Fitting Surfaces to Polygonal Meshes using Parametric Pseudo-Manifolds

Tutorial 3

SIBGRAPI 2008

XXI BRAZILIAN SYMPOSIUM ON COMPUTER GRAPHICS AND IMAGE PROCESSING

CAMPO GRANDE/MS - BRAZIL

October 12-15, 2008

Instructors

Prof. Jean Gallier, Ph.D., 1978

Department of Computer and Information Science

University of Pennsylvania

Philadelphia, PA, USA

jean@cis.upenn.edu

http://www.cis.upenn.edu/~jean

Instructors

Prof. Dimas M. Morera, Dr., 2006

Instituto de Matemática

Universidade Federal de Alagoas

Maceió, AL, Brasil

dimasmm@gmail.com

http://www.impa.br/~dimasmm

Instructors

Prof. Gustavo Nonato, Dr., 1998

Instituto de Ciências Matemáticas e de Computação

Universidade de São Paulo

São Carlos, SP, Brasil

gnonato@icmc.usp.br

http://www.icmc.usp.br/~gnonato

Instructors

Prof. Marcelo Siqueira, Ph.D., 2006

Departamento de Computação e Estatística

Universidade Federal de Mato Grosso do Sul

Campo Grande, MS, Brasil

marcelo@dct.ufms.br

http://www.dct.ufms.br/~marcelo

Instructors

Prof. Luiz Velho, Ph.D., 1994

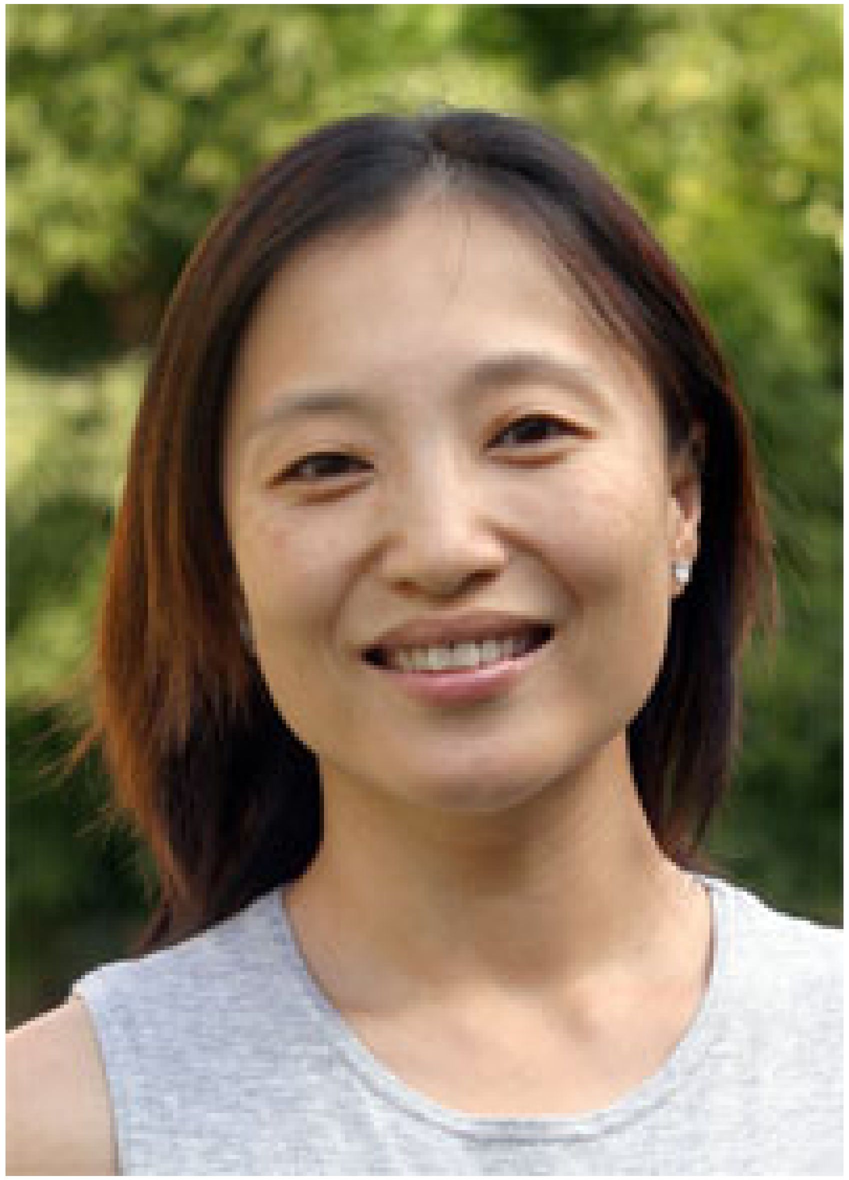
Instituto de Matemática Pura e Aplicada (IMPA)

Rio de Janeiro, RJ, Brasil

lvelho@impa.br

http://w3.impa.br/~lvelho/

Instructors



Prof. Dianna Xu, Ph.D., 2002

Computer Science Department

Bryn Mawr College

Bryn Mawr, PA, USA

dxu@cs.brynmawr.edu

http://www.cs.brynmawr.edu/~dxu

Introduction

Marcelo Siqueira
UFMS

Outline

- The Surface Fitting Problem
- Traditional Approaches
- The Manifold-Based Approach
- What's Next?

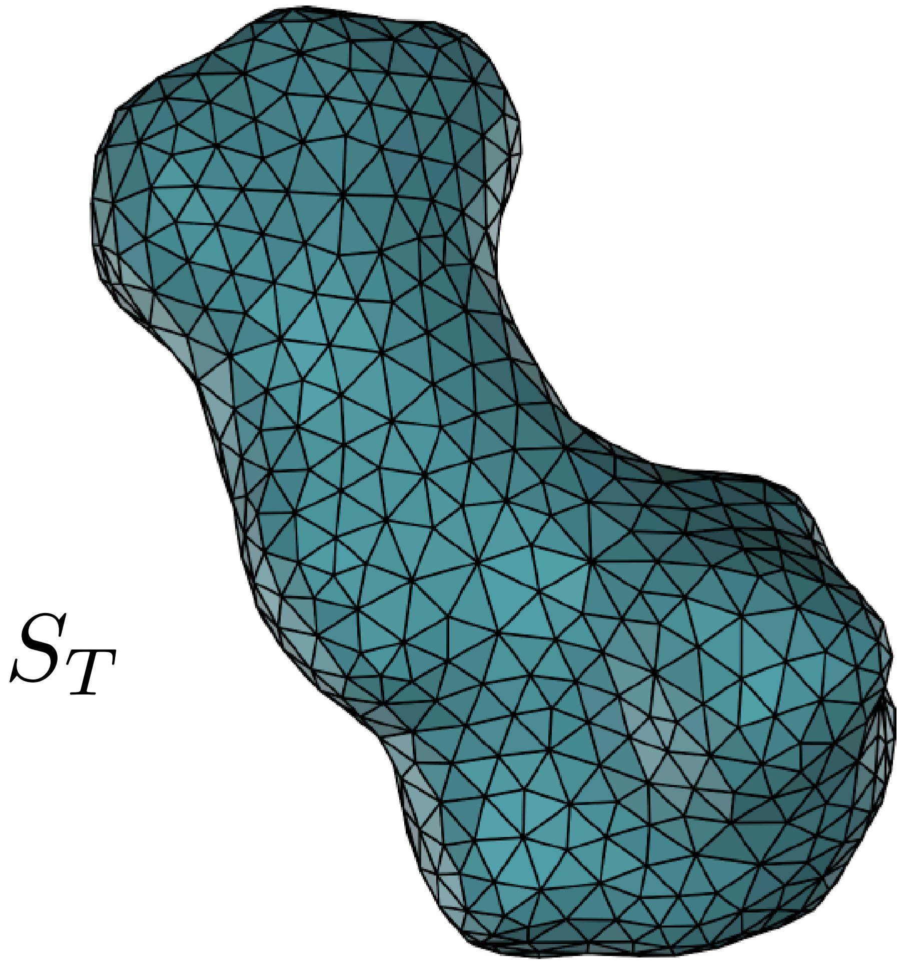
The Surface Fitting Problem

The Surface Fitting Problem

We are a given a piecewise-linear surface, S_T, in $\mathbb{R}^3$, with an empty boundary, a positive integer k, and a positive number ϵ, . . .

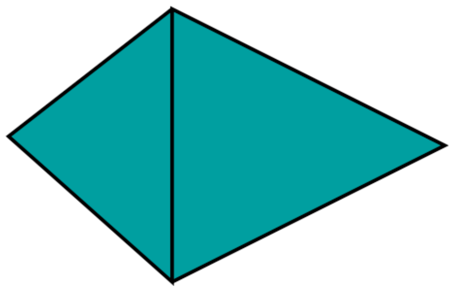
The Surface Fitting Problem

We are a given a piecewise-linear surface, S_T, in $\mathbb{R}^3$, with an empty boundary, a positive integer k, and a positive number ϵ, ...



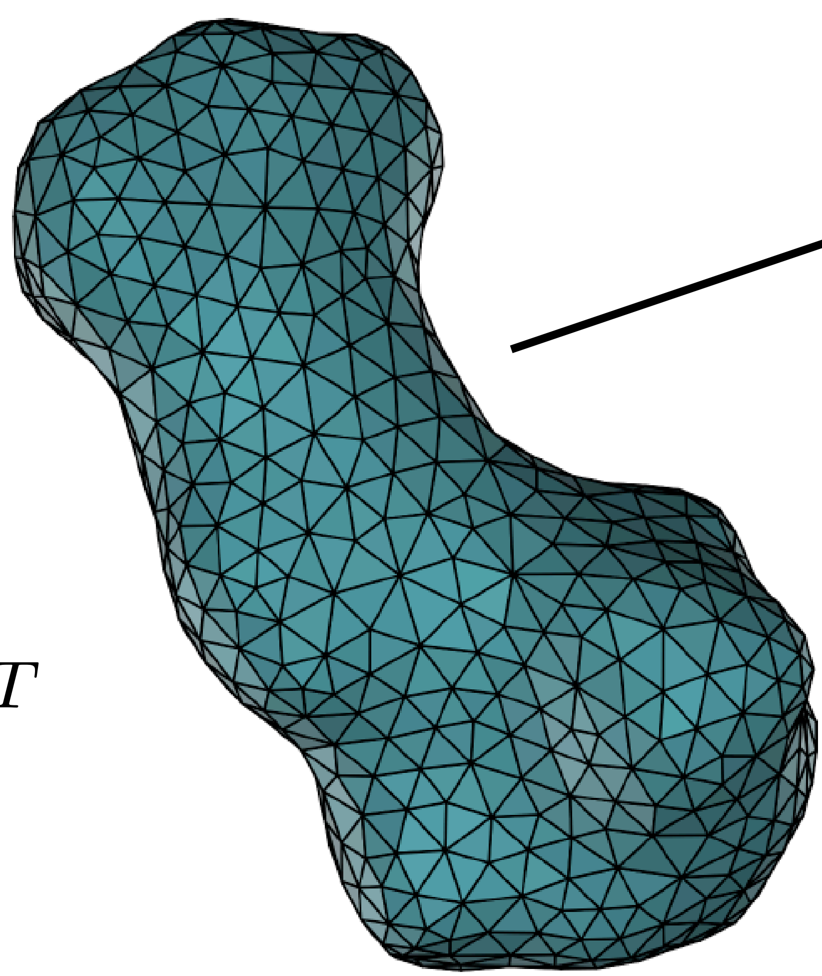
The Surface Fitting Problem

We are a given a piecewise-linear surface, S_T, in $\mathbb{R}^3$, with an empty boundary, a positive integer k, and a positive number ϵ, . . .



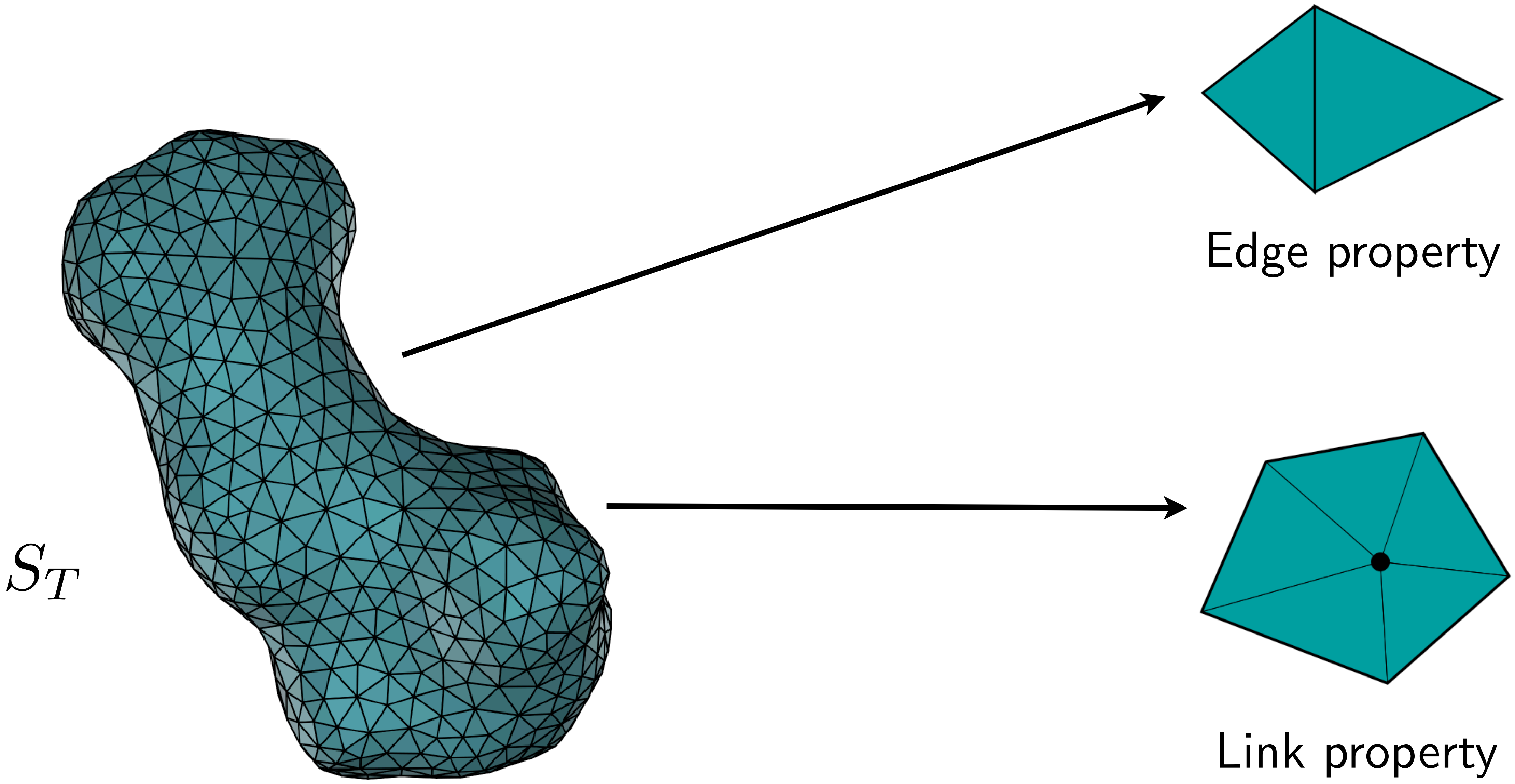
Edge property

S_T



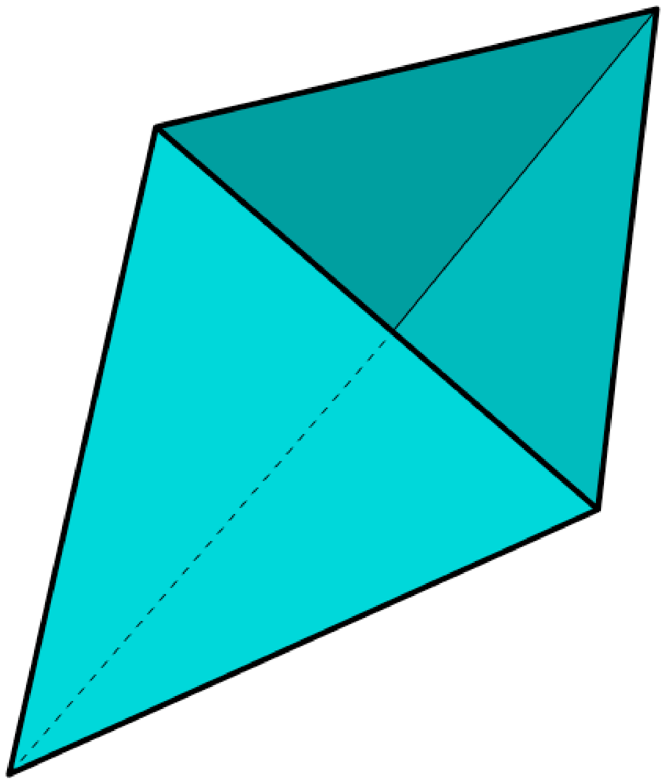
The Surface Fitting Problem

We are a given a piecewise-linear surface, S_T, in $\mathbb{R}^3$, with an empty boundary, a positive integer k, and a positive number ϵ, . . .

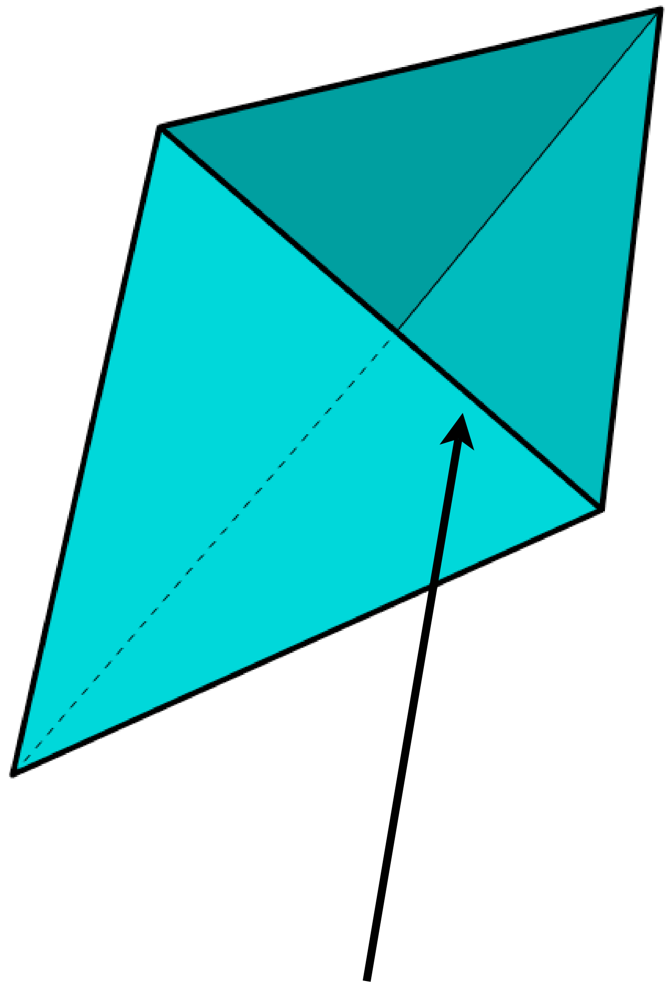


The Surface Fitting Problem

The Surface Fitting Problem

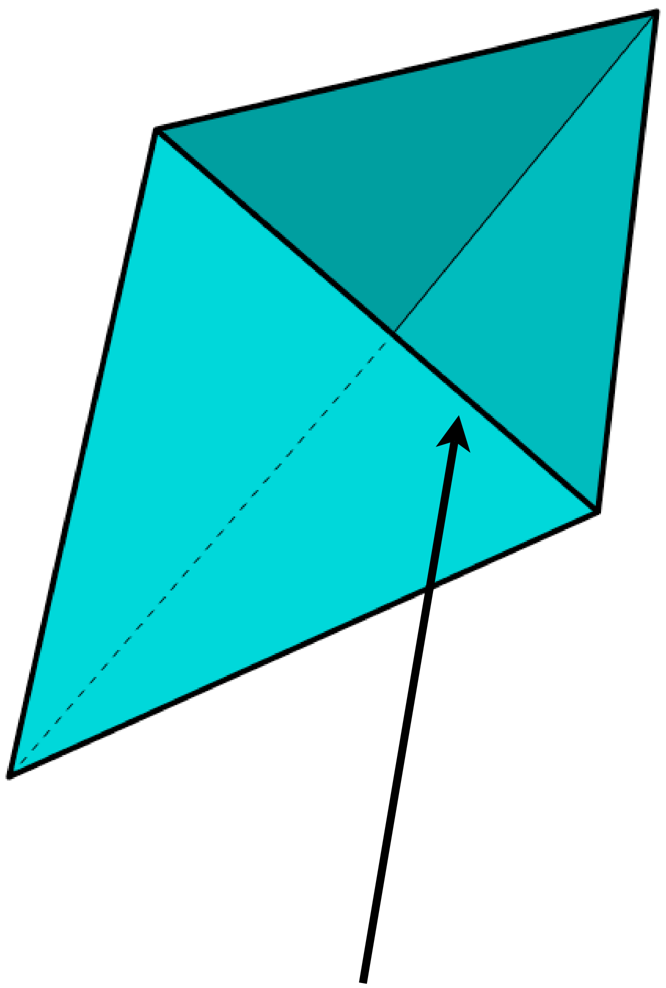


The Surface Fitting Problem

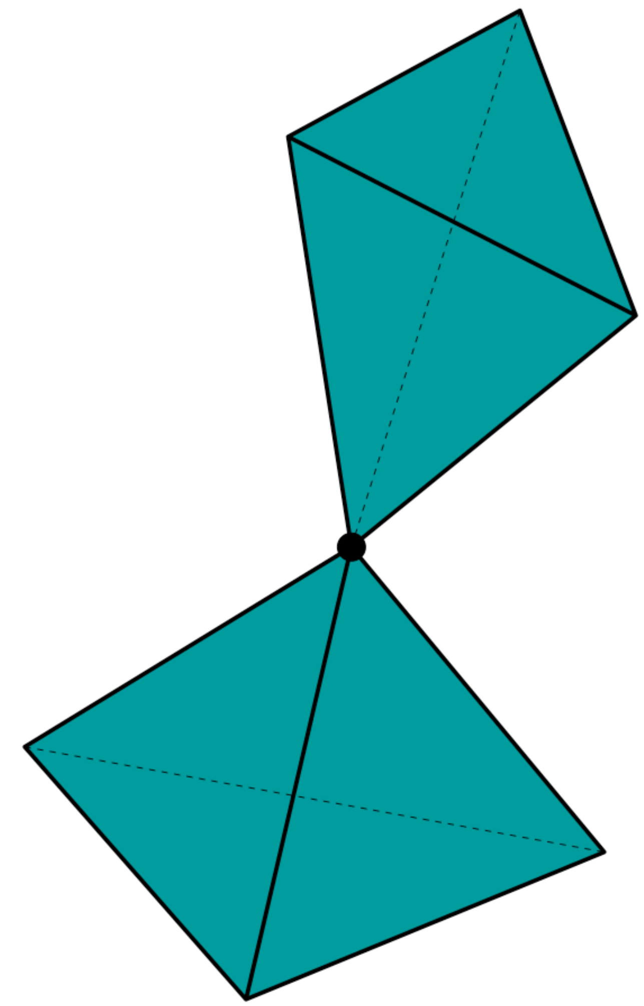


Violates edge property!

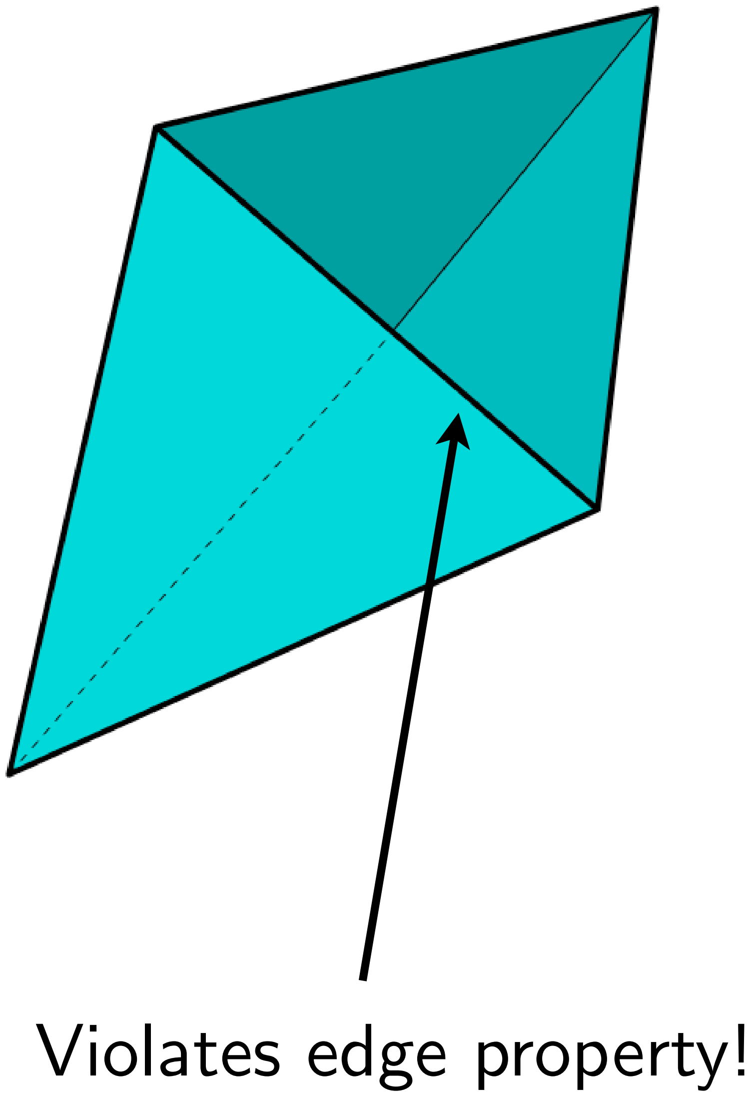
The Surface Fitting Problem

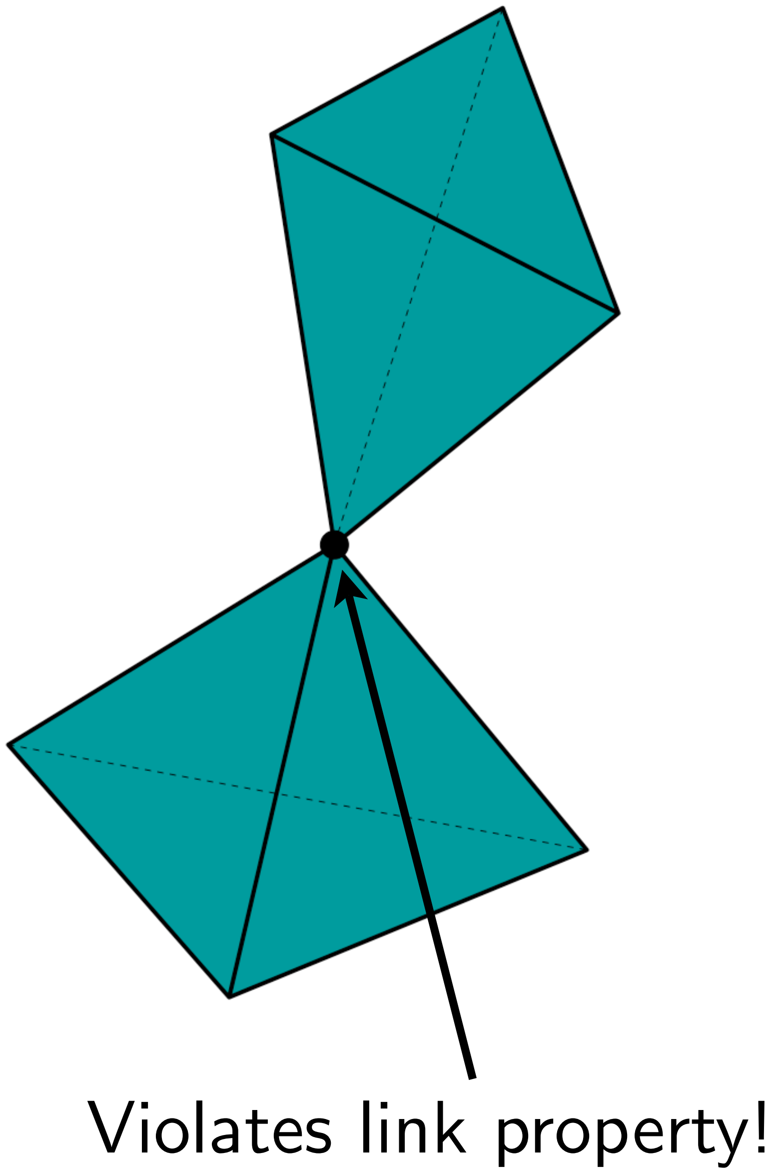


Violates edge property!

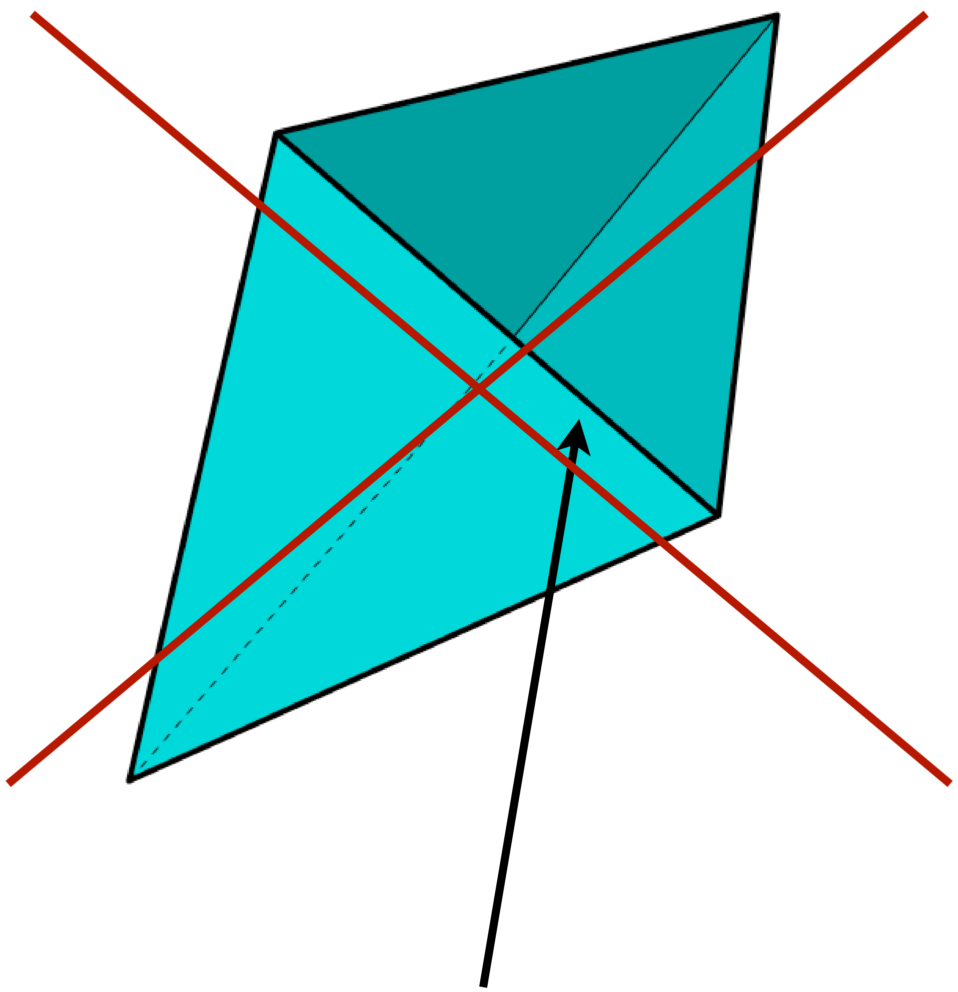


The Surface Fitting Problem

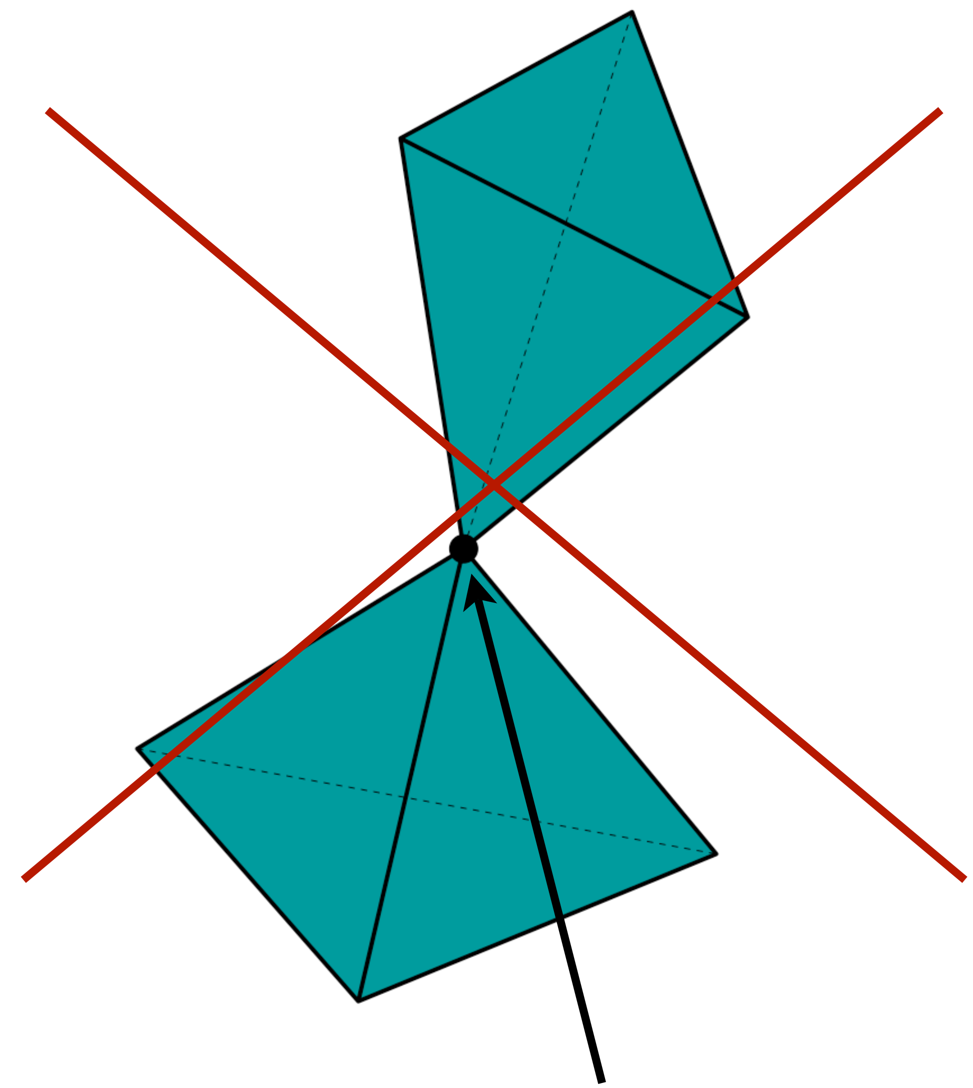




The Surface Fitting Problem



Violates edge property!



Violates link property!

They are NOT piecewise-linear surfaces

The Surface Fitting Problem

The Surface Fitting Problem

We want to find a C^k surface $S \subset \mathbb{R}^3$ ...

