\mathcal{B}^{n-1})} = \varphi_j(\overline{\mathcal{B}^{n-1}})$ and $\varphi_i^{-1} \circ \varphi_j$ is a homeomorphism, which is impossible for similar reasons. □

Definition 1.1.6. *(Faces and cofaces) From the previous theorem, the boundary of every n-cell σ_i of a regular complex is the closure of a subset of $(n-1)$-cells. For every $(n-1)$-cell σ_j in the boundary of σ_i (i.e., $\varphi_j(\mathcal{B}^{n-1}) \subseteq \varphi_i(\mathcal{S}^{n-1})$), we say that σ_j is a face of σ_i, denoted $\sigma_j < \sigma_i$, and conversely, we say that σ_i is a coface of σ_j, denoted $\sigma_i > \sigma_j$.*

Definition 1.1.7. *(Maximal cell) A maximal cell of a regular complex $\mathcal{K}$ is a cell that has no cofaces.*

Definition 1.1.8. *(Free face) A free face of a regular complex $\mathcal{K}$ is a face of a maximal cell that has no other cofaces.*

Example 1.1.2. *Figure 1.1 depicts a CW-complex decomposition of the sphere, composed of points, edges between points, and areas bounded by these edges, which are, respectively, the images of the underlying* 0*-cells,* 1*-cells, and* 2*-cells. A similar decomposition can be obtained by projecting the faces of an inscribed convex polyhedron.*

Now we introduce two particularly simple cases of regular complexes: simplicial complexes and cubical complexes. The former are the natural choice for triangulating continuous spaces, while the latter are commonly used in computer vision, as they naturally model the arrangement of pixels on a grid in digital images. An example of a simplicial complex is shown in Figure 1.2, and an example of a cubical complex is shown in Figure 1.3.

Definition 1.1.9. *(Simplicial complex) A simplex* $\Delta^k \subseteq \mathbb{R}^N$ *of dimension* $k \leq N$ *is the convex hull of* $k+1$ *affinely independent points* $\{x_i\}_{i=1}^{k+1}$*, denoted* $\Delta^k = [x_1, \ldots, x_{k+1}]$*. Examples include a point, segment, triangle, or tetrahedron for k from* 0 *to* 3*. A simplicial complex* $\mathcal{K}$ *is a regular complex such that: (1) every k-cell* σ *is the interior of a k-simplex* Δ^k*; (2) if* $[x_1, \ldots, x_{k+1}]$ *represents a k-cell, then also all the* $(k-1)$*-simplices*

$$[x_1, \ldots, x_{-i}, \ldots, x_{k+1}]$$

obtained by removing a vertex x_i*, are* $(k-1)$*-cells.*

Definition 1.1.10. *(Cubical complex) A cube is a set* $Q \subset \mathbb{R}^N$ *for which there is a set of integers* $I_Q := \{l_1, \ldots, l_N\}$ *such that* $Q = I_1 \times \cdots \times I_N$*, where each*

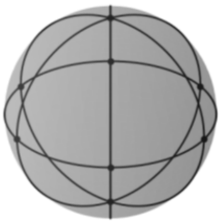


Figure 1.1: CW-complex.

$I_j = [l_j, l_j + 1]$ or $I_j = [l_j, l_j]$. When $I = [l, l]$, it is called degenerate, and the number of non-degenerate intervals in Q is its dimension. A cubical complex is a regular complex $\mathcal{K}$ where: (1) every k-cell is a cube of dimension k; (2) if $I_1 \times \cdots \times I_N$ represents a cell and $I_i = [l_i, l_i + 1]$ is a non-degenerate interval of the product, then also

$$I_1 \times \ldots [l_i, l_i] \times \ldots I_N \quad \textit{and} \quad I_1 \times \ldots [l_i + 1, l_i + 1] \cdots \times I_N$$

, the cubes where I_i has been substituted with a degenerate interval, both represent cells.

The following property will allow us to follow the simplified approach to homology theory of *(Edelsbrunner and Harer 2022)* in the next sections. We will prove that it holds for simplicial and cubical complexes so that the topic can be treated in an agnostic way.

Definition 1.1.11. *(Mod 2 boundary) We say that the regular complex $\mathcal{K}$ has the mod 2 boundary property if the fact that a $(n-2)$-cell υ is contained in the boundary of an n-cell σ, implies that it is contained in the boundary of exactly two faces of σ.*

Theorem 1.1.12. *Every simplicial complex $\mathcal{K}$ has the mod 2 boundary property.*

Proof: Given a n-cell $[x_1, \ldots, x_{n+1}]$, every $(n-2)$-cell of its boundary is of the form $[x_1, \ldots, x_{-i}, \ldots, x_{-j} \ldots, x_{k+1}]$ and is contained in exactly

$$[x_1, \ldots, x_{-i}, \ldots, x_{k+1}] \quad \text{and} \quad [x_1, \ldots, x_{-j}, \ldots, x_{k+1}].$$

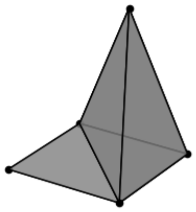


Figure 1.2: Simplicial complex.

□

Theorem 1.1.13. *Every cubical complex $\mathcal{K}$ has the mod 2 boundary property.*

Proof: Given an n-cell that is a cube, a $(n-2)$-cell contained in its boundary is a cube obtained by degenerating two intervals I_i and I_j. Thus, it is contained in both the $(n-1)$-faces with only I_i or only I_j degenerated (at the same extreme as the $(n-2)$-cell), and it is not contained in any other face. □

1.2 Homology

Homology theory associates a topological space with an algebraic invariant, specifically by assigning to the space a sequence of modules whose dimensions correspond to the number of connected components, holes, voids, and higher-dimensional volumes enclosed by the space. This is an invariant in the precise sense that if two spaces are homotopic, they are associated with isomorphic algebraic structures, capturing the same topological information. In this work, we address the topic in the context of regular complexes, which is sufficient for our purposes, though homology theory can be extended to topological manifolds in general. We follow the approach of *(Edelsbrunner and Harer 2022)* based on $\mathbb{F}_2$-homology, which provides a particularly simple treatment of the subject and suits the applications. Consult *(Hatcher 2005)* for an alternative approach to homology theory.

Although the algebraic approach to the problem may seem unnecessary at first, we emphasize that it will prove valuable by allowing us to translate qualitative

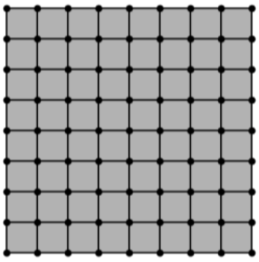


Figure 1.3: Cubical complex.

problems (e.g., how many holes does a space have?) into questions about vector spaces and linear maps, which, in turn, provide quantitative answers through simple computations. Appendix B covers basic results of abstract algebra and modules.

Definition 1.2.1. *(k-chains) The k-chains C_k is the free $\mathbb{F}_2$-module generated by the set of k-cells in a given regular complex $\mathcal{K}$. Thus, C_k consists of all possible formal sums of k-cells with coefficients in $\mathbb{F}_2$, and an element of C_k has the form $\sum_{i=1}^{m} a_i\sigma_i$, where $\mathtt{dim}\ \sigma_i = k$ and $a_i \in \mathbb{F}_2$ for each $i = 0, \ldots, m$. The k-cells of $\mathcal{K}$ form a basis for C_k, meaning the number of k-cells is equal to the dimension of the module.*

Definition 1.2.2. *(Boundary operator) For every $k > 0$, we define the boundary operator ∂_k as the linear operator that maps each k-cell in C_k to the formal sum of its faces in C_{k-1}. Since the k-cells form a basis for C_k, this definition extends linearly to the entire space without ambiguity. The boundary operator ∂_0 is defined trivially on C_0 as the map that sends every element to 0. If the underlying complex $\mathcal{K}$ has dimension n, we also define a trivial linear operator $\partial_{n+1} : \{0\} \to C_n$.*

To summarize, the k-chains form a sequence of modules connected by a sequence of linear operators:

$$\{0\} \xrightarrow{\partial_{n+1}} C_n \xrightarrow{\partial_n} C_{n-1} \xrightarrow{\partial_{n-1}} \ldots C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} \{0\}$$

This collection of algebraic structures is called the chain complex and reflects the topological relations of the underlying complex.

The following result shows that the boundary of a boundary is zero, which is the key property that leads to the definition of homology.

Theorem 1.2.3. *If the underlying complex has the property in Definition 1.1.11, then the following holds (in particular, it holds for simplicial and cubical complexes): the composition of two boundary operators results in the trivial map, which sends every element to 0, i.e., $\partial_k \circ \partial_{k+1} = 0$, for every k.*

Proof: Since $\partial_k \circ \partial_{k+1}$ is a linear map, it suffices to show that $\partial_k(\partial_{k+1}(\sigma)) = 0$ for every $(k+1)$-cell σ in $\mathcal{K}$. This follows directly from the property in Definition 1.1.11: each face of a face of σ appears exactly twice in the resulting formal sum of $(k-1)$-cells. Because we are working modulo 2, these repeated faces cancel out, resulting in 0. □

Definition 1.2.4. *(Boundaries and cycles) The image of the boundary operator ∂_{k+1} is a submodule of C_k, referred to as the space of k-boundaries and denoted by B_k. The kernel of the boundary operator ∂_k is a submodule of C_k, called the space of k-cycles, denoted by Z_k. Notice that the previous theorem establishes that B_k is a submodule of Z_k, i.e., $B_k \subseteq Z_k \subseteq C_k$.*

Definition 1.2.5. *(Homology modules) The k-homology module of a given complex is defined as the quotient module $H_k = Z_k/B_k$. Two k-cycles $a, c \in Z_k$ are said to be homologous if they differ by a k-boundary, i.e., there exists $b \in B_k$ such that $a = c + b$. The elements of H_k are precisely the equivalence classes of homologous cycles.*

Definition 1.2.6. *(Betti numbers) The k-th Betti number β_k is the dimension of the k-homology module: $\beta_k = \texttt{dim}\ H_k = \texttt{dim}\ Z_k - \texttt{dim}\ B_k$.*

On a two-dimensional surface, the presence of a hole is directly related to the existence of a loop that cannot be contracted to a point—formally, a closed curve that is not homotopic to a point. Furthermore, around the same hole, there are many different loops that can be deformed into one another. Thus, what matters when counting holes is how many distinct non-homotopic loops exist. This idea is captured by the 1-cycles and homologous 1-cycles in C_1. Moreover, around two holes in the space, we can draw three distinct loops, including one that circumnavigates both holes, clearly requiring an additional condition. What we are truly looking for is the number of linearly independent, non-homologous 1-cycles. This reasoning generalizes to higher dimensions. Indeed, the Betti number β_k, which counts the number of non-homologous k-cycles, tells us how many distinct k-dimensional volumes are enclosed by the complex.

Example 1.2.1. *Consider the cubical complexes represented in Figure 1.4, labeled a, b, c, and d from left to right. The first two-dimensional complex, a, corresponds*

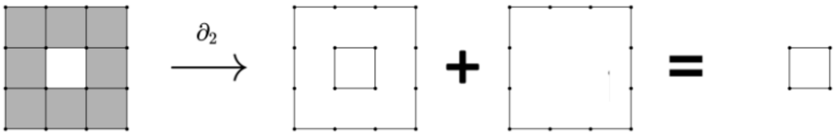


Figure 1.4: Homologous cycles.

to the entire space and can be viewed as representing a 2*-chain, which is the sum of all the* 2*-cells. The* 1*-complex b represents the* 1*-boundary $\partial_2(a)$, depicted by removing all the* 2*-cells along with all the* 1*-cells that are faces of two cells. The complexes c and d are homologous* 1*-cycles, as $d = b + c$. This sum can be visualized by drawing all the cells of b and c, then removing the ones that appear in both. Indeed, the entire complex is homotopic to an annulus with one hole, and we cannot find two non-homologous* 1*-cycles.*

1.3 Persistent Homology

Persistent homology extends homology theory by adding a temporal dimension, allowing us to study how topological features such as connected components and holes are created, persist, and eventually disappear across a sequence of topologies. Persistent homology has found significant applications in data analysis, where it is used to relate a discrete sample (a point cloud) to the topology of the manifold where the data lies. In these applications, the temporal dimension is a clever modeling artifice, and persistent homology is often employed to model noise, for instance by considering features with a short lifespan as noise, and to infer the homology of the underlying target space.

In Morse theory and discrete Morse theory, which will be discussed in the following sections, the temporal dimension arises naturally and provides a useful tool for iteratively computing the homology of a space from a sequence of subspaces. Moreover, in computer vision, persistent homology itself is of interest, underlying many watershed algorithms, and will be exploited by our methods.

Definition 1.3.1. *(Filtrations) A real-valued function defined on $\mathcal{C}_\mathcal{K}$, the set of cells in a complex $\mathcal{K}$, is called a filtration if it is non-decreasing along faces; that is, if $\tau < \sigma$ then $f(\tau) \leq f(\sigma)$. The sublevel sets of a filtration f form subcomplexes of $\mathcal{K}$. We denote by $\mathcal{K}_t = f^{-1}(-\infty, t]$ the subcomplex $\mathcal{K}_t \subseteq \mathcal{K}$, constructed from all cells σ_i for which $f(\sigma_i) \leq t$.*

Definition 1.3.2. *(Induced Morphism) If $t_1 \leq t_2$, then $\mathcal{K}_{t_1}$ is a subcomplex of $\mathcal{K}_{t_2}$. The inclusion of the cells of $\mathcal{K}_{t_1}$ into those of $\mathcal{K}_{t_2}$ defines a linear map between their k-chains, $i : C_{k,t_1} \to C_{k,t_2}$, corresponding to the inclusion of a submodule into a larger space. Since every cycle (and boundary) in C_{k,t_1} remains a cycle (or boundary, respectively) in C_{k,t_2}, the inclusion map induces a well-defined linear map between classes of homologous cycles. Consequently, it induces a linear map between their homology modules, $i^* : H_{k,t_1} \to H_{k,t_2}$, which we call the induced morphism.*

Given a filtration function and a set of times $\{t_i\}_{i=0}^m$, we obtain a sequence of homology modules for the respective sublevel set complexes, connected by a sequence of induced morphisms:

$$H_{k,t_0} \longrightarrow H_{k,t_1} \longrightarrow \ldots \longrightarrow H_{k,t_{m-1}} \longrightarrow H_{k,t_m}$$

Throughout this sequence, new homology classes may appear (with no preimage), disappear (mapped to zero), merge (mapped to a common image), or remain unchanged. The following example illustrates some of these behaviors.

Example 1.3.1. *Figure 1.5 shows two examples based on simplicial complexes. Each example includes a sequence of three filtered complexes. In each complex, a basis for H_{1,t_i} is depicted by highlighting their representative 1-cycles in different colors. The diagrams above illustrate how the induced morphism acts on the respective elements of the basis.*

In the first example, there is initially a single homology class, represented by a red 1-cycle. At the second step, a new cycle, which is not homologous to the red one, appears, indicated in blue; thus, the dimension of the homology module increases. Moving to the third step, the red cycle becomes a boundary, meaning it is mapped to zero, while the blue cycle persists, resulting in a decrease in the dimension of the homology module.

In the second example, we also begin with a single homology class, and a new one appears in the next step. As in the first example, at the third step, the dimension of the homology module decreases; however, in this case, the two cycles become homologous, meaning they are mapped to the same homology class.

The sequence of modules and morphisms can be defined for continuous time, but since the filtration function assume only a finite set of values, changes in

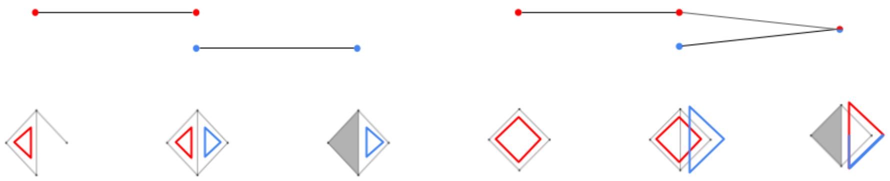


Figure 1.5: Induced morphism.

homology correspond to a finite set of critical times. The following results explain the structure of this sequence.

Definition 1.3.3. *(Persistent homology module) Given a filtration on a regular complex, the related k-th persistent homology module* $\mathbb{H}_k$ *is defined, for every* k*, as the collection of homology modules* $\mathbb{H}_k = \{H_{k,t}\}_{t\in\mathbb{R}}$*, related by the collection of induced morphisms* $\{i^*_{k,(t_1,t_2)}\}_{t_1\leq t_2}$*, which has the following composition property: if* $t_1 \leq t_2 \leq t_3$ *then* $i^*_{k,(t_2,t_3)} \circ i^*_{k,(t_1,t_2)} = i^*_{k,(t_1,t_3)}$*.*

Theorem 1.3.4. *An interval* $\mathbb{F}_2$*-modules* $\mathbb{I}_{[b,d)}$*, with* $0 \leq b < d$ *and* d *possibly equal to* $+\infty$*, is a collection of one-dimensional modules* $\mathbb{I}_{[b,d)} = \{F_t\}_{t\in\mathbb{R}}$*, where* $F_t = \mathbb{F}_2$ *for every* $b \leq t < d$ *and* $F_t = \{0\}$ *otherwise, together with the collection of morphisms* $\{h_{(t_1,t_2)}\}_{t_1\leq t_2}$ *that are: the identity* $h_{t_1,t_2} = \mathtt{id}$ *for* $b \leq t_1 \leq t_2 < d$ *or the (only possible) trivial linear map otherwise.*

Every persistent homology module $\mathbb{H}_k$ *is isomorphic to the direct sum of simple interval modules. Moreover, this decomposition is unique up to a permutation of the summands:*

$$\mathbb{H}_k \cong \bigoplus \mathbb{I}_{[b_i,d_i)}.$$

This sum is element-wise, meaning

$$\bigoplus \mathbb{I}_{[b_i,d_i)} = \{F_{0,t} \times \cdots \times F_{m,t}\}_{t\in\mathbb{R}}$$

and $h_{(t_1,t_2)} = h_{0,(t_1,t_2)} \times \cdots \times h_{m,(t_1,t_2)}$*. The isomorphism means that we can find a collection of invertible linear maps* $\{A_t\}_{t\in\mathbb{R}}$ *such that the following diagram commutes for every choice of times:*

$$\begin{tikzcd}
H_{k,t_1} \arrow[r] \arrow[d, "A_{t_1}"'] & H_{k,t_2} \arrow[r] \arrow[d, "A_{t_2}"'] & \dots \arrow[r] & H_{k,t_n} \arrow[d, "A_{t_n}"'] \\
F_{0,t_1} \times \cdots \times F_{m,t_1} \arrow[r] & F_{0,t_2} \times \cdots \times F_{m,t_2} \arrow[r] & \dots \arrow[r] & F_{0,t_n} \times \cdots \times F_{m,t_n}
\end{tikzcd}$$

meaning that starting from a specific element we find the same image regardless of which sequence of morphisms we use to reach the codomain.