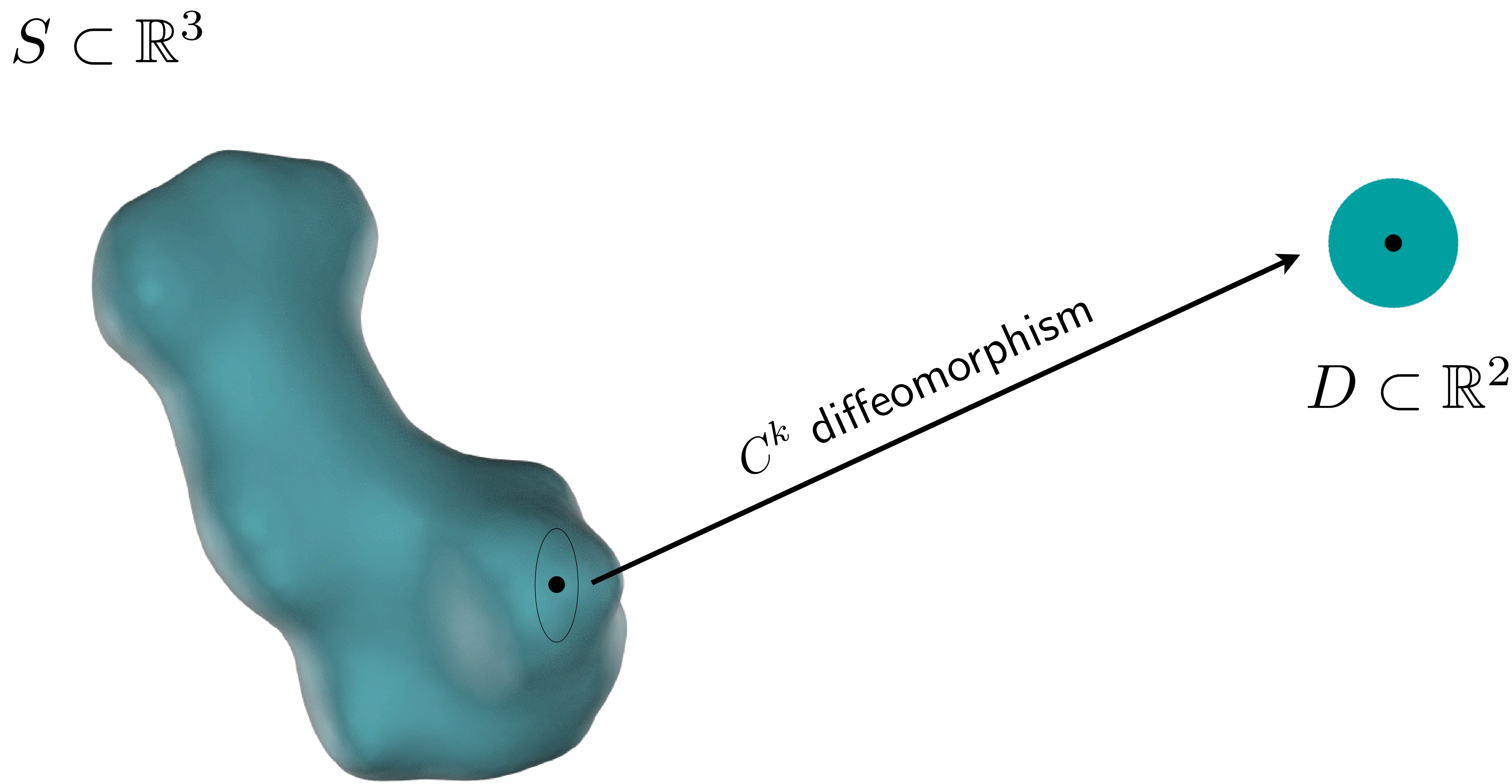
The Surface Fitting Problem

We want to find a C^k surface $S \subset \mathbb{R}^3$ ...

$S \subset \mathbb{R}^3$

The Surface Fitting Problem

We want to find a C^k surface $S \subset \mathbb{R}^3$ ...



The Surface Fitting Problem

The Surface Fitting Problem

such that there exists a homeomorphism, $h : S \to |S_T|$, satisfying

$$\|h(v) - v\| \leq \epsilon \, ,$$

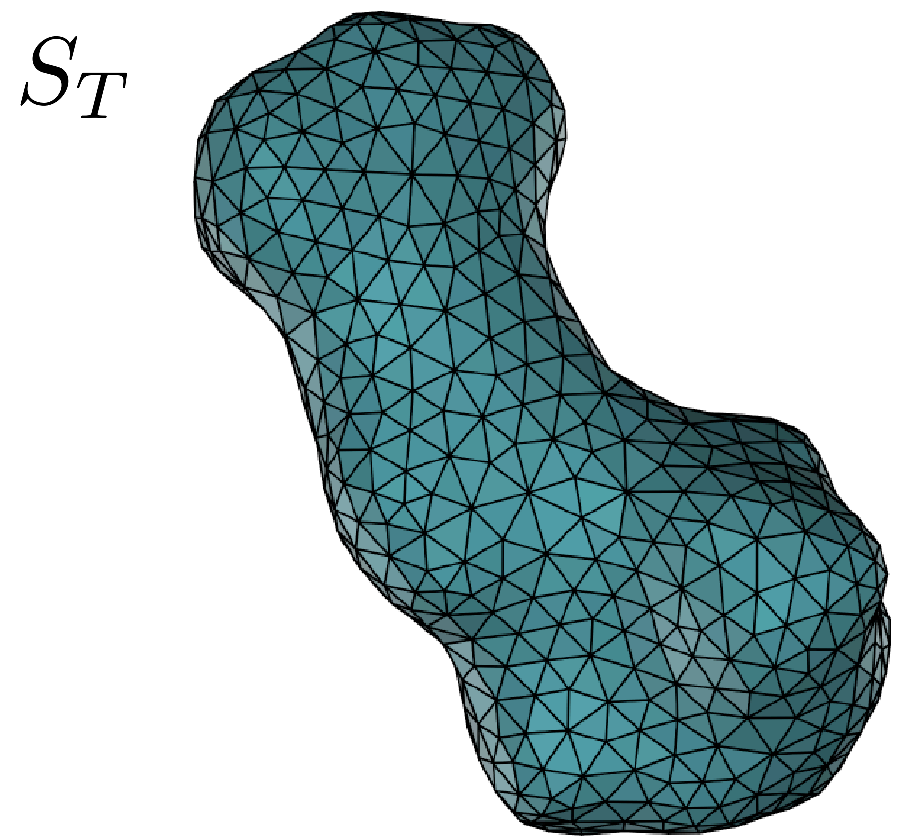
for every vertex v of S_T.

The Surface Fitting Problem

such that there exists a homeomorphism, $h : S \to |S_T|$, satisfying

$$\|h(v) - v\| \leq \epsilon \,,$$

for every vertex v of S_T.

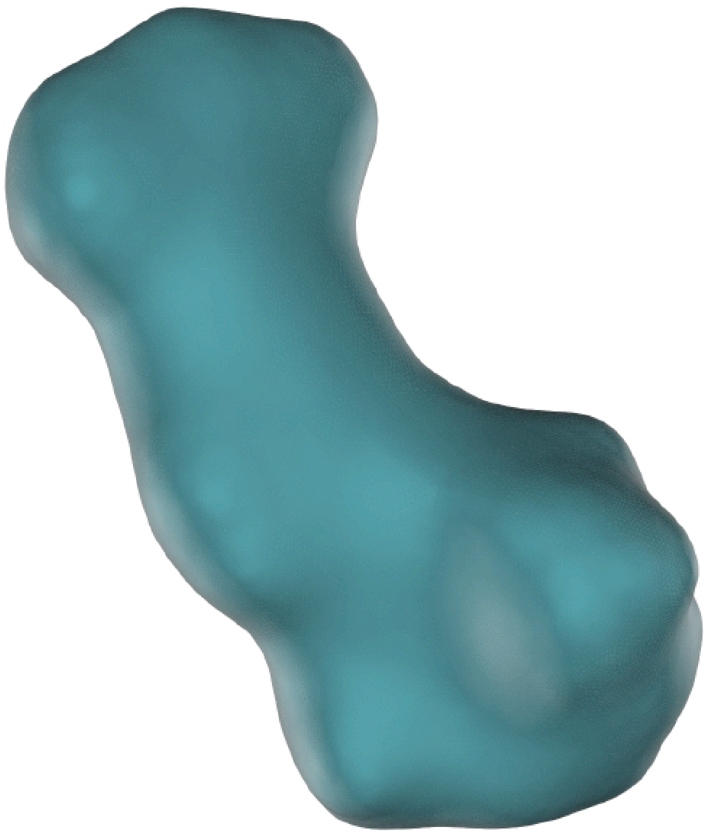


The Surface Fitting Problem

such that there exists a homeomorphism, $h : S \to |S_T|$, satisfying

$$\|h(v) - v\| \leq \epsilon \, ,$$

for every vertex v of S_T.

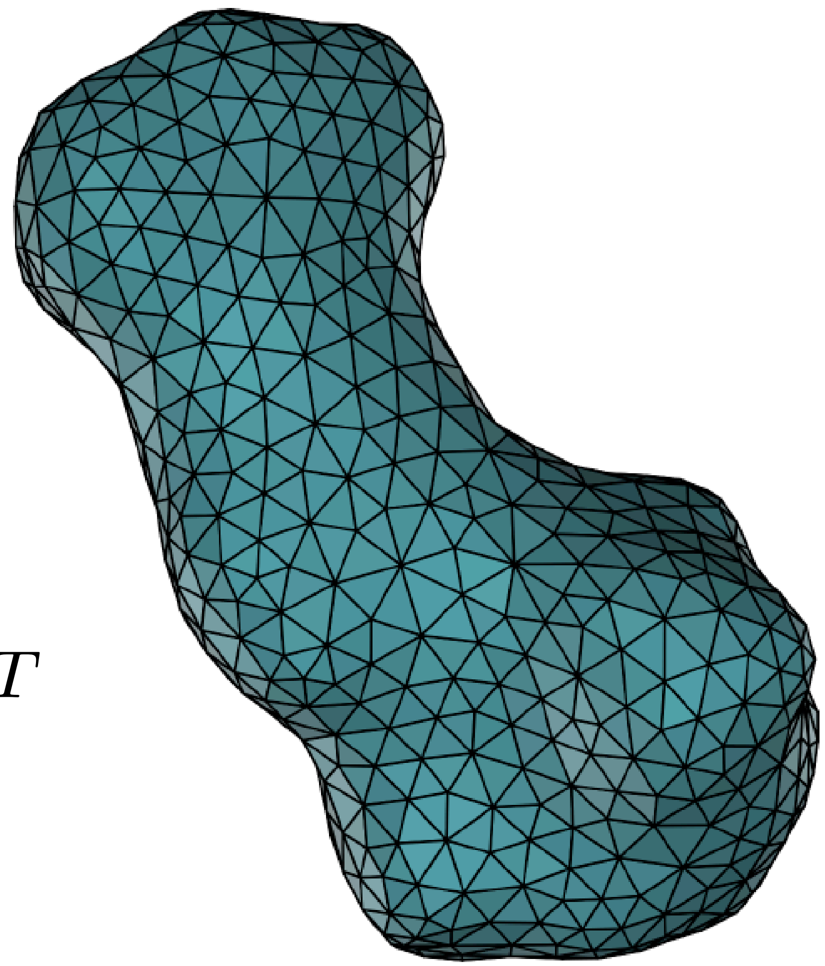


Topological and geometric guarantees!

The Surface Fitting Problem

The Surface Fitting Problem

From now on, we will refer to S_T as a **polygonal mesh**.



S_T

The Surface Fitting Problem

The Surface Fitting Problem

- It is a well-known and fundamental problem in CAGD.

The Surface Fitting Problem

- It is a well-known and fundamental problem in CAGD.
- Reasonably well-solved for $k = 1, 2$, **but not higher**.

The Surface Fitting Problem

- It is a well-known and fundamental problem in CAGD.
- Reasonably well-solved for $k = 1, 2$, **but not higher**.
- Higher values of k are desirable in many applications.

Traditional Approaches

Traditional Approaches

The most popular approach is certainly the parametric surface one.

Traditional Approaches

The most popular approach is certainly the parametric surface one.

Key idea:

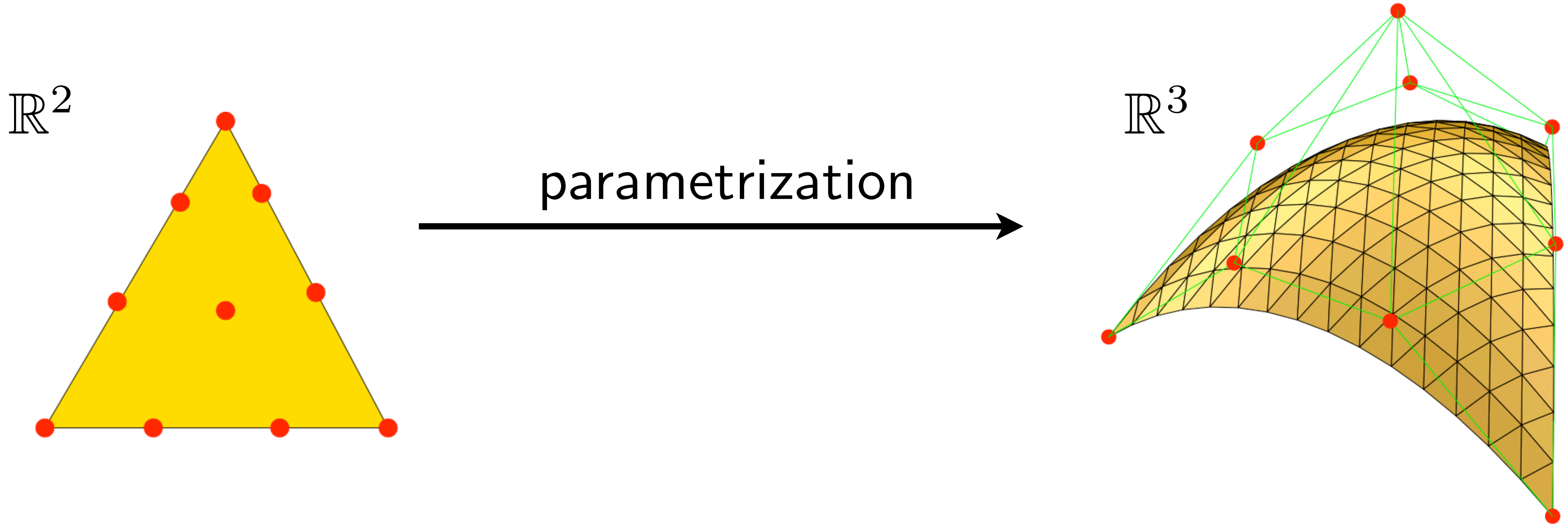
- Assign a parametric patch to each triangle of S_T.

Traditional Approaches

The most popular approach is certainly the parametric surface one.

Key idea:

- Assign a parametric patch to each triangle of S_T.



Traditional Approaches

Traditional Approaches

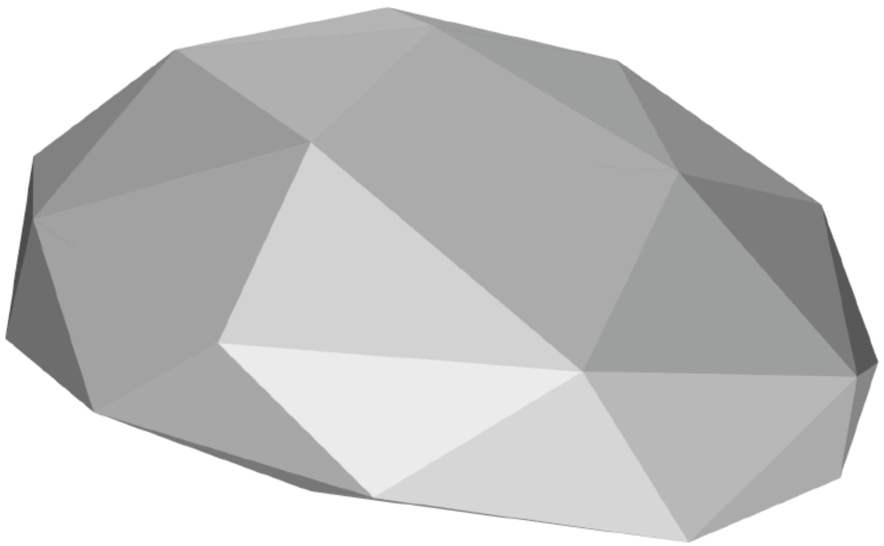
and

- stitch the patches together along their common edges and vertices.

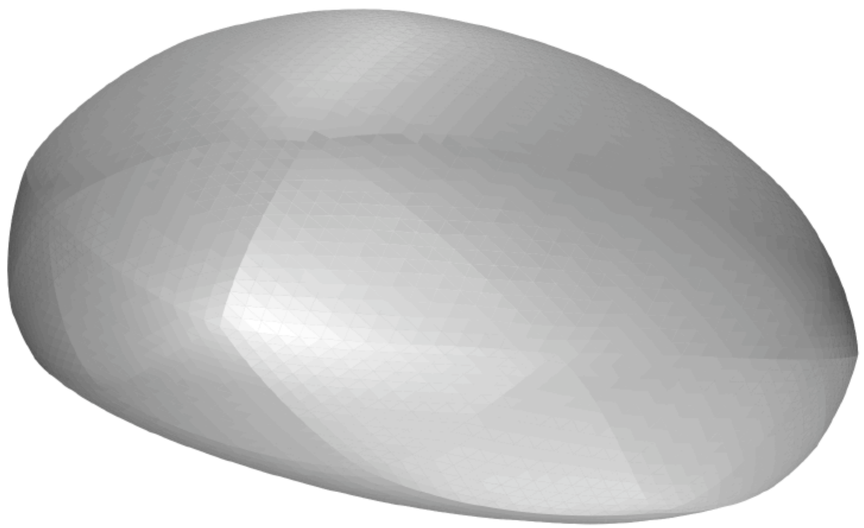
Traditional Approaches

and

- stitch the patches together along their common edges and vertices.



S_T



S

Continuity is enforced by control point placement!

Traditional Approaches

Traditional Approaches

There are several drawbacks with this approach:

- The degree d of the patches depends on k and grows rapidly with it.

Traditional Approaches

There are several drawbacks with this approach:

- The degree d of the patches depends on k and grows rapidly with it.
- Large values of d yield surfaces of poor visual quality.

Traditional Approaches

There are several drawbacks with this approach:

- The degree d of the patches depends on k and grows rapidly with it.

- Large values of d yield surfaces of poor visual quality.

- The larger d is, the larger the number of control points.

Traditional Approaches

Traditional Approaches

- The larger d is, the larger the number of control points and the more difficult the problem of control point placement.

Traditional Approaches

- The larger d is, the larger the number of control points and the more difficult the problem of control point placement.

- Local control of geometry is not very flexible.

Traditional Approaches

- The larger d is, the larger the number of control points and the more difficult the problem of control point placement.

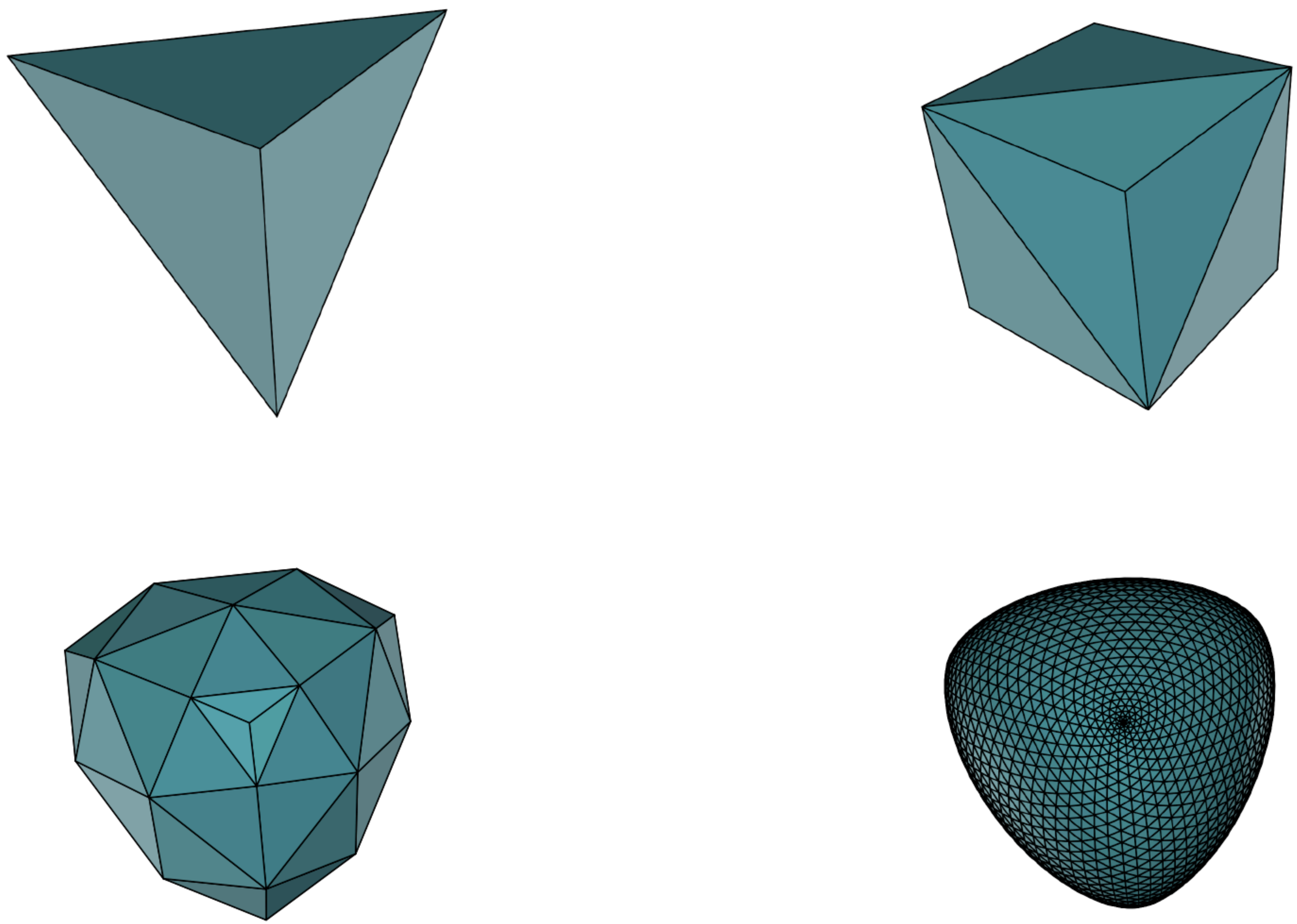
- Local control of geometry is not very flexible.

[Loop and DeRose, 1989], [Seidel, 1994], [Prautzsch, 1997], and [Reif, 1998] give C^k parametric approaches for arbitrary k.

Traditional Approaches

Traditional Approaches

Another popular approach consists of using subdivision surfaces.



Traditional Approaches

Traditional Approaches

Subdivision surfaces are probably the easiest and more intuitive solution for the problem **whenever the smoothness degree, k, is not large**.

Traditional Approaches

Subdivision surfaces are probably the easiest and more intuitive solution for the problem **whenever the smoothness degree, k, is not large**.

For large values of k, the few existing schemes are rather complex.

Traditional Approaches

Subdivision surfaces are probably the easiest and more intuitive solution for the problem **whenever the smoothness degree, k, is not large**.

For large values of k, the few existing schemes are rather complex.

See [Warren, 2002].

Traditional Approaches

Traditional Approaches

Implicit surfaces can also be used to solve the problem.

Traditional Approaches

Implicit surfaces can also be used to solve the problem.

They can naturally define C^∞ surfaces.

Traditional Approaches

Implicit surfaces can also be used to solve the problem.

They can naturally define C^∞ surfaces.

In general, the fitting problem is made into an interpolation problem.

Traditional Approaches

Implicit surfaces can also be used to solve the problem.

They can naturally define C^∞ surfaces.

In general, the fitting problem is made into an interpolation problem.

Then, one can use RBF, MPU, moving least squares, etc.

Traditional Approaches

Traditional Approaches

The main drawback of this implict surface-based approach is that the topological condition becomes a lot harder to satisfy.

Traditional Approaches

The main drawback of this implict surface-based approach is that the topological condition becomes a lot harder to satisfy.

More recent results **might** overcome this difficulty.

Traditional Approaches

The main drawback of this implict surface-based approach is that the topological condition becomes a lot harder to satisfy.

More recent results **might** overcome this difficulty.

See [Shen, O'Brien, and Shewchuk, 2004] and [Kolluri, 2005].

Traditional Approaches

The main drawback of this implict surface-based approach is that the topological condition becomes a lot harder to satisfy.

More recent results **might** overcome this difficulty.

See [Shen, O'Brien, and Shewchuk, 2004] and [Kolluri, 2005].

Implicit and parametric surfaces have complementary features.

The Manifold-Based Approach

The Manifold-Based Approach

An often neglected approach, the **manifold-based** one, has the potential to easily produce C^k surfaces, for an arbitrary k (including $k = \infty$).

The Manifold-Based Approach

An often neglected approach, the **manifold-based** one, has the potential to easily produce C^k surfaces, for an arbitrary k (including $k = \infty$).

The manifold approach has also some advantages over the traditional approaches when it comes to certain applications, such as texture synthesis and the solution of equations on surfaces.

The Manifold-Based Approach

The Manifold-Based Approach

Here, we

The Manifold-Based Approach

Here, we

- describe the manifold-based approach for the surface fitting problem,

The Manifold-Based Approach

Here, we

- describe the manifold-based approach for the surface fitting problem,
- review the main existing solutions and their limitations, and

The Manifold-Based Approach

Here, we

- describe the manifold-based approach for the surface fitting problem,

- review the main existing solutions and their limitations, and

- point out some applications and research challenges in Computer Graphics, Image Processing, and Computer Vision that can be more naturally tackled by using manifolds.

What's Next?

What's Next?

II. Manifolds

What's Next?

II. Manifolds

III. Constructing Manifolds

What's Next?

II. Manifolds

III. Constructing Manifolds

IV. Fitting Surfaces to Polygonal Meshes – Part I

What's Next?

II. Manifolds

III. Constructing Manifolds

IV. Fitting Surfaces to Polygonal Meshes – Part I

Coffee break

What's Next?

What's Next?

V. Fitting Surfaces to Polygonal Meshes – Part II

What's Next?

V. Fitting Surfaces to Polygonal Meshes – Part II

VI. Adaptive Manifold Fitting – Part I

What's Next?

V. Fitting Surfaces to Polygonal Meshes – Part II

VI. Adaptive Manifold Fitting – Part I

V. Adaptive Manifold Fitting – Part II

What's Next?

V. Fitting Surfaces to Polygonal Meshes – Part II

VI. Adaptive Manifold Fitting – Part I

V. Adaptive Manifold Fitting – Part II

VIII. Applications of Manifolds and Research Challenges

Manifolds

Jean Gallier
UPenn

Outline

- Manifolds: Brief History
- Informal definition
- Formal definition
- Examples
  - The Sphere
  - Real Projective Space
- Conclusions

Origins of Manifolds

Origins of Manifolds

- Around 1860, Mobius, Jordan, and Dyck studied the topology of surfaces.

Origins of Manifolds

- Around 1860, Mobius, Jordan, and Dyck studied the topology of surfaces.

- In a famous paper published in 1888, Dyck already uses the term **manifold** (in German).

Origins of Manifolds

- Around 1860, Mobius, Jordan, and Dyck studied the topology of surfaces.

- In a famous paper published in 1888, Dyck already uses the term **manifold** (in German).

- In the early 1900's, Dehn, Heegaard, Veblen and Alexander routinely used the term **manifold**.

Origins of Manifolds

- Around 1860, Mobius, Jordan, and Dyck studied the topology of surfaces.
- In a famous paper published in 1888, Dyck already uses the term **manifold** (in German).
- In the early 1900's, Dehn, Heegaard, Veblen and Alexander routinely used the term **manifold**.
- Hermann Weyl was among the first to give a rigorous definition (1913).

Keys Contributors to the notion of manifold:

Keys Contributors to the notion of manifold:

Georg Friedrich Bernhard Riemann
1826-1866

Keys Contributors to the notion of manifold:

Georg Friedrich Bernhard Riemann
1826-1866

Hermann Klaus Hugo Weyl
1885-1955

Keys Contributors to the notion of manifold:

Keys Contributors to the notion of manifold:

Hermann Weyl (again)

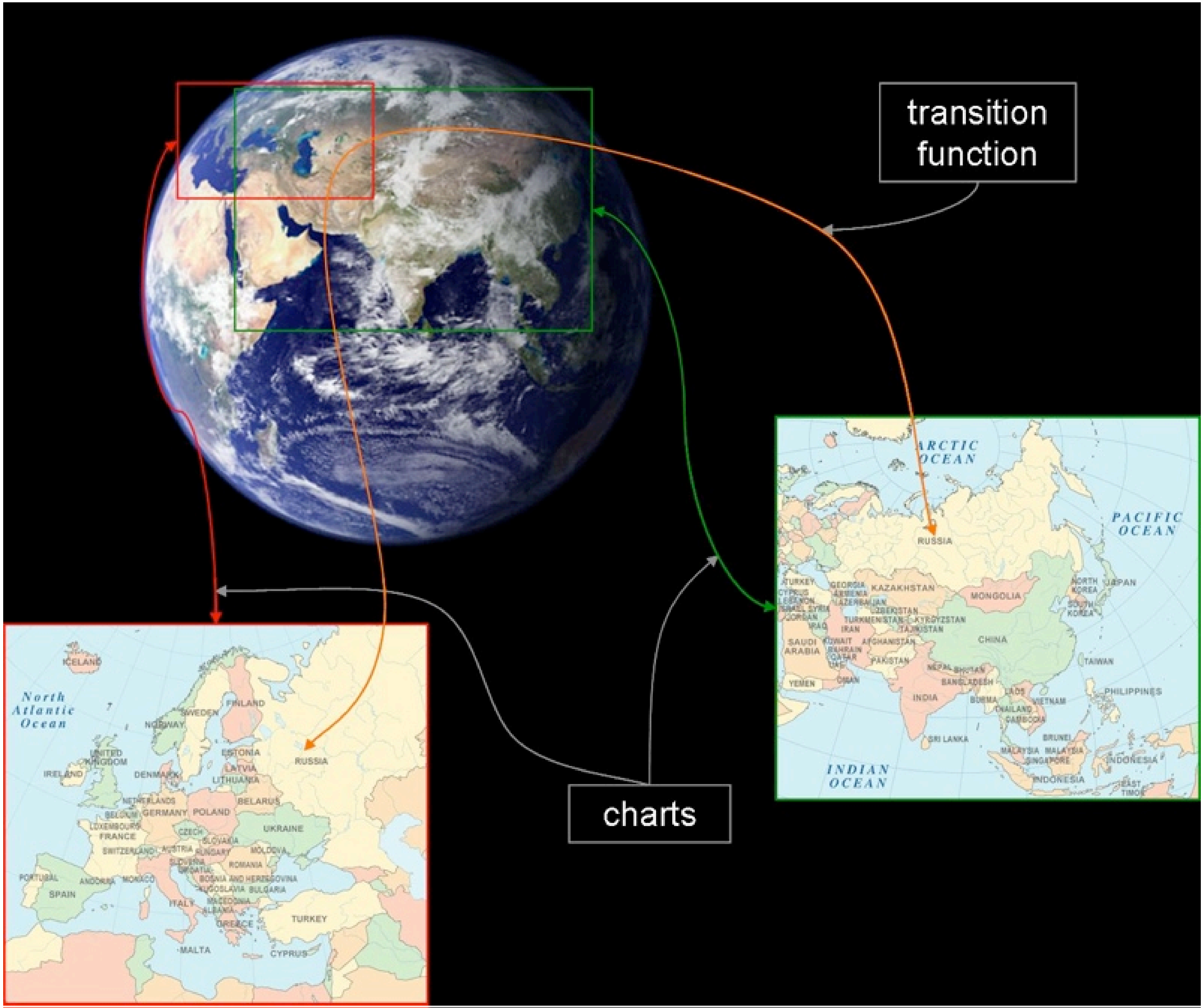
Keys Contributors to the notion of manifold:

Hermann Weyl (again)

Hassler Whitney
1907-1989

Manifold: An Intuitive Picture

Manifold: An Intuitive Picture



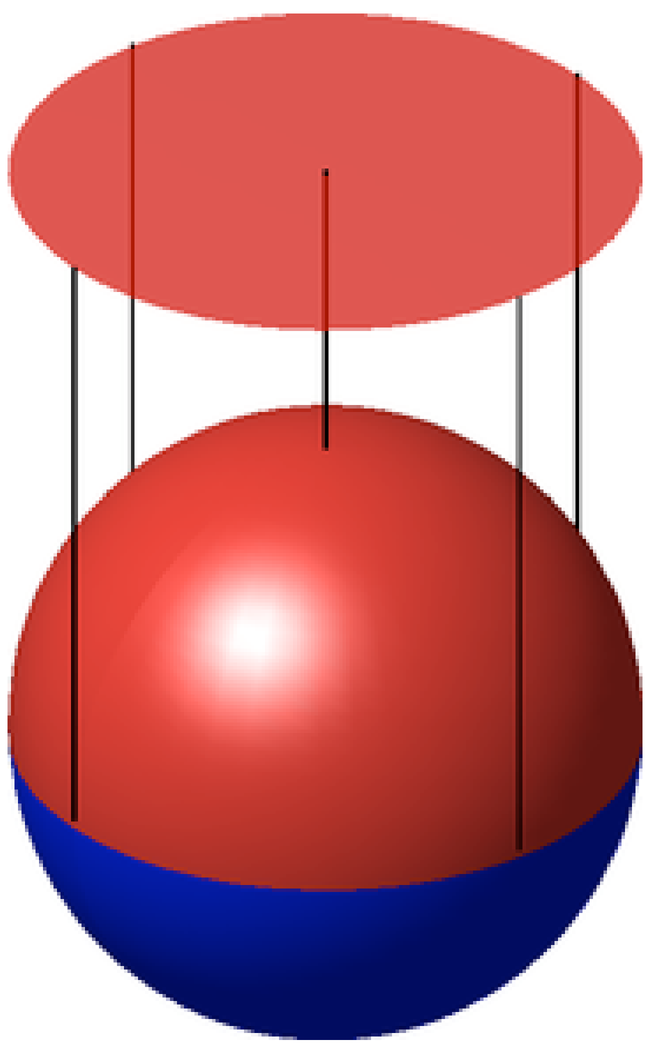
Manifolds: Informal Definition

Manifolds: Informal Definition

- A manifold is a topological space with an open cover so that every open set in this cover “looks” like an open subset of $\mathbb{R}^n$.

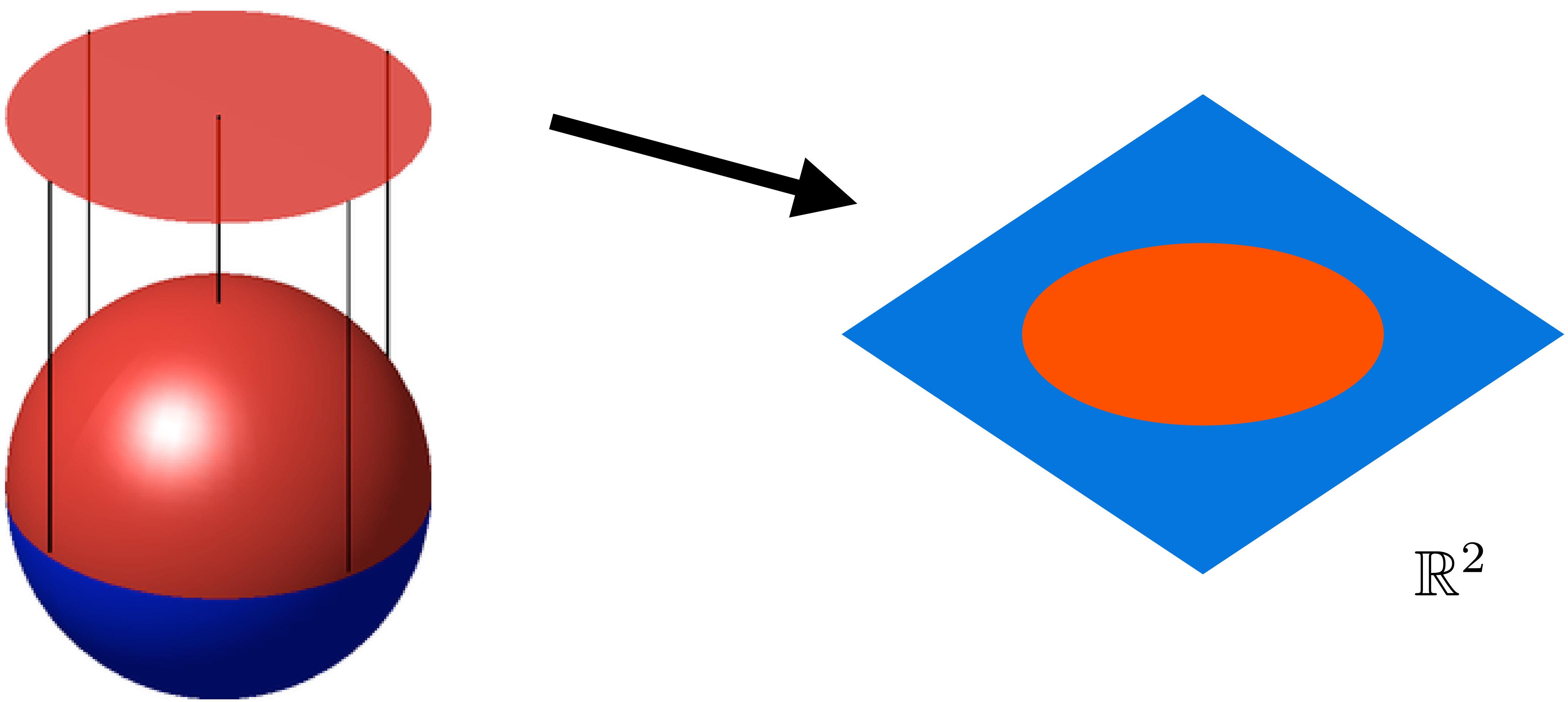
Manifolds: Informal Definition

- A manifold is a topological space with an open cover so that every open set in this cover "looks" like an open subset of $\mathbb{R}^n$.