Proof: This fundamental result in persistent homology was originally proved using graph representation theory in *(Gabriel 1972),* with an English overview provided in *(Derksen and Weyman 2005).* An adapted proof for persistent homology is available in *(Botnan and Crawley-Boevey 2020).* □

Definition 1.3.5. *(Barcode) We denote by* $\overline{\mathbb{R}}_+ = [0, +\infty]$ *the set of positive real numbers extended to include infinity. The barcode of a persistent homology module,* $\mathtt{Bar}(\mathbb{H}_k) \subseteq \overline{\mathbb{R}}^2_+ \times \mathbb{N}$*, is the collection of time intervals corresponding to the interval modules present in the decomposition, counted with their multiplicity in case two intervals are equal, uniquely characterized by the previous theorem.*

Definition 1.3.6. *(Persistence) The persistence of a bar $e = (b, d)$, an element of the barcode $e \in \texttt{Bar}(\mathbb{H}_k)$, is defined as $\texttt{Pers}(e) = d - b$ if $d < +\infty$, or $+\infty$ otherwise. The times b and d are called, respectively, the birth time and death time of the bar e.*

The decomposition in Theorem 1.3.4 reduces to a linear algebra problem and is, essentially, achieved through a careful choice of basis for the underlying homology modules. Although we have not provided a technical proof of the theorem, the following example helps to clarify the concept.

Example 1.3.2. *(Barcode decomposition) In Example 1.3.1, we demonstrated two samples of filtrations and their respective induced morphisms. Notice that, in the first case, the sequence already takes the form of a composition of two simple interval modules; that is, each of the two homology classes appears at the birth time of a bar and disappears, being mapped to zero at the respective death time. The second case, however, does not initially have this simple form, as the two selected homology classes merge at a certain point. In Figure 1.6, we show how the desired decomposition is achieved for the same scenario by making an appropriate choice of basis cycles.*

To sum up, the barcode encapsulates the topological information of the persistent module. Notice that bars with infinite persistence describe the homology of the entire space (the whole complex). We can also interpret an element of the barcode as a homological class existing between its birth and death times, with persistence

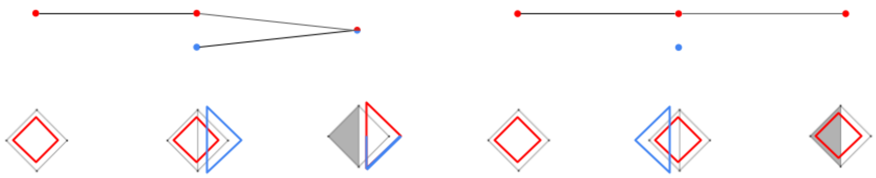


Figure 1.6: Barcode decomposition.

representing the lifespan of this class. By choosing a basis that does not merge, we eliminate any ambiguity in this interpretation.

1.4 Morse Theory

Morse theory demonstrates how the homology, and more generally the homotopy type, of a smooth manifold is intrinsically linked to, and can be fully explored through, the structure of the critical points of almost any real-valued smooth function defined on it. The discrete counterpart, discrete Morse theory, is more immediately relevant from our perspective and will be thoroughly covered in the next section. However, in this section, we aim to provide a brief overview of the continuous case, as it motivates the methods and intuition behind the discrete setting. For these reasons, we will only present the main results without covering the proofs. For a general introduction to the topic of differential topology and Morse functions, we refer the reader to *(Guillemin and Pollack 2010),* while *(Milnor 1963)* covers Morse homology in depth.

Definition 1.4.1. *(Morse function) Given a smooth manifold $\mathcal{X}$ and a real-valued smooth function $f : \mathcal{X} \to \mathbb{R}$, if $f(x) = y$ and $\partial f_x = 0$, we say that x and y are, respectively, a critical point and a critical value of f. We say that f is Morse if, around every critical point x, we can find a local parametrization φ such that the Hessian matrix of the composition at $\varphi^{-1}(x)$, i.e., $H(f \circ \varphi)(\varphi^{-1}(x))$, is invertible. Notice that this condition does not depend on the specific choice of parametrization; hence being a Morse function is a property of the function f alone.*

Theorem 1.4.2. *(Isolated critical points) If f is a Morse function on $\mathcal{X}$, then it has only isolated critical points, meaning that around every critical point x we can find an open set U of $\mathcal{X}$, i.e., $U \subseteq \mathcal{X}$, for which x is the only critical point contained in the set.*

Theorem 1.4.3. *(Genericity) If $\mathcal{X} \subseteq \mathbb{R}^N$ is a smooth manifold embedded in Euclidean space and f is a real-valued smooth function $f : \mathcal{X} \to \mathbb{R}$ defined on the manifold, then for almost every $a \in \mathbb{R}^N$, the function $f_a(x) = f(x) + \langle x, a \rangle$ is a Morse function on $\mathcal{X}$, where $\langle x, a \rangle$ is the scalar product in $\mathbb{R}^N$. This implies that every smooth function f is Morse up to an infinitesimal linear perturbation.*

Example 1.4.1. *Consider the function $f(x) = x^3$, which has only one critical point at $x = 0$, where the gradient $f'(x) = 3x^2$ vanishes. The function is not Morse since the Hessian, which in this case reduces to the second derivative $f^{''}(x) = 6x$, is zero at the origin. For every $\epsilon \in \mathbb{R} - \{0\}$, the perturbed function $f_\epsilon(x) = x^3 + \epsilon x$ has the same second derivative $f_\epsilon^{''}(x) = 6x$, which vanishes only at $x = 0$; however,* *it is not a critical point since $f_\epsilon^{'}(0) = \epsilon$. Consequently, $f_\epsilon(x)$ is Morse. Notice that we can choose ϵ arbitrarily small.*

Example 1.4.2. *Figure 1.7 depicts a torus with a single point resting on the xy plane. The height function on the z axis, i.e., $f(x, y, z) = z$ restricted to the torus points, defines a Morse function. In the image, the four isolated critical points are shown: a minimum at u, saddle points at v and w, and a maximum at z. In this case, we chose the direction $e_3 = (0,0,1)$ to define a height function, i.e., $f(x, y, z) = \langle (x, y, z), e_3 \rangle$. In general, any other choice of direction to define the height leads to a Morse function with the same configuration of four critical points, except for directions parallel to the y axis. In that case, the torus lies fully on the $y = 0$ plane, and, for example, a whole circumference corresponds to a set of non-isolated local minima.*

Theorem 1.4.4. *(Morse lemma) Given a scalar function f defined on a smooth manifold $\mathcal{X}$ of dimension n, if f is Morse, then around every one of its critical points p, there is a local parametrization on which*

$$f \circ \varphi(x) = f(p) + \sum_{i=1}^{n} \epsilon_i x_i^2$$

, where each $\epsilon_i = \pm 1$. Notice that this representation is unique up to a permutation of the elements in the summation.

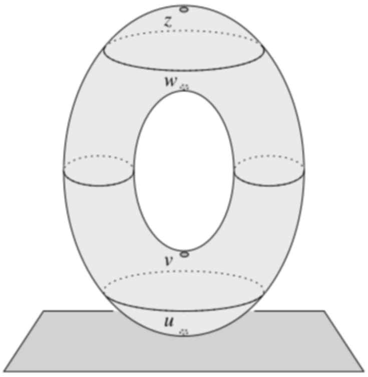


Figure 1.7: Morse function, picture taken from *(Edelsbrunner and Harer 2022).*

Definition 1.4.5. *(Morse index) The index of a critical point p of a Morse function f defined on a smooth manifold $\mathcal{X}$ of dimension n is the number of coefficients ϵ_i in the previous theorem that equal -1. A critical point p with an index of 0 is a local minimum; if the index equals n, it is a local maximum. Generally, if the index is k, we say that p is a k-saddle.*

Figure 1.8 depicts a classical example from Morse theory, originally borrowed from *(Milnor 1963).* The picture shows the changes in the topology of the sublevel sets $\mathcal{X}_t = \{x \in \mathcal{X} \mid f(x) \leq t\}$ of the torus filtered by the height function, as in Example 1.4.2. We can see that qualitative changes that alter the homotopy type of the sublevel set occur at critical values. Before the minimum $f(u)$, the set is empty; between $f(u) \leq t < f(v)$, the set is homotopic to a point or an open disk. Reaching the level $t = f(v)$ is equivalent to attaching a 1-cell to the sublevel set. The same holds for $t = f(w)$, while at the maximum $f(z)$, a 2-cell is added.

Morse theory not only proves that every topological change occurs at a critical value, but it also provides an exact CW-complex decomposition of a smooth manifold $\mathcal{X}$ based on the gradient flow defined by $-\partial f_x$, where the structure of the stable and unstable manifolds of every equilibrium, which is a critical point, is viewed as a cell. In particular, every k-saddle corresponds to a k-cell in this construction. We do not cover this construction in the continuous case; instead, in the following section, the discrete analog for finite complexes will be discussed in detail.

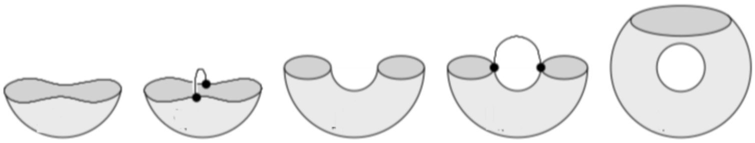


Figure 1.8: Morse decomposition, picture taken from *(Edelsbrunner and Harer 2022).*

1.5 Discrete Morse Theory

In this section, we develop the discrete analog of continuous Morse theory and present an algorithm that, by mimicking gradient descent in a discrete setting, computes the birth and death pairs of the barcode. In the smooth case, changes in the topology of sublevel sets occur at critical values, which are associated with critical points. Similarly, in the discrete setting, we will associate each critical time and change in homology with a corresponding critical cell. The primary reference for discrete Morse theory is *(Forman 2002);* in the following discussion, we also draw on *(Robins et al. 2011),* specifically applied to computer vision, and *(Lingareddy 2018)* for a detailed persistent homology perspective.

Definition 1.5.1. *(Discrete Morse function) A filtration f on the cells of a given regular complex is a discrete Morse function if for every cell σ the following conditions on its faces and cofaces hold:*

1. $\#\{\tau > \sigma | f(\tau) \leq f(\sigma)\} \leq 1$
2. $\#\{\tau < \sigma | f(\sigma) \leq f(\tau)\} \leq 1$

Definition 1.5.2. *(Critical cell) A cell σ is a critical cell if the following conditions on its faces and cofaces hold:*

1. $\#\{\tau > \sigma | f(\tau) \leq f(\sigma)\} = 0$
2. $\#\{\tau < \sigma | f(\sigma) \leq f(\tau)\} = 0$

If a cell is not critical, we say that the cell is regular.

Theorem 1.5.3. *If a cell σ is regular (i.e., it is not critical), conditions 1 and 2 in Definition 1.5.1 are mutually exclusive in the following sense: either σ has exactly one coface $\tau > \sigma$ with $f(\tau) \leq f(\sigma)$, in which case σ has a higher filtration value than all its faces; or σ has exactly one face $\upsilon < \sigma$ with $f(\upsilon) \geq f(\sigma)$, in which case σ has a lower filtration value than all its cofaces.*

Proof: Suppose there is a cell σ that has both a coface $\tau > \sigma$ for which $f(\tau) \leq f(\sigma)$ and a face $\upsilon < \sigma$ such that $f(\upsilon) \geq f(\sigma)$. Since υ would be contained twice in the boundary of τ, there must be another cell η such that $\tau > \eta > \upsilon$. Moreover, τ and υ already have, respectively, a face with a greater or equal value and a coface with a lesser or equal value; thus, it must hold that $f(\tau) > f(\eta) > f(\upsilon)$. But this leads to the absurdity $f(\tau) > f(\upsilon) \geq f(\sigma) \geq f(\tau)$. □

The above theorem shows that regular cells are added in pairs across the filtration $\mathcal{K}_t = f^{-1}(-\infty, t]$, specifically as a pair of a maximal face and its free face as defined in Definitions 1.1.7 and 1.1.8. In contrast, a critical cell has a filtration value greater than all its faces and lower than all its cofaces, meaning it appears in isolation. The next results demonstrate that the filtration value of a critical cell represents a critical time, as changes in homology occur only with the introduction of critical cells.

Theorem 1.5.4. *If $t_1 < t_2$ and $f^{-1}(t_1, t_2]$ contains only regular cells, then $\mathcal{K}_{t_1}$ has the same homology as $\mathcal{K}_{t_2}$.*

Proof: From the previous theorem, regular cells form pairs of a coface σ^k and face τ^{k-1} with the same filtration value. At the time of their insertion, they must be a maximal face, as σ^k cannot be a face of a coface with the same filtration value, and a free face, as τ^{k-1} cannot have another coface with the same filtration value. To convey the basic idea, we consider what happens when we add only one pair of a maximal cell and a free face to the complex. Introducing σ^k, a new k-chain appears, $\texttt{dim}\ C_{k,t_2} = \texttt{dim}\ C_{k,t_1} + 1$, which does not belong to any boundary, nor to any cycle; thus, $\texttt{dim}\ H_{k,t_2} = \texttt{dim}\ H_{k,t_1}$. Since this extra dimension does not contribute to the kernel, it contributes to the image of the boundary operator: $\texttt{dim}\ B_{k-1,t_2} = \texttt{dim}\ B_{k,t_1} + 1$, now containing an element with τ^{k-1}. However, since σ^k is a cell, this boundary must also be a cycle, so $\texttt{dim}\ H_{k-1,t_2} = \texttt{dim}\ B_{k,t_1} + 1 - \texttt{dim}\ C_{k,t_1} - 1 = \texttt{dim}\ B_{k,t_1} - \texttt{dim}\ C_{k,t_1}$, with no change. For a complete proof of the general case, see *(Forman 2002).* □

Theorem 1.5.5. *If $t_1 < t_2$ and $f^{-1}(t_1, t_2]$ contains only a critical k-cell σ, then there is a change in homology and only one of the following two situations can occur:*

1. *$\beta_{k,t_2} = \beta_{k,t_1} + 1$. In this case, $f(\sigma)$ is the birth time of a bar in $\texttt{Bar}(\mathbb{H}_k)$, and σ is said to be a creator cell.*

2. *$\beta_{k-1,t_2} = \beta_{k,t_1} - 1$. In this case, $f(\sigma)$ is the death time of a bar in $\texttt{Bar}(\mathbb{H}_{k-1})$, and σ is said to be a destructor cell.*

Proof: A critical k-cell σ has a filtration value below each of its cofaces; hence, it cannot be part of a boundary at the time of its insertion. Thus, σ creates a new dimension $\texttt{dim}\ C_{k,t_2} = \texttt{dim}\ C_{k,t_1} + 1$, which cannot be a boundary, so $\texttt{dim}\ B_{k,t_2} = \texttt{dim}\ B_{k,t_1}$. Only two cases arise: either this new dimension contributes to the kernel of ∂_k, hence $\beta_{k,t_2} = \beta_{k,t_1} + 1$, or it contributes to the image, giving $\beta_{k-1,t_2} = \beta_{k-1,t_1} - 1$, as the $(k-1)$-boundary gains one dimension in C_{k-1}. □

Example 1.5.1. *(Creator and destructor cells) Figure 1.9 provides examples for both the one-dimensional and two-dimensional cases. The insertion of a critical cell, marked in red, modifies the homotopy type of the complex. A destructor 1-cell reduces the number of connected components in the complex by creating a bridge between two existing components, thus decreasing the dimension of H_0.*

Conversely, a creator 1*-cell opens a loop, forming a new connection within a component, increasing the dimension of* H_1 *by one. In the two-dimensional case, a destructor* 2*-cell fills a hole, transforming the surrounding loops into boundaries and reducing the dimension of* H_1*, while a creator* 2*-cell encloses a volume, creating a new homological class in* H_2*.*

Notice that the previous results relate to a fundamental result of topology: there is no retraction of a ball onto its boundary sphere (see Appendix A). This implies that inserting a cell in isolation must change the homotopy type of a complex. In contrast, we can easily find a retraction of a ball and part of its boundary onto the rest of the boundary. The following result summarizes our construction so far and relies on the fact that we can iteratively retract pairs of regular cells onto the remainder of their boundaries, so that only critical cells are needed to determine the homotopy type of a complex.

Theorem 1.5.6. *(Fundamental theorem of discrete Morse theory) Every regular CW-complex* $\mathcal{K}$ *is homotopic to a CW-complex* $\mathcal{M}$ *with exactly one* k*-cell for every critical* k*-cell of* $\mathcal{K}$*.*

Proof: The proof is based on Theorems 1.5.4 and 1.5.5. A complete proof can be found in *(Forman 2002).* □

The end of this section presents an algorithm for computing persistent homology and the associated pairs of creator and destructor cells by analyzing the structure of the Morse complex. The key ingredient is a discrete analog of a vector field,

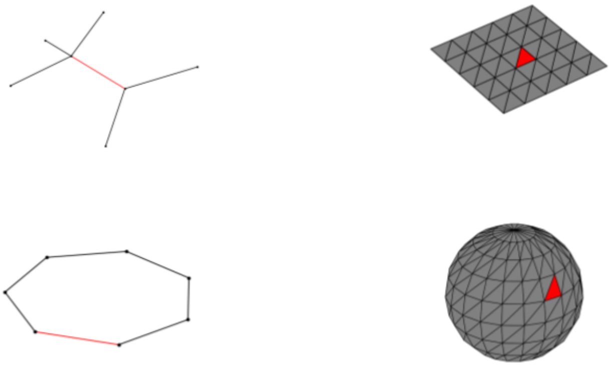


Figure 1.9: Creator and destructor cells.

the discrete vector field, on which we can perform a form of gradient descent algorithm. By following the descending or ascending paths from each critical cell, we can identify the critical cells and their boundary relations, providing an abstract description of the Morse complex.

Definition 1.5.7. *(Discrete vector field) A discrete vector field V on a complex $\mathcal{K}$ is a collection of pairs (σ^k, τ^{k-1}) composed of a k-cell σ^k and one of its faces τ^{k-1} (where the dimension is highlighted for clarity), such that each cell belongs to at most one pair.*

Theorem 1.5.8. *A discrete Morse function f defines a discrete vector field V_f on a complex.*

Proof: From Theorem 1.5.3, we have seen that regular cells form unique pairs of $\tau < \sigma$, a face and coface with the same filtration value $f(\tau) = f(\sigma)$. This is exactly the partial matching we seek. Critical cells remain unpaired. □

Definition 1.5.9. *(V-path) If V is a discrete vector field, a V-path is a sequence of cells*

$$\tau_0^{k-1}, \sigma_0^k, \tau_1^{k-1}, \sigma_1^k, \ldots, \tau_r^{k-1}, \sigma_r^k, \tau_{r+1}^{k-1}$$

, such that $(\sigma_i^k, \tau_i^{k-1}) \in V$ and $\sigma_i^k > \tau_{i+1}^{k-1}$ for every $i = 0, \ldots, r$. If $r \geq 1$ and $\tau_0^{k-1} = \tau_{r+1}^{k-1}$, the path is said to be a non-trivial closed path.

Theorem 1.5.10. *The discrete vector field V_f of a Morse function f does not have a non-trivial closed path.*

Proof: Every V-path is a sequence of pairs τ_i^{k-1}, σ_i^k of regular cells such that f is constant on the pair $f(\tau_i^{k-1}) = f(\sigma_i^k)$. Since $\sigma_i^k > \tau_{i+1}^k$ and σ_i^k already has a face with the same value, it must hold that $f(\tau_{i+1}^k) < f(\sigma_i^k)$. Thus, the path is decreasing on pairs. □

Theorem 1.5.11. *If V is a discrete vector field that has no closed path, there is a Morse function f such that $V = V_f$.*

Proof: A proof can be found in *(Forman 2002)*. □

A vector field imposed by the gradient of a function has no closed orbits, except at equilibrium points. The previous result demonstrates the same property in the discrete setting. Essentially, every V-path can be completed so that it starts and ends at a critical cell. By following these gradient paths, we can establish the boundary relationships between the critical cells of the Morse complex.

In the final section of this chapter, we specialize our discussion and the presented algorithms to the case of 2D cubical complexes, which are particularly relevant for computer vision applications. This focus allows us to simplify the topic and provide a more targeted presentation with relevant examples. For an algorithm that applies to regular complexes in general, see *(King et al. 2005)*.

1.5.1 Applications to Digital Images

We will explain the algorithm of *(Robins et al. 2011),* which serves as the basis for implementing discrete Morse theory in our applications. The original work handles both 2D and 3D data structures; here, we limit the exposition to applications involving 2D cubical complexes and grayscale digital images.

The starting point is a digital grayscale image I, represented as a $W \times H$ matrix. It is convenient to model this image as a subset of $\mathbb{R}^2$: the pixel locations are represented as a set $D = \{(i,j) \mid i,j \in \mathbb{N},\ 0 \leq i \leq W, 0 \leq j \leq H\}$, where a function $f : D \to \mathbb{R}$ is defined so that $g(i,j) = I[i,j]$ (the grayscale value).

To study the topological structure of the image, it is necessary to extend this set to a complex. Specifically, we consider the 2D cubical complex $\mathcal{K}$ whose 0-cells are the points of D, and the 1-cells and 2-cells are all edges and faces formed by adjacent lower-dimensional cells (see Figure 1.3).

The lower cut of an image, D_t, is the set of pixels with grayscale values at most t, i.e., $D_t = \{(i,j) \in D \mid f(i,j) \leq t\}$. The goal is to build a filtration (specifically, its discrete vector field) on the complex such that $D_t \subseteq \mathcal{K}_t$. Moreover, this filtration should minimize the number of critical cells to capture only the topological features induced by the pixel values, avoiding arbitrary additions. This is achieved by considering the lower-star filtration.

Definition 1.5.12. *(Lower-star filtration) Given a regular complex $\mathcal{K}$ and a scalar function g defined on its 0-cells, we recursively define a function f that extends g to all cells of the complex: (1) for every 0-cell σ, we set $f(\sigma) = g(\sigma)$; (2) for every k-cell σ, the function has the value of the maximum of its faces, $f(\sigma) = \max_{\tau < \sigma} f(\tau)$.*

Definition 1.5.13. *(Lower star of a pixel) We define the lower star of a 0-cell x, a pixel, as the set of faces containing the pixel that have the same filtration value: $L(x) = \{\sigma \subseteq \mathcal{K} \mid x \in \sigma,\ f(\sigma) = f(x)\}$. More precisely, it includes the pixel itself, together with all edges and faces for which x is the vertex with the highest filtration value.*

Along the lower-star filtration, the lower star of each pixel appears simultaneously, and the function value on each cell depends entirely on the pixel value. This aligns with our intuitive idea that a filtration should induce a topology that is as directly derived from the image data as possible.

However, the lower-star filtration function f is not yet a discrete Morse function; there are some additional steps needed to obtain a discrete vector field.

First, we must ensure that all pixel values are distinct. An infinitesimal perturbation is applied to break ties while preserving any other order relations.

Theorem 1.5.14. *Given a function g defined on $\mathcal{D}$, let $d = \min_{x,y \in D} |g(x) - g(y)|$. If $d > 0$, then the lower-star sets of the respective filtration f form a partition of the cells of the complex $\mathcal{K}$.*

Proof: Each cell belongs to the lower-star set of its vertices that achieve the maximum filtration value. If each vertex has a unique value, this maximum is unique. □

Theorem 1.5.15. *Let $d = \min_{x,y \in D} |g(x) - g(y)|$. If $d > 0$, then the lower-star sets form a partition. If $d = 0$, there exists an infinitesimal perturbation $g'(x)$ of $g(x)$ such that $d > 0$ and $g(x) > g(y) \implies g'(x) > g'(y)$.*

Proof: Let η be the smallest positive difference among values $|g(x) - g(y)|$. Define $g'(i,j) = g(i,j) + \epsilon \frac{i+Ij}{2IJ}$ with $0 < \epsilon < \eta$.

Since

$$0 < \frac{1}{2IJ} \leq \left| \frac{(i-i') + I(j-j')}{2IJ} \right| \leq \frac{I+IJ}{2IJ} < 1,$$

if $g(i,j) = g(i',j')$, then $|g'(i,j) - g'(i',j')| > 0$; and if $g(i,j) < g(i',j')$, we have $g'(i',j') - g'(i,j) > \eta - \epsilon > 0$. □

From now on, we assume distinct pixel values, resulting in a partition of lower stars. The following algorithm creates a discrete vector field where only cells within the same lower star are paired.

We highlight two remarks about the algorithm, which are relevant for applications:

1. The discrete vector field pairing occurs only among cells within the same lower-star set. Since all cells in this set share the same value from f, ties are broken arbitrarily to assign gradient pairs and critical cells among multiple possibilities. For instance, if a pixel is the local maximum within a neighborhood patch, this choice will only affect which specific 2-cell among four possibilities in the same lower star is designated as critical. There is a one-to-one correspondence between critical 2-cells and pixels that are local maxima within a neighborhood whose number does not depend on these choices.

2. The algorithm constructs a discrete vector field by processing the lower star of each pixel independently. Thus, it has a complexity proportional to the image size $\mathcal{O}(W \times H)$, but can be parallelized to achieve a constant-time complexity of $\mathcal{O}(1)$ with respect to image size.

Algorithm 1.5.1. *(ProcessLowerStars) The proposed algorithm processes $L(x)$, the lower star of each pixel, to return a discrete vector field consisting of critical cells C and pairs of regular cells $V[\alpha^{(p)}] = \beta^{(p+1)}$. The pseudocode from* (Robins et al. 2011) *is given in Algorithm 1. The algorithm uses a function* `num_unpaired_faces` *to track the number of paired faces of a cell. It also requires G, a complete ordering among the cells of the lower star that decreases along faces.*

Once a discrete vector field has been constructed, we know that the critical cells form part of the Morse complex. To compute the boundary relations in the Morse

Algorithm 1 ProcessLowerStars, taken from *(Robins et al. 2011)*.

```
Input: $D$ (digital image pixels), $g$ (grayscale values)
Output: $C$ (critical cells), $V$ (discrete vector field $V[\alpha^{(p)}] = \beta^{(p+1)}$)
for $x \in D$ do
    if $L(x) = \{x\}$ then
        add $x$ to $C$ ( $x$ is a local minimum)
    else
        $\delta \leftarrow$ the 1-cell in $L(x)$ such that $G(\delta)$ is minimal
        $V[x] \leftarrow \delta$
        add all other 1-cells from $L(x)$ to $PQ_{\text{zero}}$
        add all cells $\alpha \in L(x)$ to $PQ_{\text{one}}$ such that $\alpha > \delta$ and
        num_unpaired_faces$(\alpha) = 1$
        while $PQ_{\texttt{one}} \neq \emptyset$ or $PQ_{\texttt{zero}} \neq \emptyset$ do
            while $PQ_{\texttt{one}} \neq \emptyset$ do
                $\alpha \leftarrow PQ_{\texttt{one}}$.pop_front()
                if num_unpaired_faces$(\alpha) = 0$ then
                    add $\alpha$ to $PQ_{\texttt{zero}}$
                else
                    $V[$pair$(\alpha)] \leftarrow \alpha$
                    remove pair$(\alpha)$ from $PQ_{\texttt{zero}}$
                    add all cells $\beta \in L(x)$ to $PQ_{\texttt{one}}$ such that ($\beta > \alpha$ or $\beta >$ pair$(\alpha)$)
                    and num_unpaired_faces$(\beta) = 1$
                end if
            end while
            if $PQ_{\texttt{zero}} \neq \emptyset$ then
                $\gamma \leftarrow PQ_{\texttt{zero}}$.pop_front()
                add $\gamma$ to $C$
                add all cells $\alpha \in L(x)$ to $PQ_{\texttt{one}}$ such that $\alpha > \gamma$ and
                num_unpaired_faces$(\alpha) = 1$
            end if
        end while
    end if
end for
```

complex, we can trace the V-paths originating from each critical cell. Since these paths cannot form cycles, there are only two possible outcomes: either the path will reach the lower star of another critical cell, indicating that this cell is a face of the source cell in the Morse complex, or it will end trivially at the boundary of the image.

Since we need to evaluate the V-paths starting from each critical cell, the complexity of the algorithm scales with the number of critical cells. This number is approximately proportional to the image size, giving a complexity of $\mathcal{O}(W \times H)$. However, because each V-path can be evaluated in parallel, the algorithm could be parallelized over V-paths to achieve performance improvements depending on the actual implementation and computer architecture.

Algorithm 1.5.2. *(ExtractMorseComplex) The algorithm takes as input V, the discrete vector field pairing, and C, the critical cells as computed from the previous method, and returns* `Facelist`*, a list of tuples made by a cell of the Morse complex and the list of its faces. In Algorithm 2, we present the pseudocode from* (Robins et al. 2011).

The retrieved Morse complex may not be regular; for example, it could have a 2-cell with a boundary composed of only one 1-cell (it has only one face). However, it allows us to compute the homology of the former complex in a straightforward way:

1. Every 2-cell σ is the destructor cell of an element in the barcode $\texttt{Bar}(\mathbb{H}_1)$, of which the 1-cell τ in the boundary of σ, i.e., $\tau < \sigma$, with the highest filtration value, is the creator cell.

2. Every 1-cell σ, left unpaired by the previous step, is the destructor cell of an element in the barcode $\texttt{Bar}(\mathbb{H}_0)$, of which the 0-cell τ, in the boundary of σ, i.e., $\tau < \sigma$, with the highest filtration value, is the creator cell.

3. The 0-cell σ, left unpaired, obtains the global minimum of the filtration function and corresponds to the bar in $\texttt{Bar}(\mathbb{H}_0)$ with infinite length, representing the connected component of the entire complex.

This scheme also provides a straightforward method for an algorithm that computes the persistence pairs from a given Morse complex.

Example 1.5.2. *(Discrete Morse Theory on images) We demonstrate an application of the methods developed in this section to a handwritten digit image from the MNIST dataset. Figure 1.10 shows, in order: (1) the input grayscale image, where the background corresponds to values close to zero, and the digit corresponds to values close to one; (2) the discrete vector field computed with the previous algorithms, where cell pairings are represented as arrows from a coface to a face, and the critical*

Algorithm 2 ExtractMorseComplex, taken from *(Robins et al. 2011).*

```
Input: $V$ (discrete vector field), $C$ (critical cells of $V$)
Output: $M$ (cells in the Morse chain complex), Facelist (cell adjacencies of the Morse chain complex)
for $p \in \{0, 1, 2\}$ do
  for $\gamma^{(p)} \in C$ do
    create a new $p$-cell $\tilde{\gamma} \in M$
    if $p > 0$ then
      for $\alpha^{(p-1)} < \gamma^{(p)}$ do
        if $V[\alpha] \neq \emptyset$ then
          $Q_{\text{bfs}}$.push_back($\alpha$)
        end if
      end for
      while $Q_{\text{bfs}} \neq \emptyset$ do
        $\alpha \leftarrow Q_{\text{bfs}}$.pop_front()
        $\beta^{(p)} \leftarrow V[\alpha]$
        for $\delta^{(p-1)} < \beta^{(p)}$ s.t. $\delta \neq \alpha$ do
          if $\delta \in C$ then
            add $\tilde{\delta}$ to Facelist($\tilde{\gamma}$)
          else if $V[\delta] \neq \emptyset$ then
            $Q_{\text{bfs}}$.push_back($\delta$)
          end if
        end for
      end while
    end if
  end for
end for
```

cells are depicted in red, blue, and green for critical 2-cells, 1-cells, and 0-cells, respectively; (3) a set of V-paths that fully describe the boundary relations between their originating critical cells and the sinks; note that some V-paths may end at the image boundary and do not establish a boundary link; (4) a representation of the derived Morse complex, where each point corresponds to a critical cell with the previous color notation, and cofaces are linked to their faces by an edge. The points are positioned according to the vertex with the maximum value, i.e., the pixel associated with the critical cell; (5) the persistence diagram of the barcode $\texttt{Bar}(\mathbb{H}_1)$*, a Cartesian diagram where each bar is represented as a point with its birth and death time coordinates. Notice that there is only one bar, i.e., only one 1-cycle, with significantly high persistence. This corresponds to the fact that the digit forms a single connected component, representing a hole in the background that was created early in the sublevel set filtration and was closed at the end (when* $t = 1$*).*

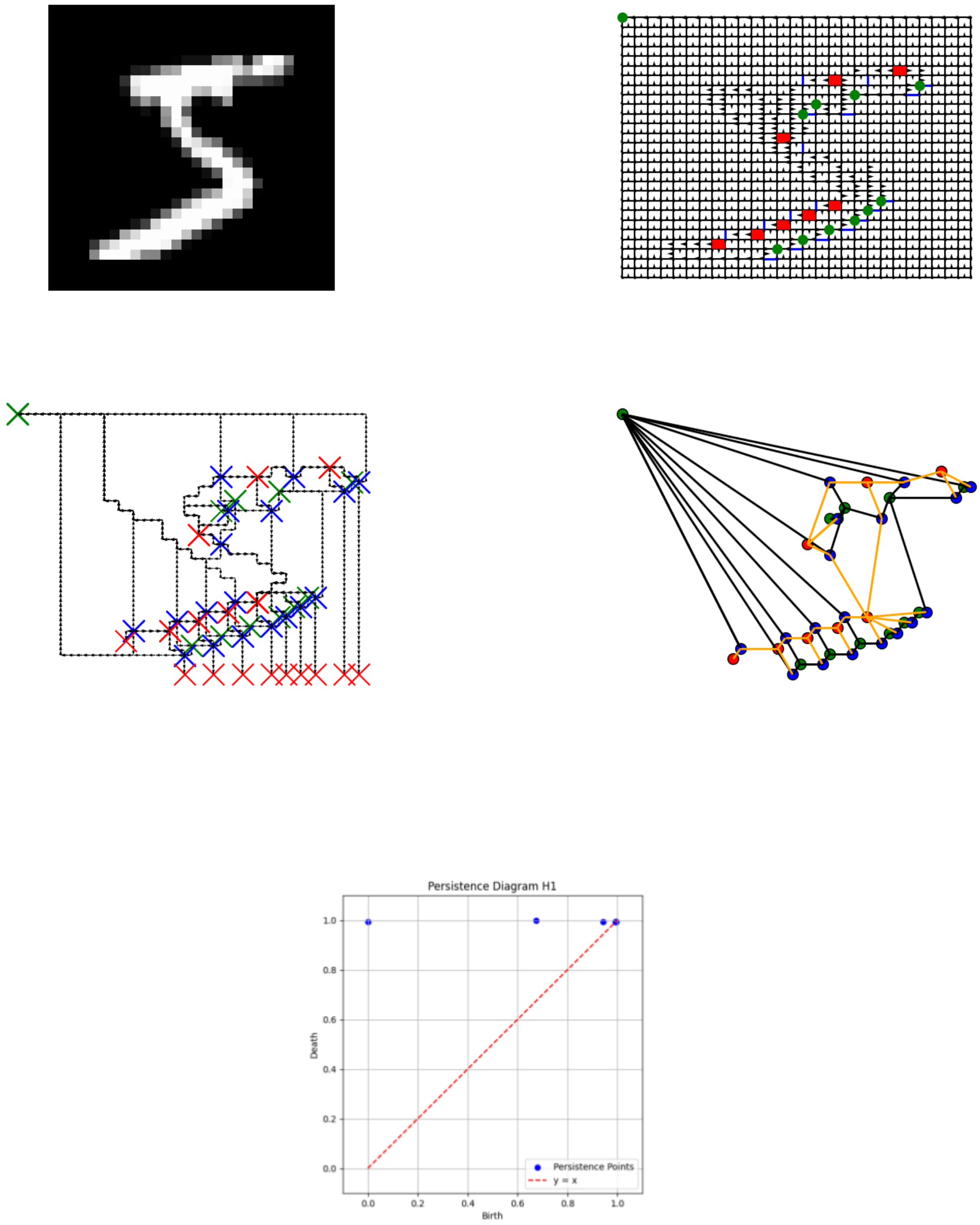


Figure 1.10: Discrete Morse Theory.

Chapter 2

Computer Vision

In this chapter, we revisit some concepts and problems in computer vision, particularly within image matching, which is our field of application. A fundamental challenge in these areas is understanding the relationship between different views of the same subject. For example, a sequence of photos of the ground captured from a moving probe can be used to infer the probe's relative trajectory and facilitate a successful landing. Conversely, if the subject is moving, multiple frames can be used to estimate its velocity and future position, as long as the object can be accurately matched between frames. Image matching can also enhance the accuracy of retrieval tasks in image-based search engines; we refer to one of our previous works published in *(Barbarani, Mostafa, et al. 2023)* as an example.

To provide a sufficiently complete understanding of the foundations of our topics, we begin with the pinhole camera model, the simplest and most instructive abstraction of the physical model underlying a camera and the process of capturing digital images, discussed in Section 2.1. In Section 2.2, we introduce some notions of projective geometry and demonstrate how the core aspects of the problem and solutions can be approached by working with the projective plane and homographies. Our primary references for this subject are *(Richter-Gebert 2011)* for mathematical results on projective geometry and *(Fusiello 2024)* for the applicative context.

Once we have developed the necessary tools, the initial problem of understanding the relationship between two images reduces to the task of estimating a homography. Section 2.3 explains how current methodologies address this task through a set of correspondences, which may be uncertain, derived from the image data. Many algorithms have been proposed to estimate these correspondences within the keypoints-and-descriptors paradigm: a set of reliable pixel locations (keypoints) is identified, along with descriptors of the represented features. Numerous algorithms have been proposed for this task, ranging from theoretically grounded hand-crafted solutions to data-intensive deep learning models.

Examples of these methodologies, as well as the overall paradigm of image matching, are covered in Section 2.4, with a special focus on some prototypical examples. Specifically, we explore the scale-space theory framework, following the comprehensive work of *(Lindeberg 2013),* and its primary application, SIFT, an effective hand-crafted method proposed by *(Lowe 2004).* Among recent deep learning approaches, we discuss R2D2, introduced by *(Revaud et al. 2019),* while additional deep learning methods are briefly covered in Appendix C.

2.1 The Pinhole Camera Model

The earliest photographs were taken with the help of a camera obscura, a dark environment where light rays could enter exclusively through a small pinhole, imprinting themselves on a photoreactive plate placed on the opposite side. The pinhole camera model is an abstraction that captures how the primary aspects of perspective work in this setup. By thinking of the pupil (or the lens of a modern camera) as the pinhole, and the retina (or sensor) as the plate, this model serves as a reasonable abstraction for vision systems in general.

The model consists of a point C, called the center of projection, which represents the position of the pinhole in space, and an image plane Q, which represents the plane where the plate lies. The focal length f is the distance between the center of projection and the image plane. To explain the geometry of perspective, we introduce a Cartesian coordinate system with C as the origin. The model is depicted in Figure 2.1. As shown in the figure, a source point $P = (X, Y, Z)$ emits a ray of light that is projected onto a two-dimensional image point $p = (x, y)$. By similarity of triangles, we can explicitly derive the relationship between their components:

$$\begin{cases} x = -f\frac{X}{Z} \\ y = -f\frac{Y}{Z} \end{cases}$$

From these equations, we can quickly retrieve some intuitive aspects of perspective: the farther an object is from the center of projection (increasing Z), the less space it will occupy in the image. Conversely, increasing the focal length f, as with an adjustable zoom in photography, will make the object appear larger.