Manifolds: Informal Definition

- A manifold is a topological space with an open cover so that every open set in this cover “looks” like an open subset of $\mathbb{R}^n$.



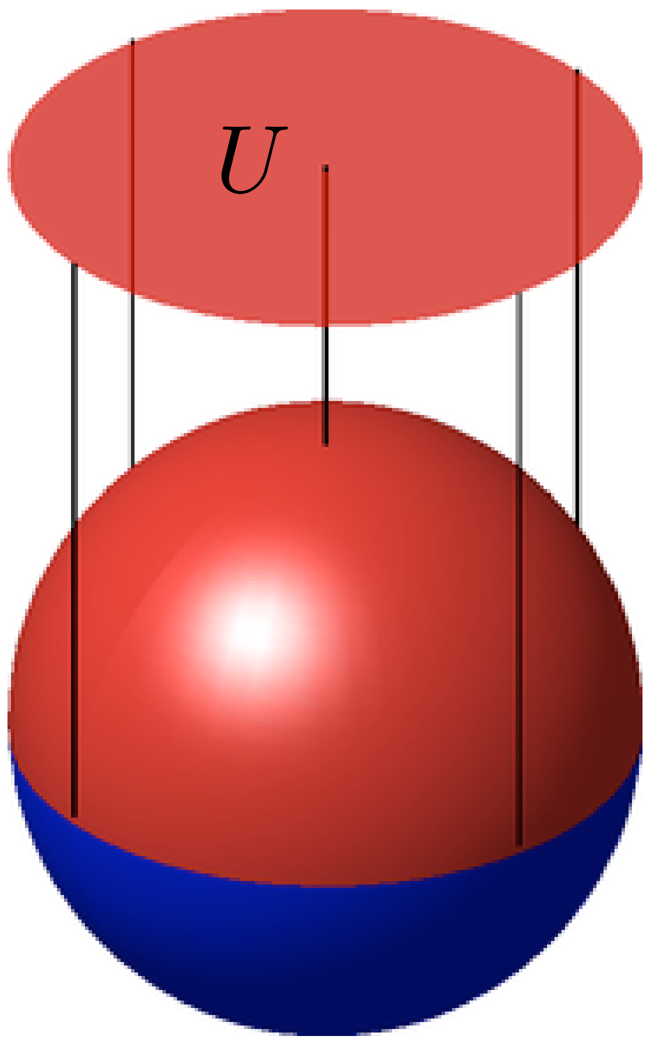
Manifolds: Informal Definition

Manifolds: Informal Definition

- To make our informal notion precise, we use homeomorphisms, $\varphi : U \to \Omega$, where $\Omega \subseteq \mathbb{R}^n$ is an open subset of $\mathbb{R}^n$.

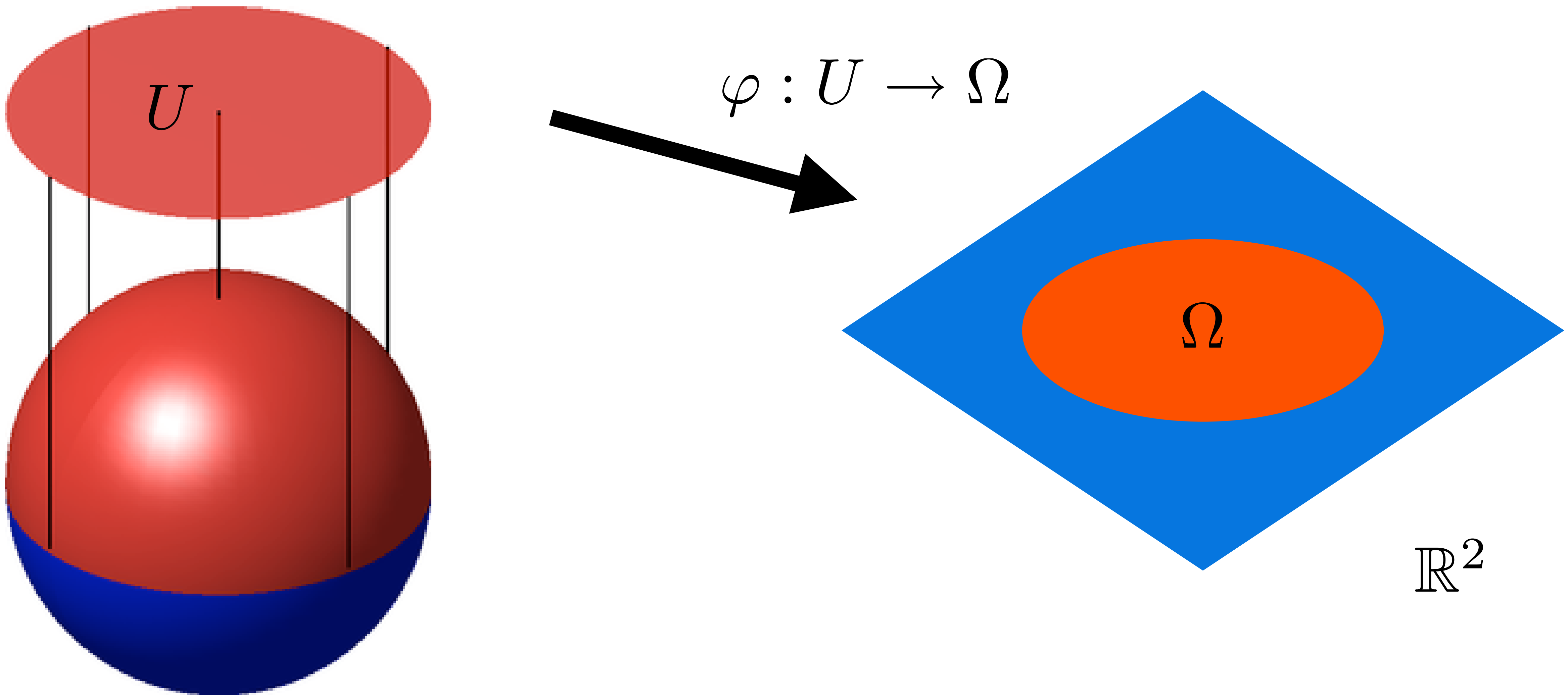
Manifolds: Informal Definition

- To make our informal notion precise, we use homeomorphisms, $\varphi : U \to \Omega$, where $\Omega \subseteq \mathbb{R}^n$ is an open subset of $\mathbb{R}^n$.



Manifolds: Informal Definition

- To make our informal notion precise, we use homeomorphisms, $\varphi : U \to \Omega$, where $\Omega \subseteq \mathbb{R}^n$ is an open subset of $\mathbb{R}^n$.



Manifolds: Informal Definition

Manifolds: Informal Definition

- We also want to be able "to do calculus" on our manifolds. For this we need some conditions on **overlaps** of open sets.

Manifolds: Informal Definition

Manifolds: Informal Definition

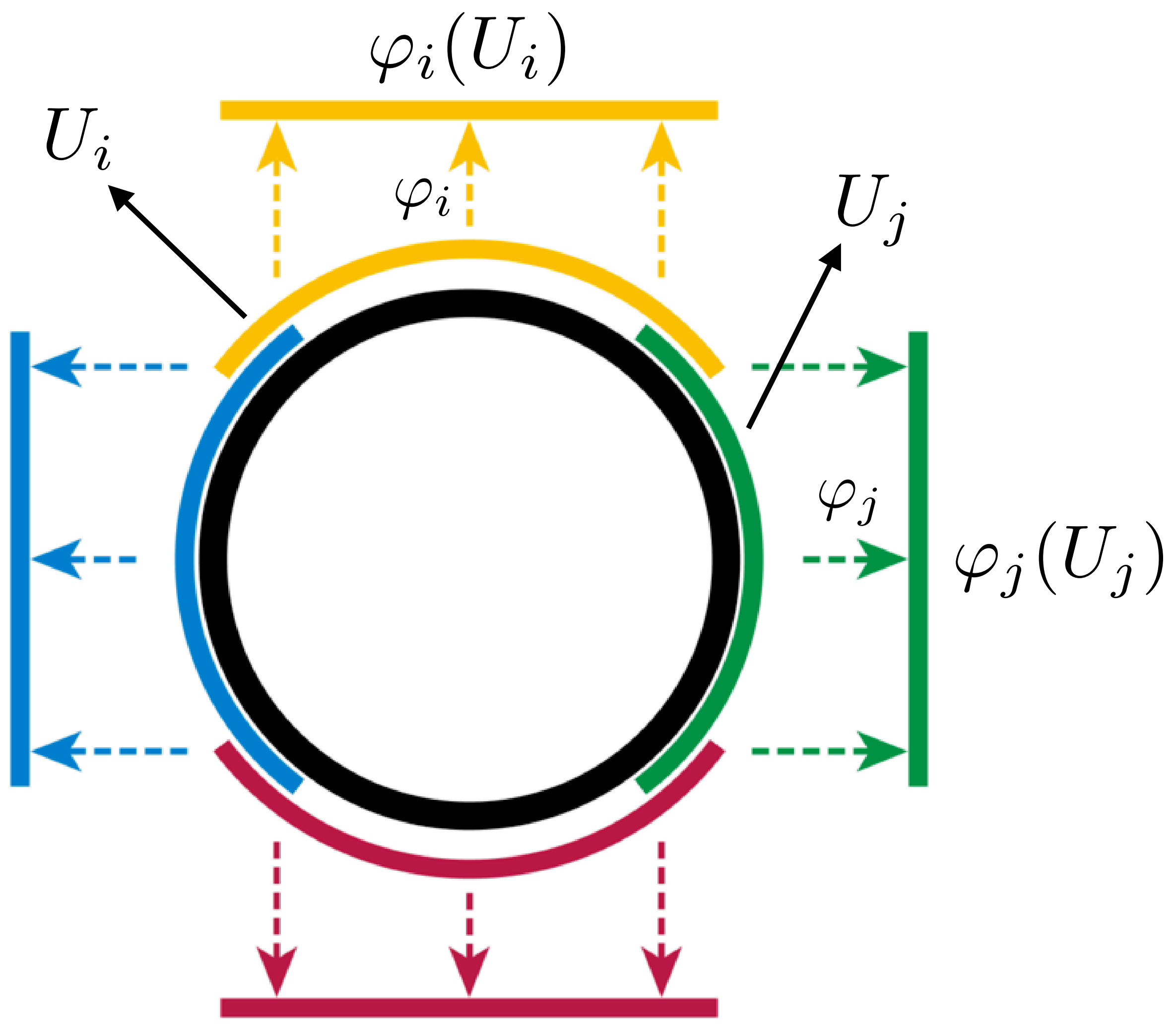
- Whenever $U_i \cap U_j \neq \emptyset$, we need some condition on the **transition function**,

$$\varphi_{ji} = \varphi_j \circ \varphi_i^{-1} : \varphi_i(U_i \cap U_j) \to \varphi_j(U_i \cap U_j)\,.$$

Manifolds: Informal Definition

- Whenever $U_i \cap U_j \neq \emptyset$, we need some condition on the **transition function**,

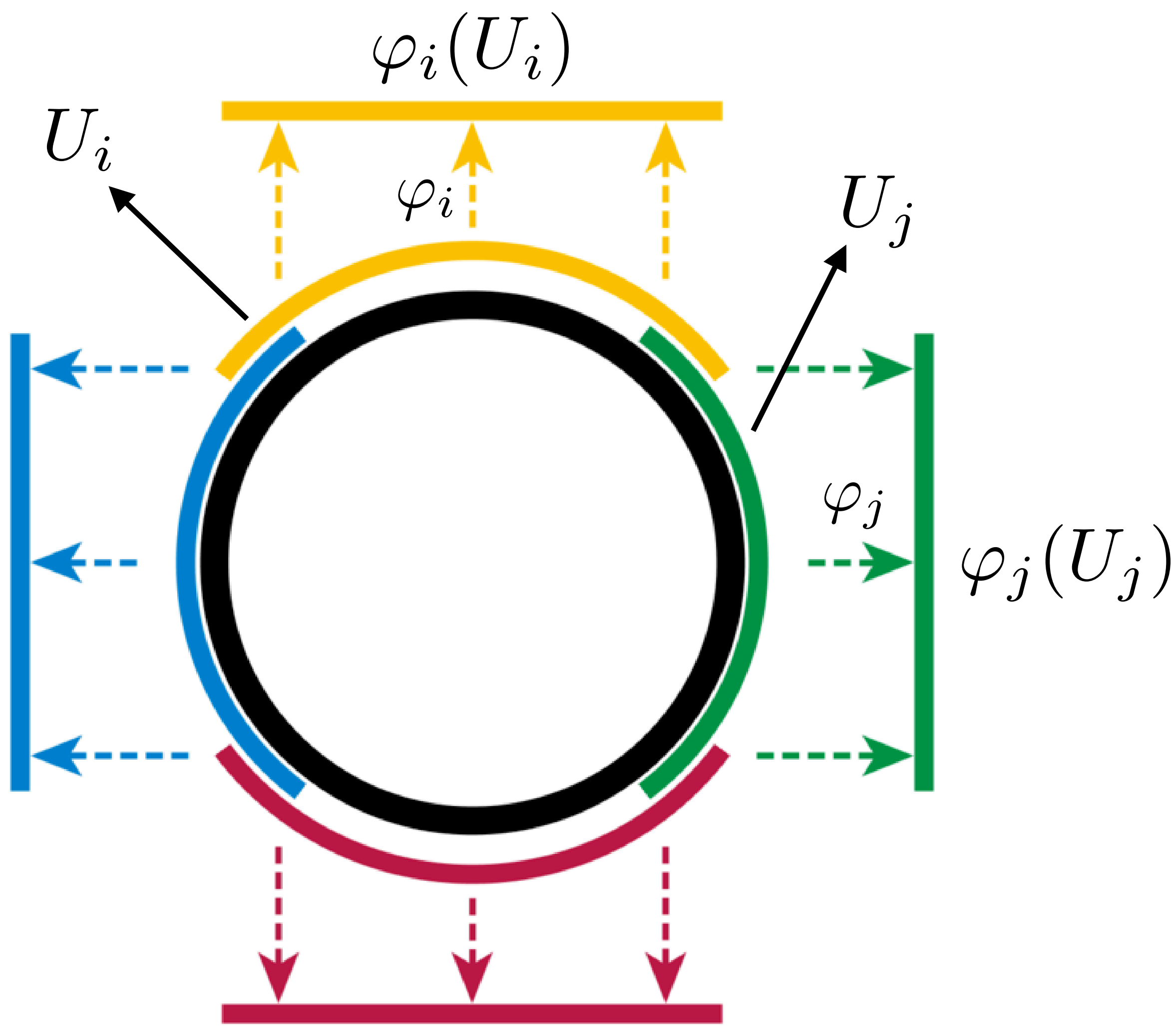
$$\varphi_{ji} = \varphi_j \circ \varphi_i^{-1} : \varphi_i(U_i \cap U_j) \to \varphi_j(U_i \cap U_j)\,.$$

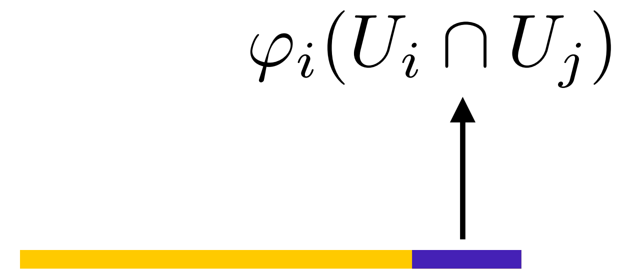


Manifolds: Informal Definition

- Whenever $U_i \cap U_j \neq \emptyset$, we need some condition on the **transition function**,

$$\varphi_{ji} = \varphi_j \circ \varphi_i^{-1} : \varphi_i(U_i \cap U_j) \to \varphi_j(U_i \cap U_j)\,.$$

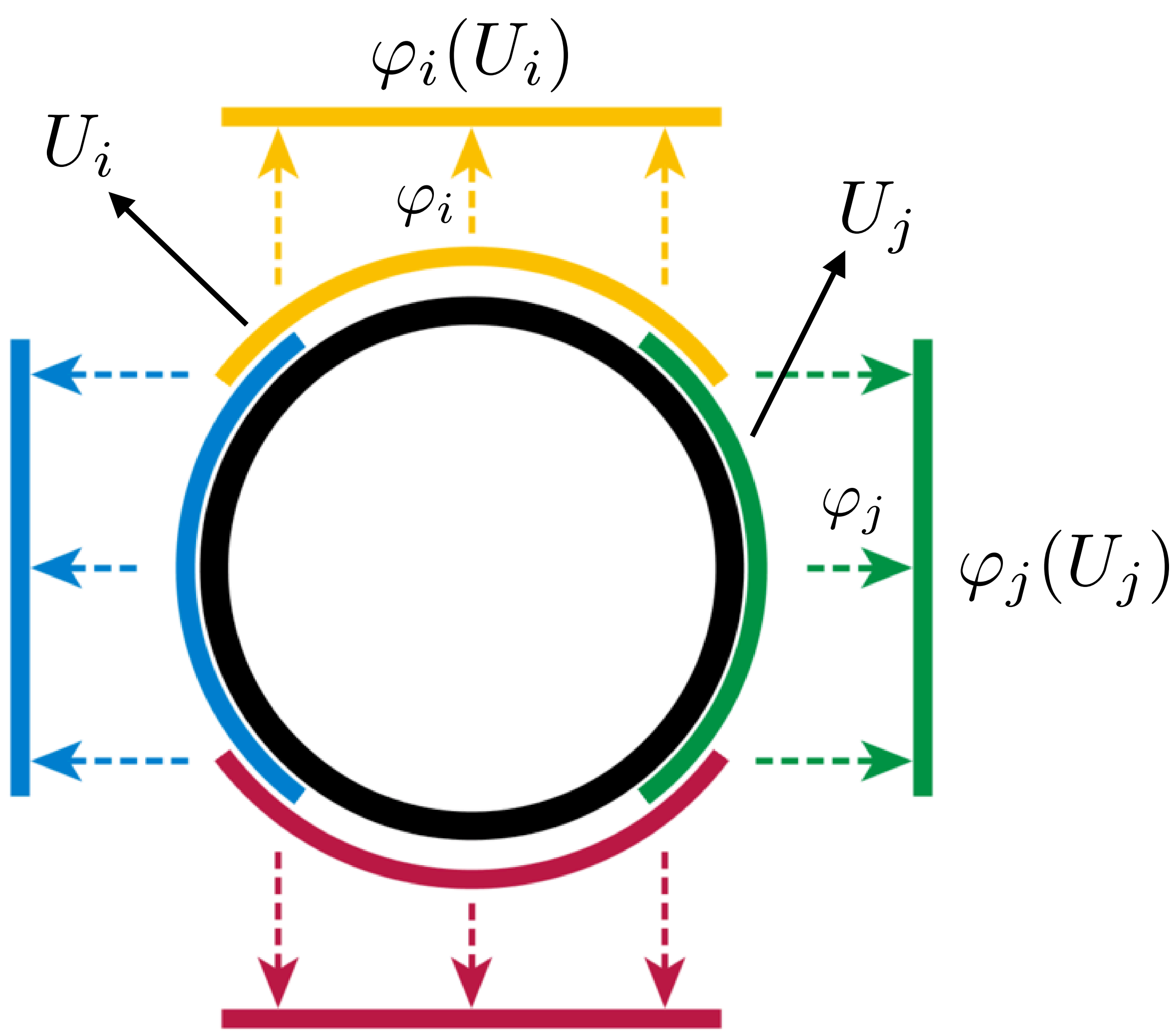


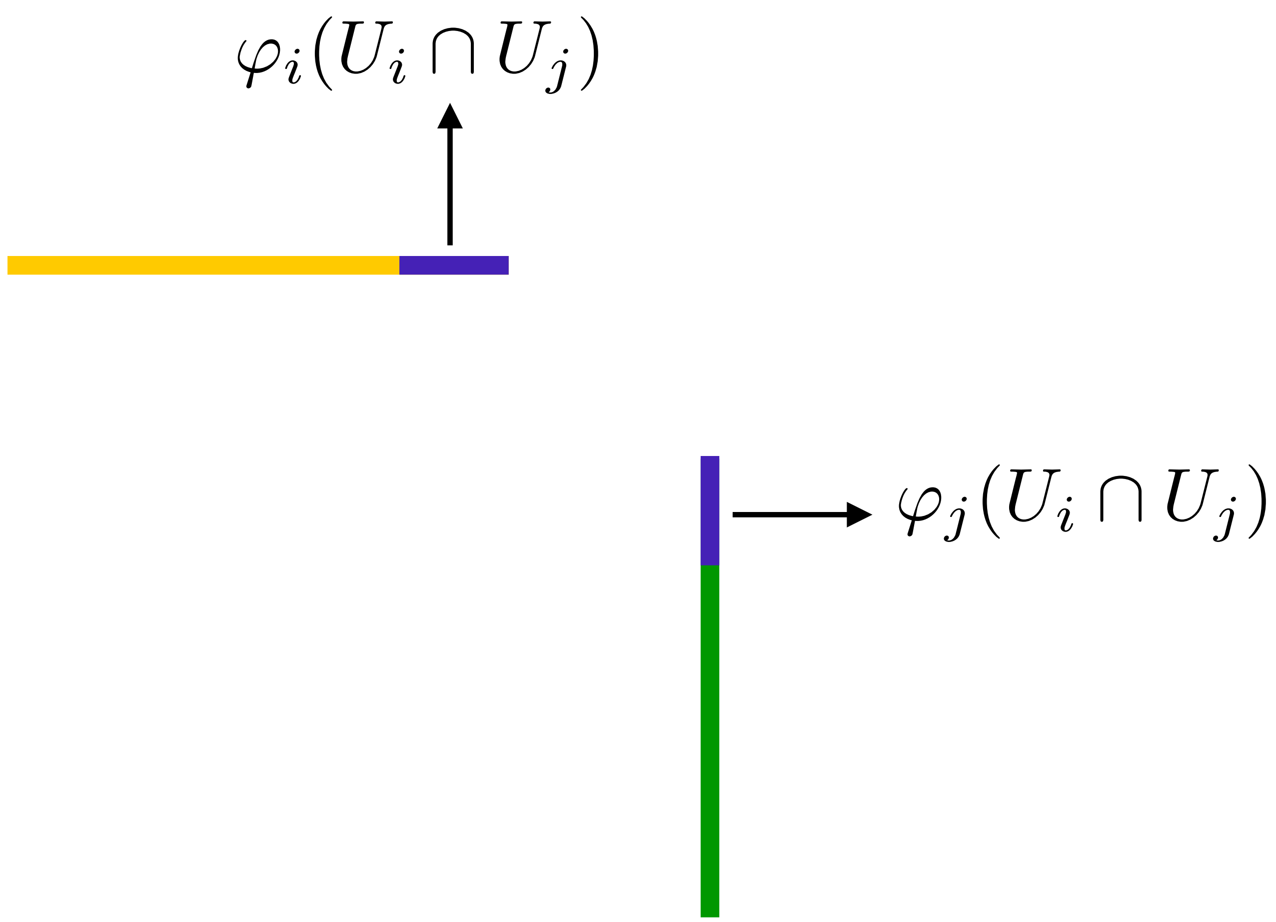


Manifolds: Informal Definition

- Whenever $U_i \cap U_j \neq \emptyset$, we need some condition on the **transition function**,

$$\varphi_{ji} = \varphi_j \circ \varphi_i^{-1} : \varphi_i(U_i \cap U_j) \to \varphi_j(U_i \cap U_j)\,.$$

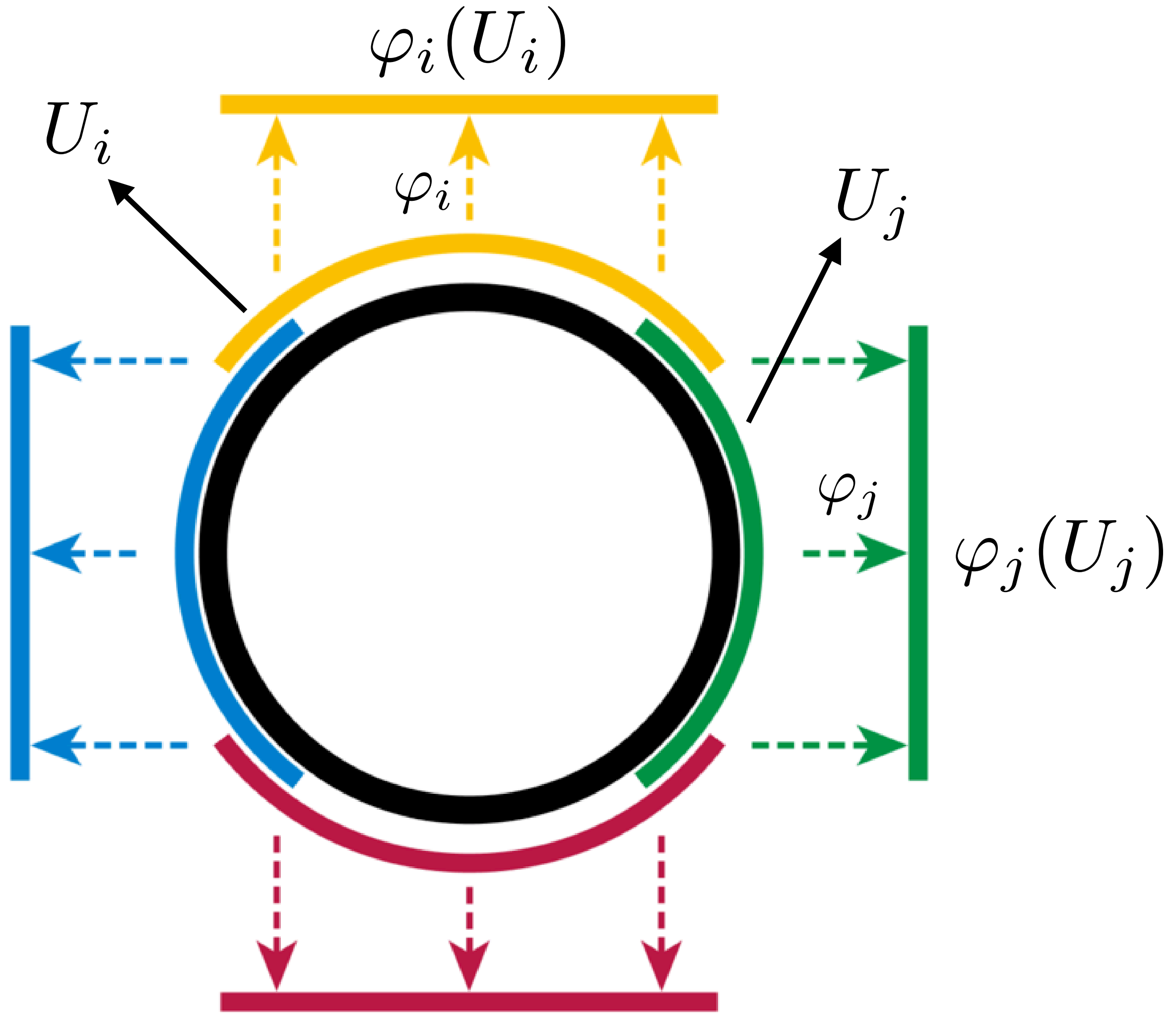


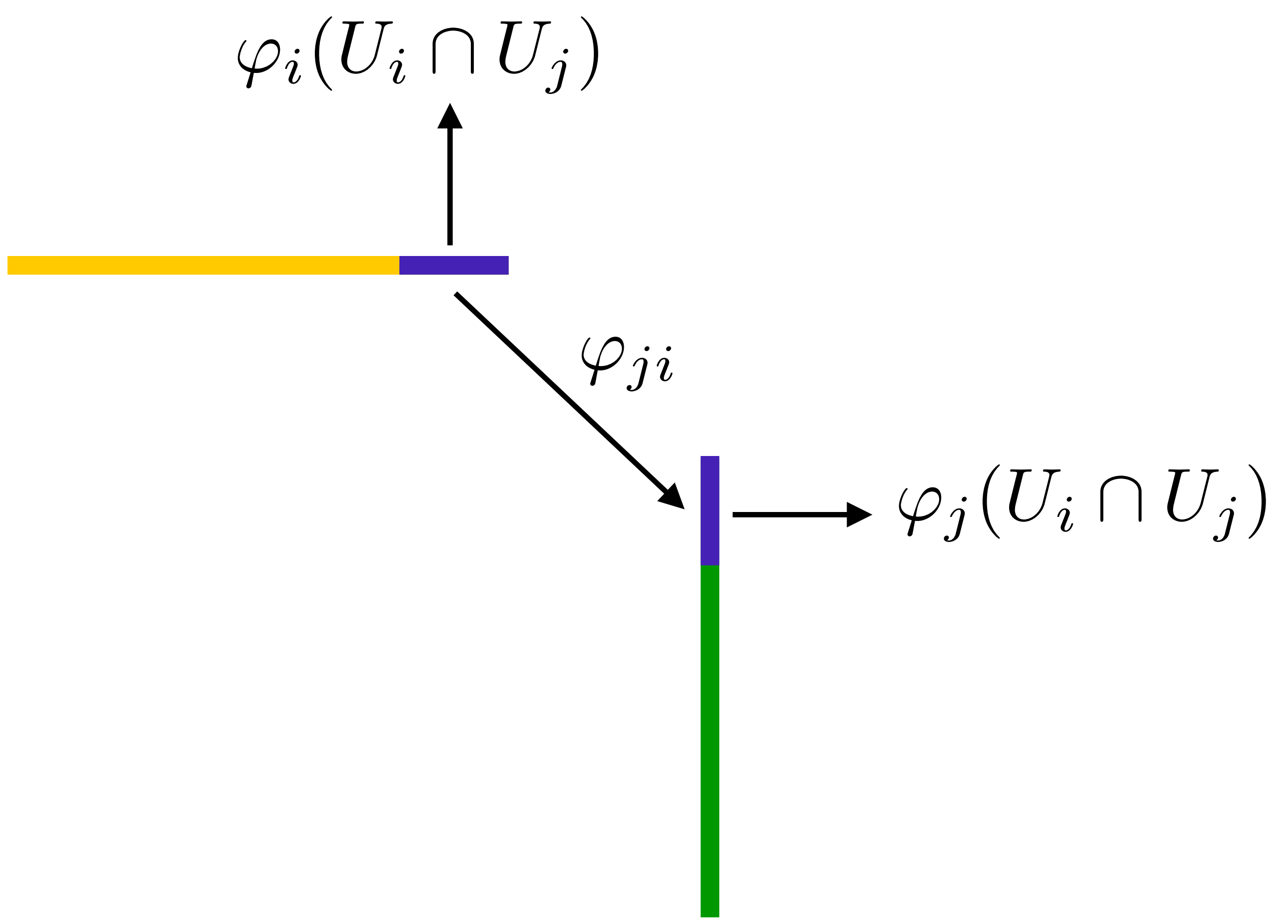


Manifolds: Informal Definition

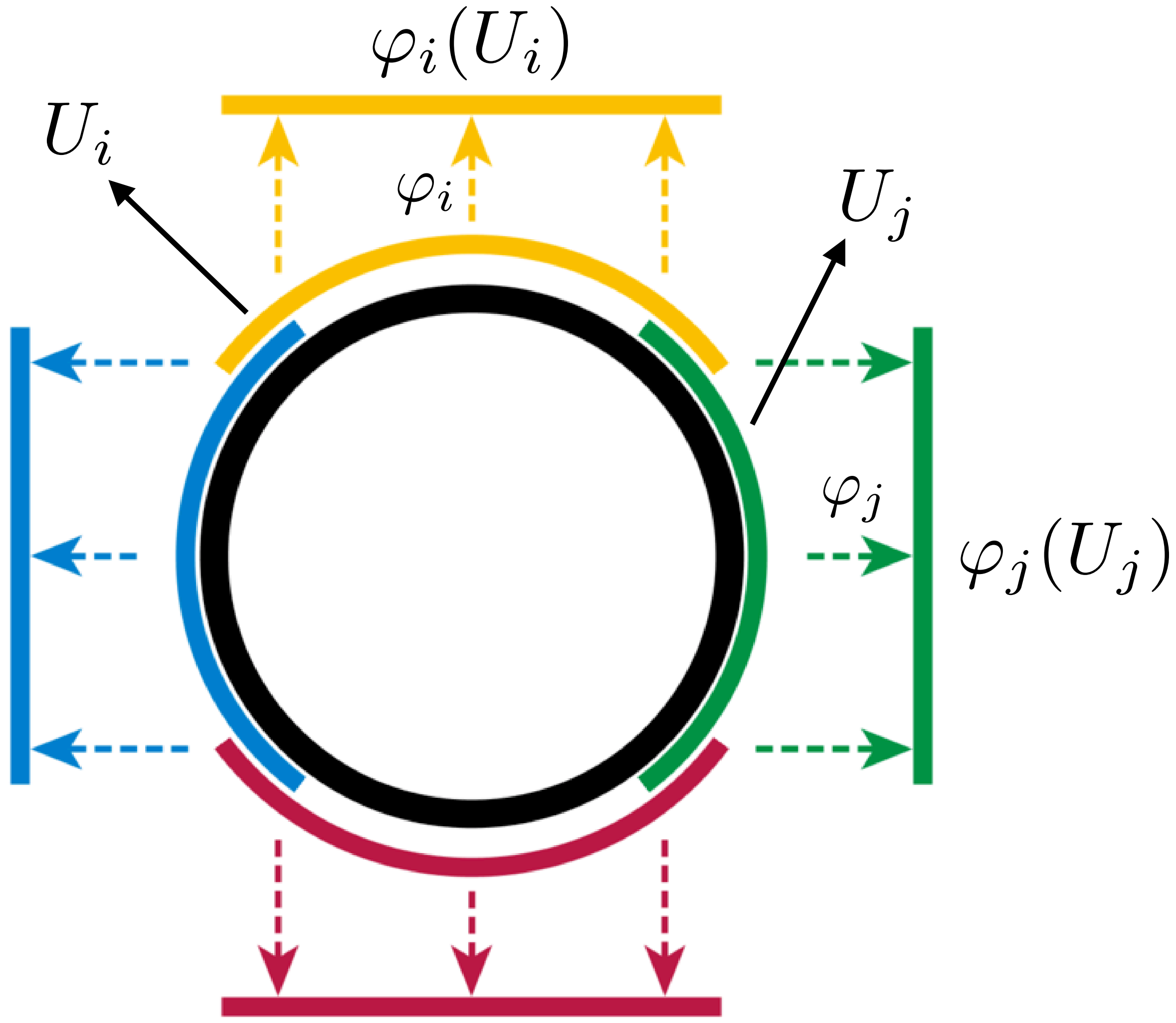
- Whenever $U_i \cap U_j \neq \emptyset$, we need some condition on the **transition function**,

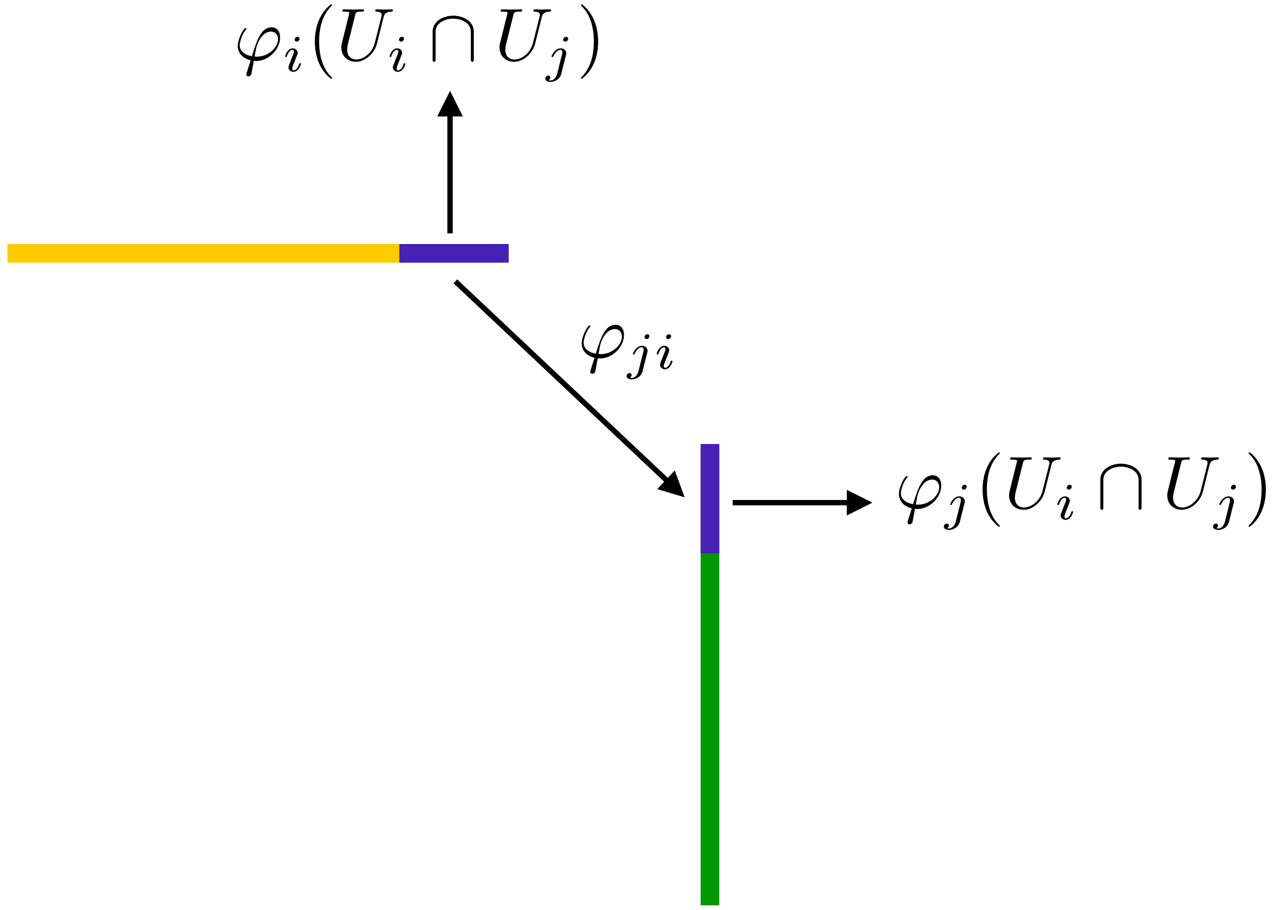
$$\varphi_{ji} = \varphi_j \circ \varphi_i^{-1} : \varphi_i(U_i \cap U_j) \to \varphi_j(U_i \cap U_j)\,.$$