To understand where the object at P will be mapped in a digital image, we also need to account for an independent rescaling to pixel coordinates, an axis reversal to convert the negative image plane to a positive one, and a translation to align with the top-left corner as the pixel origin.

$$\begin{cases} x = fk_x\frac{X}{Z} + c_x \\ y = fk_y\frac{Y}{Z} + c_y \end{cases}$$

The set of parameters f, k_x, k_y, c_x, and c_y are called intrinsic parameters, as they depend only on the camera and not on its position in space.

Before continuing, we introduce some simplifications to make the treatment of the problem easier, both in notation and complexity.

First, we assume the focal length to be one, $f = 1$, by default. Additionally, we consider our image plane to be in front of the center of projection, so we do not need to flip the image. Specifically, we set the image plane as the xy-plane translated to $z = 1$. Note that these assumptions do not alter the generality of the model: in real systems, the sensor lies behind the camera, but by considering a positive translation in z instead, we achieve the necessary axis reversal. As for focal length adjustments, we will demonstrate how to handle changes in the camera's internal settings and coordinates.

A stronger assumption we make in our treatment is to consider only planar objects; in other words, we assume the scene to be a specific plane in space. This assumption is effective for many applications, as it reasonably approximates flat surfaces like building facades, the ground viewed from above, and objects at sufficient distance. Moreover, this simplified model is instructive in general, as more complex scenarios can often be handled by considering multiple planes.

In this context, every point on the plane emits radial light with a specific value, which can be a scalar intensity or an RGB vector—either representation suits the general problem.

Our main assumptions is captured in the following definition.

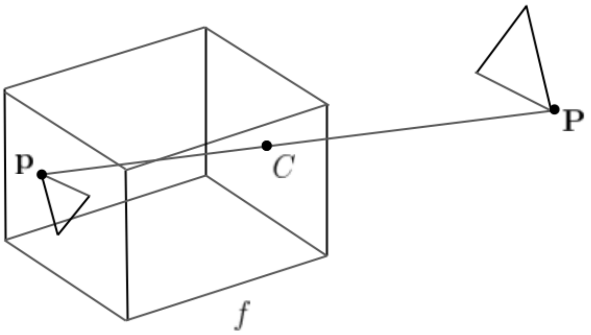


Figure 2.1: The pinhole camera model.

Definition 2.1.1. *(Object) The object $\mathcal{O}$ is an affine subspace of $\mathbb{R}^3$ that is the domain of a function $F : \mathcal{O} \to \mathcal{V}$ (we are not interested in specifying the codomain).*

Consider the (positive) image plane of a camera with a center of projection C that necessarily does not lie on the object $\mathcal{O}$.

When a ray of light from a point on the plane reaches the center of projection C and strikes the image plane, it imprints the same light value as the originating point. Thus, the function F induces a function on a subset of points in the image plane, F_I, which in the camera coordinate system is given by

$$F_I\left(\frac{X}{Z}, \frac{Y}{Z}\right) = F(X, Y, Z)$$

for every $(X, Y, Z) \in \mathcal{O}$.

As shown in Figure 2.2, if we move our camera to a new position or orientation, formally changing the coordinate system via an isometry $T(P) = R(P) + t$, where $R \in SO(3)$ and $t \in \mathbb{R}^3$, there will be a relationship between the pixel values depicted in our new image plane I' and the former one. In the coordinate system of the new camera, this relationship is given by

$$F_{I'}\left(\frac{X}{Z}, \frac{Y}{Z}\right) = F_I\left(\frac{T^{-1}(X, Y, Z)_1}{T^{-1}(X, Y, Z)_3}, \frac{T^{-1}(X, Y, Z)_2}{T^{-1}(X, Y, Z)_3}\right)$$

for every $(X, Y, Z) \in \mathcal{O}$.

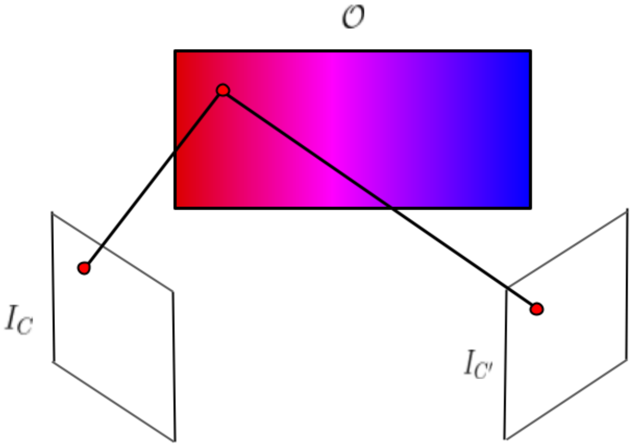


Figure 2.2: Planar object.

In the next section, we will formalize these relations and concepts more precisely and show how, under certain conditions, we can generate a second image without the need to capture a new photo.

2.2 The Projective Plane

In this chapter, we introduce the formalism of projective geometry to model perspective in Euclidean space. Our main goal is to explain how the relationship between different images of the same planar object can be reduced to linear algebra problems and solved with well-known tools.

Definition 2.2.1. *(Bundle of rays) $\mathcal{P}_{\mathbb{R}}$ is the collection of equivalence classes of non-zero vectors in $\mathbb{R}^3$ that differ by a scalar multiplication. In formula, an element $[\lambda] \in \mathcal{P}_{\mathbb{R}}$ is a one-dimensional subspace $[v] = \{\lambda v \mid v \in \mathbb{R}^3 - \{0\}, \lambda \in \mathbb{R} - \{0\}\}$ with the 0 element excluded. Note that orthogonality relations are preserved by the classes, i.e., $v \perp w \iff [v] \perp [w]$.*

Definition 2.2.2. *(Homogeneous coordinates) For every point $(x, y, z) \in \mathbb{R}^3$ such that $z \neq 0$, we define its homogeneous coordinates to be the point $\left(\frac{x}{z}, \frac{y}{z}, 1\right)$. Note that if the homogeneous coordinates are defined for a representative v of a class $[v] \in \mathcal{P}_{\mathbb{R}}$, then all representatives of the same class have the same homogeneous coordinates. Thus, we can refer to the homogeneous coordinates of the class $[v]$ without ambiguity.*

The object $\mathcal{P}_{\mathbb{R}}$ that has just been introduced can be thought of as the bundle of rays passing through the origin. In the previous section, we showed how, in a camera model, the color or light value of a point in the image plane is inherited from the ray that hits it—specifically, from the unique ray passing through the center of projection and a point chosen on the image plane. This defines, at most, a bijection between the bundle of lines $\mathcal{P}_{\mathbb{R}}$ passing through the origin, set as our center of projection C, and the points in the image plane $\mathcal{Q}$, namely the plane over $(x, y, 1)$. Every point of the image $\mathcal{Q}$ thus represents the homogeneous coordinates of a class in $\mathcal{P}_{\mathbb{R}}$. However, there is a puzzle to solve in this discussion: lines with directions parallel to $\mathcal{Q}$ (i.e., those with $z = 0$) do not correspond to any point, as they represent rays parallel to the image plane that never intersect it. If we consider a sequence of points in homogeneous coordinates $(a + rx, b + ry, 1)$ with increasing r, they correspond to a sequence of rays with directions $\left(\frac{a}{r} + x, \frac{b}{r} + y, \frac{1}{r}\right)$ in $\mathcal{P}_{\mathbb{R}}$, which tend toward the direction $(x, y, 0)$ as $r \to \infty$. Thus, $\mathcal{P}_{\mathbb{R}}$ can be seen as an extension of the image plane with the addition of limiting points $(x, y, 0)$, called points at infinity.

Let $\mathcal{L}_{\mathbb{R}}$ be, for notational purposes, an identical but distinct copy of $\mathcal{P}_{\mathbb{R}}$. We can use the same set to model lines on the image plane $\mathcal{Q}$. Each line can be defined by its normal as the set $(a, b, 1) \in \mathcal{Q}$ such that $(a, b, 1) \perp (x, y, z)$ for a fixed $(x, y, z) \neq 0$. This set remains the same for any non-zero multiple of the normal vector, meaning each class $[v] \in \mathcal{L}_{\mathbb{R}}$ defines a line on the image plane. The line with normal direction $(0, 0, z)$ do not correspond to any line in the image plane, but can be thought of as the line passing through all points at infinity, as $(x, y, 0) \perp (0, 0, z)$. Points $(x, y, 0)$ in our extended image plane $\mathcal{P}_{\mathbb{R}}$ represent the limit of all the sequences like $(a + rx, b + ry, 1) = (a, b, 1) + r(x, y, 0)$, which can be understood as the intersection points of parallel lines in the image plane with direction $(x, y, 0)$.

We have seen that $\mathcal{P}_{\mathbb{R}}$ and its copy $\mathcal{L}_{\mathbb{R}}$ extend both the image plane $\mathcal{Q}$ and its one-dimensional affine subspaces with the addition of points at infinity and a line passing through them. From a formal point of view, this extension is consistent and preserves all Euclidean geometry properties of interest, such as orthogonality and incidence relations. For an in-depth treatment, see *(Ghrist 2014)*. Practically, there is nothing inherently special about points at infinity—they are an artifact of perspective. As shown in Figure 2.3, different positions and orientations of the camera in space lead to different points at infinity. This extension is summarized in the following definition.

Definition 2.2.3. *(Projective plane) Given $\mathcal{P}_{\mathbb{R}}$ as defined in Definition 2.2.1, and its identical but distinct copy $\mathcal{L}_{\mathbb{R}}$, the triplet $\mathbb{RP}^2 = (\mathcal{P}_{\mathbb{R}}, \mathcal{L}_{\mathbb{R}}, \mathcal{I}_{\mathbb{R}})$ is called the projective plane, where $\mathcal{P}_{\mathbb{R}}$ and $\mathcal{L}_{\mathbb{R}}$ are its points and lines, respectively, and $\mathcal{I}_{\mathbb{R}}$ is*

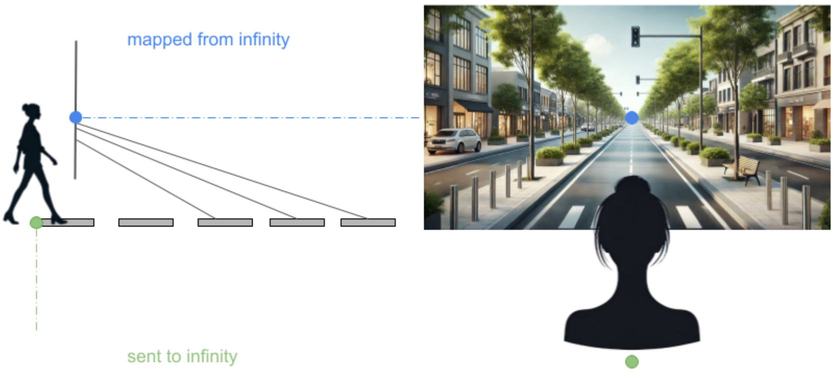


Figure 2.3: Point at infinity.

the set of incidence relations:

$$\mathcal{I}_{\mathbb{R}} \subseteq \mathcal{P}_{\mathbb{R}} \times \mathcal{L}_{\mathbb{R}} \quad \textit{such that} \quad ([v],[w]) \in \mathcal{I}_{\mathbb{R}} \iff [v] \perp [w]$$

The projective plane $\mathbb{RP}^2$ can be thought of as the extension of the image plane $\mathcal{Q}$ with its extended points and lines. Every subset of its points, lines, and incidence relations that has an immediate interpretation in terms of homogeneous coordinates behaves as points and lines of $\mathcal{Q}$ in Euclidean space.

Definition 2.2.4. *(Homography) A homography, or projective transformation $\mathcal{T} = (\mathcal{T}_{\mathcal{P}_{\mathbb{R}}}, \mathcal{T}_{\mathcal{L}_{\mathbb{R}}})$, is an isomorphism of the projective plane $\mathbb{RP}^2$. This means a pair of bijective functions*

$$\mathcal{T}_{\mathcal{P}_{\mathbb{R}}} : \mathcal{P}_{\mathbb{R}} \to \mathcal{P}_{\mathbb{R}} \quad \textit{and} \quad \mathcal{T}_{\mathcal{L}_{\mathbb{R}}} : \mathcal{L}_{\mathbb{R}} \to \mathcal{L}_{\mathbb{R}}$$

such that

$$([v],[w]) \in \mathcal{I}_{\mathbb{R}} \implies (\mathcal{T}_{\mathcal{P}_{\mathbb{R}}}([v]), \mathcal{T}_{\mathcal{L}_{\mathbb{R}}}([w])) \in \mathcal{I}_{\mathbb{R}}.$$

Theorem 2.2.5. *Every 3×3 invertible matrix M defines a homography. If two 3×3 invertible matrices M and N differ by a scalar multiplication, i.e., $M = \lambda N$ with $\lambda \neq 0$, then they define the same transformation. We denote by $[M]$ the class of matrices that differ by a non-zero scalar multiple, i.e., that represent the same transformation of the projective plane.*

Proof: Both M and $\left(M^T\right)^{-1}$ are linear maps, thus they induce a pair of well-defined functions on equivalence classes of $\mathcal{P}_{\mathbb{R}}$ and $\mathcal{L}_{\mathbb{R}}$, respectively. As the matrices are invertible, both functions are bijections. Finally, it holds that

$$\langle x, y \rangle = 0 \iff \langle Mx, \left(M^T\right)^{-1} y \rangle = 0,$$

hence incidence relations are mapped consistently. If $M = \lambda N$, then both M and N induce the same bijection on classes, and the same holds for the inverse of their transposes. □

Theorem 2.2.6. *(Fundamental theorem of projective geometry) Every isomorphism of the projective plane $\mathbb{RP}^2$ is induced by a 3×3 invertible matrix H.*

Proof: The proof is highly technical and requires a more complete exposition from abstract algebra; see *(Richter-Gebert 2011)* for details. □

As a result of the fundamental theorem, in the context of the real projective plane $\mathbb{RP}^2$, every projective transformation is induced by a matrix, and the terms homography, isomorphism, and projective transformation are commonly used interchangeably. Likewise, an invertible 3×3 matrix is often directly referred to as a homography or homography matrix. In our context, we will adopt this common terminology.

As we have seen, the projective plane naturally extends the image plane and its geometry, and we can think of the color of a point in $\mathcal{Q}$ as the color of a ray in $\mathcal{P}_{\mathbb{R}}$ interchangeably. Based on this, we define an image as follows:

Definition 2.2.7. *(Image) An image is a function defined on a subset I of $\mathcal{P}_{\mathbb{R}}$, i.e., $F : I \subseteq \mathcal{P}_{\mathbb{R}} \to \mathcal{V}$. An initial image can be defined by the object (Definition 2.1.1), seen as the homogeneous coordinates for a coordinate system in which the plane $\mathcal{O}$ coincides with the image plane. In this case, F is defined at every point except those at infinity. Alternatively, we may have an initial photo of the object, in which case the image is defined on a rectangular region $\mathcal{R} = [a, b] \times [c, d]$ of the image plane that corresponds to rays in direction $(x, y, 1)$ for every $(x, y) \in \mathcal{R}$.*

The following 3 × 3 matrices show how the intrinsic transformations discussed in the previous section act on the image in terms of the projective plane and homographies:

$$K = \begin{bmatrix} -k_x & 0 & c_x \\ 0 & -k_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad I_f = \begin{bmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad H_{\text{intr}} = KI_f$$

To map an image F defined on our ideal image plane coordinate system to the real image defined on the real coordinates and physical units of the sensor, we compute $F_f = F \circ I_f^{-1}$. In practice, F_f is defined on the rectangular region representing the sensor $\mathcal{R}_f = [-a, a] \times [-b, b]$. To convert to pixel coordinates, starting from the top-left corner as commonly used in computer vision (i.e., $\mathcal{R} = [0, H] \times [0, W]$), we choose parameters in K so that $\mathcal{R} = K(\mathcal{R}_f)$, ensuring correct axis orientation. The digital image will then be a discretization of $F_f \circ K^{-1}$. Overall, the homography matrix H_{intr} represents a transformation that depends solely on the intrinsic parameters of the physical system or data representation conventions.

Note that K, I_f, and H_{intr} map homogeneous coordinates to homogeneous coordinates, so if the input is in planar coordinates, the output is immediately interpretable as an image plane coordinate. For a generic homography matrix, however, further division by the third component of the output is needed to locate it on the image plane.

Now we address the main problem presented in the introductory section and depicted in Figure 2.2. We consider a camera with center of projection C and image plane $\mathcal{Q}$, and a second camera with center of projection $C' = R(C) + t$ and image plane $\mathcal{Q}' = R(\mathcal{Q}) + t$, where $T(x) = R(x) + t$ is an isometry with $R \in SO(3)$ and $t \in \mathbb{R}^3$. A ray passing through the center of projection C and a specific point on the image plane $\mathcal{Q}$ that intersects the object at P will have the same color as the ray passing through C' and $\mathcal{Q}'$ that intersects the object at the same point P. This defines a bijection between a subset of the bundle of rays at C and a subset of those at C', excluding rays parallel to the image plane, which would have no color in either system. However, this bijection is also determined by the position of the object in space, as shown in Figure 2.4. Moving the plane $\mathcal{O}$ to the plane $\mathcal{O}'$ changes the pairing of rays from H to H'. Thus, a projective transformation describing the mapping from the image plane in the coordinate system of the first camera to the second depends on the relative position of the object in space.

The following theorem formalizes this relationship. Note that in the theorem, we assume the coordinate system of the first camera, which defines the domain of the homography. Thus, R is expressed in the coordinates of this system.

Theorem 2.2.8. *The homography matrix H representing the change of perspective obtained by moving the camera location and orientation via an isometry $T(x) = Rx + t$, where R is a 3×3 invertible matrix and $t \in \mathbb{R}^3$, is given by $H = R + \frac{tn^T}{d}$, where n and d are the outward normal and distance from the origin, respectively, of the object plane $\mathcal{O}$.*

Proof: The theorem is a well-known result in computer vision and projective geometry. For a step-by-step proof, see *(Fusiello 2024).* The proof presented here follows the language and methods of this context, primarily based on linear algebra and the reasoning developed so far.

We know how H acts on lines represented by points on the object plane, as $[Hx] = [Rx + t]$ for every $x \in \mathcal{O}$. Since the center of projection (the origin) lies outside the object plane, we can find three linearly independent points on the plane using the outward normal n and the distance from the origin d. We know that $a = dn$ lies on the plane. Let m and w be vectors in the basis of the orthogonal

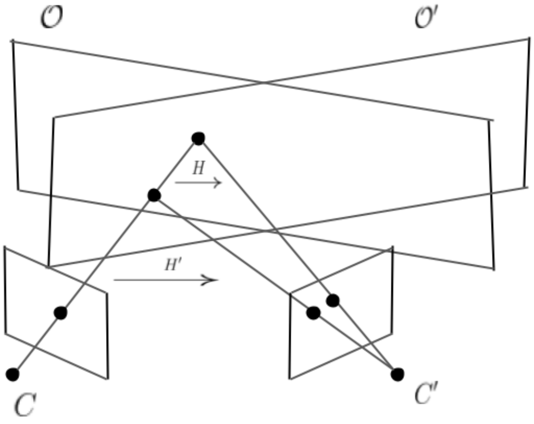


Figure 2.4: Relative Homographies.

complement $\{n\}^{\perp}$, so that n, m, and w form an orthonormal basis. We can find two more points on the object plane, $b = dn + m$ and $c = dn + w$; together, a, b, and c form a basis for $\mathbb{R}^3$, and we denote its basis matrix as $B = [a, b, c]$. Define the matrix

$$M = RB + t\mathbb{1}^T,$$

where M is invertible and will be shown to equal H_B, our desired homography matrix in the basis B.

In B-coordinates, any point on the plane can be expressed as $(1 - \lambda - \eta, \lambda, \eta)$, as:

$$\begin{gathered} n^T((1 - \lambda - \eta)a + \lambda b + \eta c) = \\ n^T(d(1 - \lambda - \eta)n + \lambda(dn + m) + \eta(dn + w)) = d. \end{gathered}$$

Thus, if $x \in \mathcal{O}$, it can be expressed as $(1 - \lambda - \eta, \lambda, \eta)$ in B for some choice of parameters. Notice how M acts on x in B-coordinates:

$$M \begin{pmatrix} 1 - \lambda - \eta \\ \lambda \\ \eta \end{pmatrix} = RB \begin{pmatrix} 1 - \lambda - \eta \\ \lambda \\ \eta \end{pmatrix} + t\mathbb{1}^T \begin{pmatrix} 1 - \lambda - \eta \\ \lambda \\ \eta \end{pmatrix} = Rx + t.$$

Therefore, $H_B = M$ acts as the desired map in B-coordinates. To express it in the usual basis, we right-multiply M by B^{-1}:

$$H = MB^{-1} = R + t\mathbb{1}^T B^{-1}.$$

Finally, we verify that $(B^{-1})^T \mathbb{1} = \frac{n}{d}$. Using the transpose properties:

$$\begin{gathered} \langle \left(B^{-1}\right)^T \mathbb{1}, n\rangle = \langle \mathbb{1}, B^{-1}n\rangle = \langle \mathbb{1}, \frac{a}{d}\rangle = \langle \mathbb{1}, \frac{e_1}{d}\rangle = \frac{1}{d} \\ \langle \left(B^{-1}\right)^T \mathbb{1}, m\rangle = \langle \mathbb{1}, B^{-1}(b - a)\rangle = \langle \mathbb{1}, e_2 - e_1\rangle = 0 \\ \langle \left(B^{-1}\right)^T \mathbb{1}, w\rangle = \langle \mathbb{1}, B^{-1}(c - a)\rangle = \langle \mathbb{1}, e_3 - e_1\rangle = 0 \end{gathered}$$

Thus $\left(B^{-1}\right)^T \mathbb{1}$ has the same expression of $\frac{n}{d}$ in the orthonormal system n, m, w and they are equal. □

The theorem shows that we can depict how an image would look if taken from a different position using an initial image and additional information about the object's location. If we know the homography matrix H, we can evaluate $F \circ H^{-1}$ on the rectangular region $\mathcal{R}$ representing our picture. In practice, only covisible regions (i.e., points lying in $\mathcal{R} \cap H(I)$, where I is the domain of the initial image) can be reconstructed. This technique is widely used, for instance, to generate synthetic data for neural network training or to adjust photos of buildings taken from the ground to reduce distortion for aesthetic purposes. Examples are shown below.

Example 2.2.1. *(Random homographies) Synthetic perspective changes are widely adopted for data augmentation in neural network training for computer vision tasks. This technique is used to prevent overfitting and to make classifiers robust to perspective changes. Figure 2.5 shows two pairs of images: the first images are real photos, while the second images are generated by applying random homographies. These are examples of training data used for image matching tasks.*

With knowledge of H, we can also solve the system in Theorem 2.2.8 for R and t, or for n and d. In general, the system may be underdetermined, requiring additional assumptions or observations to obtain a unique solution. However, the relative camera positions can often be recovered when the homography matrix is known with sufficient accuracy, which is particularly relevant for applications such as trajectory estimation and 3D reconstruction.

In the next chapter, we address the problem of homography estimation: knowing the visual features of two images (i.e., having two photos of the same scene), which effectively addresses the problem of recovering a homography matrix from raw image data.

2.3 Homography Estimation

In this section, we show how a homography matrix can be estimated from two images of the same scene, using only the image data in pixel coordinates. This

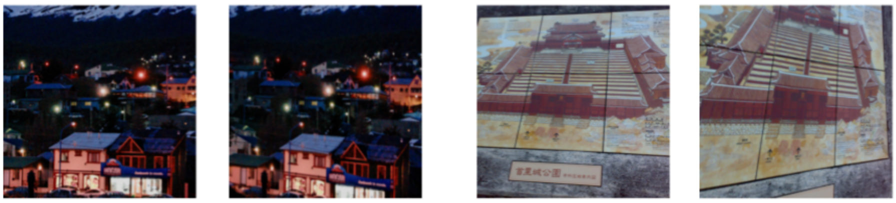


Figure 2.5: Random homographies.

provides the foundation of image matching, an application field that addresses the problem of finding correspondences between images.

The next result shows that all we need are four correspondences between non-collinear pixel locations in the first and second images. Since a homography $[H]$ is defined up to a scalar multiplication, it has eight free parameters, and four pairs of points in homogeneous coordinates provide exactly eight linear equations, which are linearly independent if we are not in a degenerate case.

Theorem 2.3.1. *Given four points $[v_i] \in \mathcal{P}_{\mathbb{R}}$, $i = 1, .., 4$, such that every subset of three points is linearly independent, and another four points $[w_i] \in \mathcal{P}_{\mathbb{R}}$ satisfying the same condition, consider a bijection $[v_i] \to [w_i]$ between them. There is a unique homography, represented by H, that extends this bijection, i.e., such that $[Hv_i] = [w_i]$ for every i.*

Proof: Consider the basis $B = [v_1, v_2, v_3]$ of $\mathbb{R}^3$. Expressed in this basis, the homography matrix H_B is simply given by representatives (i.e., multiples) of w_1, w_2, w_3 stacked by columns: $H_B = [\lambda w_1, \tau w_2, \mu w_3]$. The fourth element v_4 can be expressed in the basis coordinates as $\beta \in \mathbb{R}^3$, where $v_4 = \beta_1 v_1 + \beta_2 v_2 + \beta_3 v_3$, on which H_B must act as

$$H_B \beta = w_4.$$

Given our assumptions, this system is solvable and we can find λ, τ, and μ. Finally, we obtain H in the usual basis by $H = H_B B^{-1}$. □

Given a set of matchings, i.e., correspondences, between pixel locations (whether as rays, homogeneous coordinates, or image plane coordinates), denoted by

$$\mathcal{M} : A \to B,$$

where $A, B \subseteq \mathcal{P}_{\mathbb{R}}$ are discrete subsets and $\mathcal{M}$ is a bijection between them. Theorem 2.3.1 suggests how an algorithm for homography estimation can be implemented in a straightforward manner, provided the matchings satisfy certain minimal constraints. One simple approach is to compute H by minimizing its squared error over the matchings, as is common in regression tasks. However, such estimators are not robust to outliers, and computer vision typically requires higher accuracy. For these reasons, a more robust approach is often preferred in practice, namely an algorithm based on Random Sample Consensus (RANSAC), which iteratively seeks the best four pairs of matchings.

Algorithm 2.3.1. *(RANSAC) The pseudocode for an implementation of RANSAC is provided in Algorithm 3. For a complete reference on the algorithm, we refer to* (Fischler and Bolles 1981). *The algorithm takes as inputs a list of at least four matchings* $\mathbf{M}$*, represented as pairs of points in $\mathbb{R}^3$ (i.e., $M[i] = (p_i, p_i'))$, an error tolerance threshold ϵ, and the number of iterations T to be run.*

The function `RandomSampling` *returns a random sample of four pairs of matching points. The algorithm makes use of* `CheckCollinearity`*, a function that evaluates four matchings and returns* `True` *if they satisfy the non-collinearity conditions of Theorem 2.3.1, or* `False` *otherwise. The function* `FindHomography` *takes the sample as input and returns the homography matrix H (represented as a 3×3 array), following the steps of the previous theorem. The function* `CountInliers` *takes the matchings, the threshold, and a homography matrix as inputs. It applies H to every point p_i and checks if the distance from p'_i in homogeneous coordinates is less than ϵ; if so, the pair is counted as an inlier. This function returns the number of inliers of the given homography H. Finally, the algorithm returns the homography matrix $H_{\texttt{best}}$ with the maximum number of inliers among those evaluated.*

Algorithm 3 RANSAC

```
Input: $\mathbf{M}$ (list of matchings), $\epsilon$ (tolerance), $T$ (number of iterations)
Output: Best homography $H_{\texttt{best}}$
$H_{\texttt{best}} \leftarrow 0$, $n_{\texttt{best}} \leftarrow 0$
for $i$ from 1 to $T$ do
    $m \leftarrow$ RandomSampling($\mathbf{M}$)
    if CheckCollinearity($m$) then
        $H \leftarrow$ FindHomography($m$)
        $n \leftarrow$ CountInliers($\mathbf{M}, H, \epsilon$)
        if $n > n_{\texttt{best}}$ then
            $H_{\texttt{best}} \leftarrow H$
            $n_{\texttt{best}} \leftarrow n$
        end if
    end if
end for
return $H_{\texttt{best}}$
```

In conclusion, we have shown how to obtain a robust and reliable estimation of the homography matrix relating two different images from a set of (possibly noisy) correspondences. In the next section, we will delve into the current methodologies for extracting these pairs of matches from raw image data.

2.4 Image Matching

Image matching refers to the broad field of techniques and challenges related to understanding relationships between different images. Homography estimation, for instance, is a common problem in this field. A comprehensive survey of the field can be found in *(Ma et al. 2021).*

The main approach in image matching is to extract a set of local features $\mathcal{F}_A = \{(p_i, d_i)\}$ from an image A. Each local feature consists of a keypoint p, representing a notable pixel location, and a descriptor $d \in \mathbb{R}^K$, capturing the visual characteristics of that point. Local features from different images, such as $\mathcal{F}_A$ and $\mathcal{F}_B$, are then matched to find correspondences. These correspondences are essential for estimating the homography matrix between images using RANSAC, as covered in the previous section. Once local features are extracted, various methods are used to establish matching relationships; the simplest is to identify mutual nearest neighbors (MNN) based on the Euclidean distance in descriptor space. Specifically, a pair of pixel locations (p_i, p_j), where $(p_i, d_i) \in \mathcal{F}_A$ and $(p_j, d_j) \in \mathcal{F}_B$, is considered a match in $\mathcal{M}_{A,B}$ if

$$d_i = \arg\min_{d_k \in \mathcal{F}_A} ||d_k - d_j|| \quad \text{and} \quad d_j = \arg\min_{d_k \in \mathcal{F}_B} ||d_k - d_i||.$$

Keypoints are essential in image matching, helping to identify unique regions within images, often in areas like corners or intersections of lines. The key requirements for effective keypoints include high repeatability, allowing them to be consistently found across different images of the same scene, even with changes in viewpoint or lighting. They should also be easily matchable with corresponding points in other images. Keypoints work together with descriptors, which capture the visual properties of the detected feature.

Various algorithms have been developed to detect local features, ranging from classic hand-crafted methods to modern deep learning approaches. These algorithms are often called "detectors" or "extractors" in image matching workflows. Detectors are typically evaluated using two approaches. The first follows the protocol from *(Mikolajczyk and Schmid 2005),* which assesses detectors individually using metrics like keypoint repeatability, designed to predict performance in multiple applications. The second approach evaluates the entire image matching pipeline directly on the downstream task, as suggested by *(Jin et al. 2021).*

In the following sections, we introduce a classic hand-crafted algorithm, SIFT, with its background in scale-space theory, and a deep learning-based detector, namely R2D2, which are particularly helpful for understanding applications relevant to our study. Additional methods are briefly described in Appendix C.

2.4.1 Scale-Space Theory

Scale-space theory is a foundational framework in computer vision that enables the analysis of image structures across multiple scales, addressing the inherent variability in object appearances depending on the observation scale. This theory acknowledges that objects exhibit meaningful features only within certain scale ranges; for instance, analyzing the texture of a leaf requires a close-up scale, while observing the overall form of a tree demands a broader view. To account for these variations, scale-space theory proposes a method to systematically represent images across a continuum of scales by progressively smoothing the image data using Gaussian filters. This approach results in a series of increasingly coarser representations of the original image, in which finer details are gradually suppressed.

More specifically, scale-space theory proposes analyzing the image through the lens of a one-parameter operator, named the scale-space operator. This operator, represented by the function $L(x, t)$, is the convolution of a grayscale image $f(x)$ with a Gaussian kernel $g(x, t)$, where t serves as a scale parameter that controls the progressive loss of details through the level of blurring:

$$L(x,t) = \int_{\mathbb{R}^2} f(x-\xi)\, g(\xi,t)\, d\xi$$

Here, $g(x,t) = \frac{1}{(2\pi t)^{n/2}} e^{-\frac{||x||^2}{2t}}$ defines the Gaussian kernel, which smooths the image and progressively removes finer details as t increases.

The choice of a Gaussian kernel is not arbitrary; it has emerged from multiple streams of research, motivated by theoretical insights and biological inspirations, as thoroughly reviewed in *(Lindeberg 2013).* Essentially, the Gaussian kernel is the only valid choice that provides an operator exhibiting strong scale-invariance properties.

The intuition that structures recognizable at a distance should already be visible in finer details suggests the need for a form of non-creation property. Formally, this implies that keypoints detected at a coarse scale should also appear at finer

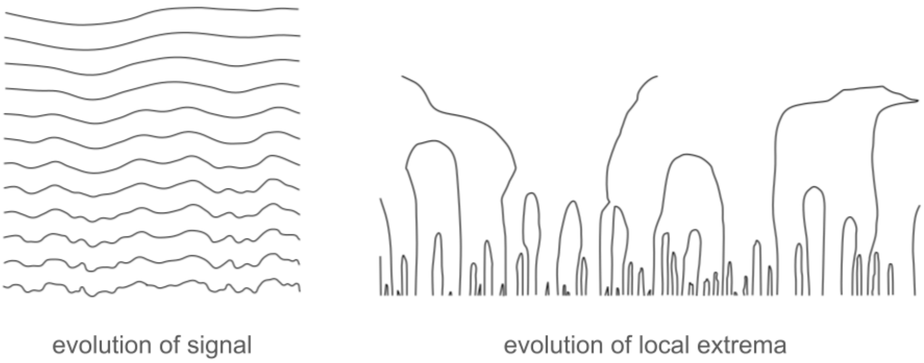


Figure 2.6: Progressive smoothing, picture taken from *(Lindeberg 2013).*

scales. For one-dimensional signals, the local extrema of the scale-space operator uphold this property: as shown in Figure 2.6, increasing the level of smoothing causes pairs of local minima and maxima to merge and cancel out, without creating any new extrema. Although this property does not extend directly to higher dimensions, the scale-space operator still preserves certain non-creation properties for multidimensional signals. In particular, as shown in *(Yuille and Poggio 1986),* it does not introduce new zero-crossings of its Laplacian, and similar conditions apply to other derived linear differential operators.

Scale Invariant Feature Transform (SIFT) is a widely recognized algorithm in computer vision based on the framework of scale-space theory, published in *(Lowe 2004).* It creates a pyramid of filters by repeatedly applying Gaussian convolutions to the input digital image with an increasing scale and multiple strides. Then, it computes a set of keypoints based on the local extrema of the differences of the resulting filters. The computational flow of SIFT is depicted in Figure 2.7. The pyramid of convolutions can be seen as a discrete implementation of the scale-space operator, while the Difference of Gaussians (DoG) serves as an approximation of its Laplacian.

The relationship between DoG and the Laplacian can be derived from the heat diffusion equation, parameterized in terms of σ rather than $t = \sigma^2$:

$$\frac{\partial g}{\partial \sigma} = \sigma \nabla^2 g.$$

From this, we see that $\nabla^2 g$ can be approximated by the finite difference of $\frac{\partial g}{\partial \sigma}$.

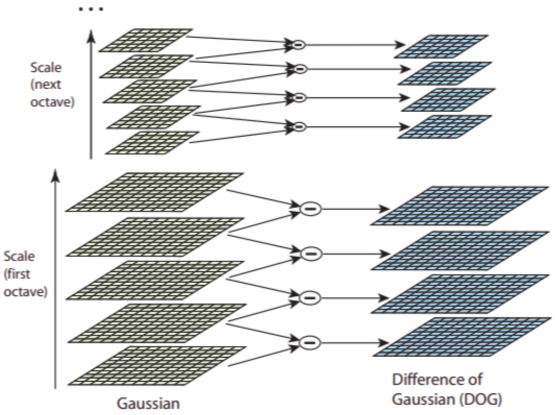


Figure 2.7: SIFT, picture taken from *(Lowe 2004).*

For a detailed explanation, see the original work in *(Lowe 2004)*.

After locating these DoG-based keypoints, SIFT assigns each keypoint an orientation based on the local image gradients, achieving rotational invariance. Finally, it generates a distinctive descriptor by analyzing gradient orientations around each keypoint.

In summary, scale-space theory introduces a notion of keypoints based on their consistency across scales, which has proven successful in practical applications. This structured approach allows keypoints to represent meaningful image features that are robust to scale variations.

2.4.2 Deep Learning

Neural networks are universal function approximators and can be effectively trained using gradient descent algorithms. If we can express our desired properties through a differentiable loss functional, then a neural network can approximate the minimizer and produce any desired output. This capability makes neural networks highly adaptable, and in computer vision, convolutional neural networks (CNNs) have specialized to leverage spatial structures in images. Compared to traditional, hand-crafted feature extraction methods, CNNs can be directly trained on data to provide representations robust to various transformations, noise, and distortions.

A notable application of deep learning to image matching is R2D2 from *(Revaud et al. 2019)*. This method implements CNNs and a set of loss functions for jointly learning keypoint detection and descriptors. By processing an input digital image $I \in \mathbb{R}^{H \times W \times C}$, the network produces three output feature maps: (1) a descriptors map $D \in \mathbb{R}^{H \times W \times K}$, (2) a scalar repeatability map $S \in \mathbb{R}^{H \times W}$, and (3) a scalar reliability map $R \in \mathbb{R}^{H \times W}$. Keypoints are selected as locations maximizing both repeatability and reliability, with the associated descriptor taken from the corresponding location in the descriptor map.

The R2D2 training pipeline involves two primary loss functions: a repeatability loss for keypoint detection and a reliability loss for descriptor discriminativeness. We will focus on the repeatability loss, as it is particularly instructive for our further development, while referring to the original work for a comprehensive explanation of descriptor training.

For repeatable keypoints, R2D2 employs a self-supervised loss to ensure that the repeatability map S remains consistent across different viewpoints and illumination changes. It uses image pairs I, I' of the same scene, related by a known homography. These image pairs can be synthetically generated, as discussed in Section 2.2 and Example 2.2.1, or real pairs with recorded camera intrinsic and extrinsic parameters. Additional data augmentation techniques, such as noise injection and random illumination changes, are applied. The homographic relation projects the repeatability map output from the second image into the coordinate system of the first image I, resulting in two aligned repeatability maps, S and S'.