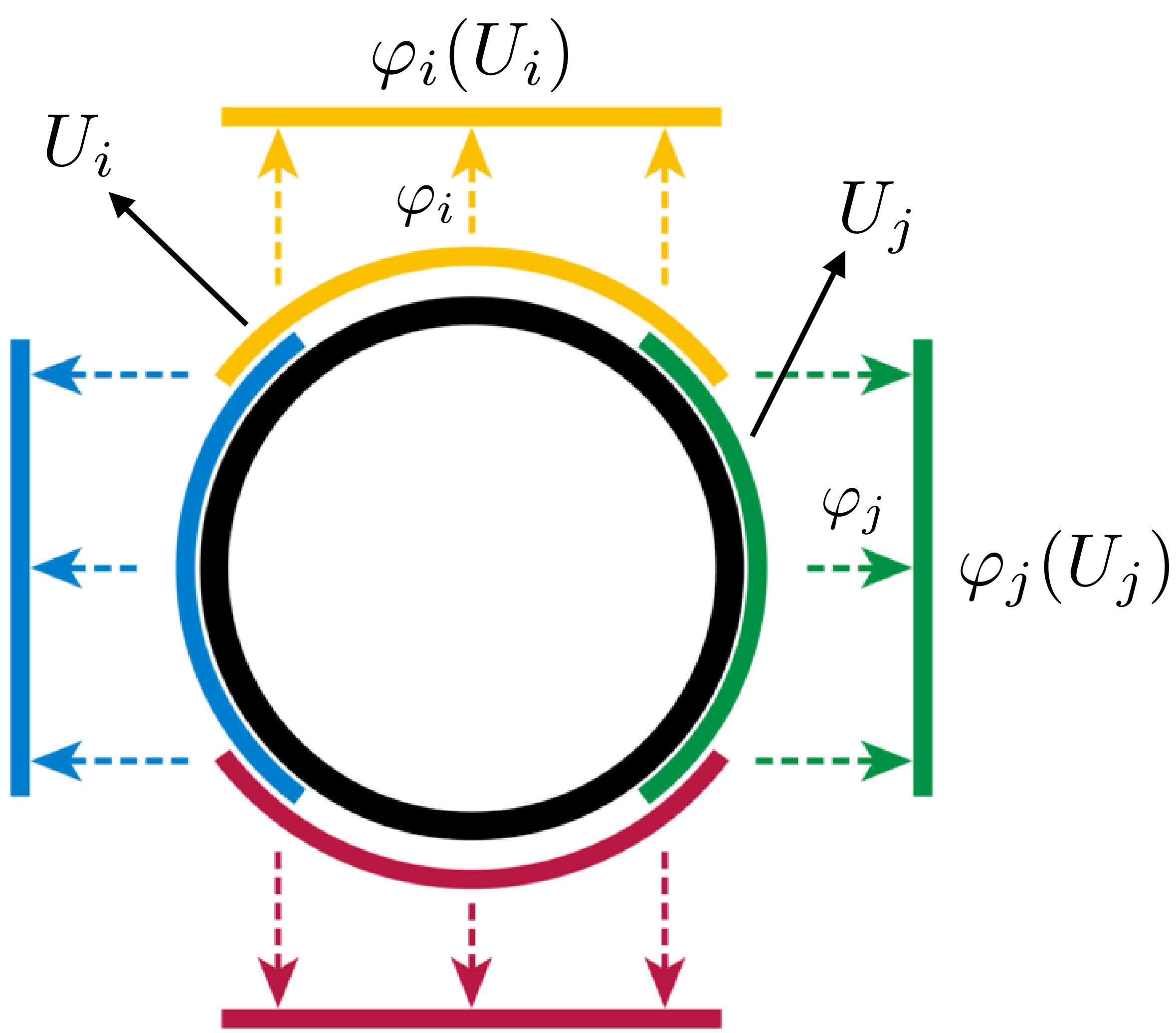
Manifolds: Informal Definition

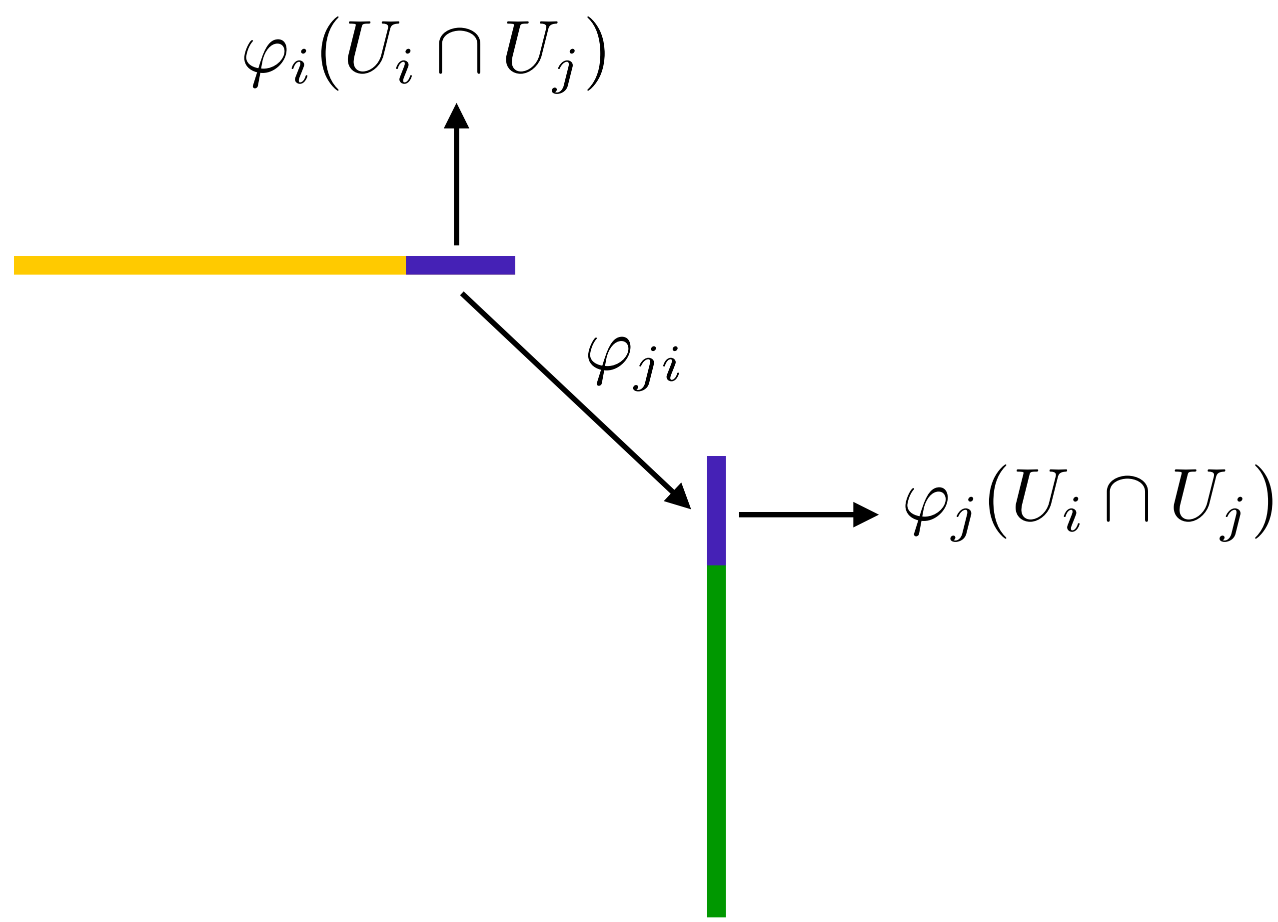




Manifolds: Informal Definition

- This is a map between two open subsets of $\mathbb{R}^n$ and we require it possess a certain amount of **smoothness**.





Manifolds: Formal Definition

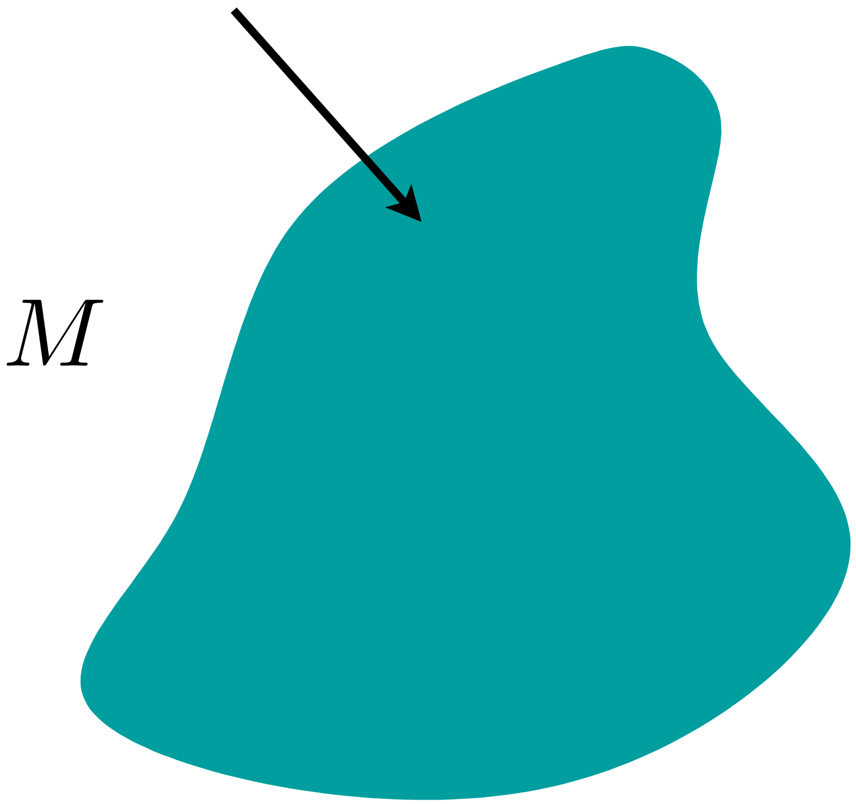
Manifolds: Formal Definition

Recall the definition of a manifold...

Manifolds: Formal Definition

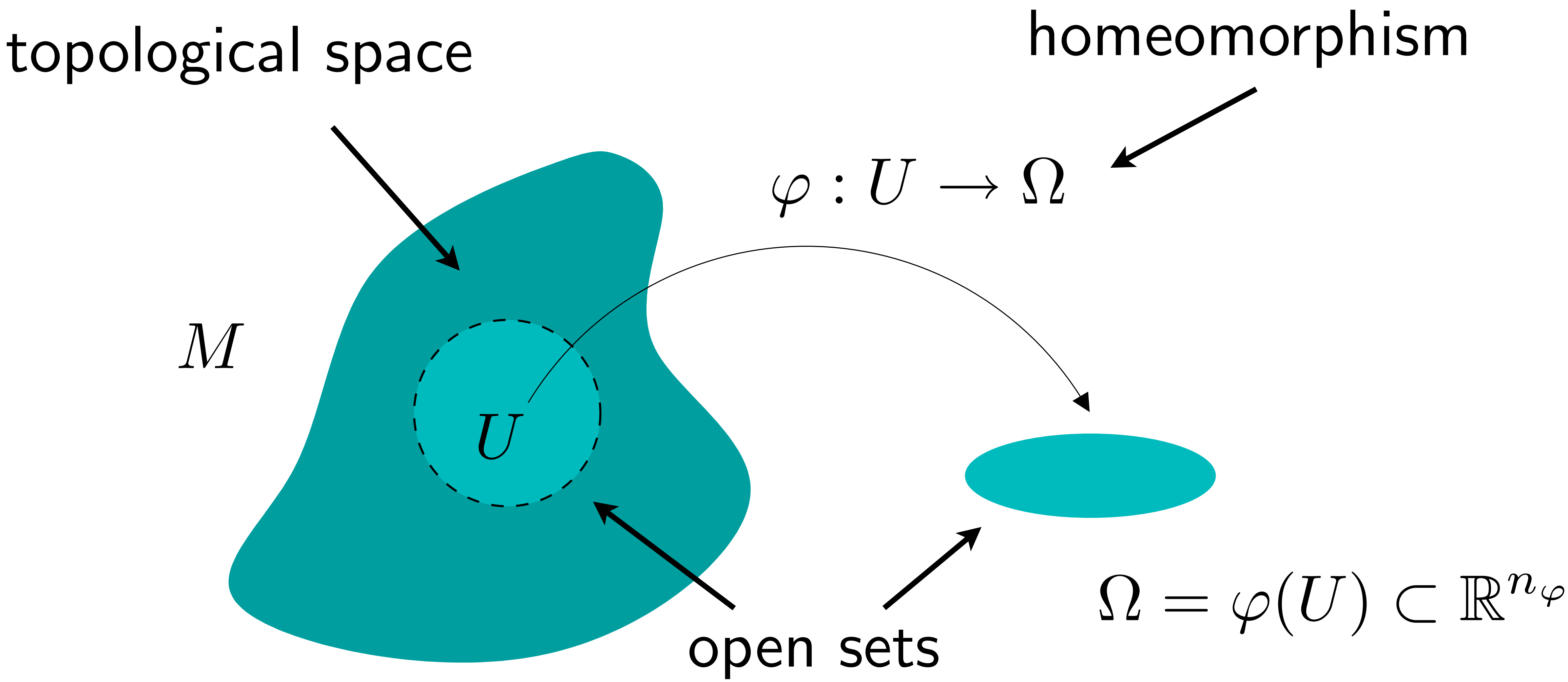
Recall the definition of a manifold...

topological space



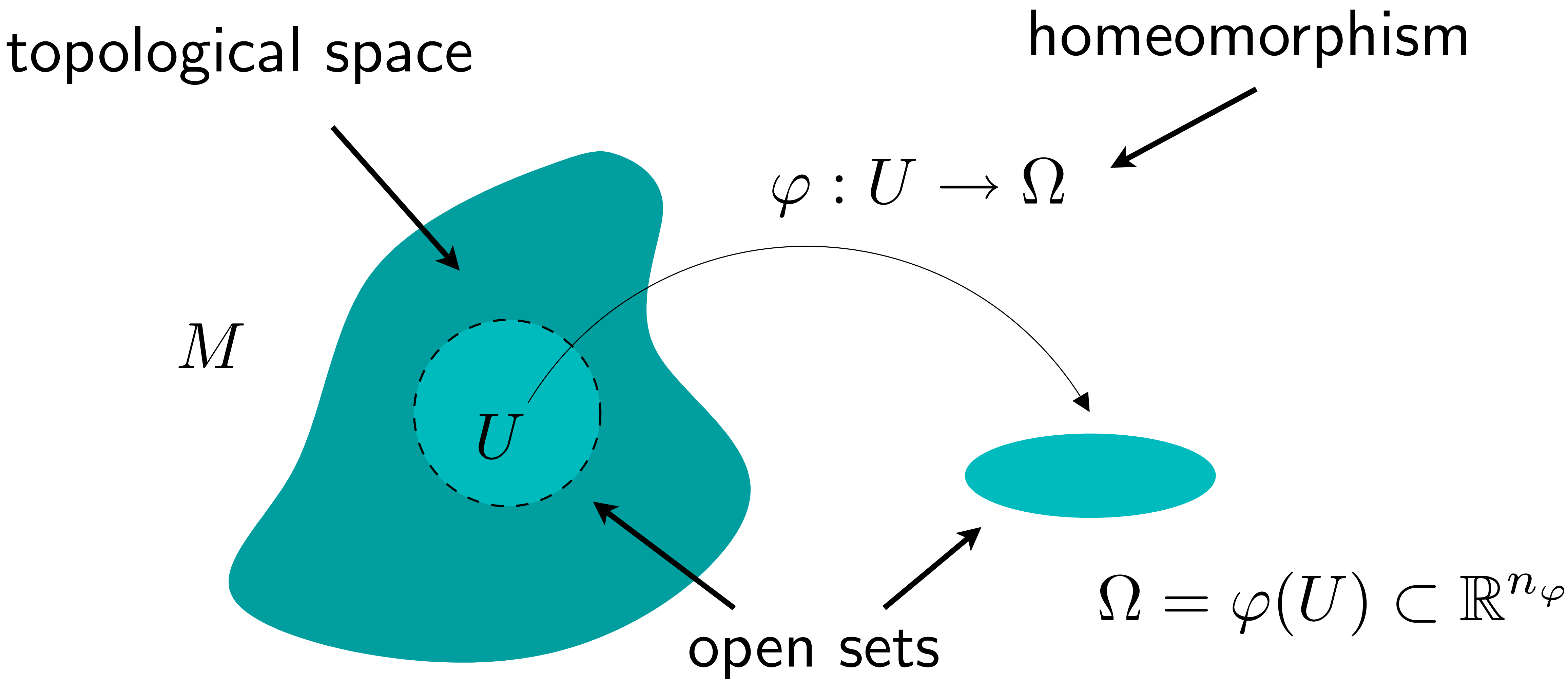
Manifolds: Formal Definition

Recall the definition of a manifold...



Manifolds: Formal Definition

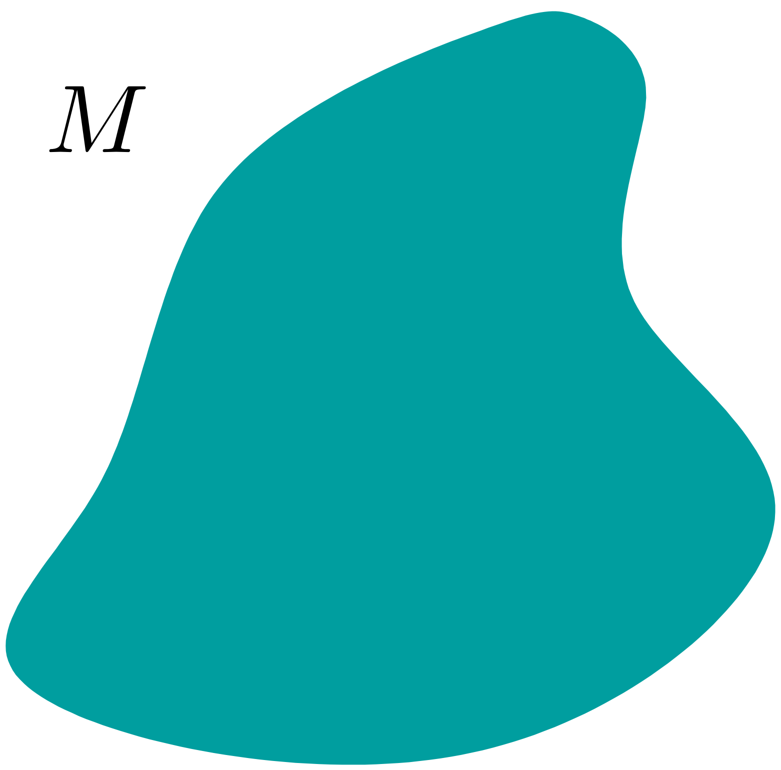
Recall the definition of a manifold...



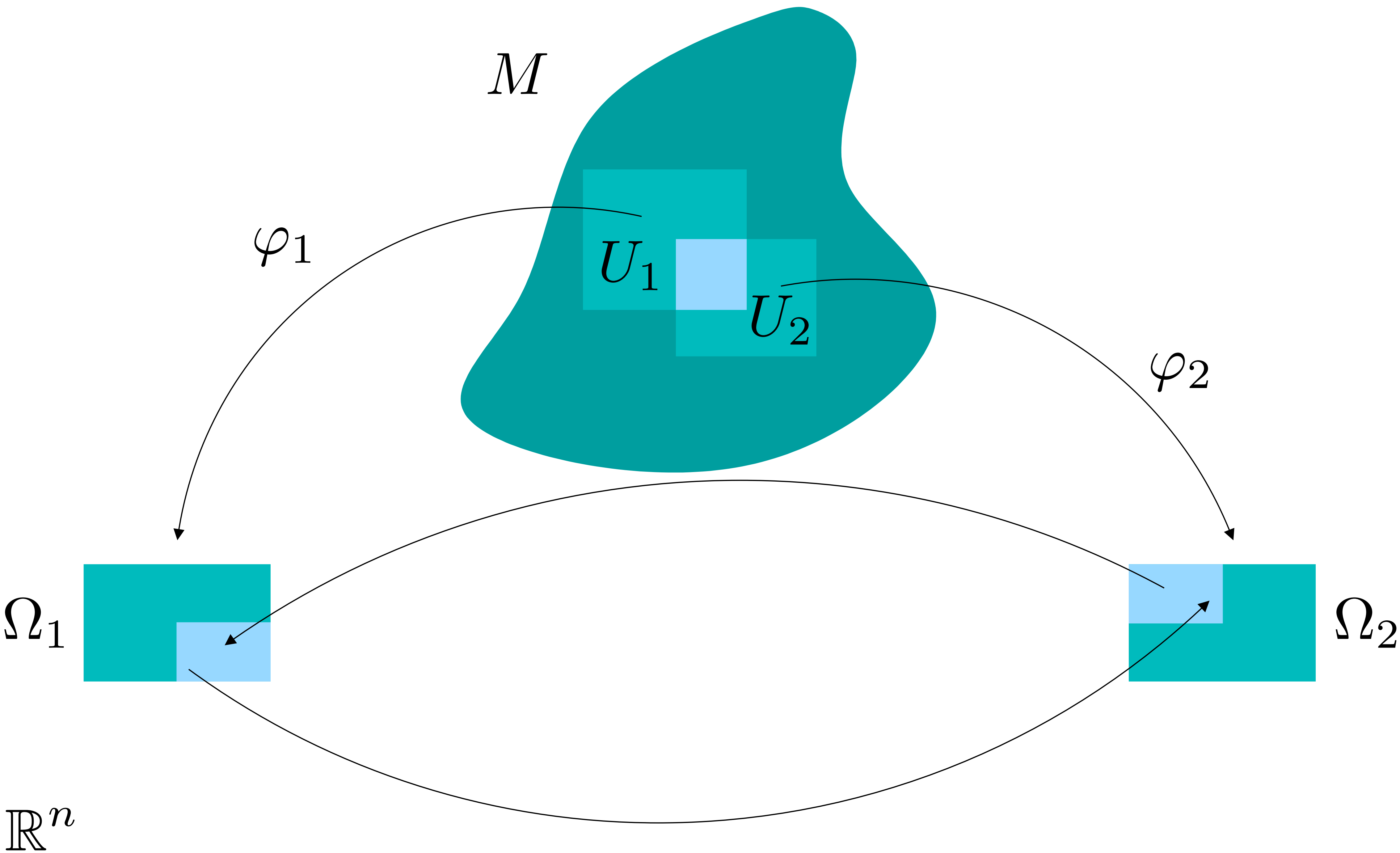
(U, φ) is called a **chart**.

Manifolds: Formal Definition

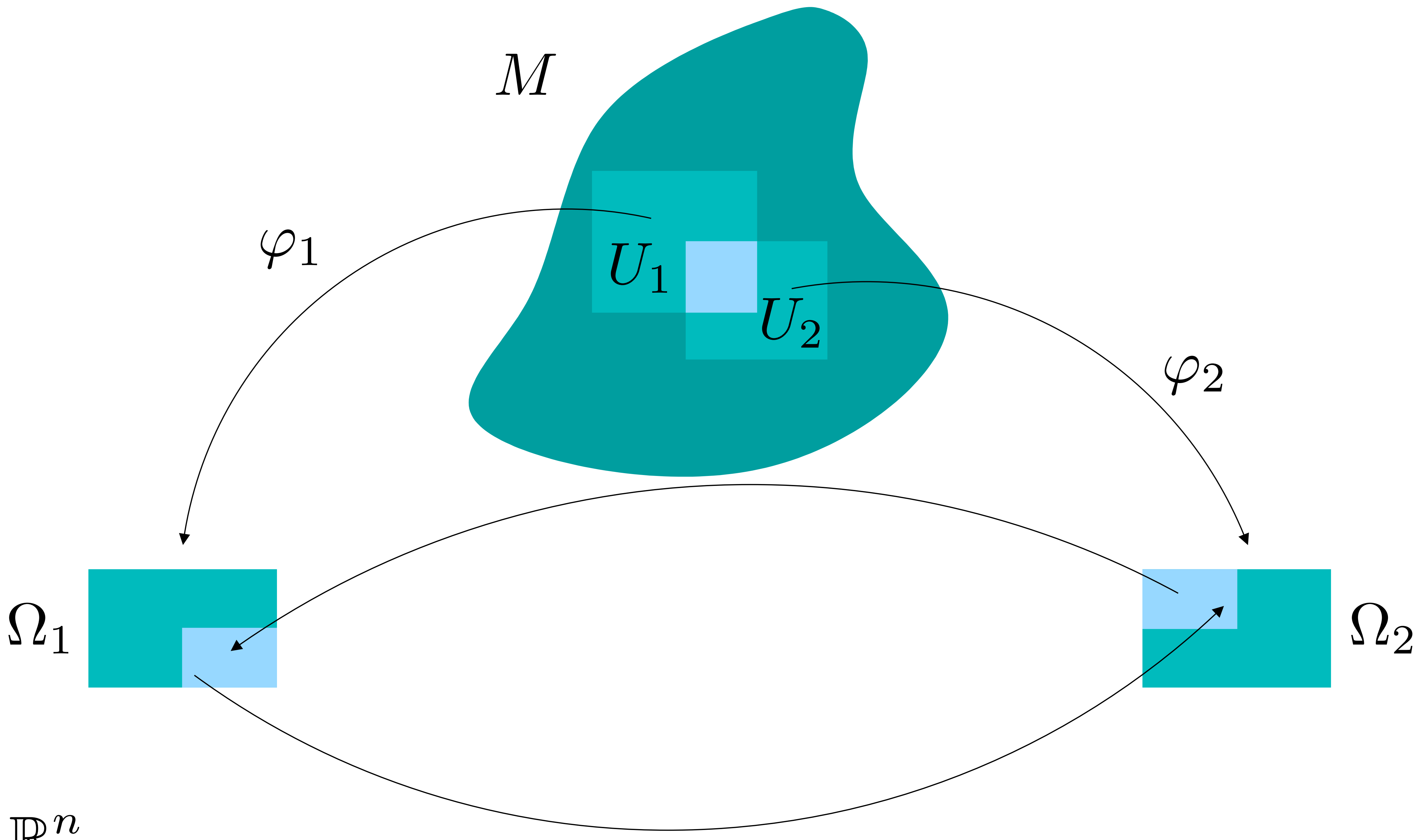
Manifolds: Formal Definition



Manifolds: Formal Definition

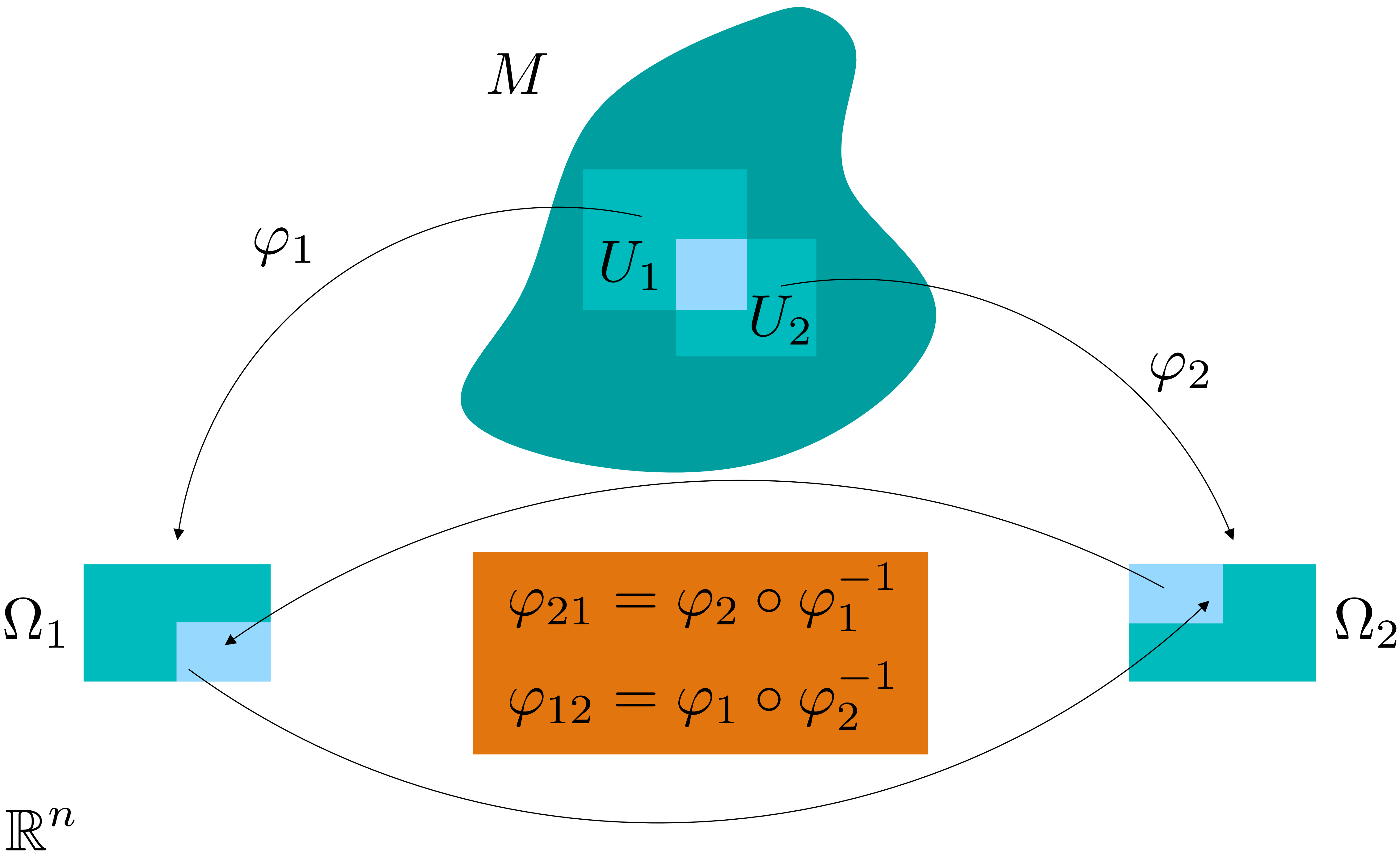


Manifolds: Formal Definition



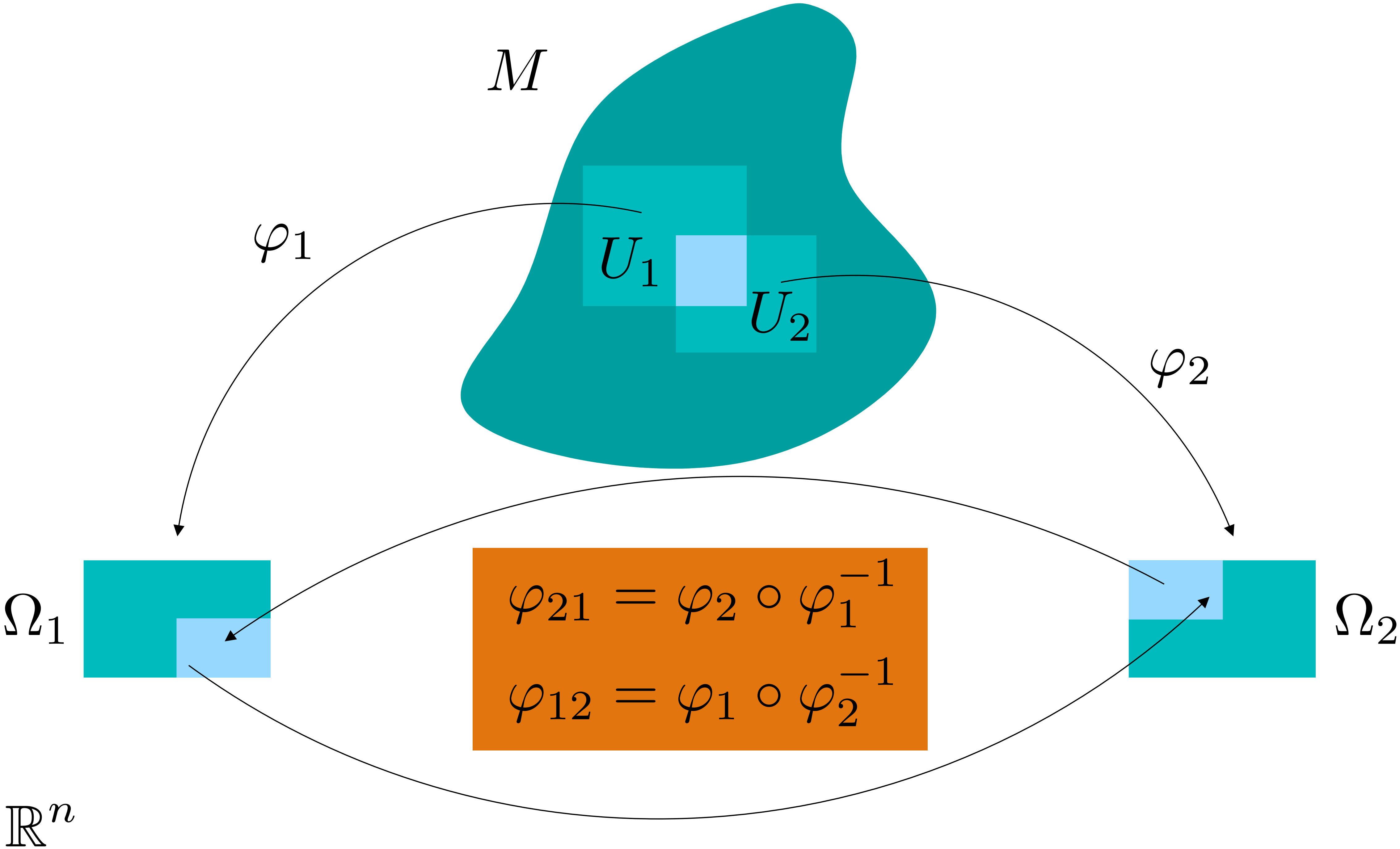
$$\varphi_{21} : \varphi_1(U_1 \cap U_2) \to \varphi_2(U_1 \cap U_2)$$
$$\varphi_{12} : \varphi_2(U_1 \cap U_2) \to \varphi_1(U_1 \cap U_2)$$

Manifolds: Formal Definition



$$\varphi_{21} : \varphi_1(U_1 \cap U_2) \to \varphi_2(U_1 \cap U_2)$$
$$\varphi_{12} : \varphi_2(U_1 \cap U_2) \to \varphi_1(U_1 \cap U_2)$$

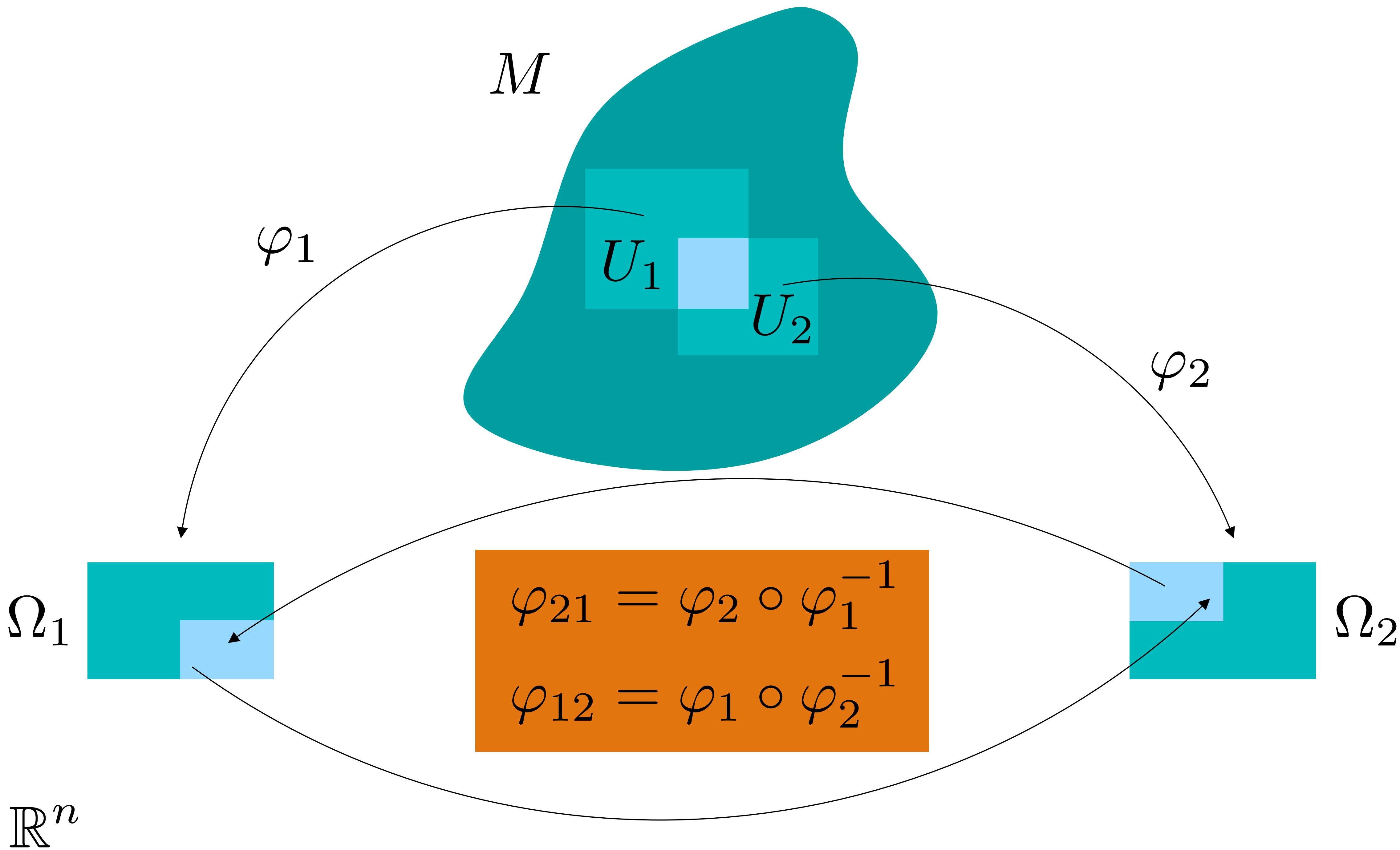
Manifolds: Formal Definition



$$\varphi_{21} = \varphi_2 \circ \varphi_1^{-1}$$

$$\varphi_{12} = \varphi_1 \circ \varphi_2^{-1}$$

Manifolds: Formal Definition



φ_{21} and φ_{12} are the **transition functions.**

Manifolds: Formal Definition

Manifolds: Formal Definition

A C^k n-**atlas** is a family of charts, $\{(U_i, \varphi_i)\}_{(i \in I)}$, where I is a non-empty countable set, and such that the following conditions hold:

Manifolds: Formal Definition

A C^k n-**atlas** is a family of charts, $\{(U_i, \varphi_i)\}_{(i \in I)}$, where I is a non-empty countable set, and such that the following conditions hold:

(1) $\varphi_i(U_i) \subseteq \mathbb{R}^n$, for all i.

Manifolds: Formal Definition

A C^k n**-atlas** is a family of charts, $\{(U_i, \varphi_i)\}_{(i \in I)}$, where I is a non-empty countable set, and such that the following conditions hold:

(1) $\varphi_i(U_i) \subseteq \mathbb{R}^n$, for all i.

(2) $M = \bigcup_{i \in I} U_i$.

Manifolds: Formal Definition

A C^k n-**atlas** is a family of charts, $\{(U_i, \varphi_i)\}_{(i \in I)}$, where I is a non-empty countable set, and such that the following conditions hold:

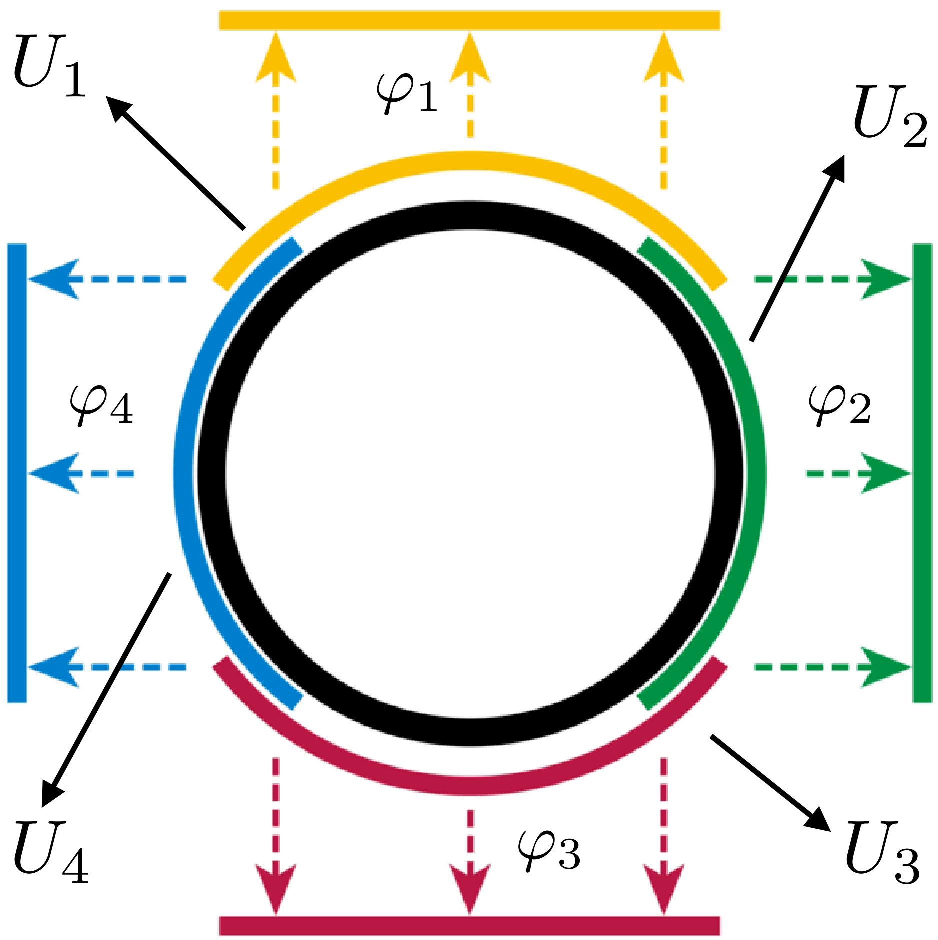
(1) $\varphi_i(U_i) \subseteq \mathbb{R}^n$, for all i.

(2) $M = \displaystyle\bigcup_{i \in I} U_i$.

(3) Whenever $U_i \cap U_j \neq \emptyset$, the transition function φ_{ji} (resp. φ_{ij}) is a C^k diffeomorphism.

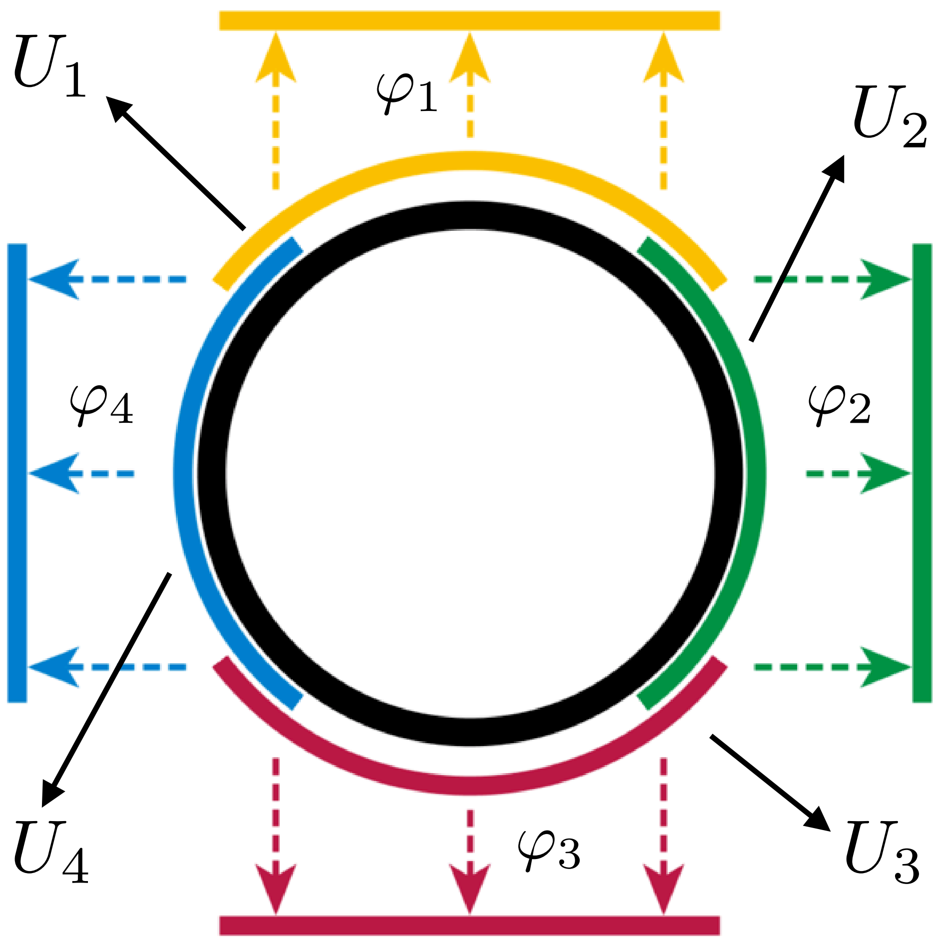
Manifolds: Formal Definition

Manifolds: Formal Definition



Atlas: $\{(U_1, \varphi_1), (U_2, \varphi_2), (U_3, \varphi_3), (U_4, \varphi_4)\}$

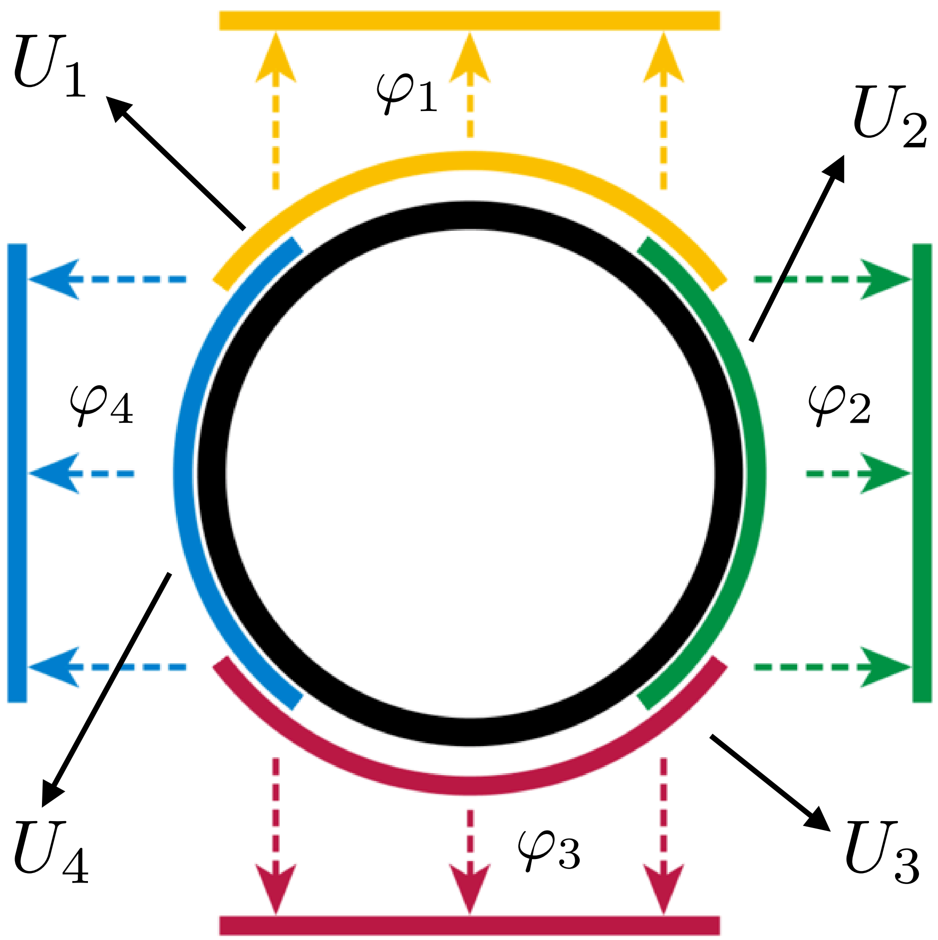
Manifolds: Formal Definition



$$M = \bigcup_{i=1}^{4} U_i$$

Atlas: $\{(U_1, \varphi_1), (U_2, \varphi_2), (U_3, \varphi_3), (U_4, \varphi_4)\}$

Manifolds: Formal Definition



$$M = \bigcup_{i=1}^{4} U_i$$

φ_i is a C^k diffeomorphism

Atlas: $\{(U_1, \varphi_1), (U_2, \varphi_2), (U_3, \varphi_3), (U_4, \varphi_4)\}$

Manifolds: Formal Definition

Manifolds: Formal Definition

The existence of a C^k atlas on a topological space, M, is sufficient to establish that M is an n-dimensional C^k manifold, but...

Manifolds: Formal Definition

The existence of a C^k atlas on a topological space, M, is sufficient to establish that M is an n-dimensional C^k manifold, but...

- there may be many choice of atlases;

Manifolds: Formal Definition

The existence of a C^k atlas on a topological space, M, is sufficient to establish that M is an n-dimensional C^k manifold, but...

- there may be many choice of atlases;
- we get around this problem by defining a notion of atlas compatibility;

Manifolds: Formal Definition

The existence of a C^k atlas on a topological space, M, is sufficient to establish that M is an n-dimensional C^k manifold, but...

- there may be many choice of atlases;
- we get around this problem by defining a notion of atlas compatibility;
- this notion induces an equivalence relation of atlases on M;

Manifolds: Formal Definition

The existence of a C^k atlas on a topological space, M, is sufficient to establish that M is an n-dimensional C^k manifold, but...

- there may be many choice of atlases;
- we get around this problem by defining a notion of atlas compatibility;
- this notion induces an equivalence relation of atlases on M;
- the set of all charts compatible with a given atlas is a maximum atlas in its class.

Manifolds: Formal Definition

Manifolds: Formal Definition

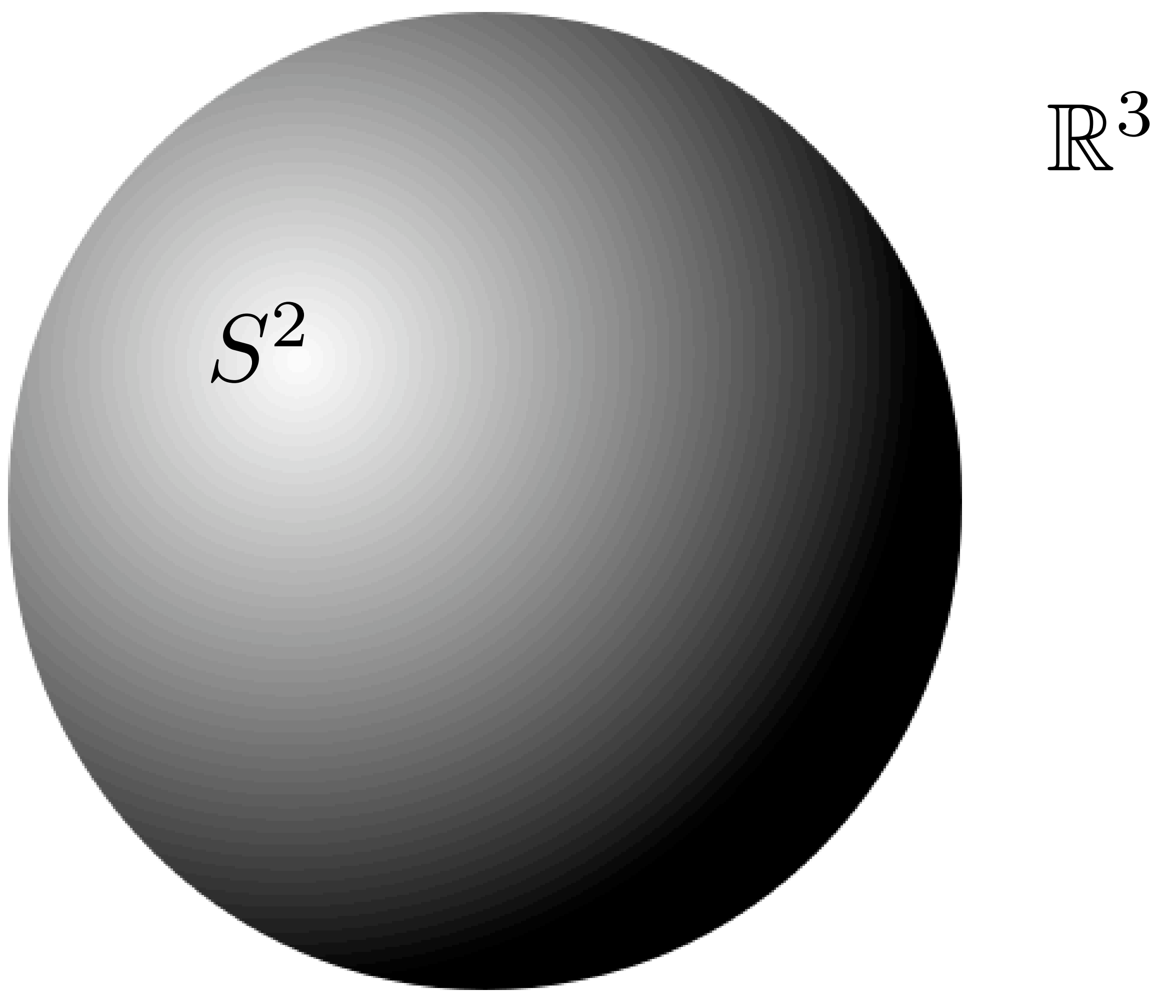
To avoid pathological cases and to ensure that a manifold is always embeddable in $\mathbb{R}^n$, for some $n \geq 1$, we further require that the topology of M be **Hausdorff** and **second-countable**.

Examples

Examples

- The sphere

$$S^{n-1} = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid x_1^2 + \cdots + x_n^2 = 1\} .$$



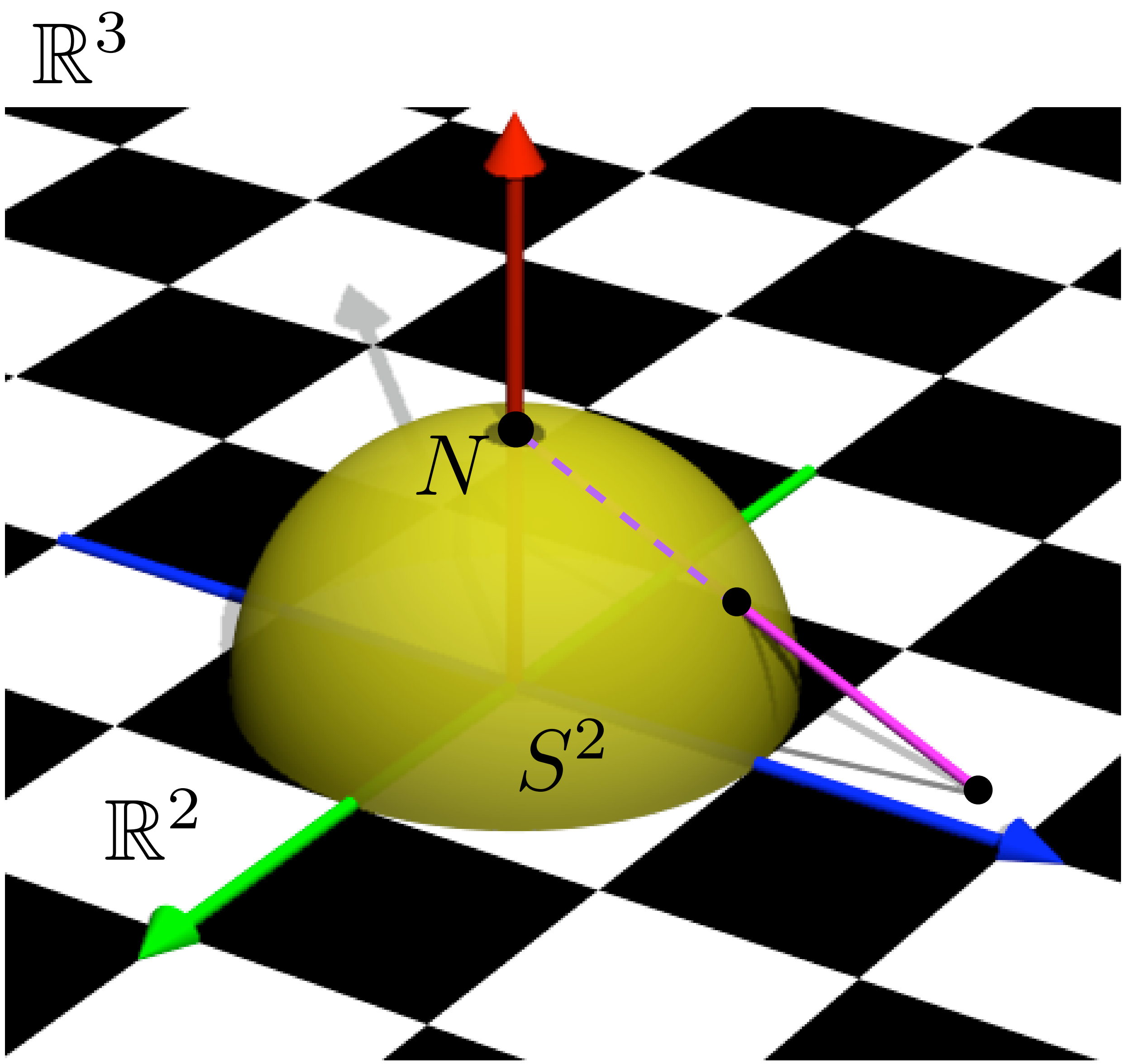
Examples

Examples

- We use stereographic projection from the north pole . . .

Examples

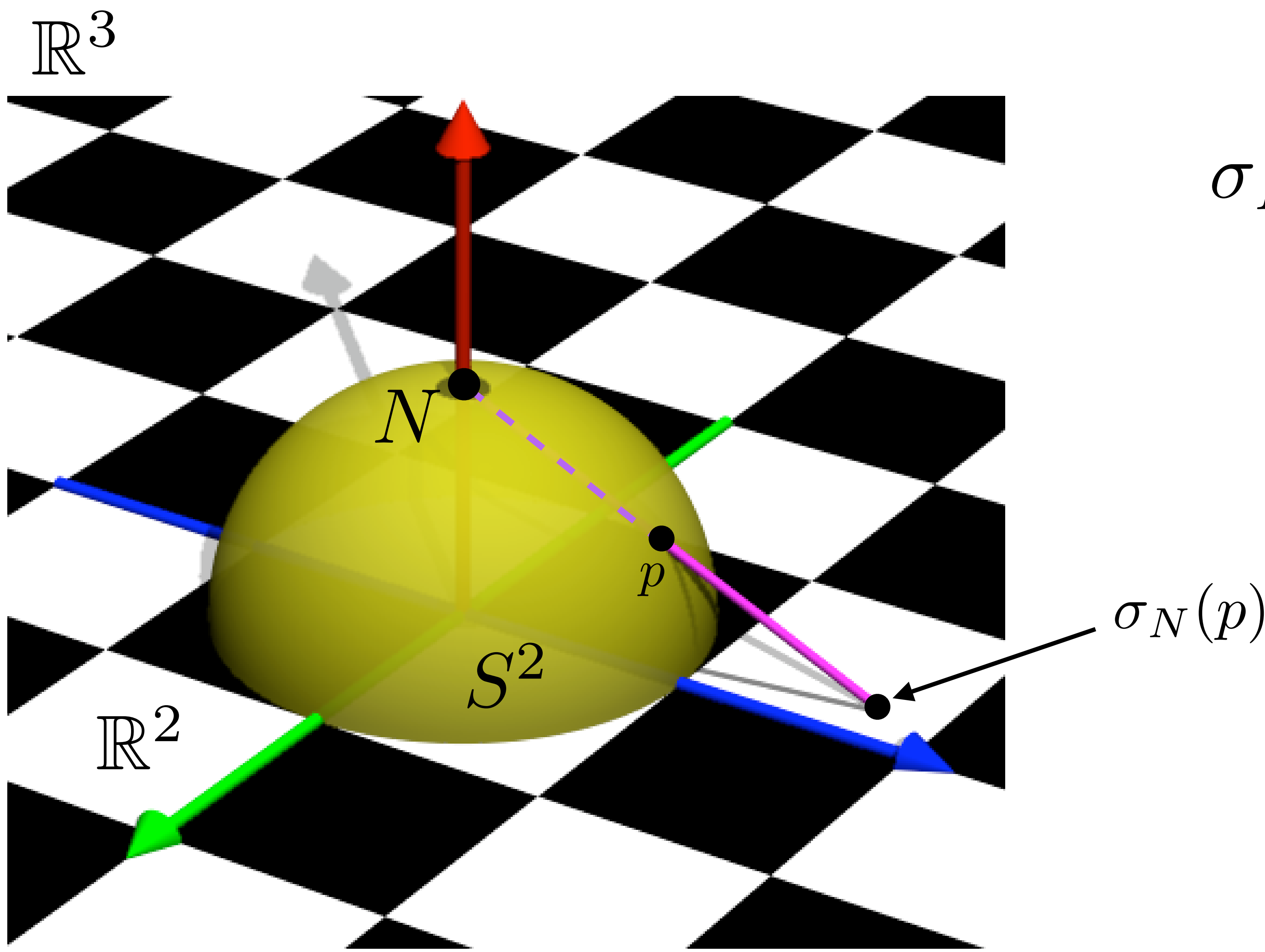
- We use stereographic projection from the north pole . . .



$$\sigma_N : S^n - \{N\} \longrightarrow \mathbb{R}^n$$

Examples

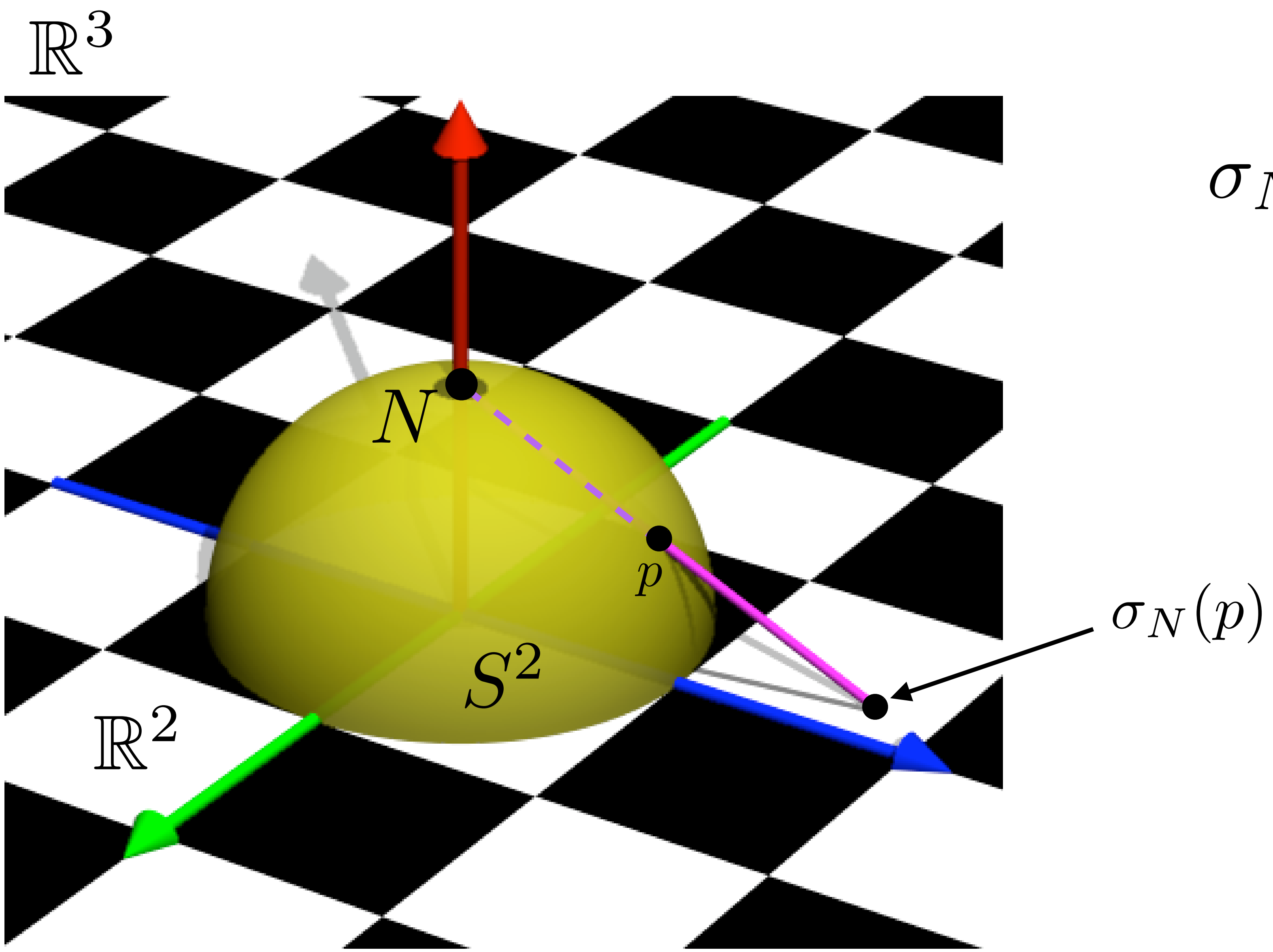
- We use stereographic projection from the north pole . . .



$$\sigma_N : S^n - \{N\} \longrightarrow \mathbb{R}^n$$

Examples

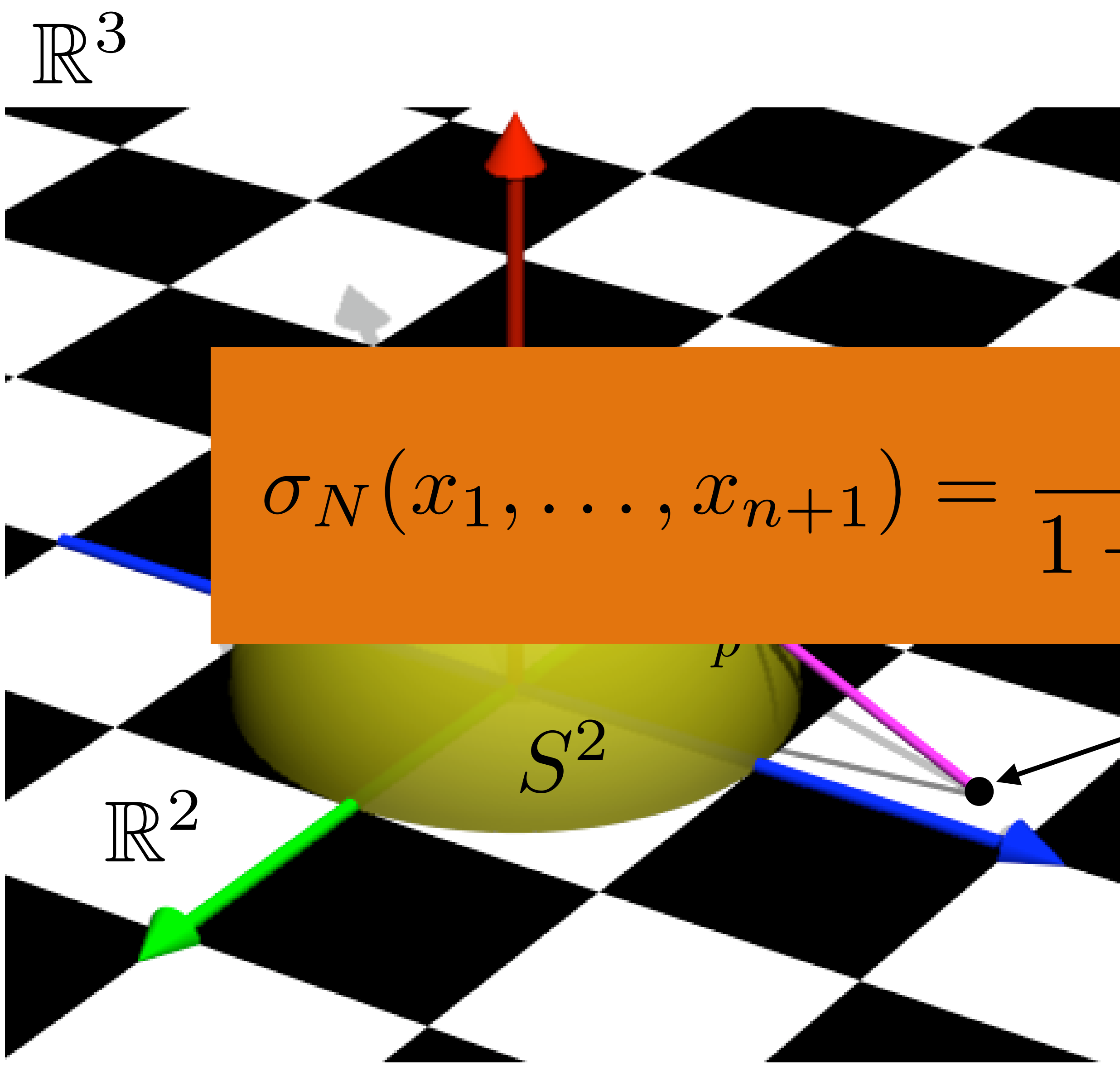
- We use stereographic projection from the north pole . . .



$$\sigma_N : S^n - \{N\} \longrightarrow \mathbb{R}^n$$

Examples

- We use stereographic projection from the north pole . . .



$$\sigma_N : S^n - \{N\} \longrightarrow \mathbb{R}^n$$

$$\sigma_N(x_1, \ldots, x_{n+1}) = \frac{1}{1 - x_{n+1}}(x_1, \ldots, x_n)$$

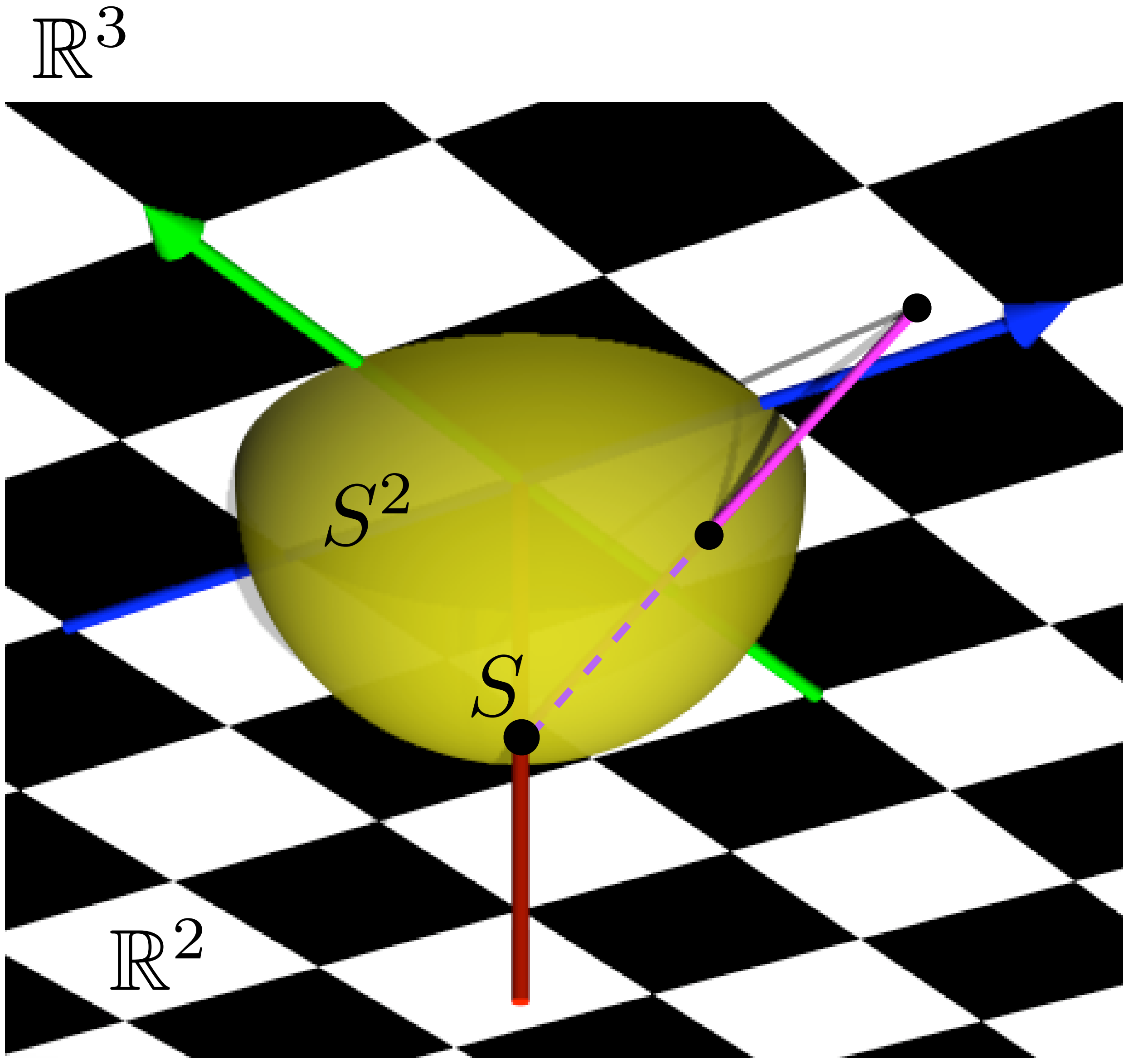
Examples

Examples

and from the south pole:

Examples

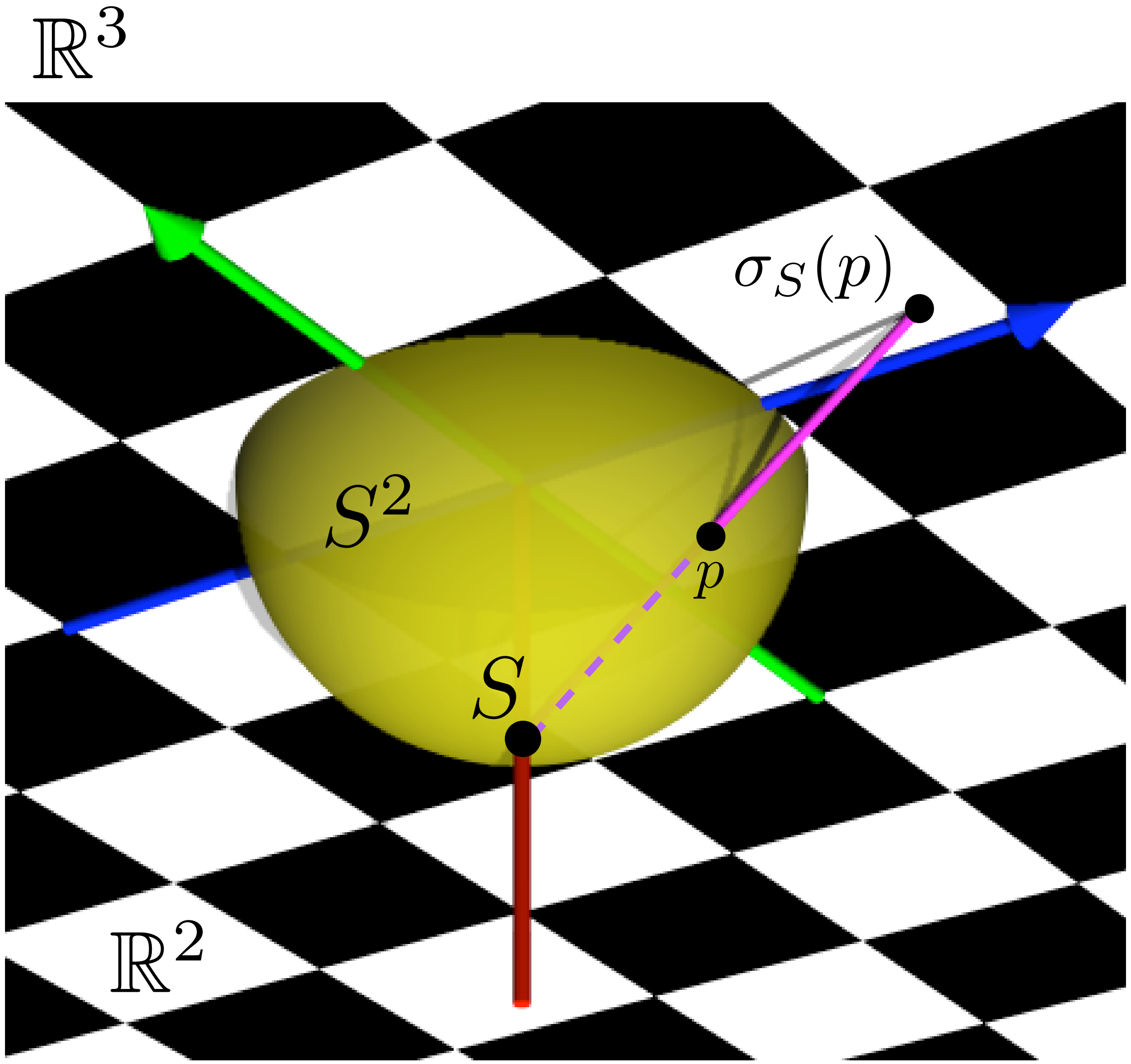
and from the south pole:



$$\sigma_S : S^n - \{S\} \longrightarrow \mathbb{R}^n$$

Examples

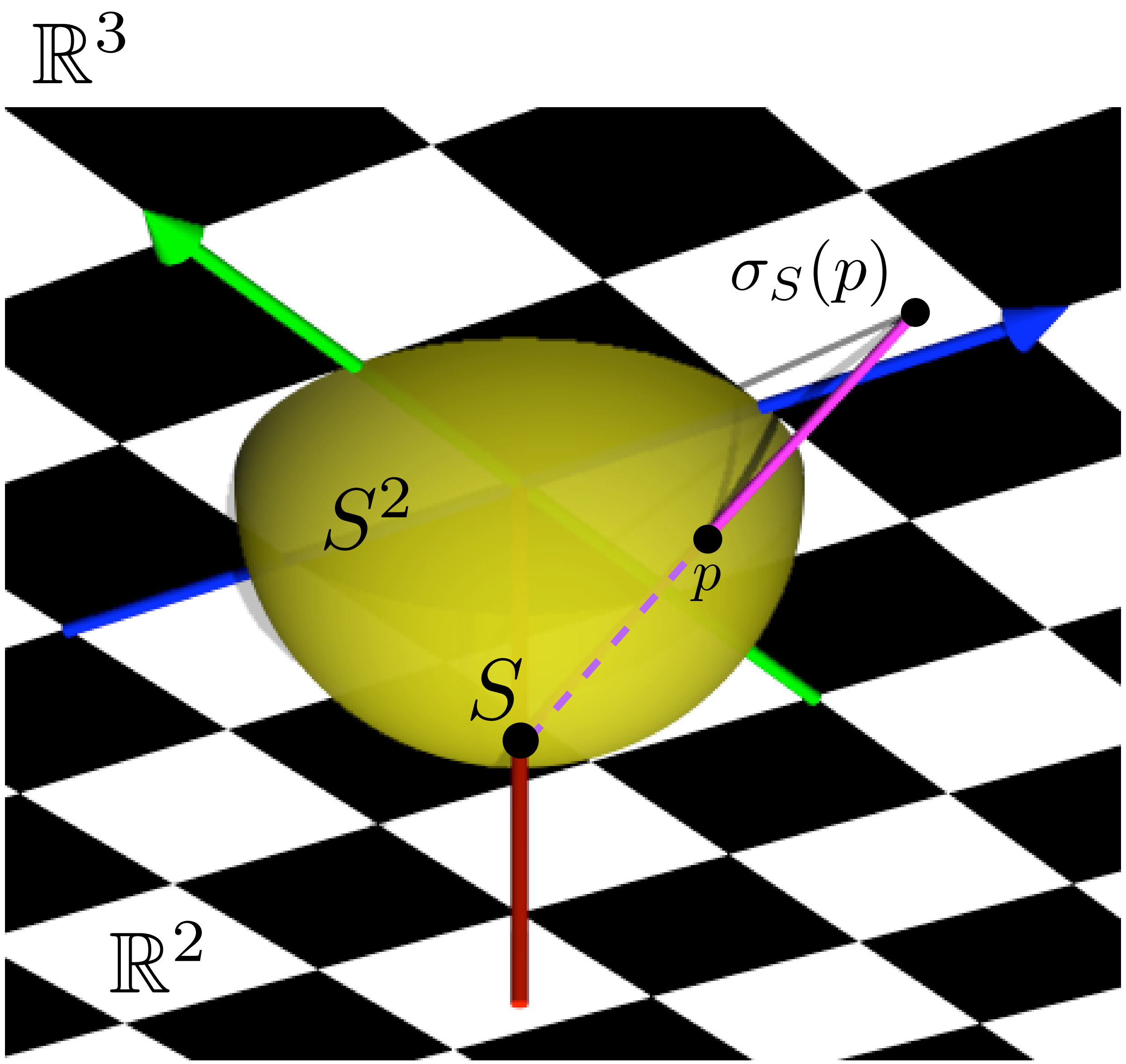
and from the south pole:



$$\sigma_S : S^n - \{S\} \longrightarrow \mathbb{R}^n$$

Examples

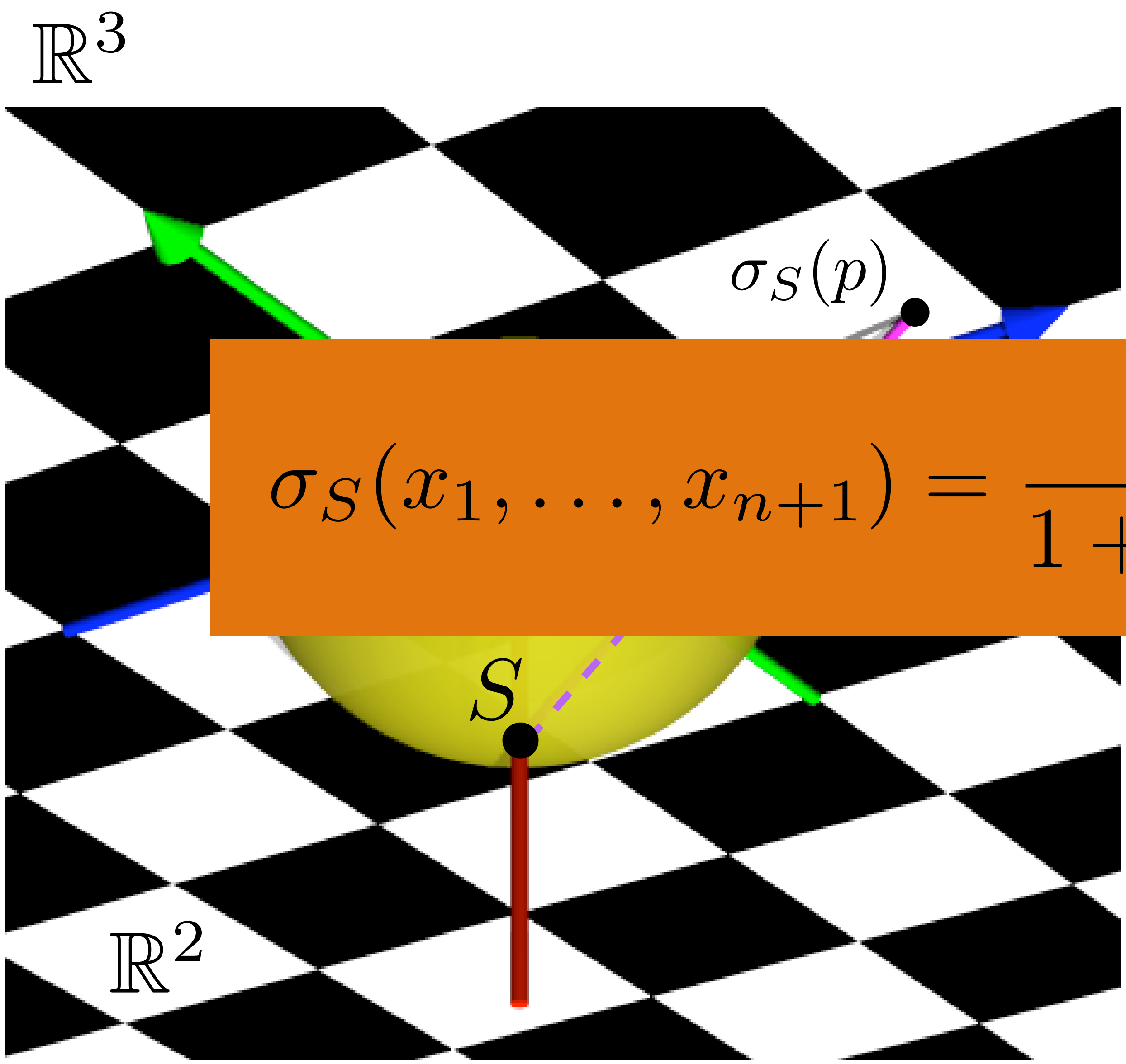
and from the south pole:



$$\sigma_S : S^n - \{S\} \longrightarrow \mathbb{R}^n$$

Examples

and from the south pole:



$$\sigma_S : S^n - \{S\} \longrightarrow \mathbb{R}^n$$

$$\sigma_S(x_1, \ldots, x_{n+1}) = \frac{1}{1 + x_{n+1}}(x_1, \ldots, x_n)$$

Examples

Examples

- Inverse stereographic projections:

Examples

- Inverse stereographic projections:

$$\sigma_N^{-1}(x_1, \ldots, x_n) = \frac{1}{\left(\sum_{i=1}^n x_i^2\right) + 1} \left(2x_1, \ldots, 2x_n, \left(\sum_{i=1}^n x_i^2\right) - 1\right)$$

Examples

- Inverse stereographic projections:

$$\sigma_N^{-1}(x_1, \ldots, x_n) = \frac{1}{\left(\sum_{i=1}^n x_i^2\right) + 1} \left(2x_1, \ldots, 2x_n, \left(\sum_{i=1}^n x_i^2\right) - 1\right)$$

and

Examples

- Inverse stereographic projections:

$$\sigma_N^{-1}(x_1, \ldots, x_n) = \frac{1}{\left(\sum_{i=1}^n x_i^2\right) + 1} \left(2x_1, \ldots, 2x_n, \left(\sum_{i=1}^n x_i^2\right) - 1\right)$$

and

$$\sigma_S^{-1}(x_1, \ldots, x_n) = \frac{1}{\left(\sum_{i=1}^n x_i^2\right) + 1} \left(2x_1, \ldots, 2x_n, -\left(\sum_{i=1}^n x_i^2\right) + 1\right).$$

Examples

Examples

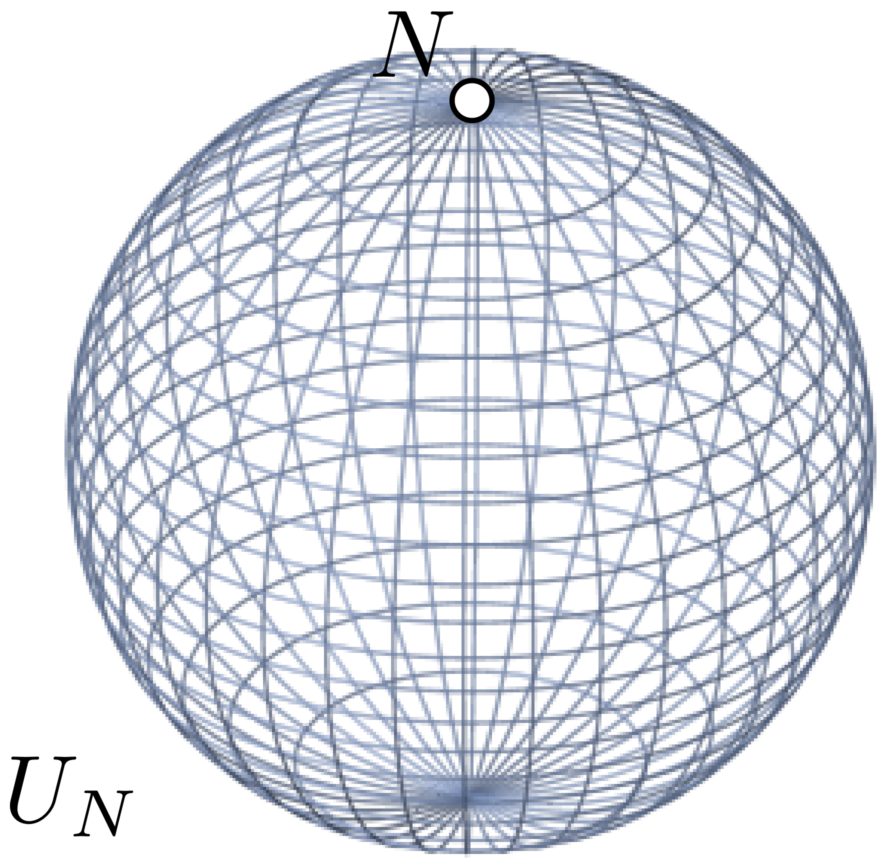
- Consider the open cover consisting of

$$U_N = S^n - \{N\} \qquad \text{and} \qquad U_S = S^n - \{S\}\,.$$

Examples

- Consider the open cover consisting of

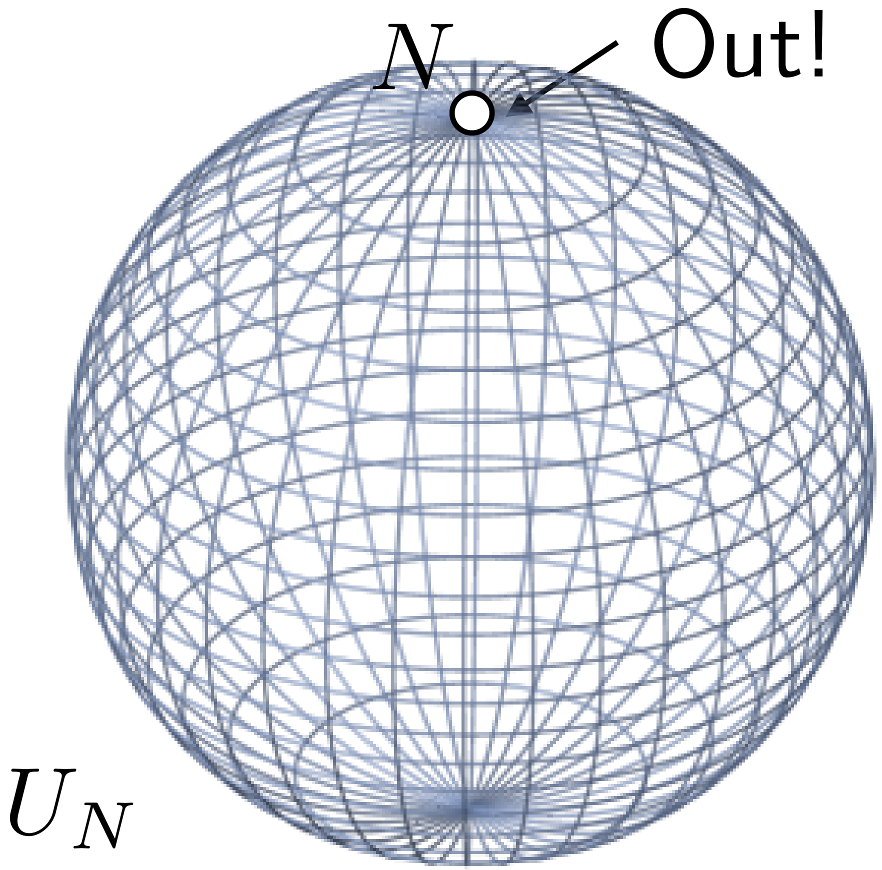
$$U_N = S^n - \{N\} \qquad \text{and} \qquad U_S = S^n - \{S\}\,.$$



Examples

- Consider the open cover consisting of

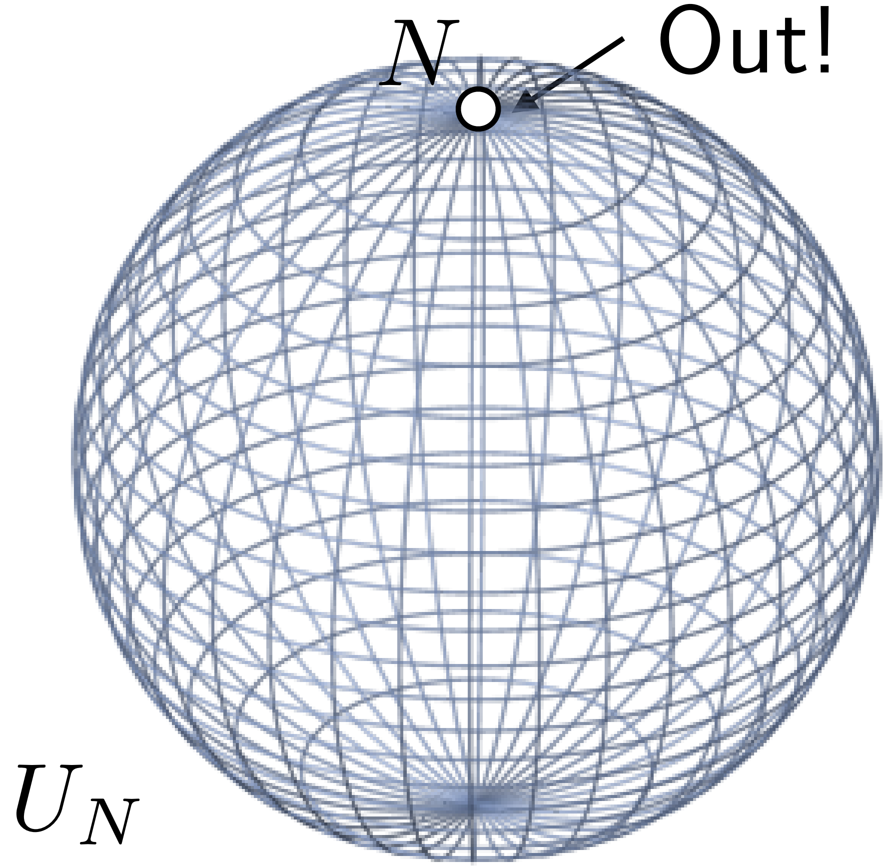
$$U_N = S^n - \{N\} \qquad \text{and} \qquad U_S = S^n - \{S\}\,.$$

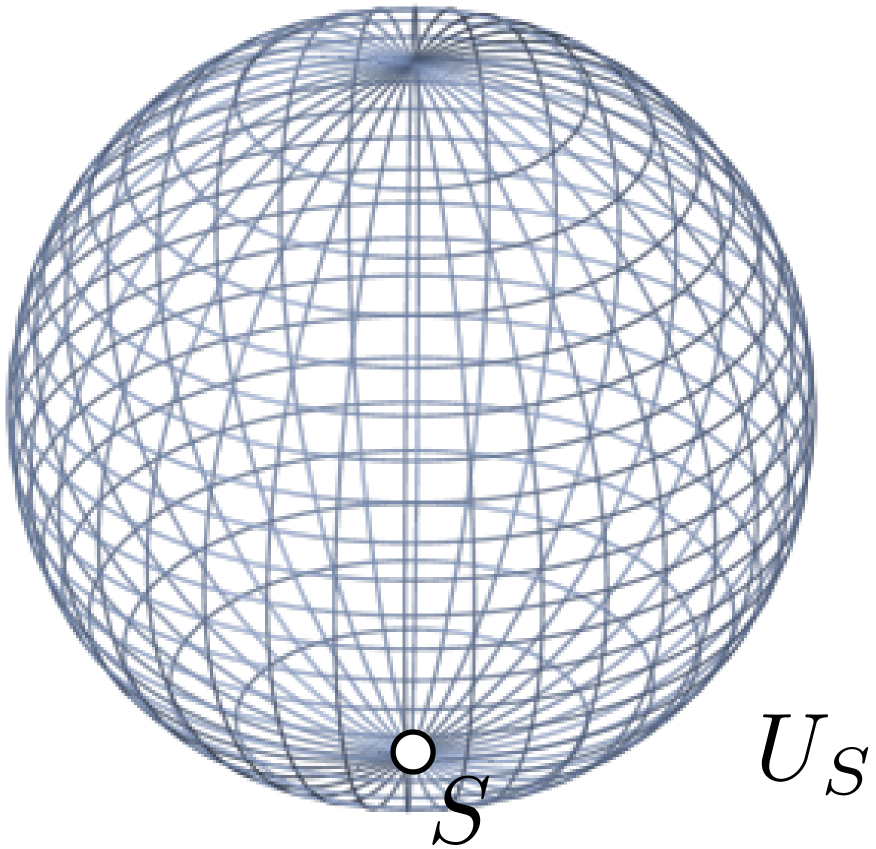


Examples

- Consider the open cover consisting of

$$U_N = S^n - \{N\} \qquad \text{and} \qquad U_S = S^n - \{S\}\,.$$

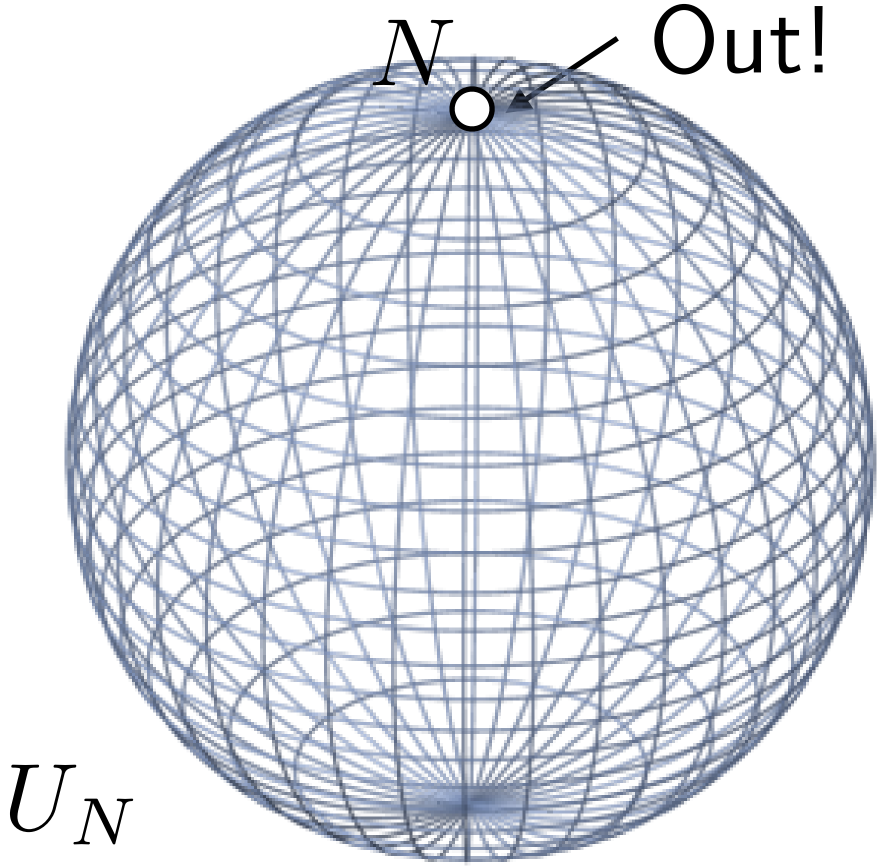


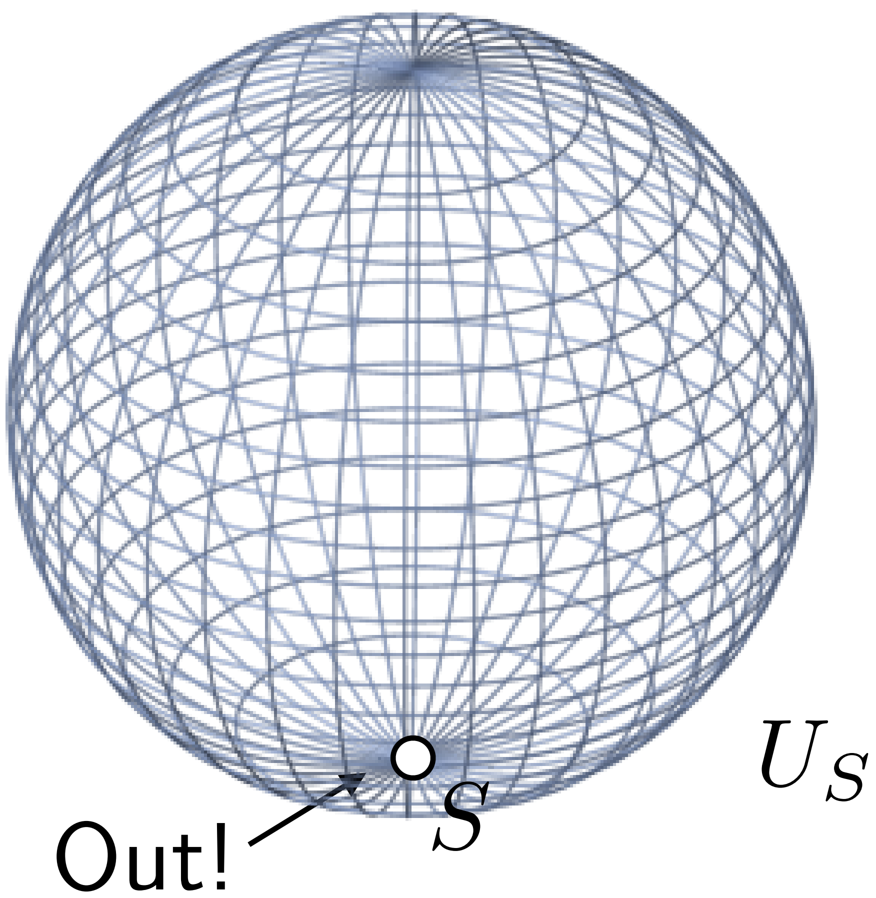


Examples

- Consider the open cover consisting of

$$U_N = S^n - \{N\} \qquad \text{and} \qquad U_S = S^n - \{S\}\,.$$





Examples

Examples

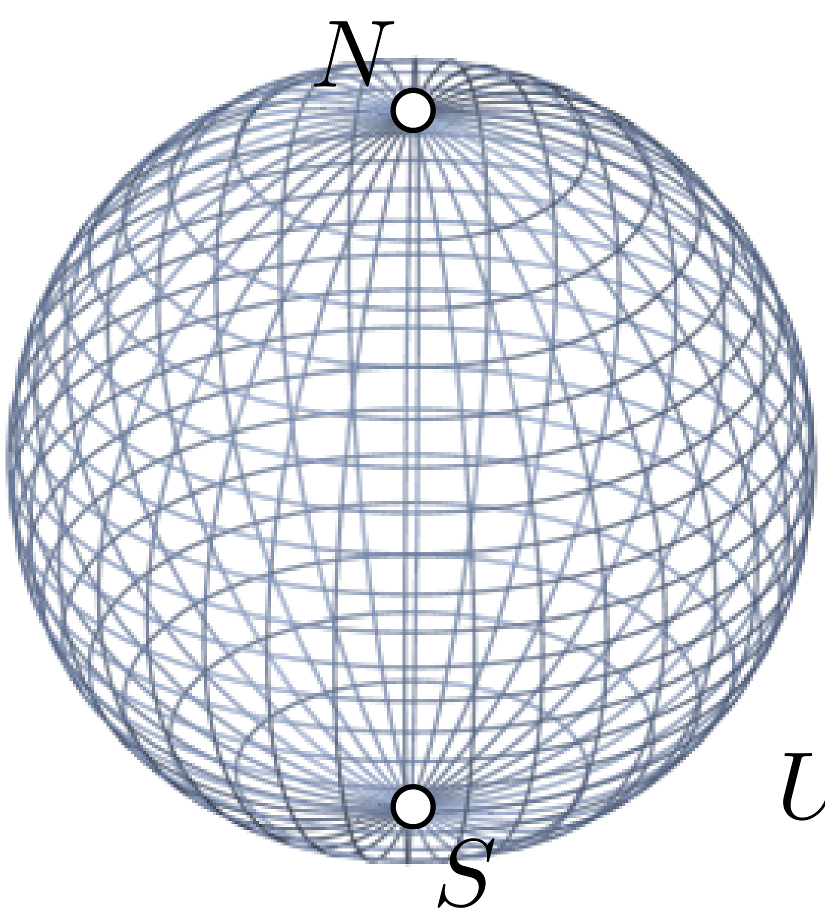
- On the overlap,

$$U_N \cap U_S = S^n - \{N, S\} \, .$$

Examples

- On the overlap,

$$U_N \cap U_S = S^n - \{N, S\}\,.$$

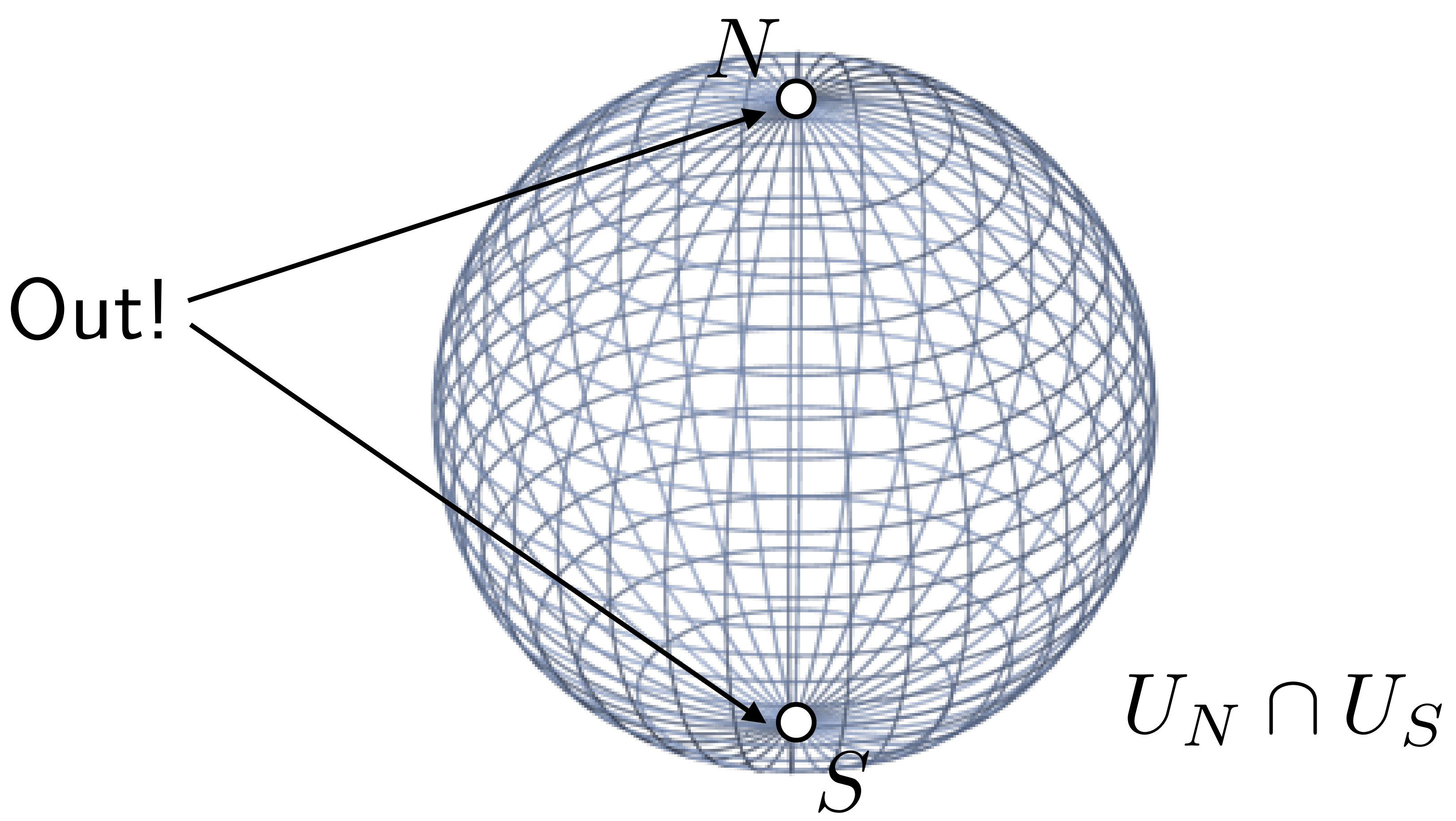


$U_N \cap U_S$

Examples

- On the overlap,

$$U_N \cap U_S = S^n - \{N, S\}\,.$$



Examples

Examples

- The transition maps

$$\sigma_S \circ \sigma_N^{-1} = \sigma_N \circ \sigma_S^{-1}$$

are given by

$$(x_1, \ldots, x_n) \mapsto \frac{1}{\sum_{i=1}^n x_i^2}(x_1, \ldots, x_n)\,.$$

Examples

Examples

- Consequently,

$$(U_N, \sigma_N) \quad \text{and} \quad (U_S, \sigma_S)$$

form a smooth atlas for S^n .