To enforce consistency between S and S', R2D2 uses cosine similarity computed over image patches:

$$L_{\text{cosim}}(I, I') = 1 - \frac{1}{|P|} \sum_{p \in P} \text{cosim}(S[p], S'[p])$$

where P represents overlapping patches in the images, and $S[p]$ is the vectorized $N \times N$ patch of S. This ensures that local maxima in S and S' align under transformations. To avoid a trivial constant solution, a peakiness loss encourages distinct local maxima in S and S':

$$L_{\text{peaky}}(I) = 1 - \frac{1}{|P|} \sum_{p \in P} \left(\max_{(i,j) \in p} S_{ij} - \text{mean}_{(i,j) \in p} S_{ij} \right)$$

The final repeatability loss L_{rep} combines these two terms:

$$L_{\text{rep}}(I, I') = L_{\text{cosim}}(I, I') + \lambda \Big(L_{\text{peaky}}(I) + L_{\text{peaky}}(I') \Big)$$

Moreover, at inference time, R2D2 processes an input image at multiple scales to enforce scale invariance of the detected keypoints, identifying keypoints by finding local maxima in the combined repeatability and reliability maps.

In summary, the method implements a system of different loss functions and uses both real and synthetic data to train the CNN outputs to be equivariant to homographies and invariant to noisy transformations.

Chapter 3

Applications

In this chapter, we present our applications of algebraic topology to the field of computer vision, already published in *(Barbarani, Vaccarino, et al.)* In particular, we aim to address a gap in the field of image matching, specifically the lack of a scale-independent notion of keypoints. These motivations are thoroughly explained in Section 3.1. In Section 3.2, we introduce a framework for scale-free keypoint detection based on deep learning and an unsupervised loss function. This proposed methodology builds on all the material covered in the previous chapters and represents a novel contribution to computer vision. In Section 3.3, we empirically demonstrate the validity of our approach with experiments on common benchmarks. At the end of the chapter, readers will find our final remarks and conclusions on the current state of the research and directions for future work.

3.1 Motivations

In Section 2.4, we explained the importance of keypoint detection—the consistent extraction of points from an image across different views—which is a fundamental task in computer vision and serves as a crucial preliminary step for many complex applications.

A theoretical framework for this problem is provided by scale-space theory (see Section 2.4.1). In this context, keypoints of an image $I \in \mathbb{R}^{H \times W}$ are modeled as the collection of local extrema (maxima and minima) of a one-parameter operator related to the Laplacian of the scale-space operator. The guiding principle for designing this operator is the *non-creation property*: features noticeable at a coarse scale should have been already visible in finer details at smaller scales. Therefore, an ideal operator should remain consistent across scales, identifying keypoints at a larger scale s_2 as a subset of the keypoints at a smaller scale $s_1 < s_2$. Many classical handcrafted keypoint detectors exploit this scale-space theoretical framework, with the most notable example being SIFT, also discussed in Section 2.4.1.

Recently, several learning-based detectors have been introduced. In the spirit of deep learning, these methods replace a formal definition of keypoints with a data-driven approach, teaching a neural network to select salient points. Inspired by scale-space theory, they model keypoints at inference time as local maxima of a scalar map produced by a trained convolutional neural network. However, at training time, these methods require several relaxations to define a differentiable loss function. A common approach is to consider local maxima within a fixed-size $N \times N$ sliding window. As an example, we have detailed the loss function of a deep learning method, R2D2, in Section 2.4.2.

Despite these recent innovations in deep learning, classical handcrafted solutions remain competitive and often outperform their learnable counterparts. We hypothesize that a major reason for this is the current formulation of keypoints in the deep learning literature, which relies on a fixed-size, patch-wise, differentiable relaxation of the concept of local maxima. This approach encourages models to detect keypoints at a specific frequency, introducing a scale dependency that contradicts the non-creation property, a requirement established as crucial in earlier literature. As depicted in Figure 3.1, a training objective based on a fixed-size $N \times N$ sliding window may encourage the model to find keypoints in large, untextured areas or to miss multiple keypoints that are close to one another. Overall, this approach could be inconsistent when the same subject is presented at different resolutions.

We thus outline the necessity for a new methodology, suitable for gradient-based optimization, that does not rely on any relaxation. This approach should model

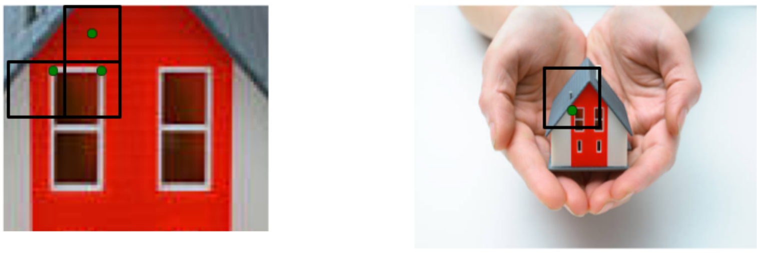


Figure 3.1: Scale inconsistency.

and optimize the structure of local maxima of an output scalar map in its generality, independent of feature scale.

To address this, we propose a framework based on homology theory. Persistent homology is particularly suitable for gradient-based methods, as formally established by *(Leygonie et al. 2021)* and *(Carriere et al. 2021),* and empirically demonstrated in applications such as neural network regularization *(Chen et al. 2019),* deep learning autoencoders *(Moor et al. 2020),* and image segmentation *(Hu et al. 2021; Gupta et al. 2024).* Moreover, persistent homology, as deeply explored in Morse theory and discrete Morse theory (see Section 1.4 and Section 1.5), establishes a bijection between local extrema and the evolution of topological features along a sublevel-set filtration. Specifically, each local maximum corresponds to a loop that appears at a critical value associated with a saddle point and closes at the maximum value itself, as illustrated in Figure 3.2. This formulation introduces no model-specific choices to represent local maxima and is inherently scale-independent, as it quantitatively tracks only the persistence of features. This persistence, being topological in nature, does not depend on the size of the region that the feature occupies.

3.2 Methods

In this section, we introduce **MorseDet**, a keypoint detector model based on deep learning that falls within the image matching paradigm described in Section

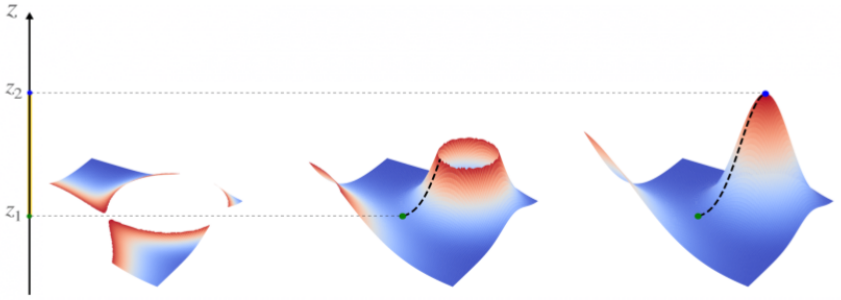


Figure 3.2: Local maximum.

2.4. As is common in the field, it models keypoints as local maxima; however, as discussed in the previous section, our method is built upon persistent homology and its connection to local maxima, as established by discrete Morse theory, which is treated in Section 1.5.

Our model utilizes a convolutional neural network backbone F_θ that, given an input image $I \in \mathbb{R}^{H \times W \times C}$, generates discrete pixel locations $\{k_i\} \in \mathbb{R}^2$ representing keypoints. Specifically, we use the convolutional neural network described in *(Tian et al. 2017),* with the same modifications from *(Revaud et al. 2019),* as a backbone.

To incorporate a topological approach, we modify the final layer of the backbone to output a single channel, yielding a scalar map for each image. This map, denoted $F_\theta(I) = \mathfrak{H} \in \mathbb{R}^{H \times W}$, serves as a unified representation of spatial features, or *height map*, analogous to the concept of a height function used in Morse theory.

We adopt the same training data pipeline as R2D2, covered in Section 2.4.2. Every training instance consists of images I_1 and I_2, along with a ground-truth correspondence map $U \in \mathbb{R}^{H \times W \times 2}$, which encodes pixel-level correspondences within co-visible regions. The map specifies, for instance, that $U[i, j] = (i', j')$ if pixel (i', j') in I_2 corresponds to pixel (i, j) in I_1.

The objective function $\mathcal{L}_{\mathtt{det}}(\mathfrak{H}_1, \mathfrak{H}_2)$, a detection loss that operates on the height maps $\mathfrak{H}_1$ and $\mathfrak{H}_2$ produced by forwarding I_1 and I_2 through the model, is optimized using a common approach based on stochastic gradient descent on mini-batches. The loss function $\mathcal{L}_{\mathtt{det}}$ is the key innovation of our approach, as it enforces local peakness of the height maps while ensuring reproducibility at topologically relevant locations. The design of the loss function will be treated in detail in the next

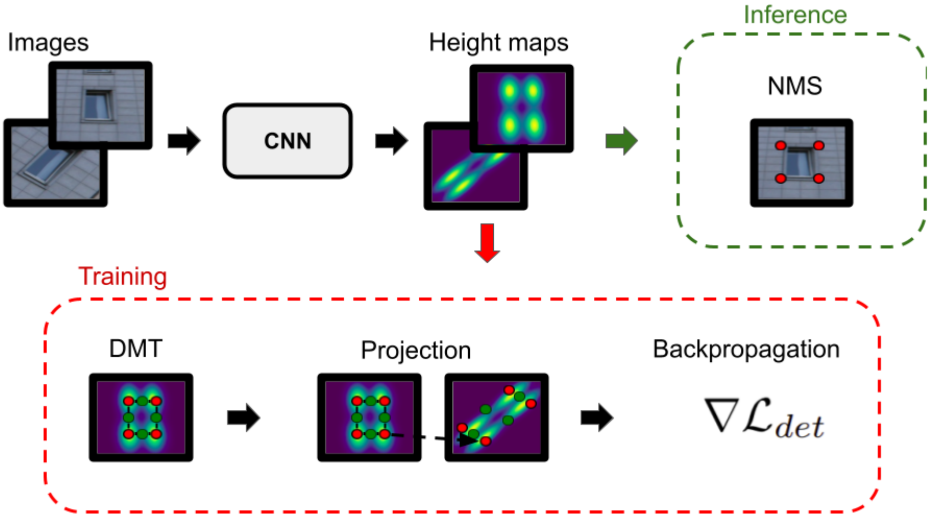


Figure 3.3: Pipeline.

section.

At inference time, MorseDet performs keypoint detection based on local maxima in a straightforward manner that does not involve discrete Morse theory. Given a new image, we proceed as follows:

1. The input image I is processed through the trained feature extractor F_θ, producing a height map $\mathfrak{H} \in \mathbb{R}^{H \times W}$.

2. Local maxima are identified by performing a Non-Maximum Suppression (NMS) algorithm on $\mathfrak{H}$, which rapidly finds all local maxima by comparing each pixel with its neighbors.

3. To filter out noisy features, only local maxima that exceed a threshold γ are selected as keypoints. We set $\gamma = 0.7$ in our experiments.

A summary of the training and inference pipeline is provided in Figure 3.3.

3.2.1 Loss Function

In contrast to previous heuristic methods, during training, we model keypoints bijectively with the local maxima of the feature map. We *refer* to a local maximum via the associated topological feature.

Formally, we construct $\mathcal{K}(\mathfrak{H}_1)$, the 2D cubical complex associated with $\mathfrak{H}_1$, the output scalar height map obtained by the backbone convolutional neural network from the input image I_1. In Section 1.5, we described the methodology for extending the pixel values of a grayscale image through lower-star filtration and for deriving a discrete vector field on the cubical complex. Unlike traditional applications of discrete Morse theory in computer vision, our method works on the complex derived from the output feature map $\mathfrak{H}_1$, which can still be viewed as an $H \times W$ image, rather than from the raw input image.

Let $\mathbb{H}_1(\mathfrak{H}_1)$ denote the persistent homology module given by the filtration on the complex $\mathcal{K}(\mathfrak{H}_1)$. From the theory developed in Section 1.3, each element $e \in \mathtt{Bar}(\mathbb{H}_1(\mathfrak{H}_1))$ can be seen as a cycle that appears at a specific birth time $b(e)$ and disappears at a death time $d(e)$ along the filtration, with persistence defined as $\mathtt{Pers}(e) = d(e) - b(e)$. Since the filtration is determined by a discrete Morse function, Theorem 1.5.5 establishes a correspondence between $d(e)$ and $b(e)$ with a critical 2-cell and a critical 1-cell, respectively.

Through the construction of the lower-star filtration and the discrete vector field, each critical cell belongs uniquely to the lower star of a single pixel location where the input entry value matches the critical time, allowing us to directly associate critical times with "critical pixels." As discussed in Section 1.5, this establishes a bijection between the death times $d(e_i)$ of elements in $\mathtt{Bar}(\mathbb{H}_1(\mathfrak{H}_1))$ and the

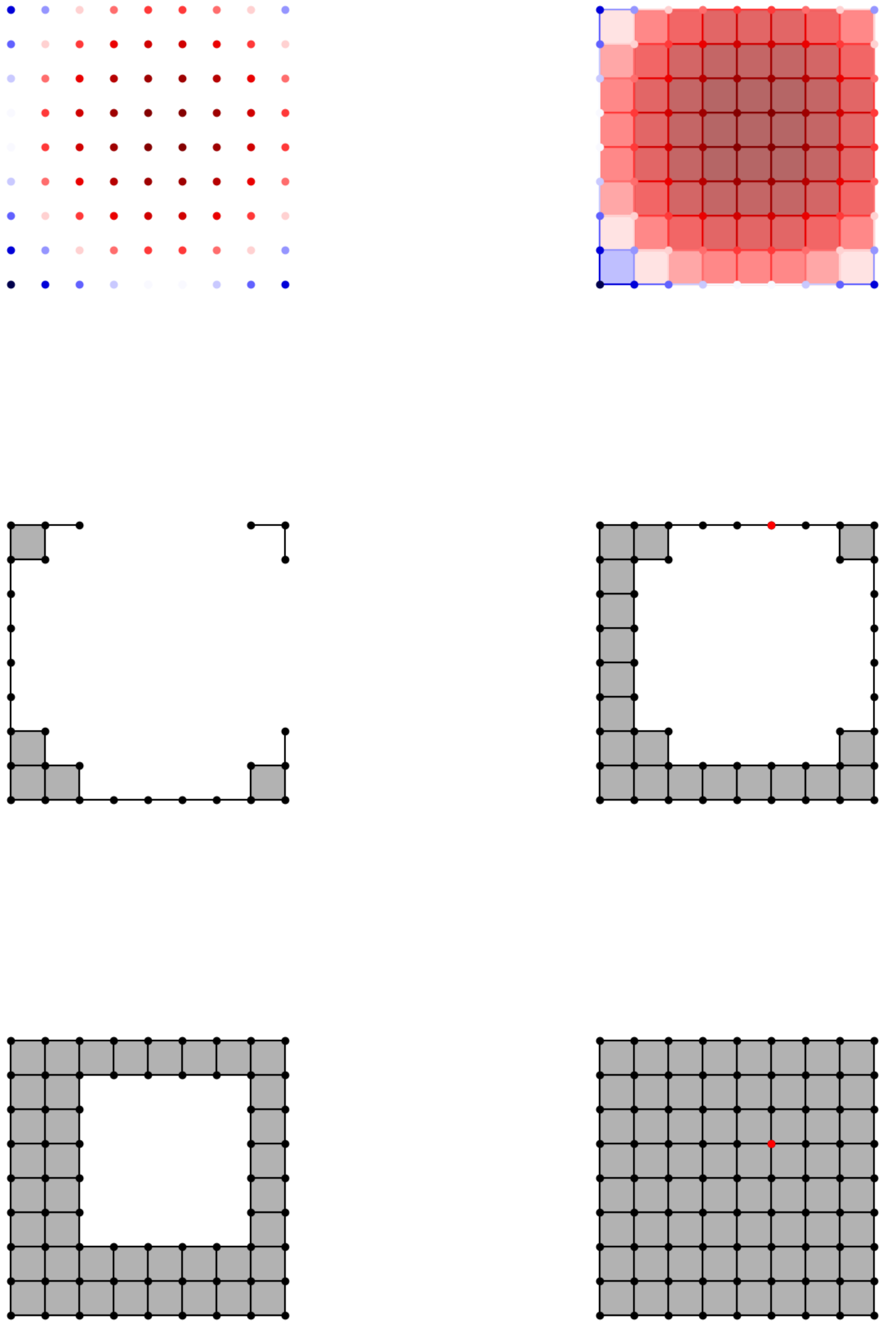


Figure 3.4: Critical pixels.

entries of $\mathfrak{H}_1$ that, when viewed as pixel locations, represent local maxima across neighboring patches.

An example is shown in Figure 3.4, which illustrates: (1) $\mathfrak{H}_1$ as a pixel grid colored according to the values of the entries; (2) the cubical complex $\mathcal{K}(\mathfrak{H}_1)$, where each cell is colored according to its filtration value; (3) the filtered complex $\mathcal{K}_{t_0}$ at a stage with no holes; (4) $\mathcal{K}_{t_1}$, where $t_1 = b(e)$ corresponds to a birth time, and the insertion of a saddle (a critical 1-cell) creates a hole. This cell belongs to the lower star of a specific pixel, marked in red; (5) $\mathcal{K}_{t_2}$, filtered at a regular time t_2, showing the hole shrinking but without a qualitative change; (6) $\mathcal{K}_{t_3}$, where $t_3 = d(e)$, the death time at which the introduction of a critical 2-cell closes the hole. Here, the critical cell is associated with the pixel also marked in red.

In the degenerate case where multiple entries of $\mathfrak{H}_1$ have the same value, this construction requires an infinitesimal perturbation, as described in Theorem 1.5.15. The exact implementation of this perturbation can be regarded as a design choice; we discuss the role of such perturbation in the training process from a theoretical perspective at the end of this section. However, once this design choice has been fixed, we can always define a function that associates each birth time $b(e)$ and death time $d(e)$ with a corresponding saddle pixel $s(e)$ and maximum pixel $m(e)$, such that $b(e) = \mathfrak{H}[s(e)]$ and $d(e) = \mathfrak{H}[m(e)]$ (the critical times match the values of the input feature map at the respective critical pixel locations).

The entries of the height map are functions of the neural network parameters that depend on the input image, which ultimately allows us to define an objective function based on the critical times that supports gradient backpropagation.