Examples

- Consequently,

$$(U_N, \sigma_N) \quad \text{and} \quad (U_S, \sigma_S)$$

form a smooth atlas for S^n .

- So, the sphere is a smooth manifold.

Examples

Examples

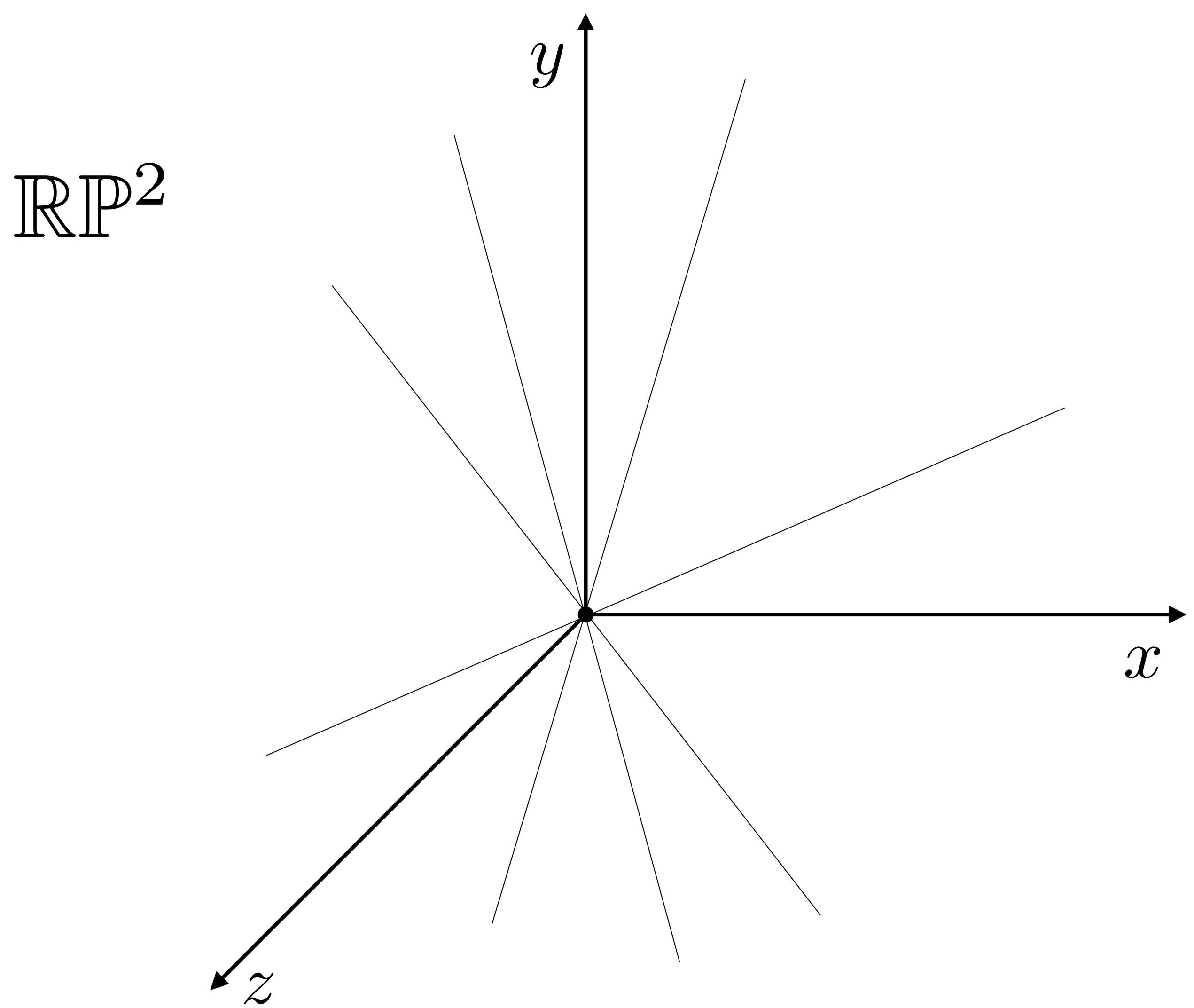
- The real projective space, $\mathbb{RP}^n$.

Examples

- The real projective space, $\mathbb{RP}^n$.
- This is the space of all lines through the origin of $\mathbb{R}^{n+1}$.

Examples

- The real projective space, $\mathbb{RP}^n$.
- This is the space of all lines through the origin of $\mathbb{R}^{n+1}$.



Examples

Examples

- Equivalent definition:

  Define an equivalence relation on nonzero vector in $\mathbb{R}^{n+1}$ as follows:

$$u \sim v \quad \text{iff} \quad v = \lambda u, \text{ for some } \lambda \neq 0 \in \mathbb{R}\,.$$

Examples

- Equivalent definition:

  Define an equivalence relation on nonzero vector in $\mathbb{R}^{n+1}$ as follows:

  $$u \sim v \quad \text{iff} \quad v = \lambda u, \text{ for some } \lambda \neq 0 \in \mathbb{R}\,.$$

- Denote the equivalence class of $(x_1, \ldots, x_{n+1})$ by

  $$(x_1 : \cdots : x_{n+1})$$

  also called homogeneous coordinates.

Examples

Examples

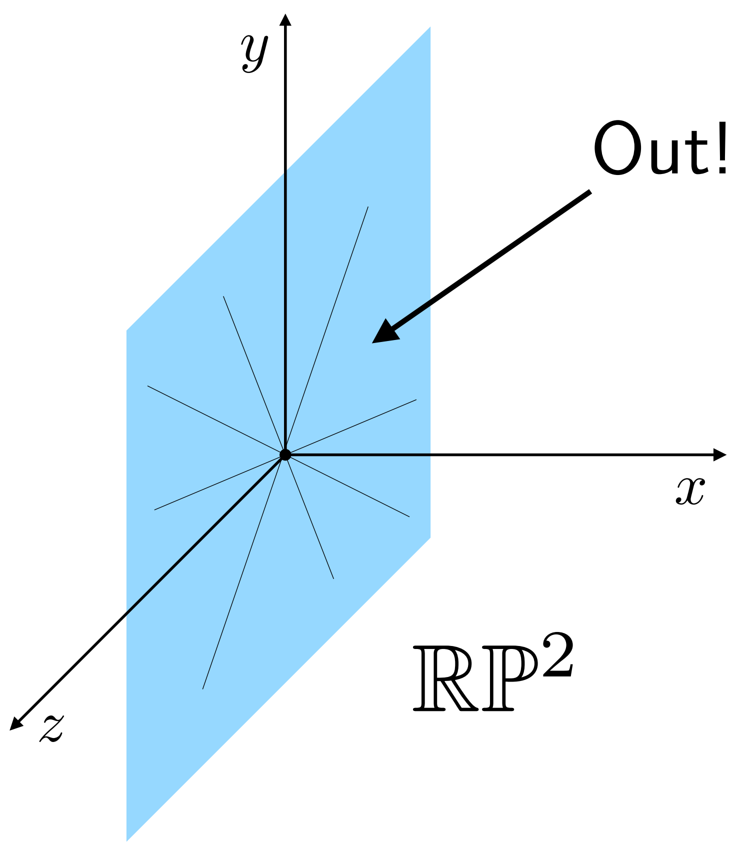
- Let

$$U_i = \{(x_1 : \cdots : x_{n+1}) \in \mathbb{RP}^n \mid x_i \neq 0\} .$$

Examples

- Let

$$U_i = \{(x_1 : \cdots : x_{n+1}) \in \mathbb{RP}^n \mid x_i \neq 0\}.$$

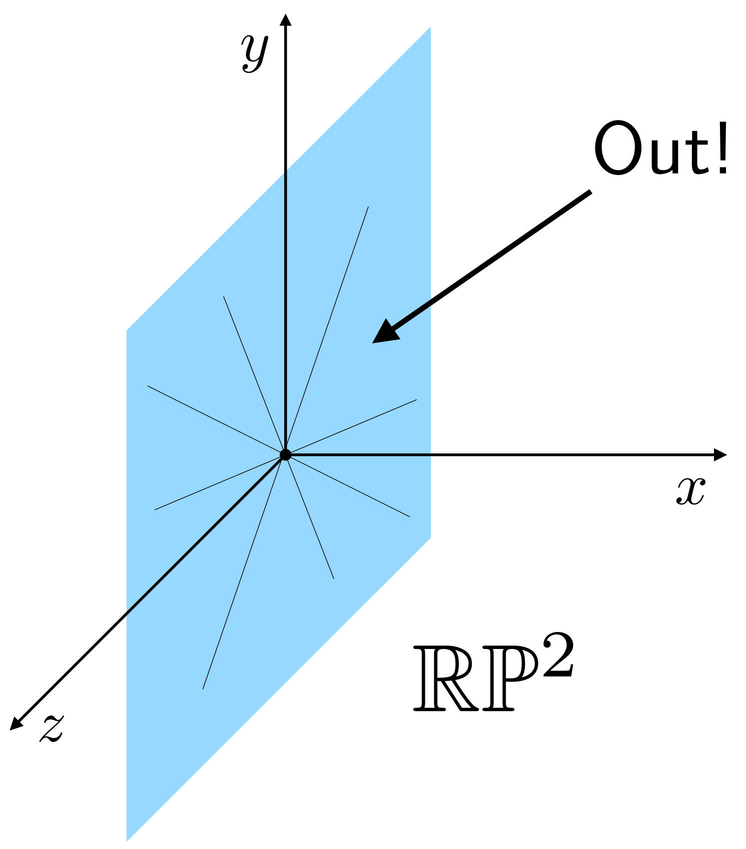


$U_1 = \{(x : y : z) \in \mathbb{RP}^2 \mid x \neq 0\}$

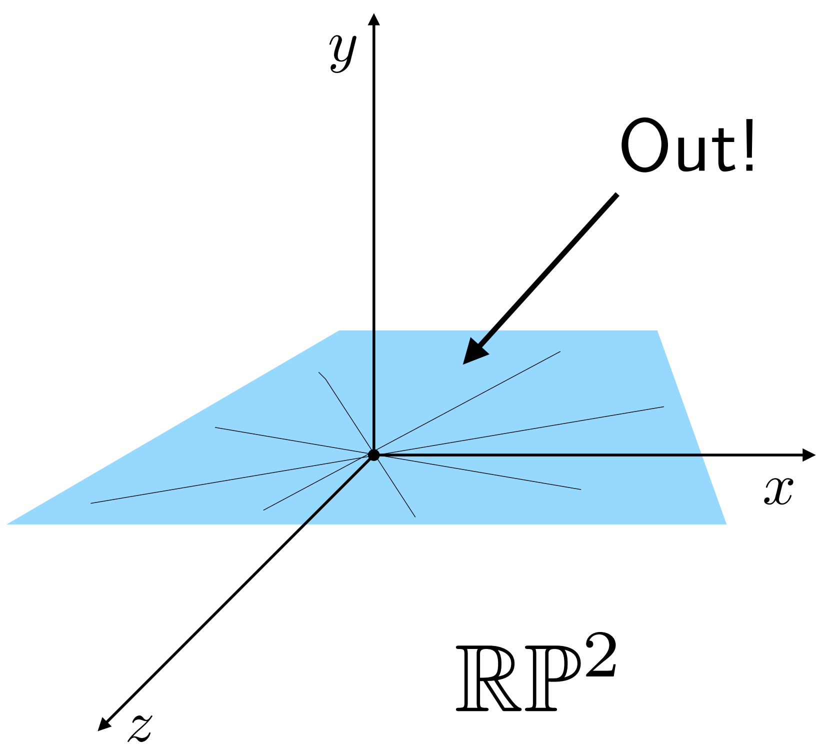
Examples

- Let

$$U_i = \{(x_1 : \cdots : x_{n+1}) \in \mathbb{RP}^n \mid x_i \neq 0\}\,.$$



$U_1 = \{(x : y : z) \in \mathbb{RP}^2 \mid x \neq 0\}$

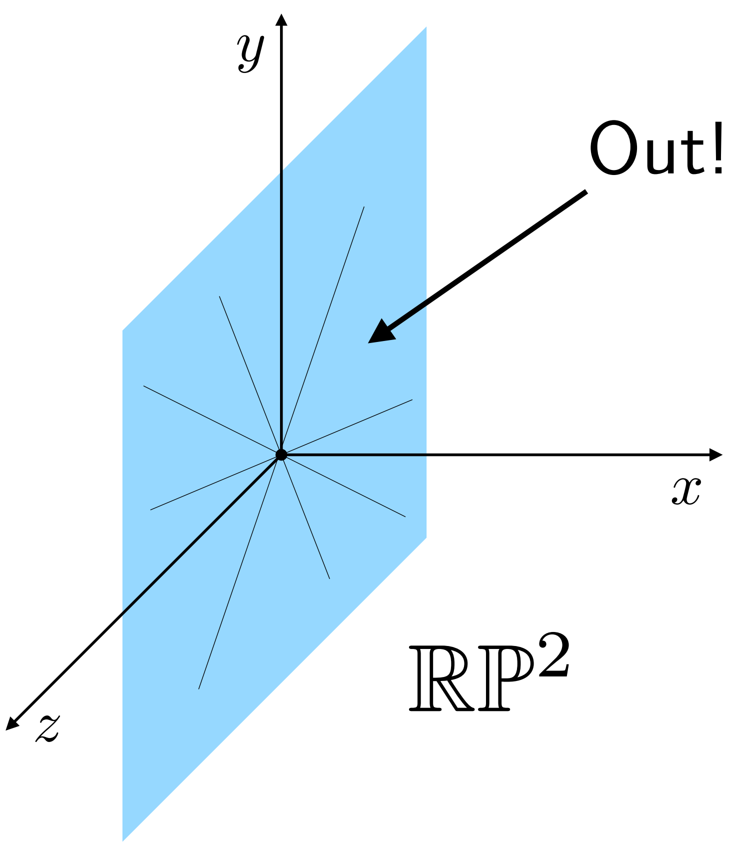


$U_2 = \{(x : y : z) \in \mathbb{RP}^2 \mid y \neq 0\}$

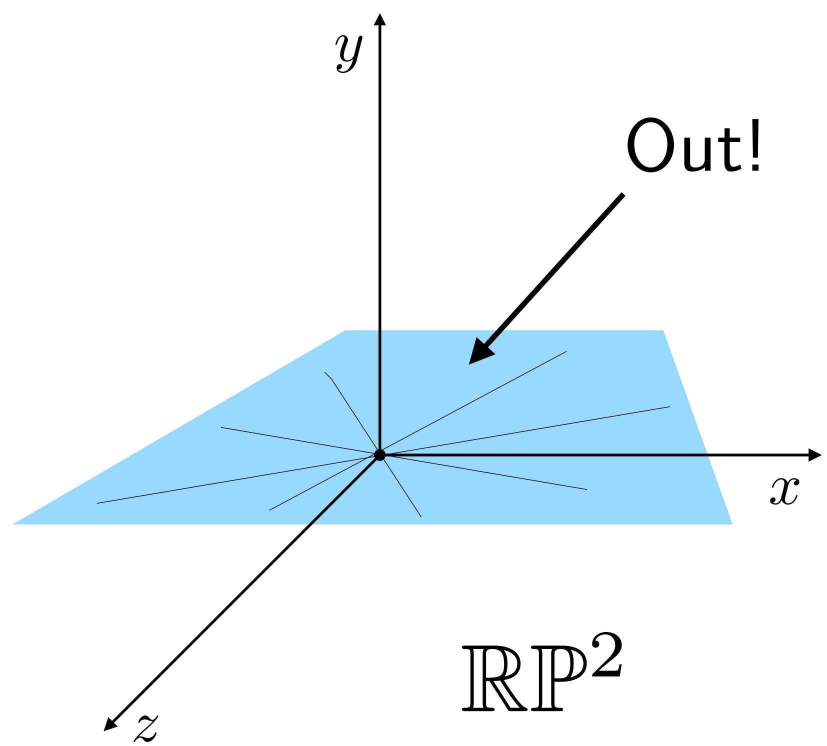
Examples

- Let

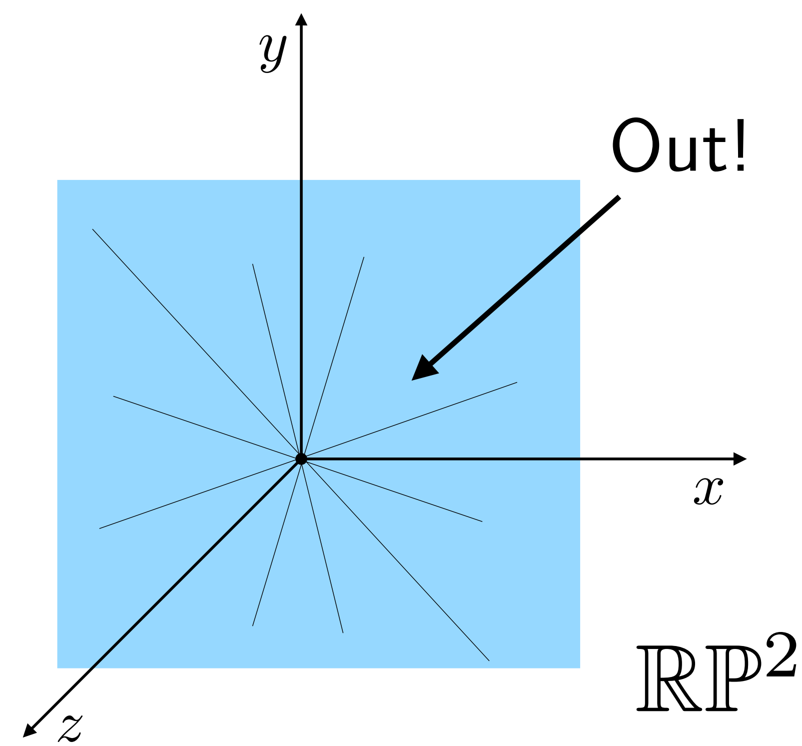
$$U_i = \{(x_1 : \cdots : x_{n+1}) \in \mathbb{RP}^n \mid x_i \neq 0\}\,.$$



$U_1 = \{(x : y : z) \in \mathbb{RP}^2 \mid x \neq 0\}$



$U_2 = \{(x : y : z) \in \mathbb{RP}^2 \mid y \neq 0\}$



$U_3 = \{(x : y : z) \in \mathbb{RP}^2 \mid z \neq 0\}$

Examples

Examples

- Define $\varphi_i : U_i \to \mathbb{R}^n$ by

$$\varphi_i(x_1 : \cdots : x_{n+1}) = \left(\frac{x_1}{x_i}, \ldots, \frac{x_{i-1}}{x_i}, \frac{x_{i+1}}{x_i}, \ldots, \frac{x_{n+1}}{x_i} \right).$$

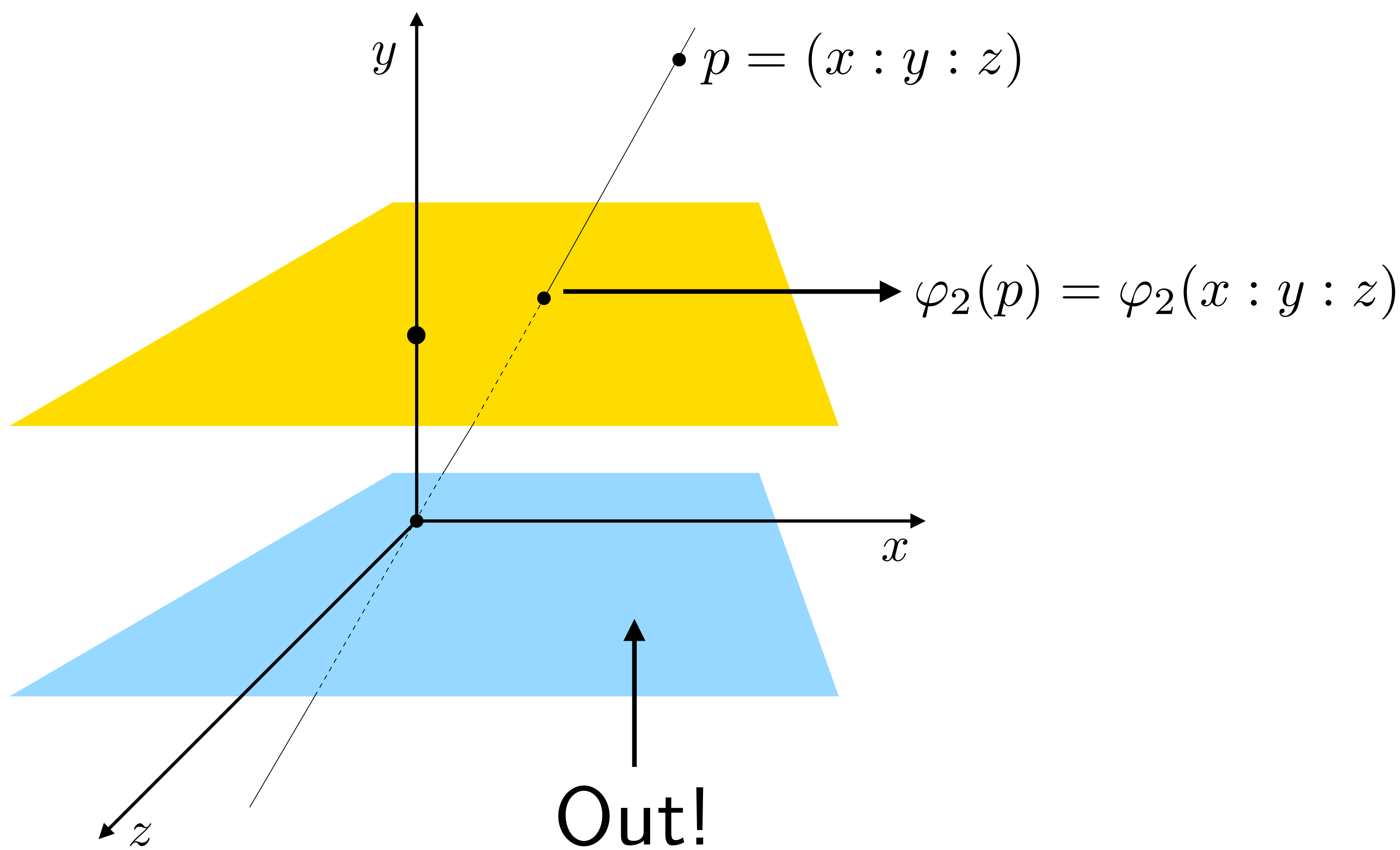
Examples

- Define $\varphi_i : U_i \longrightarrow \mathbb{R}^n$ by

$$\varphi_i(x_1 : \cdots : x_{n+1}) = \left(\frac{x_1}{x_i}, \ldots, \frac{x_{i-1}}{x_i}, \frac{x_{i+1}}{x_i}, \ldots, \frac{x_{n+1}}{x_i} \right).$$

$\varphi_2(x : y : z)$

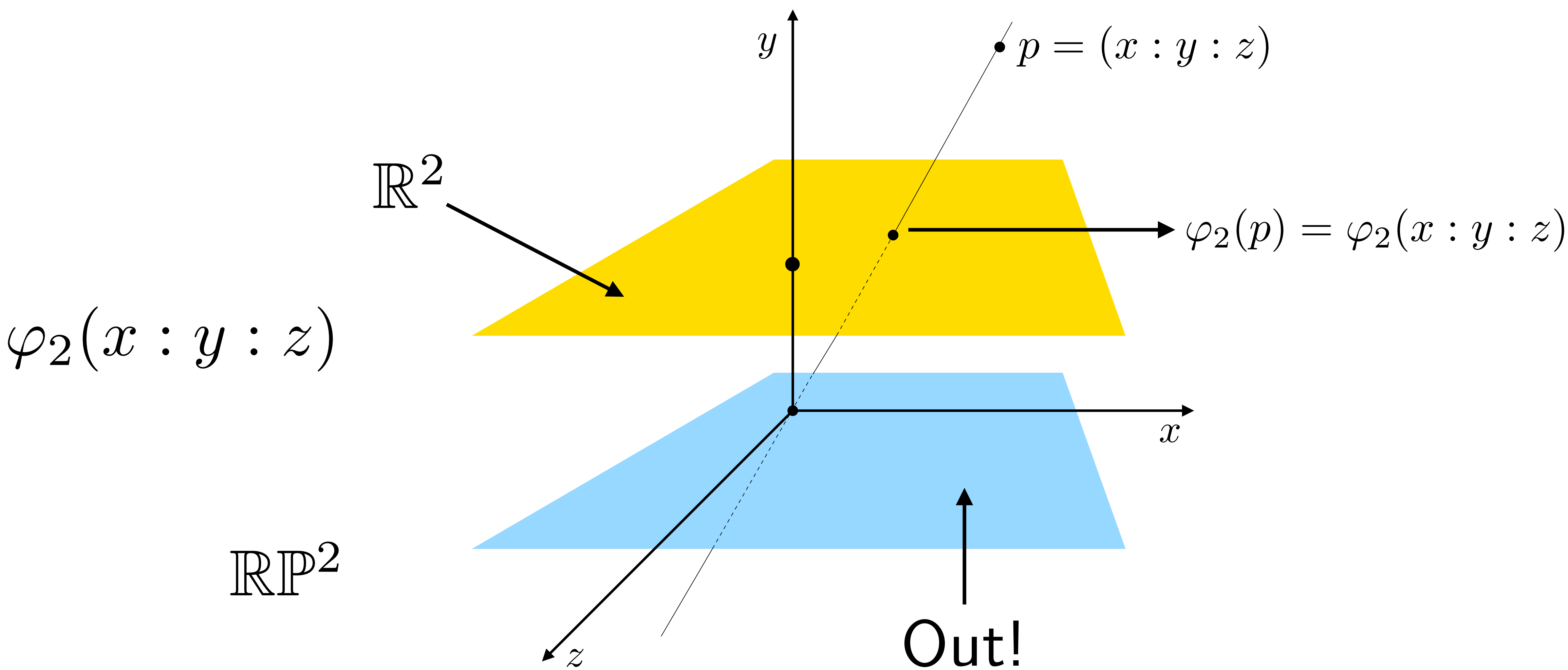
$\mathbb{RP}^2$



Examples

- Define $\varphi_i : U_i \longrightarrow \mathbb{R}^n$ by

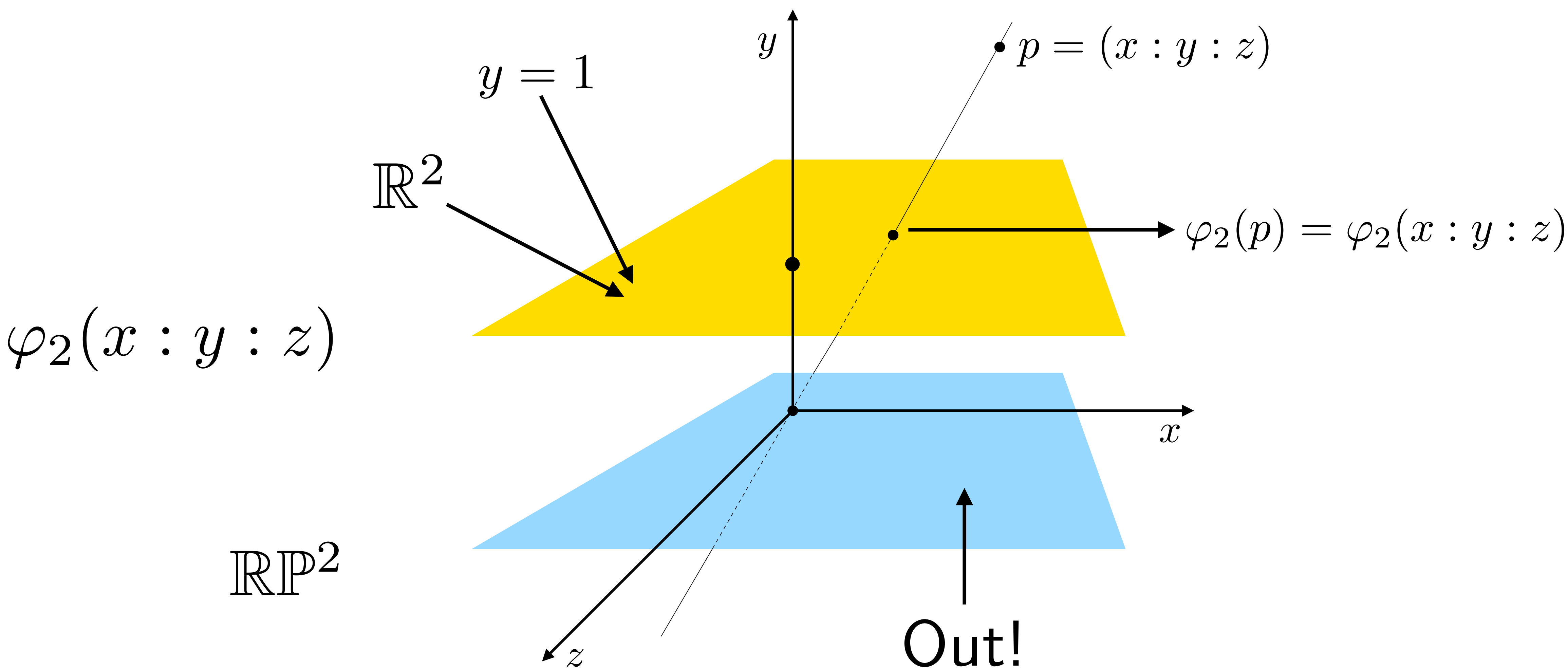
$$\varphi_i(x_1 : \cdots : x_{n+1}) = \left(\frac{x_1}{x_i}, \ldots, \frac{x_{i-1}}{x_i}, \frac{x_{i+1}}{x_i}, \ldots, \frac{x_{n+1}}{x_i}\right).$$



Examples

- Define $\varphi_i : U_i \longrightarrow \mathbb{R}^n$ by

$$\varphi_i(x_1 : \cdots : x_{n+1}) = \left(\frac{x_1}{x_i}, \ldots, \frac{x_{i-1}}{x_i}, \frac{x_{i+1}}{x_i}, \ldots, \frac{x_{n+1}}{x_i} \right) .$$



Examples

Examples

- The inverse maps are given by

$$\psi_i(x_1, \ldots, x_n) = (x_1 : \cdots : x_{i-1} : 1 : x_{i+1} : \cdots : x_n)\,.$$

Examples

- The inverse maps are given by

$$\psi_i(x_1, \ldots, x_n) = (x_1 : \cdots : x_{i-1} : 1 : x_{i+1} : \cdots : x_n)\,.$$

- On the overlap, $U_i \cap U_j$,

$$(\varphi_j \circ \varphi_i^{-1})(x_1, \ldots, x_n) = \left(\frac{x_1}{x_j}, \ldots, \frac{x_{i-1}}{x_j}, \frac{1}{x_j}, \frac{x_i}{x_j}, \ldots, \frac{x_{j-1}}{x_j}, \frac{x_{j+1}}{x_j}, \ldots, \frac{x_n}{x_j}\right)$$

Examples

- The inverse maps are given by

$$\psi_i(x_1, \ldots, x_n) = (x_1 : \cdots : x_{i-1} : 1 : x_{i+1} : \cdots : x_n)\,.$$

- On the overlap, $U_i \cap U_j$,

$$(\varphi_j \circ \varphi_i^{-1})(x_1, \ldots, x_n) = \left(\frac{x_1}{x_j}, \ldots, \frac{x_{i-1}}{x_j}, \frac{1}{x_j}, \frac{x_i}{x_j}, \ldots, \frac{x_{j-1}}{x_j}, \frac{x_{j+1}}{x_j}, \ldots, \frac{x_n}{x_j}\right)$$

- As these maps are smooth, real projective space is a smooth manifold.

Conclusions

Conclusions

- In the next part of the tutorial, we will show that a manifold can be **reconstructed** from its transition functions.

Conclusions

- In the next part of the tutorial, we will show that a manifold can be **reconstructed** from its transition functions.

- Such a construction was first proposed by Andre Weil around 1944 in his book, *Foundations of Algebraic Geometry*.

Conclusions

- In the next part of the tutorial, we will show that a manifold can be **reconstructed** from its transition functions.

- Such a construction was first proposed by Andre Weil around 1944 in his book, *Foundations of Algebraic Geometry*.

- A similar approach was used to construct fiber bundles in the 1950's (Steenrod).

Constructing Manifolds from Sets of Gluing Data

Jean Gallier
UPenn

Outline

- Motivations
- Sets of gluing data
- Transition functions
- The cocyle condition
- Parametric pseudo manifolds (PPM's)
- Conclusions

Motivations

Motivations

- Recall that **we want to define a surface** S that approximates the underlying surface, $|S_T|$, of a given polygonal surface (mesh), S_T.

Motivations

- Recall that **we want to define a surface** S that approximates the underlying surface, $|S_T|$, of a given polygonal surface (mesh), S_T.

- More specifically, we want to build a C^k two-dimensional manifold in $\mathbb{R}^3$.

Motivations

- Recall that **we want to define a surface** S that approximates the underlying surface, $|S_T|$, of a given polygonal surface (mesh), S_T.

- More specifically, we want to build a C^k two-dimensional manifold in $\mathbb{R}^3$.

- Our plan is to define S constructively by **building a manifold**.

Motivations

Motivations

A LITTLE PROBLEM:

Our definition of manifold is not constructive: it states what a manifold is by assuming it already exists! So, for our purposes, it is not useful!

Motivations

A LITTLE PROBLEM:

Our definition of manifold is not constructive: it states what a manifold is by assuming it already exists! So, for our purposes, it is not useful!

THE KEY IDEA:

The notion of a **set of gluing data**.

Sets of Gluing Data

Sets of Gluing Data

Let n and k be integers such that $n \geq 1$ and $k \geq 1$ (or $k = \infty$).

Sets of Gluing Data

Let n and k be integers such that $n \geq 1$ and $k \geq 1$ (or $k = \infty$).

A set of gluing data is a triple

$$\mathcal{G} = \big((\Omega_i)_{i \in I}, (\Omega_{ij})_{(i,j) \in I \times I}, (\varphi_{ji})_{(i,j) \in K \times K}\big)$$

satisfying the following properties, where I and K are countable sets and I is non-empty:

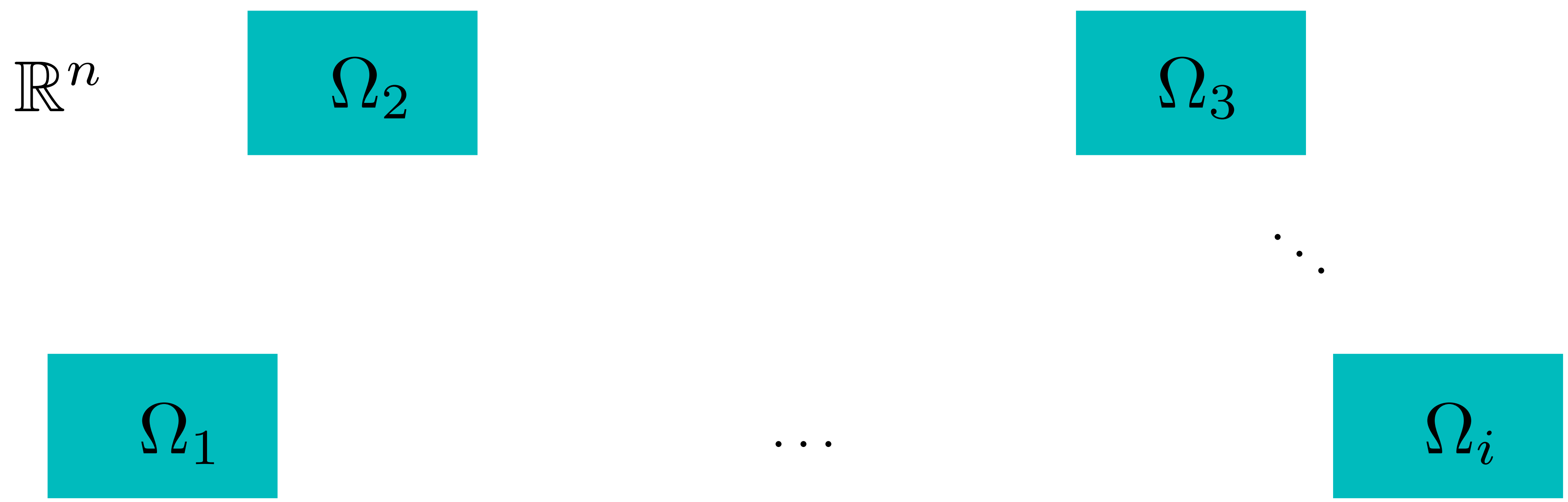
Sets of Gluing Data

Sets of Gluing Data

(1) For every $i \in I$, the set Ω_i is a non-empty open subset of $\mathbb{R}^n$ called **parametrization domain**, for short, p-**domain**, and the Ω_i are pairwise disjoint (i.e., $\Omega_i \cap \Omega_j = \emptyset$ for all $i \neq j$).