Given a training instance (I_1, I_2, U), which consists of two images of the same scene and a correspondence map U defined on co-visible regions, we define the error matrix between two height maps $\mathfrak{H}_1$ and $\mathfrak{H}_2$ as follows:

$$E[i,j] = \mathfrak{H}_1[i,j] - \mathfrak{H}_2[U[i,j]]$$

if U is defined at (i,j); otherwise, $E[i,j] = 0$. We introduce a new term, the *boundary similarity*, to account for differences in $\mathfrak{H}_1$ and $\mathfrak{H}_2$ at topologically relevant positions. For each $e \in \texttt{Bar}(\mathbb{H}_1(\mathfrak{H}_1))$, the boundary similarity term is defined as:

$$\texttt{Sim}(e) = E[s(e)]^2 + E[m(e)]^2$$

Given a positive constant α, the proposed detector loss for keypoint detection is finally defined as

$$\mathcal{L}_{\texttt{det}}(\mathfrak{H}_1, \mathfrak{H}_2) = -\sum_{e \in \texttt{Bar}(\mathbb{H}_1(\mathfrak{H}_1))} \texttt{Pers}(e)\left[\texttt{Pers}(e) - \alpha \texttt{Sim}(e)\right] \tag{3.1}$$

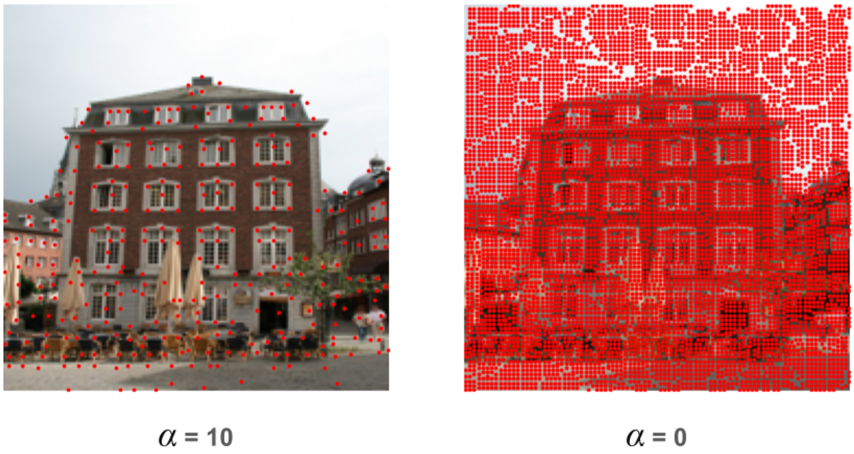
To understand our objective function, consider the case when $\alpha = 0$. In this scenario, the optimization of the loss function corresponds to maximizing

$\sum \texttt{Pers}(e)^2$, the total squared persistence of $\mathcal{K}_1$, which leads to a trivial and uninformative solution. Without the $\texttt{Sim}(e)$ term, the loss function drives the output feature map to contain as many local maxima as possible, resulting in a grid of 1s surrounded by 0s within every 3×3 patch, disregarding the input image values.

Figure 3.5 compares models trained with $\alpha = 0$ and $\alpha = 10$. The model trained with $\alpha = 0$ produces an almost ideal grid pattern of local maxima, while the model trained with $\alpha = 10$ generates repeatable local maxima that align with image corners and edge endpoints, effectively capturing meaningful keypoints. Indeed, the boundary similarity term serves as a regularizing constraint, promoting an increase in the persistence term $\texttt{Pers}(e)$ only when the height maps are reproducible at the corresponding critical pixel locations (i.e., if $\mathfrak{H}_1$ and $\mathfrak{H}_2$ have approximately the same values at $m(e)$ and $s(e)$). The strength of this regularization is controlled by the hyperparameter α, which is set to 10 in our experiments.

Notice that $\mathcal{L}_{\texttt{det}}$, compared to previous approaches in the image matching literature, does not involve any scale parameter. Our loss is capable of modeling the local maxima of the output feature map $\mathfrak{H}_1$ in their generality, without imposing any fixed frequency. Indeed, it depends solely on quantities that are topological in nature.

We conclude by discussing some theoretical aspects of the loss function and its optimization. Previous works *(Leygonie et al. 2021; Carriere et al. 2021)* have shown that adopting a perturbation to compute critical cells corresponds to choosing a directional derivative in the filtration values. Simpler objectives using


α = 10

α = 0

Figure 3.5: Boundary similarity.

only persistence terms and critical times are Lipschitz continuous, ensuring the convergence of gradient methods to a local optimum. In practice, the choice of an infinitesimal perturbation (see Theorem 1.5.5) determines which pixel is assigned a critical time among multiple pixels with the same value.

In our loss function, the boundary similarity terms introduce discontinuities. The loss is still differentiable almost everywhere, specifically when $\mathfrak{H}_1$ has distinct values. However, in cases where $\mathfrak{H}_1$ has multiple entries with the same value, the function value and gradient depend on the corresponding location in $\mathfrak{H}_2$. Indeed, it is always possible to define a direction in the filtration values along which the function and its gradient can be extended continuously, but they may not agree across different choices.

Although our training objective belongs to a family of functions that are wild and may hinder the convergence of gradient methods, the loss function was designed to meet specific application requirements. Empirically, we observe convergence to high-quality optima, as shown in the next section.

3.3 Experiments

Regarding the evaluation of the presented methods, our main concern is comparing the quality of the extracted keypoints for different detectors. Following the protocol established by *(Mikolajczyk and Schmid 2005),* repeatability is a key metric in the literature for assessing the detection of reproducible keypoints. In our experiments, we adopt the version from *(DeTone et al. 2018),* which aligns with current point-based prediction methods. For clarity, we detail this metric: given two sets of predicted keypoints A, B from a pair of images I_1, I_2 related by a homography U, a keypoint $x \in A$ is positively referenced in B if $\min_{y \in B} ||x - U^{-1}(y)||$ is less than a threshold ϵ, where $U^{-1}(y)$ is the projection of y through the ground truth homography. The repeatability score is the average number of keypoints with a positive reference, typically assessed within covisible areas, acknowledging that detectors do not know a priori which regions will match.

However, repeatability is influenced by the number of extracted keypoints. For instance, a uniformly distributed grid of keypoints can artificially inflate the score. To mitigate this, we varied the maximum number of keypoints in our experiments, as in *(Revaud et al. 2019).* Despite this, the metric may still favor detectors that produce clustered keypoints, as noted by *(Rey-Otero et al. 2015; Lenc and Vedaldi 2018).* The work in *(Lenc and Vedaldi 2018)* suggested a method that restricts keypoints to match at most once: computing repeatability based on matches from an optimally constructed bipartite graph, minimizing the sum of a cost function based on distance, with a proposed greedy approximation for this optimization problem.

Thus, we employ a revised version of the repeatability metric, which further requires keypoints to be mutually nearest neighbors, i.e.,

$$x = \underset{x' \in A}{\arg\min} \, ||y - U(x')|| \tag{3.2}$$

and

$$y = \underset{y' \in B}{\arg\min} \, ||x - U^{-1}(y')|| \tag{3.3}$$

In the following, we compare the performance of MorseDet, our method, in terms of the repeatability metric against a comprehensive set of baselines, namely: SIFT, a handcrafted approach discussed in Section 2.4.1; R2D2, a deep learning method covered in Section 2.4.2; and D2-Net *(Dusmanu et al. 2019),* SuperPoint *(DeTone et al. 2018),* DISK *(Tyszkiewicz et al. 2020),* and ALIKED *(Zhao et al. 2023),* which are briefly introduced in Appendix C.

3.3.1 Viewpoint and Illumination

We assessed the capability of our method to predict repeatable keypoints using the well-established HPatches benchmark *(Balntas et al. 2017).* This dataset comprises 116 scenes, split into 696 images, with the first 57 scenes emphasizing variations in illumination and the subsequent 59 containing changes in viewpoint. Each sequence in the dataset comprises image pairs of increasing difficulty. We focus on this dataset, given that it represents a classical, longstanding benchmark for the task of keypoint detection, to assess the validity of our framework.

We present results for various maximum values of detected keypoints. These results are shown in Table 3.1, with metrics averaged across all thresholds up to 5 pixels. In each column, the best result is marked in bold text, and the second-best result is underlined.

Method	Illumination					Viewpoint				
	250	500	1000	2000	4000	250	500	1000	2000	4000
D2-Net	21.1	22.0	23.6	26.4	28.7	12.1	13.6	19.5	18.6	22.1
R2D2	27.3	28.6	29.8	30.5	30.7	24.3	25.5	26.5	27.6	28.3
SIFT	34.9	37.2	38.8	40.4	41.2	37.8	38.9	39.9	40.7	40.4
SuperPoint	42.4	**47.7**	49.8	49.5	49.4	27.5	36.0	43.6	**46.8**	46.4
DISK	42.2	45.9	49.8	**54.2**	**57.4**	30.6	35.0	39.3	44.0	**47.6**
ALIKED	14.8	24.4	37.3	47.0	51.9	6.5	10.6	18.1	29.7	43.1
MorseDet (ours)	**44.3**	47.3	**50.3**	53.4	55.2	**40.6**	**42.8**	**44.6**	46.1	47.2

Table 3.1: HPatches repeatability.

We can see that MorseDet achieves consistently good performance, regardless of the number of keypoints or settings (i.e., illumination and viewpoint changes), being either best or second-best across the table.

Some other methods perform competitively with MorseDet under specific settings, although none is competitive in all cases. Notably, DISK shows strong results with a high number of keypoints, and SIFT is second-best with fewer keypoints under viewpoint changes but performs poorly under illumination changes. On average, SuperPoint is second-best.

3.3.2 Scale Shift

We posit that models employing a fixed-size window approach for keypoint modeling during training learn to predict keypoints at a specific frequency. Building on this premise, such models may struggle to consistently replicate keypoints under rescaling transformations. To study this idea in isolation, we designed the following experiment using the images of HPatches. We evaluated for every method the repeatability metric between every image resized to 1000×1000, and the image resized to smaller sizes to have approximately 75%, 50%, and 25% the pixel area of the original image. As the number of keypoints deeply influences repeatability, we limit keypoints to 500, to ensure that every method uses the same number of keypoints at every scale for fair comparisons, thus also measuring how the methods can prioritize their most robust keypoints. The metrics are summarized in the tab. 3.2 by their average above all the thresholds till 5px.

The results show that MorseDet obtains second-best results on average after SIFT. In particular, MorseDet shines with 75% image resize (i.e. to images of 750×750), outperforming the second best method, SIFT, by 6.3 points. For extreme scale changes (i.e., 25% of the original resolution), the best model is SIFT,

Method	Avg	75%	50%	25%
D2-Net	24.6	31.9	19.2	22.8
R2D2	48.5	55.7	56.2	33.7
SIFT	**63.6**	75.9	**64.8**	**50.2**
SuperPoint	60.6	73.3	63.0	45.6
DISK	56.0	71.8	57.4	38.8
ALIKED	18.7	24.2	16.5	15.4
MorseDet (ours)	62.2	**82.2**	63.0	41.3

Table 3.2: Scale shift repeatability.

which is a handcrafted detector built to be scale-invariant, followed by SuperPoint and MorseDet. Overall, the only learnable model competitive with MorseDet is SuperPoint, which benefits from a human-informed prior on keypoints (see Appendix C). Notably, despite SIFT being proposed nearly two decades ago, it still outperforms modern detectors in this setup; MorseDet performs significantly better than every other learnable method in this task. This is a direct consequence of the fact that previous learnable methods lack a principled framework for modeling local maxima, which is our core contribution.

3.3.3 Qualitative Results

Fig. 3.6 shows, in order: (1) the height map of MorseDet; (2) the keypoints detected by MorseDet; (3) the repeatability map produced by R2D2; and (4) the keypoints detected by R2D2.

The repeatability map of R2D2 shows a bias towards detecting keypoints at a fixed resolution, resulting in the exclusion of some features and the creation of artifacts, especially along edges and in untextured areas. In contrast, MorseDet adapts its keypoints to the image content, effectively detecting both large-scale corners and fine-grained details without creating artifacts in low-textured regions.

This comparison demonstrates the validity of the topological formulation, highlighting the limitations of a fixed-size sliding window relaxation.

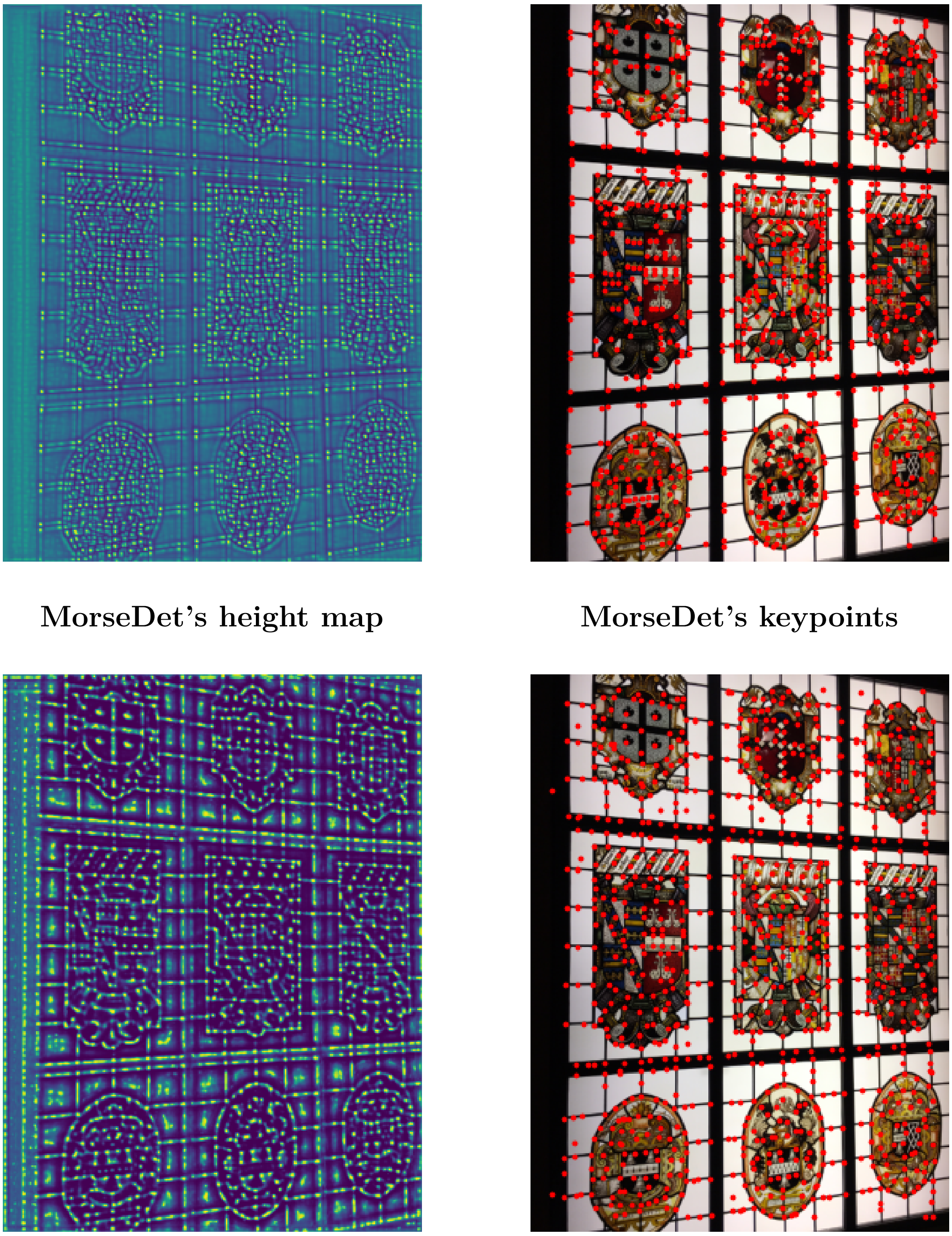
3.4 Conclusions

In this chapter, we introduced an application of algebraic topology to the field of computer vision, specifically leveraging a topological characterization of the concept of local maxima to model scale-agnostic keypoints in the context of image matching.

In these final remarks, we want to emphasize some innovative features of our approach. MorseDet, our model, is the first method in the deep learning literature to operate with a training objective purely defined on topological quantities, without relying on other loss functions. Indeed, we can say that MorseDet is the first topology-based learning model for feature detection.

We have already demonstrated some advantages of this approach, such as the ability to model a set of features in an unsupervised manner, independent of their cardinality or scale within the input image. In a certain sense, through topology, we have managed to define a more high-level loss function, free from scale parameters that are necessary for previous methods but not inherent to the problem.

However, the use of persistent homology in differentiable deep learning applications to this extent, being novel, presents numerous challenges. As we have seen, for example with the concerns expressed at the end of Section 3.2.1, many



MorseDet's height map

MorseDet's keypoints

R2D2's repeatability map

R2D2's keypoints

Figure 3.6: Qualitative results.

new questions arise. Further efforts will be necessary to clearly define the fields of application, the potential, and the limitations of this approach.

Appendix A

Topology

In this section, we provide some basic definitions and notions of topology that are assumed to be familiar in our discussion, particularly for the treatment of complexes.

Definition A.0.1. *(Topological space) A topological space is a set X together with a collection $\mathcal{T}$ of subsets of X satisfying the following conditions:*

1. *$\emptyset \in \mathcal{T}$ and $X \in \mathcal{T}$.*
2. *The union of any collection of sets in $\mathcal{T}$ is also in $\mathcal{T}$.*
3. *The intersection of any finite number of sets in $\mathcal{T}$ is also in $\mathcal{T}$.*

The collection $\mathcal{T}$ is called a topology *on X, and the elements of $\mathcal{T}$ are called* open sets*.*

Definition A.0.2. *(Closed set) A subset $A \subseteq X$ is* closed *if its complement $X - A$ is open.*

Definition A.0.3. *(Base of a topology) A* base *(or* basis*) for a topology on a set X is a collection $\mathcal{B}$ of open sets in X such that:*

1. *Every open set in X can be written as a union of sets from $\mathcal{B}$.*
2. *For any $B_1, B_2 \in \mathcal{B}$ and any point $x \in B_1 \cap B_2$, there exists a $B_3 \in \mathcal{B}$ such that $x \in B_3 \subseteq B_1 \cap B_2$.*

Definition A.0.4. *(Subspace topology) Let $(X, \mathcal{T})$ be a topological space and let $Y \subseteq X$ be a subset of X. The subspace topology on Y, denoted by $\mathcal{T}_Y$, is defined as follows: a set $U \subseteq Y$ is open in Y (i.e., $U \in \mathcal{T}_Y$) if and only if there exists an open set $V \in \mathcal{T}$ such that $U = V \cap Y$.*

In other words, the open sets of the subspace topology on Y are precisely the intersections of open sets in X with the subset Y.

Definition A.0.5. *(Usual topology on $\mathbb{R}^n$) The* usual topology *on $\mathbb{R}^n$ is the topology induced by the Euclidean metric d, where $d(x, y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}$. In this topology,* open balls *$B_r(x) = \{y \in \mathbb{R}^n : d(x, y) < r\}$, with $r > 0$ and $x \in \mathbb{R}^n$, form a basis.*

Theorem A.0.6. *(Characterization of open sets in $\mathbb{R}^n$) A set $U \subset \mathbb{R}^n$ is open if and only if for every $x \in U$, there exists an open ball $B_r(x) \subset U$ containing x.*

Definition A.0.7. *(Boundary in $\mathbb{R}^n$) The* boundary *of a subset $A \subset \mathbb{R}^n$, denoted ∂A, is defined as the set of points $x \in \mathbb{R}^n$ such that every open neighborhood of x intersects both A and $\mathbb{R}^n - A$.*

Theorem A.0.8. *(Characterization of closed sets in $\mathbb{R}^n$) A set $F \subset \mathbb{R}^n$ is closed if and only if it contains its boundary, i.e., $\partial F \subset F$.*

Definition A.0.9. *(Dense set) A subset $A \subset \mathbb{R}^n$ is said to be* dense *in $\mathbb{R}^n$ if every open set in $\mathbb{R}^n$ contains at least one point of A, or equivalently, the closure of A is equal to $\mathbb{R}^n$.*

Definition A.0.10. *(Compact set) A subset $K \subset X$ is* compact *if every open cover of K has a finite subcover. That is, if for every collection of open sets $\{U_\alpha\}_{\alpha \in I}$ such that $K \subset \bigcup_{\alpha \in I} U_\alpha$, there exists a finite subcollection $\{U_{\alpha_1}, \ldots, U_{\alpha_m}\}$ such that $K \subset \bigcup_{i=1}^{m} U_{\alpha_i}$.*

Theorem A.0.11. *(Characterization of compact sets in $\mathbb{R}^n$) A subset $K \subset \mathbb{R}^n$ is compact if and only if it is closed and bounded.*

Definition A.0.12. *(Continuous function) Let X and Y be topological spaces. A function $f : X \to Y$ is* continuous *if the preimage of every open set in Y is open in X. That is, for every open set $V \subset Y$, we have $f^{-1}(V) \subset X$ is open.*

Definition A.0.13. *(Homeomorphism) A function $f : X \to Y$ between two topological spaces X and Y is a* homeomorphism *if f is continuous, bijective, and its inverse $f^{-1} : Y \to X$ is also continuous. If such a function exists, X and Y are said to be* homeomorphic*, meaning they are topologically equivalent.*

Definition A.0.14. *(Embedding) Let X and Y be topological spaces. A function $f : X \to Y$ is called an* embedding *if f is a homeomorphism onto its image $f(X)$, where $f(X)$ is endowed with the subspace topology from Y. In other words, f is an injective continuous map such that the inverse $f^{-1} : f(X) \to X$ is also continuous.*

Definition A.0.15. *(Topological manifold) A* topological manifold *of dimension n is a topological space M such that:*

1. *Every point in M has an open neighborhood homeomorphic to an open subset of $\mathbb{R}^n$.*

2. M is a Hausdorff space (any two distinct points have disjoint neighborhoods).

3. M has a countable basis (is second countable).

Theorem A.0.16. *The hyper-sphere $S^{n-1} = \{x \in \mathbb{R}^n : \|x\| = 1\}$ is an $(n-1)$-dimensional manifold.*

Theorem A.0.17. *The open ball $B^n = \{x \in \mathbb{R}^n : \|x\| < 1\}$ is an n-dimensional manifold.*

Definition A.0.18. *(Homotopy) Let X and Y be topological spaces, and let $f,g : X \to Y$ be continuous functions. A homotopy from f to g is a continuous map $H : X \times [0,1] \to Y$ such that $H(x,0) = f(x)$ and $H(x,1) = g(x)$ for all $x \in X$. We say that f and g are homotopic if such a homotopy exists, denoted by $f \sim g$. Homotopy is an equivalence relation on the set of continuous maps from X to Y.*

Definition A.0.19. *(Homotopy type) Let X and Y be topological spaces, X and Y are said to be homotopic if there exist continuous maps $f : X \to Y$ and $g : Y \to X$ such that $f \circ g \sim id_Y$ and $g \circ f \sim id_X$. In this case, we write $X \sim Y$.*

Definition A.0.20. *(Retraction) Let X be a topological space and $A \subseteq X$. A continuous map $r : X \to A$ is called a retraction if $r(a) = a$ for all $a \in A$, meaning that r restricted to A is the identity map on A. If such a map exists, it follows that X and A are homotopic.*

Theorem A.0.21. *There is no retraction from the closed unit ball $\mathcal{B}^n$ in $\mathbb{R}^n$ to its boundary, the sphere $\mathcal{S}^{n-1}$.*

Theorem A.0.22. *(Invariance of domain) Let U be an open subset of $\mathbb{R}^n$, and let $f : U \to \mathbb{R}^n$ be an injective continuous map. Then $f(U)$ is open in $\mathbb{R}^n$ and f is a homeomorphism onto its image.*

Theorem A.0.23. *(Corollary of invariance of domain) An open subset of $\mathbb{R}^n$ cannot be homeomorphic to an open subset of $\mathbb{R}^m$ if $n \neq m$.*

Definition A.0.24. *(Connected set) A subset $A \subseteq \mathbb{R}^n$ is said to be connected if it cannot be partitioned into two non-empty disjoint open subsets in the subspace topology on A. In other words, there do not exist two open sets $U, V \subseteq \mathbb{R}^n$ such that $A \cap U$ and $A \cap V$ are both non-empty, disjoint, and satisfy $A = (A \cap U) \cup (A \cap V)$.*

Definition A.0.25. *(Connected component) A connected component of a subset $A \subseteq \mathbb{R}^n$ is a maximal connected subset of A. This means that a connected component $C \subset A$ is connected, and there is no larger connected subset $D \subset A$ containing C as a subset. Every point in A lies in exactly one connected component of A.*

Definition A.0.26. *(Loop) In a topological space X, a loop at a point $x_0 \in X$ is a continuous map $\gamma : [0,1] \to X$ such that $\gamma(0) = \gamma(1) = x_0$. A loop can be visualized as a path that starts and ends at the same point.*

Definition A.0.27. *(Simply connected manifold) A topological space X is said to be simply connected if it is path-connected and every loop in X can be continuously deformed (is homotopic) to a single point within X. In particular, a simply connected manifold has no "holes" that would prevent a loop from contracting to a point.*

Appendix B

Algebra

In this section, we summarize key concepts from abstract algebra necessary for our discussion on homology and persistent homology.

Definition B.0.1. *(Field) A field is a set F equipped with two operations, addition and multiplication, such that F forms an abelian group under addition, the nonzero elements of F form an abelian group under multiplication, and multiplication distributes over addition. Two commonly used fields are the real numbers $\mathbb{R}$ and the finite field $\mathbb{F}_2 = \{0, 1\}$ with addition and multiplication modulo 2.*

Definition B.0.2. *(Module) Let R be a ring. An R-module is a set M together with an addition operation and a scalar multiplication by elements of R, such that M is an abelian group under addition and scalar multiplication is distributive over both module addition and ring addition, and associative with ring multiplication. In this work, we primarily consider modules over fields, which are known as vector spaces.*

Definition B.0.3. *(Basis) A basis of an R-module M is a set of elements in M that are linearly independent and span M. If M has a basis, then M is said to be a free module, and the cardinality of any basis of M is called the dimension of M. For modules over fields (vector spaces), any two bases have the same cardinality.*

Definition B.0.4. *(Linear map) Let M and N be R-modules. A linear map (or homomorphism) from M to N is a function $f : M \to N$ that preserves addition and scalar multiplication, i.e., for all $x, y \in M$ and $r \in R$, we have $f(x+y) = f(x)+f(y)$ and $f(rx) = rf(x)$.*

Definition B.0.5. *(Kernel and image) The kernel of a linear map $f : M \to N$ is the set of elements in M that are mapped to zero in N, denoted by $\texttt{ker}(f) = \{x \in M \mid f(x) = 0\}$. The image of f is the set of elements in N that are the images of elements of M, denoted by $\texttt{Im}(f) = \{f(x) \mid x \in M\}$. Both the kernel and image of a linear map are submodules of M and N, respectively.*

Theorem B.0.6. *(Rank-nullity) Let M and N be R-modules, and let $f : M \to N$ be a linear map. If M is finite-dimensional, then we have the following dimensionality result:*

$$\mathtt{dim}(M) = \mathtt{dim}(\mathtt{ker}(f)) + \mathtt{dim}(\mathtt{Im}(f))$$

Definition B.0.7. *(Coset) Let $f : M \to N$ be a linear map between R-modules, and let $\mathtt{ker}(f)$ be the kernel of f. A coset of $\mathtt{ker}(f)$ in M is defined as $\{x + y \mid y \in \mathtt{ker}(f)\}$ for a fixed $x \in M$, and is denoted by $x + \mathtt{ker}(f)$. The cosets form a partition of M into equivalence classes on which f has the same value.*

Definition B.0.8. *(Quotient module) Let $f : M \to N$ be a linear map between R-modules, and let $\mathtt{ker}(f)$ be the kernel of f. The quotient module $M/\mathtt{ker}(f)$ is the set of cosets of $\mathtt{ker}(f)$ in M, with the module operations defined by $(x + \mathtt{ker}(f)) + (y + \mathtt{ker}(f)) = (x + y) + \mathtt{ker}(f)$ and $r(x + \mathtt{ker}(f)) = (rx) + \mathtt{ker}(f)$ for all $x, y \in M$ and $r \in R$. The map f induces a well-defined injective linear map from $M/\mathtt{ker}(f)$ to $\mathtt{Im}(f)$.*

Theorem B.0.9. *Given a quotient R-module $M/\mathtt{ker}(f)$, its dimension is given by $\mathtt{dim}(M/\mathtt{ker}(f)) = \mathtt{dim}(M) - \mathtt{dim}(\mathtt{ker}(f))$.*

Appendix C

Baselines

In this section, we explain the design of the baselines considered in our experiments that have not been discussed in the main document. All the presented models are deep learning methods that utilize convolutional neural networks to extract a set of keypoints and their descriptors from an image.

SuperPoint

SuperPoint *(DeTone et al. 2018)* is based on a fully convolutional encoder-decoder network with shared layers that split into two distinct branches: one for detecting keypoints and the other for computing descriptors.

The model training process employs a two-step approach called "homographic adaptation." Initially, SuperPoint is trained on synthetic images of simple geometric shapes, where keypoints are defined, such as intersections of edges. In the second phase, SuperPoint is fine-tuned on real images, enforcing the detected keypoints to be equivariant to homographic transformations. In practice, SuperPoint uses self-supervised learning to generalize a notion of keypoints originally defined on a simple subset of geometric shapes to natural images.

At inference time, the network outputs a heatmap where prominent local maxima represent keypoints. These keypoints are filtered using non-maximum suppression to ensure spatial separation, resulting in a set of distinct and stable detections.

D2-Net

D2-Net *(Dusmanu et al. 2019)* is a deep learning model designed to jointly learn keypoint detection and descriptor extraction by training on dense pixel-wise correspondences between images. Its loss function encourages both repeatability and distinctiveness of keypoints and descriptors.

At training time, D2-Net uses a triplet margin loss to align descriptors for corresponding pixels across image pairs while separating descriptors for non-corresponding pixels. For each pixel in an image, the model learns to minimize the distance in the descriptor space to its corresponding location in the paired image (positive match) and to maximize the distance to a set of non-corresponding locations (negative matches). This loss formulation ensures that descriptors for matching keypoints are close in feature space, while those for non-matching pixel are pushed apart, up to a specified margin.

At inference time, D2-Net processes an input image to produce dense feature maps, identifying keypoints as local maxima within these maps. Descriptors are then directly extracted from the feature maps at each detected keypoint location.

DISK

DISK *(Tyszkiewicz et al. 2020)* is a local feature extractor based on reinforcement learning. Recognizing that casting keypoint detection and matching in a differentiable manner suitable for optimization is a notoriously difficult problem without an obvious solution, DISK proposes a formulation that allows exact gradients to be computed using policy gradient techniques.

During training, DISK utilizes a CNN to process input images and generate keypoint heatmaps along with dense descriptors. Heatmaps are normalized across $N \times N$ windows to produce a keypoint probability distribution within image patches. Keypoints are sampled from these heatmaps, and their corresponding descriptors are used to establish a distribution over potential feature matches between image pairs. The method applies reinforcement learning principles to maximize the expected reward associated with correct feature matches.

At inference time, DISK processes new images through the CNN to produce keypoint heatmaps and dense descriptors. Keypoints are then sampled from the heatmaps or deterministically selected using non-maximum suppression, and their descriptors are matched across images to identify correspondences. DISK has demonstrated state-of-the-art performance on various public benchmarks, highlighting its effectiveness in local feature learning.

ALIKED

ALIKED *(Zhao et al. 2023)* is a deep learning framework that has achieved state-of-the-art performance both in terms of accuracy and computational efficiency by incorporating several innovations in keypoint detection and descriptor computation.

During training, ALIKED computes a keypoint score scalar feature map. Keypoints are selected with sub-pixel accuracy in a partially differentiable manner. Within each $N \times N$ image patch, a keypoint is identified as the position of the local maxima within the patch, combined with a weighted sum of all other pixel locations in the patch based on their normalized scores. While the first component is not differentiable, the second component can be optimized during training.

For descriptor computation, ALIKED employs deformable convolutional layers. These layers use differentiable offsets to adapt the supporting features on which the descriptors are computed, making them robust to various geometric transformations.

At inference time, ALIKED processes new images through the CNN to produce a score map and an aggregated feature map.

Bibliography

Ghrist, Robert W (2014). *Elementary applied topology.* Vol. 1. Createspace Seattle (cit. on pp. 1, 32).

Edelsbrunner, Herbert and John L Harer (2022). *Computational topology: an introduction.* American Mathematical Society (cit. on pp. 1, 5, 6, 14, 15).

Milnor, John Willard (1963). *Morse theory.* 51. Princeton university press (cit. on pp. 1, 13, 15).

Forman, Robin (2002). «A user's guide to discrete Morse theory.» In: *Séminaire Lotharingien de Combinatoire [electronic only]* 48, B48c–35 (cit. on pp. 1, 16–19).

Hatcher, Allen (2005). *Algebraic topology* (cit. on pp. 2, 6).

Gabriel, Peter (1972). «Unzerlegbare darstellungen I». In: *Manuscripta mathematica* 6, pp. 71–103 (cit. on p. 11).

Derksen, Harm and Jerzy Weyman (2005). «Quiver representations». In: *Notices of the AMS* 52.2, pp. 200–206 (cit. on p. 11).

Botnan, Magnus and William Crawley-Boevey (2020). «Decomposition of persistence modules». In: *Proceedings of the American Mathematical Society* 148.11, pp. 4581–4596 (cit. on p. 11).

Guillemin, Victor and Alan Pollack (2010). *Differential topology.* Vol. 370. American Mathematical Soc. (cit. on p. 13).

Robins, Vanessa, Peter John Wood, and Adrian P Sheppard (2011). «Theory and algorithms for constructing discrete Morse complexes from grayscale digital images». In: *IEEE Transactions on pattern analysis and machine intelligence* 33.8, pp. 1646–1658 (cit. on pp. 16, 20–24).

Lingareddy (2018). «Calculating persistent homology using discrete Morse theory». In: (cit. on p. 16).

King, Henry, Kevin Knudson, and Neža Mramor (2005). «Generating discrete Morse functions from point data». In: *Experimental Mathematics* 14.4, pp. 435–444 (cit. on p. 19).

Barbarani, Giovanni, Mohamad Mostafa, Hajali Bayramov, Gabriele Trivigno, Gabriele Berton, Carlo Masone, and Barbara Caputo (2023). «Are local features all you need for cross-domain visual place recognition?» In: *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6155–6165 (cit. on p. 27).

Richter-Gebert, Jürgen (2011). *Perspectives on projective geometry: a guided tour through real and complex geometry.* Springer (cit. on pp. 27, 33).

Fusiello, Andrea (2024). *Computer Vision: Three-Dimensional Reconstruction Techniques.* Springer (cit. on pp. 27, 35).

Lindeberg, Tony (2013). *Scale-space theory in computer vision.* Vol. 256. Springer Science & Business Media (cit. on pp. 28, 41).

Lowe, David G (2004). «Distinctive image features from scale-invariant keypoints». In: *International journal of computer vision* 60, pp. 91–110 (cit. on pp. 28, 42, 43).

Revaud, Jerome, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel (2019). «R2d2: Reliable and repeatable detector and descriptor». In: *Advances in neural information processing systems* 32 (cit. on pp. 28, 43, 48, 53).

Fischler, Martin A and Robert C Bolles (1981). «Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography». In: *Communications of the ACM* 24.6, pp. 381–395 (cit. on p. 38).

Ma, Jiayi, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan (2021). «Image matching from handcrafted to deep features: A survey». In: *International Journal of Computer Vision* 129.1, pp. 23–79 (cit. on p. 39).

Mikolajczyk, Krystian and Cordelia Schmid (2005). «A performance evaluation of local descriptors». In: *IEEE transactions on pattern analysis and machine intelligence* 27.10, pp. 1615–1630 (cit. on pp. 40, 53).

Jin, Yuhe, Dmytro Mishkin, Anastasiia Mishchuk, Jiri Matas, Pascal Fua, Kwang Moo Yi, and Eduard Trulls (2021). «Image matching across wide baselines: From paper to practice». In: *International Journal of Computer Vision* 129.2, pp. 517–547 (cit. on p. 40).

Yuille, Alan L and Tomaso A Poggio (1986). «Scaling theorems for zero crossings». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1, pp. 15–25 (cit. on p. 42).

Barbarani, Giovanni, Francesco Vaccarino, Gabriele Trivigno, Marco Guerra, Gabriele Berton, and Carlo Masone (n.d.). «Scale-Free Image Keypoints Using Differentiable Persistent Homology». In: *Forty-first International Conference on Machine Learning* (cit. on p. 45).

Leygonie, Jacob, Steve Oudot, and Ulrike Tillmann (2021). «A framework for differential calculus on persistence barcodes». In: *Foundations of Computational Mathematics*, pp. 1–63 (cit. on pp. 47, 52).

Carriere, Mathieu, Frédéric Chazal, Marc Glisse, Yuichi Ike, Hariprasad Kannan, and Yuhei Umeda (2021). «Optimizing persistent homology based functions». In: *International conference on machine learning.* PMLR, pp. 1294–1303 (cit. on pp. 47, 52).

Chen, Chao, Xiuyan Ni, Qinxun Bai, and Yusu Wang (2019). «A topological regularizer for classifiers via persistent homology». In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2573–2582 (cit. on p. 47).

Moor, Michael, Max Horn, Bastian Rieck, and Karsten Borgwardt (2020). «Topological autoencoders». In: *International conference on machine learning*. PMLR, pp. 7045–7054 (cit. on p. 47).

Hu, Xiaoling, Yusu Wang, Li Fuxin, Dimitris Samaras, and Chao Chen (2021). «Topology-Aware Segmentation Using Discrete Morse Theory». In: *International Conference on Learning Representations* (cit. on p. 47).

Gupta, Saumya, Yikai Zhang, Xiaoling Hu, Prateek Prasanna, and Chao Chen (2024). «Topology-aware uncertainty for image segmentation». In: *Advances in Neural Information Processing Systems* 36 (cit. on p. 47).

Tian, Yurun, Bin Fan, and Fuchao Wu (2017). «L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space». In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6128–6136 (cit. on p. 48).

DeTone, Daniel, Tomasz Malisiewicz, and Andrew Rabinovich (2018). «Superpoint: Self-supervised interest point detection and description». In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 224–236 (cit. on pp. 53, 54, 65).

Rey-Otero, Ives, Mauricio Delbracio, and Jean-Michel Morel (2015). «Comparing feature detectors: A bias in the repeatability criteria». In: *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 3024–3028 (cit. on p. 53).

Lenc, Karel and Andrea Vedaldi (2018). «Large scale evaluation of local image feature detectors on homography datasets». In: *BMVC* (cit. on p. 53).

Dusmanu, Mihai, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler (2019). «D2-Net: A Trainable CNN for Joint Detection and Description of Local Features». In: *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (cit. on pp. 54, 65).

Tyszkiewicz, Michał, Pascal Fua, and Eduard Trulls (2020). «DISK: Learning local features with policy gradient». In: *Advances in Neural Information Processing Systems* 33, pp. 14254–14265 (cit. on pp. 54, 66).

Zhao, Xiaoming, Xingming Wu, Jinyu Miao, Weihai Chen, Peter C. Y. Chen, and Zhengguo Li (2023). «ALIKE: Accurate and Lightweight Keypoint Detection and Descriptor Extraction». In: *IEEE Transactions on Multimedia* 25, pp. 3101–3112 (cit. on pp. 54, 66).

Balntas, Vassileios, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk (2017). «HPatches: A benchmark and evaluation of handcrafted and learned local descriptors». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5173–5182 (cit. on p. 54).