Sets of Gluing Data

(1) For every $i \in I$, the set Ω_i is a non-empty open subset of $\mathbb{R}^n$ called **parametrization domain**, for short, p-**domain**, and the Ω_i are pairwise disjoint (i.e., $\Omega_i \cap \Omega_j = \emptyset$ for all $i \neq j$).



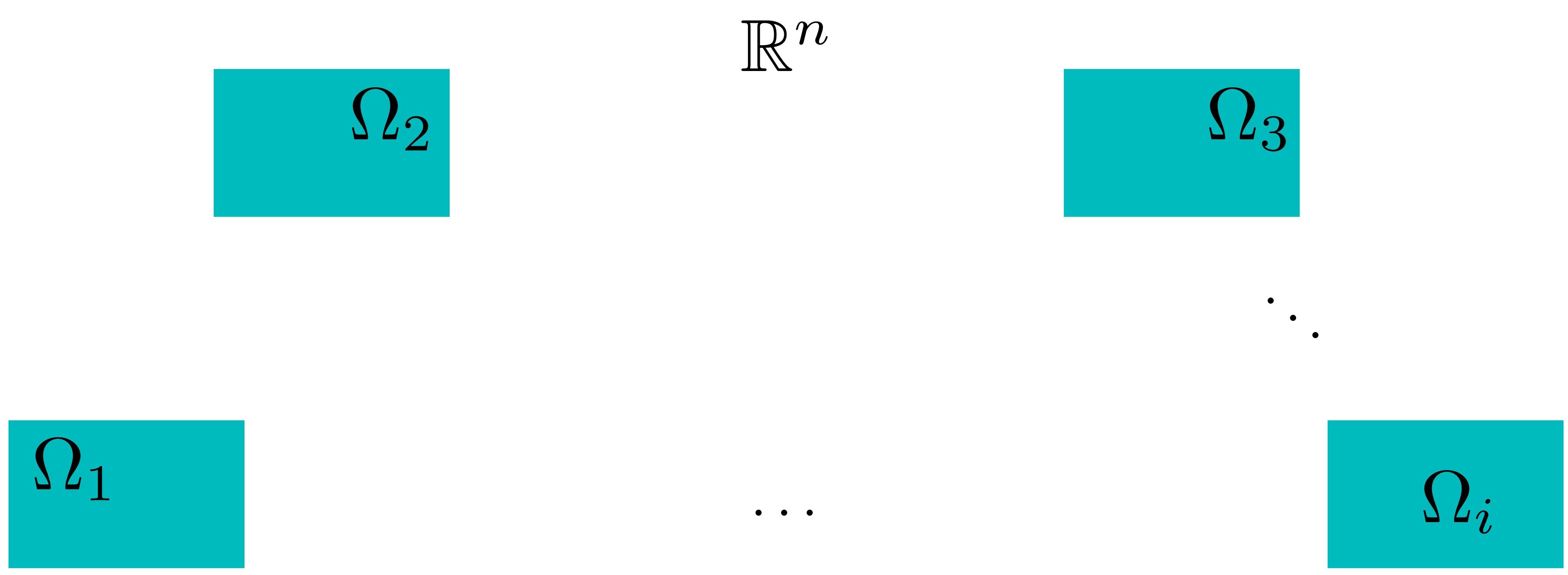
Sets of Gluing Data

Sets of Gluing Data

(2) For every pair $(i, j) \in I \times I$, the set Ω_{ij} is an open subset of Ω_i. Furthermore, $\Omega_{ii} = \Omega_i$, and $\Omega_{ji} \neq \emptyset$ if and only if $\Omega_{ij} \neq \emptyset$. Each non-empty Ω_{ij} (with $i \neq j$) is called **gluing domain**.

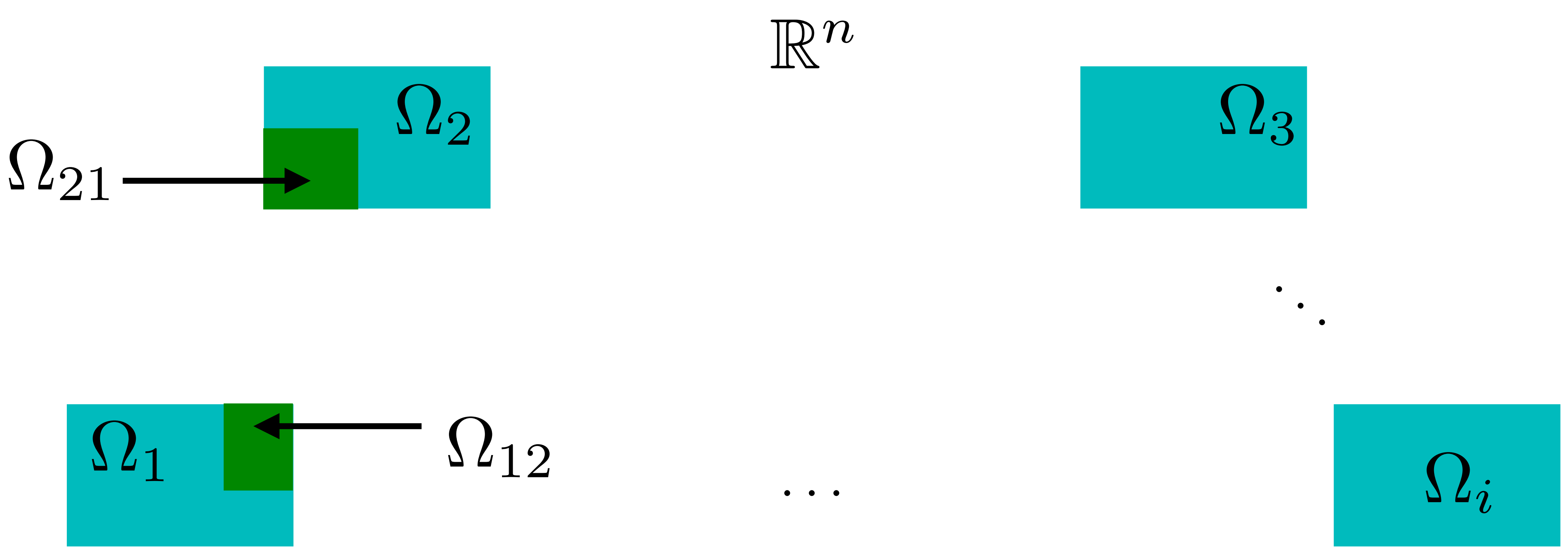
Sets of Gluing Data

(2) For every pair $(i, j) \in I \times I$, the set Ω_{ij} is an open subset of Ω_i. Furthermore, $\Omega_{ii} = \Omega_i$, and $\Omega_{ji} \neq \emptyset$ if and only if $\Omega_{ij} \neq \emptyset$. Each non-empty Ω_{ij} (with $i \neq j$) is called **gluing domain**.



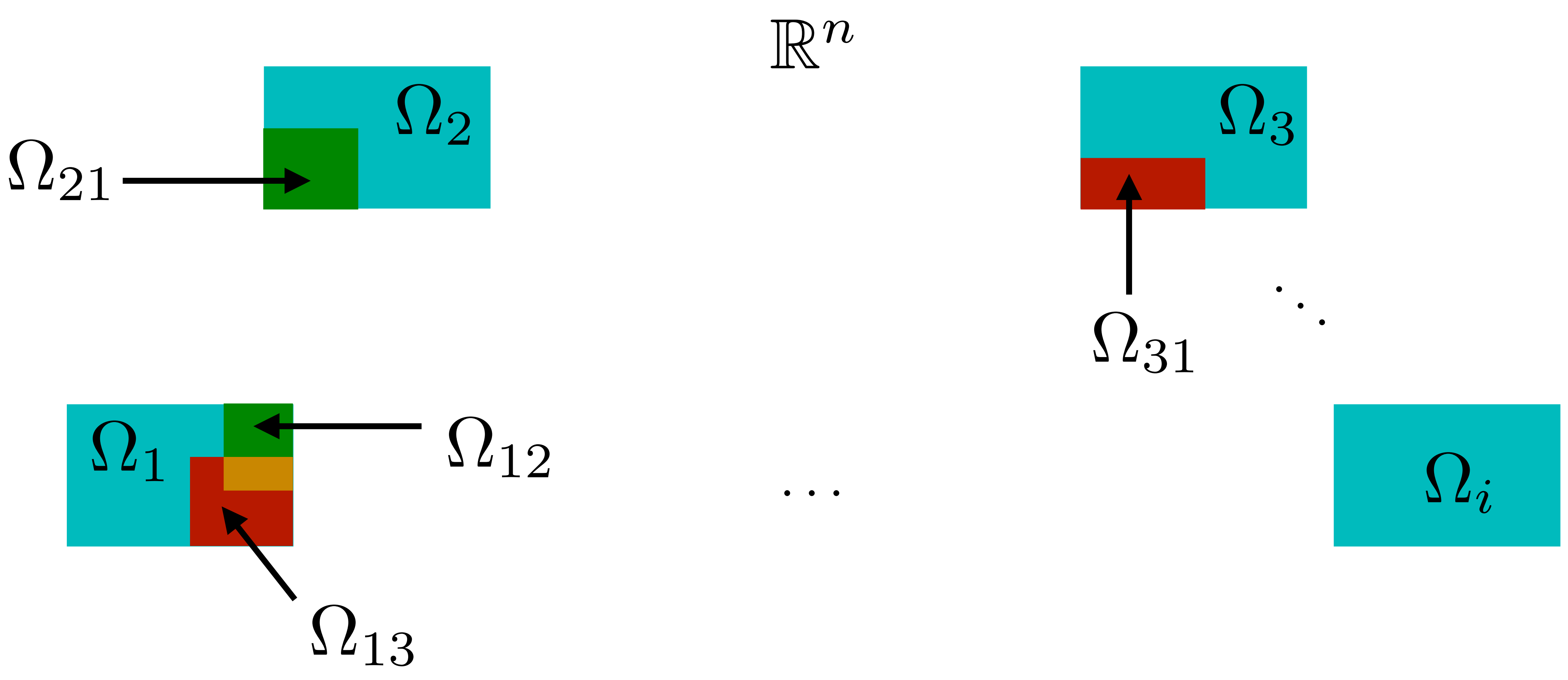
Sets of Gluing Data

(2) For every pair $(i, j) \in I \times I$, the set Ω_{ij} is an open subset of Ω_i. Furthermore, $\Omega_{ii} = \Omega_i$, and $\Omega_{ji} \neq \emptyset$ if and only if $\Omega_{ij} \neq \emptyset$. Each non-empty Ω_{ij} (with $i \neq j$) is called **gluing domain**.



Sets of Gluing Data

(2) For every pair $(i, j) \in I \times I$, the set Ω_{ij} is an open subset of Ω_i. Furthermore, $\Omega_{ii} = \Omega_i$, and $\Omega_{ji} \neq \emptyset$ if and only if $\Omega_{ij} \neq \emptyset$. Each non-empty Ω_{ij} (with $i \neq j$) is called **gluing domain**.



Sets of Gluing Data

Sets of Gluing Data

(3) If we let

$$K = \{(i,j) \in I \times I \mid \Omega_{ij} \neq \emptyset\},$$

then

$$\varphi_{ji} \colon \Omega_{ij} \longrightarrow \Omega_{ji}$$

is a C^k bijection for every $(i,j) \in K$, called a **transition function** or **gluing function**.

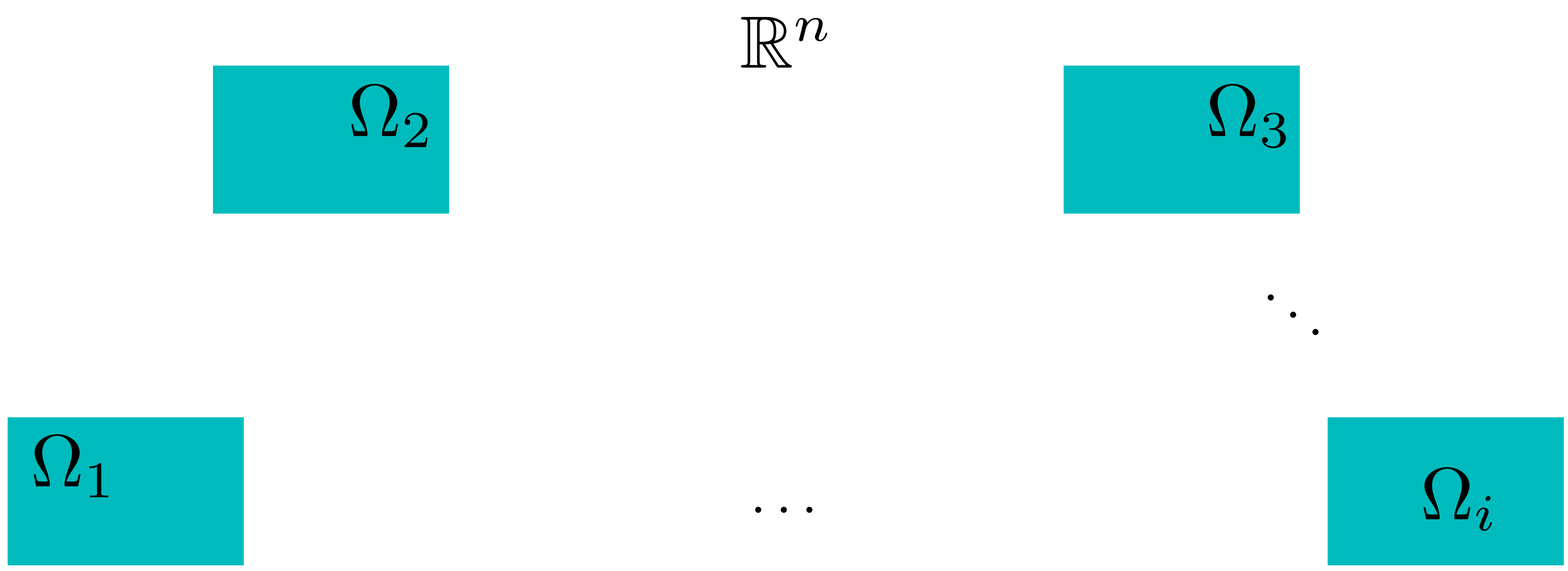
Transition Functions

Transition Functions

- The transition functions tell us how to glue the p-domains.

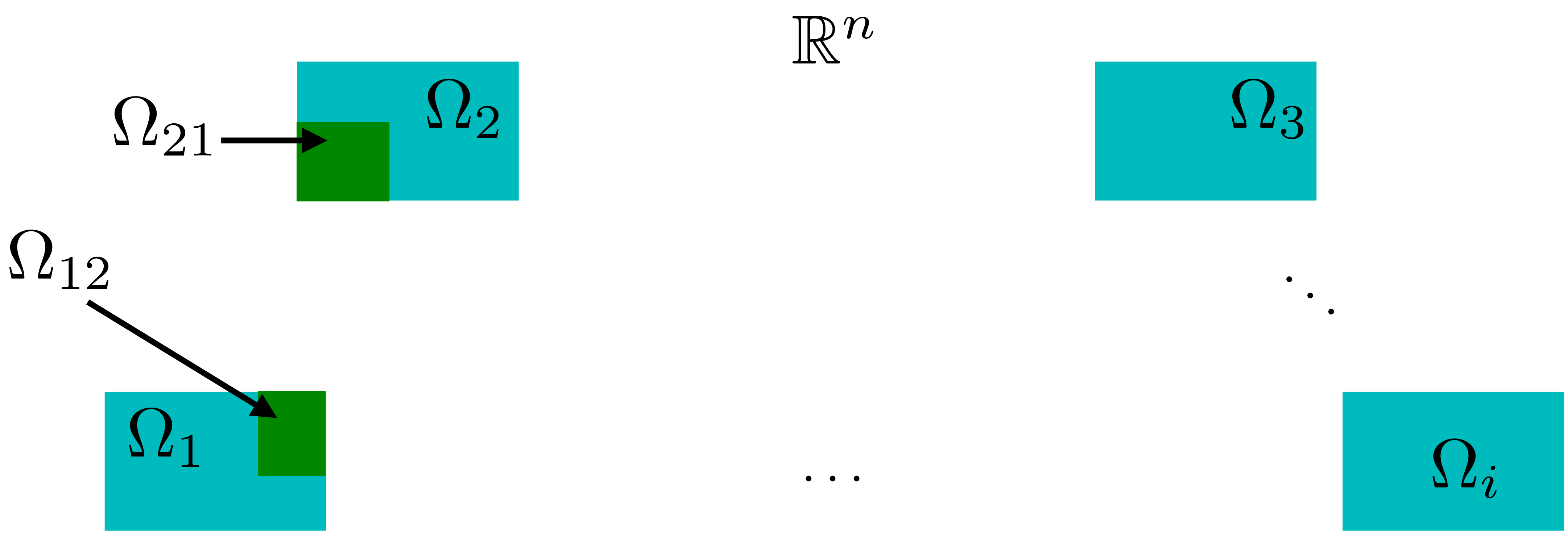
Transition Functions

- The transition functions tell us how to glue the p-domains.



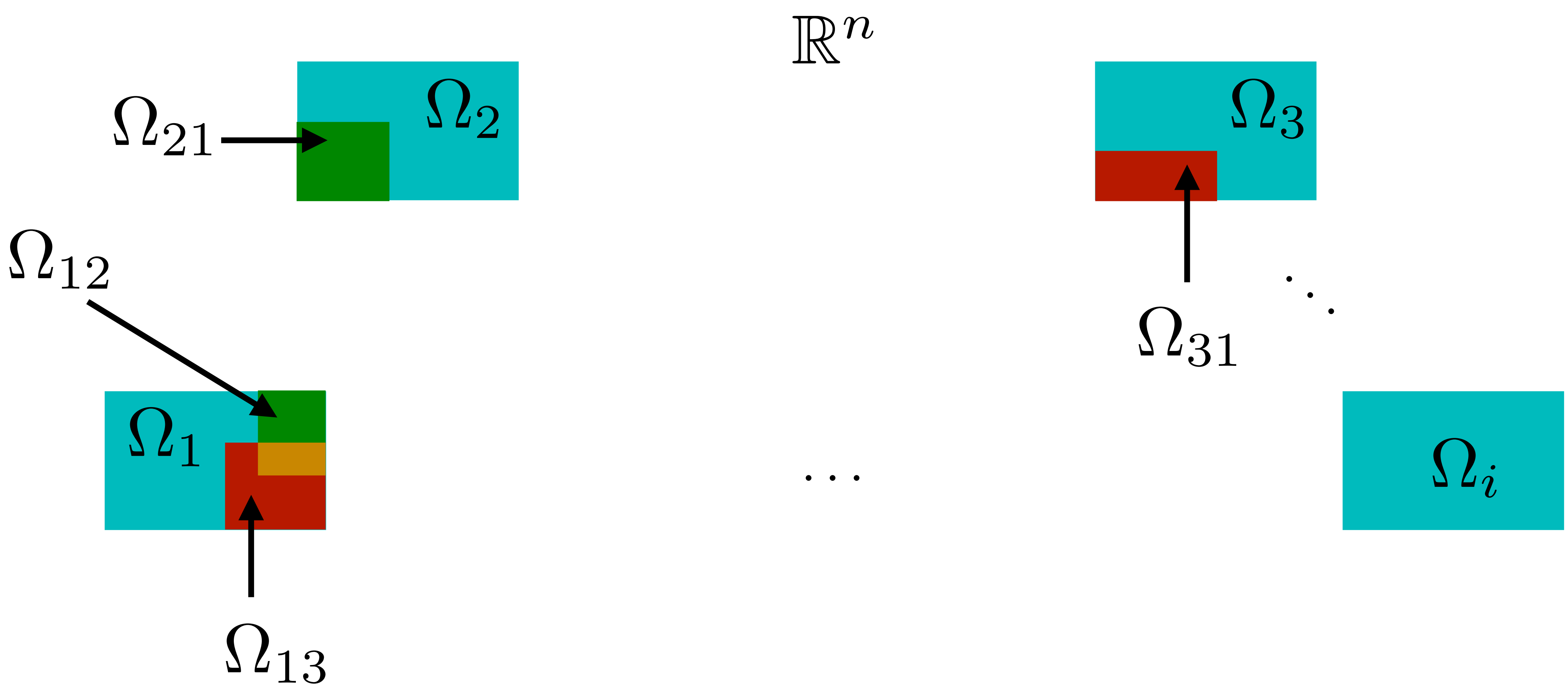
Transition Functions

- The transition functions tell us how to glue the p-domains.



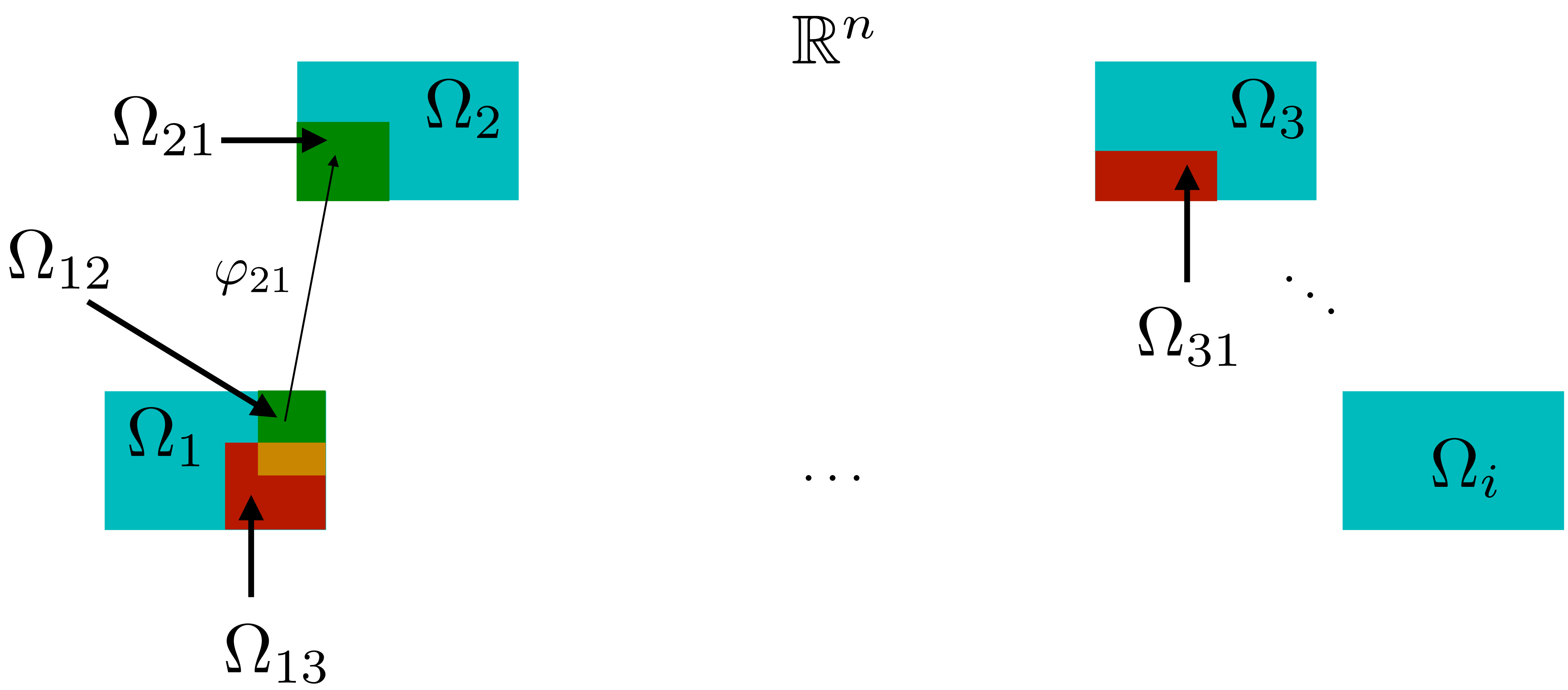
Transition Functions

- The transition functions tell us how to glue the p-domains.



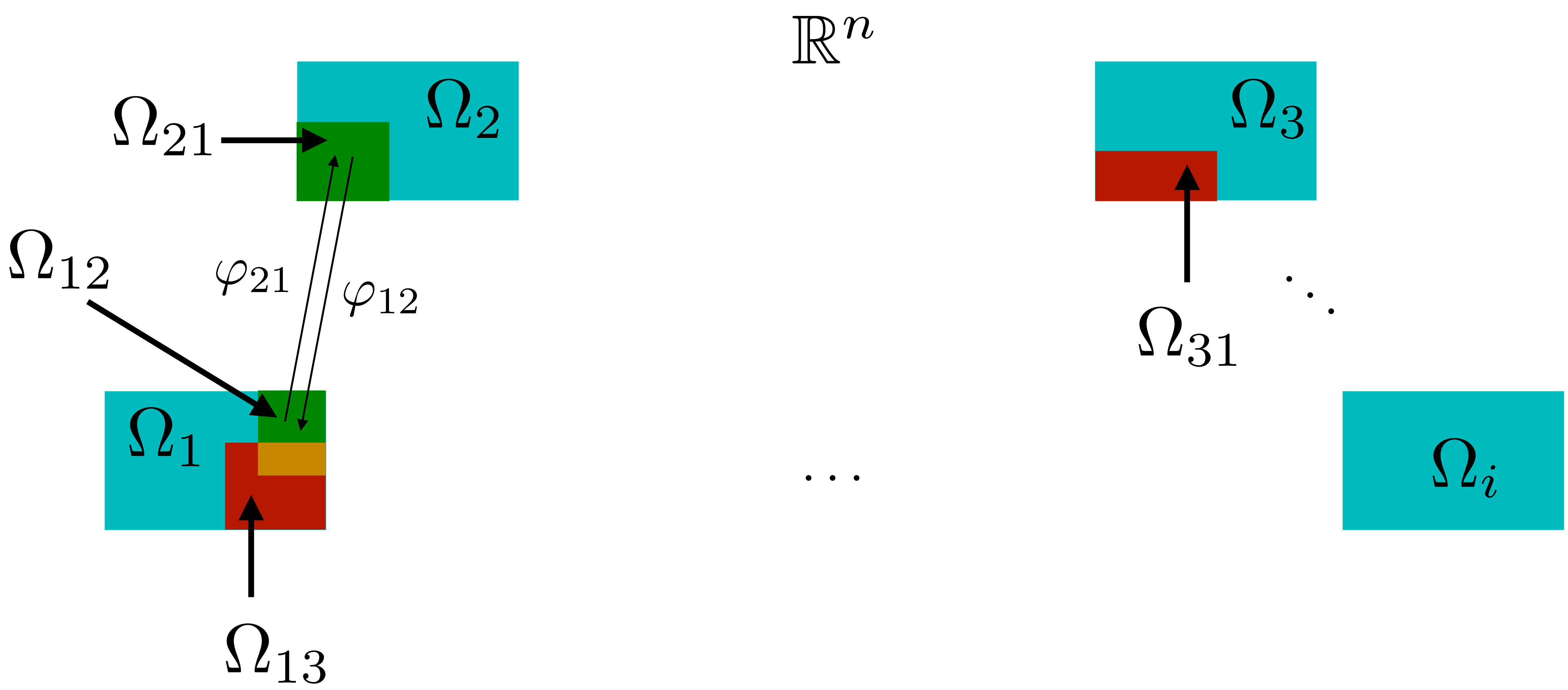
Transition Functions

- The transition functions tell us how to glue the p-domains.



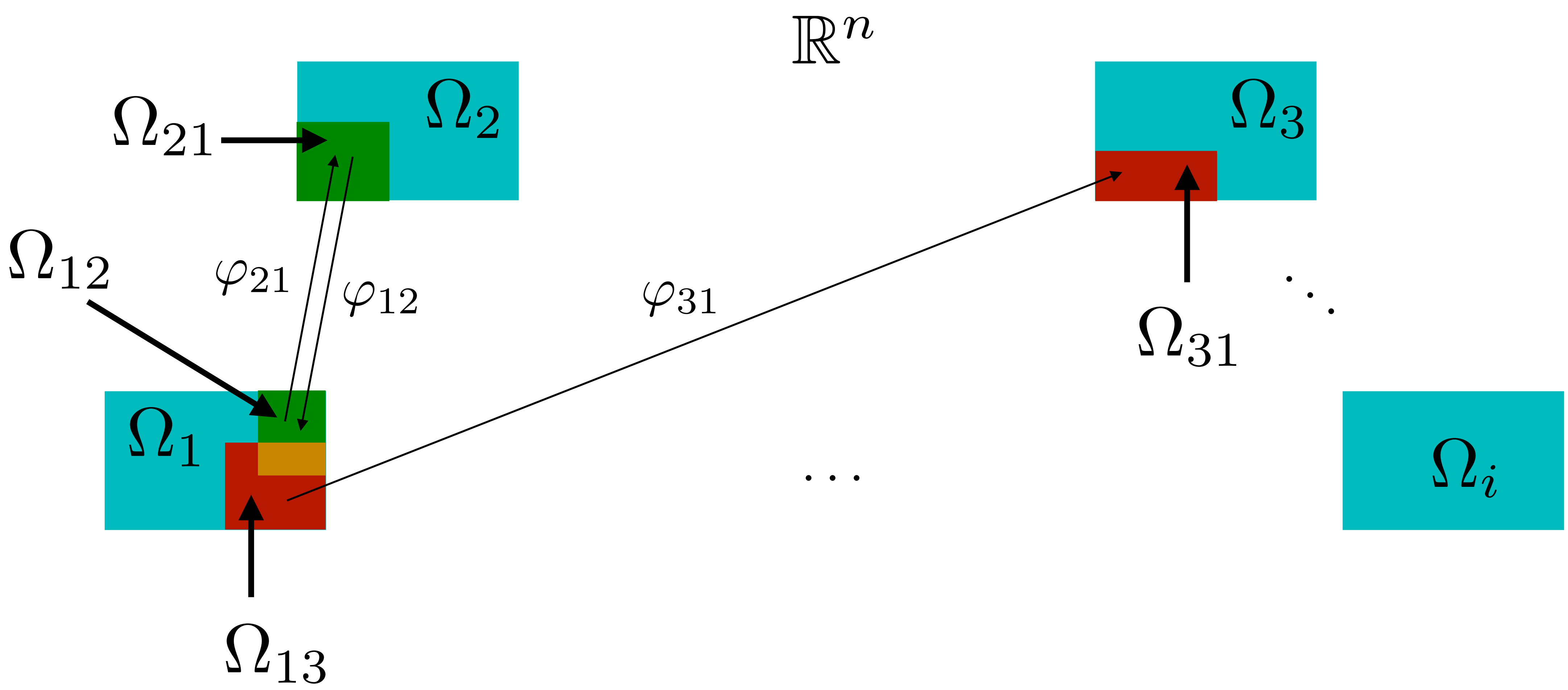
Transition Functions

- The transition functions tell us how to glue the p-domains.



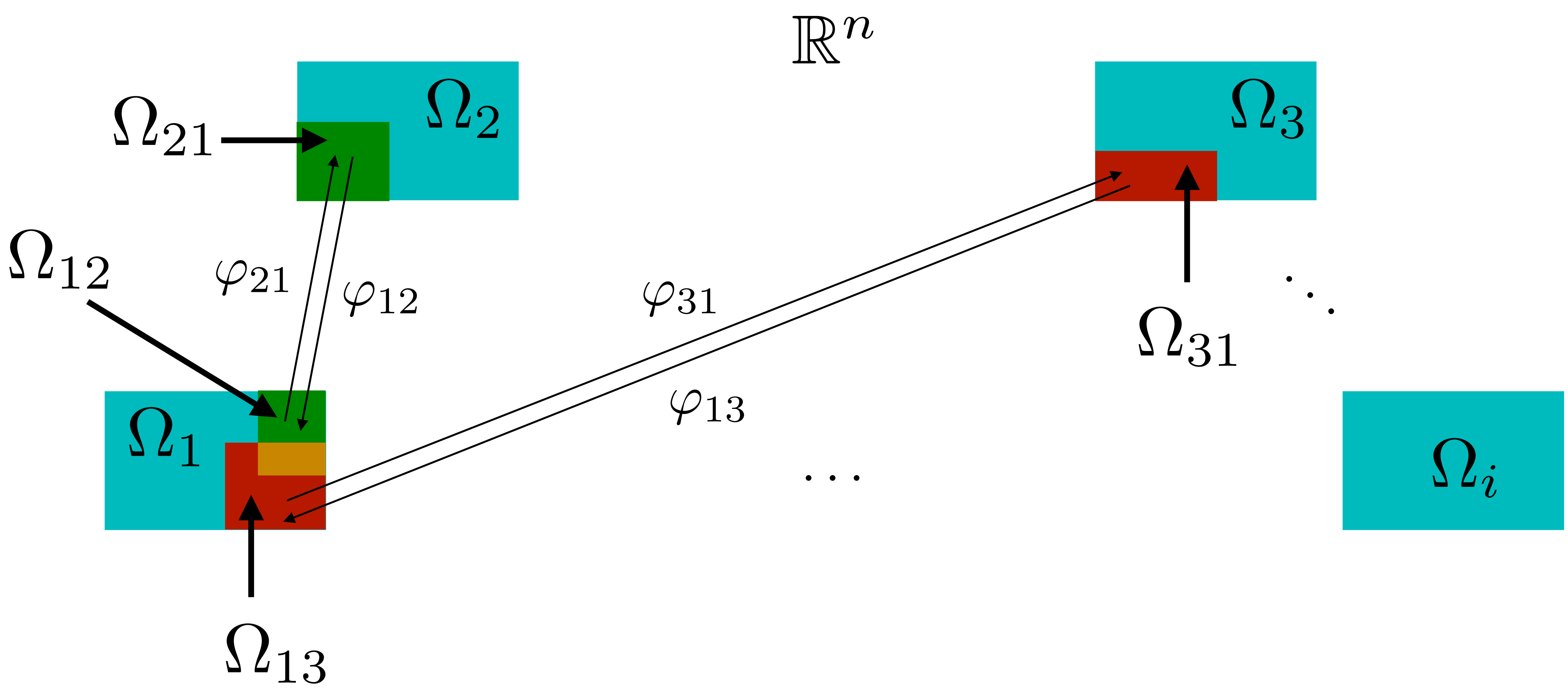
Transition Functions

- The transition functions tell us how to glue the p-domains.



Transition Functions

- The transition functions tell us how to glue the p-domains.



Transition Functions

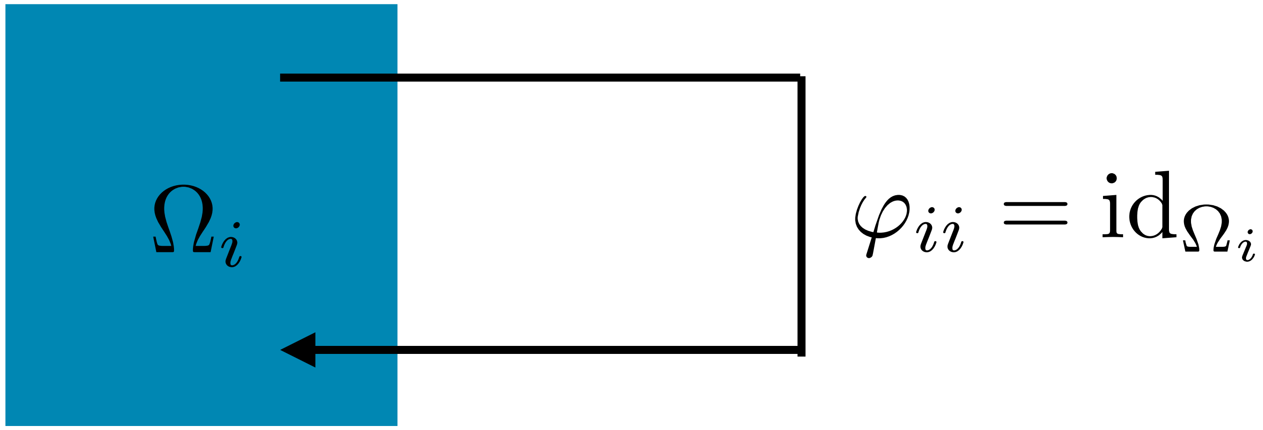
Transition Functions

The transition functions must satisfy the following conditions:

Transition Functions

The transition functions must satisfy the following conditions:

(a) $\varphi_{ii} = \mathrm{id}_{\Omega_i}$, for all $i \in I$,



Transition Functions

Transition Functions

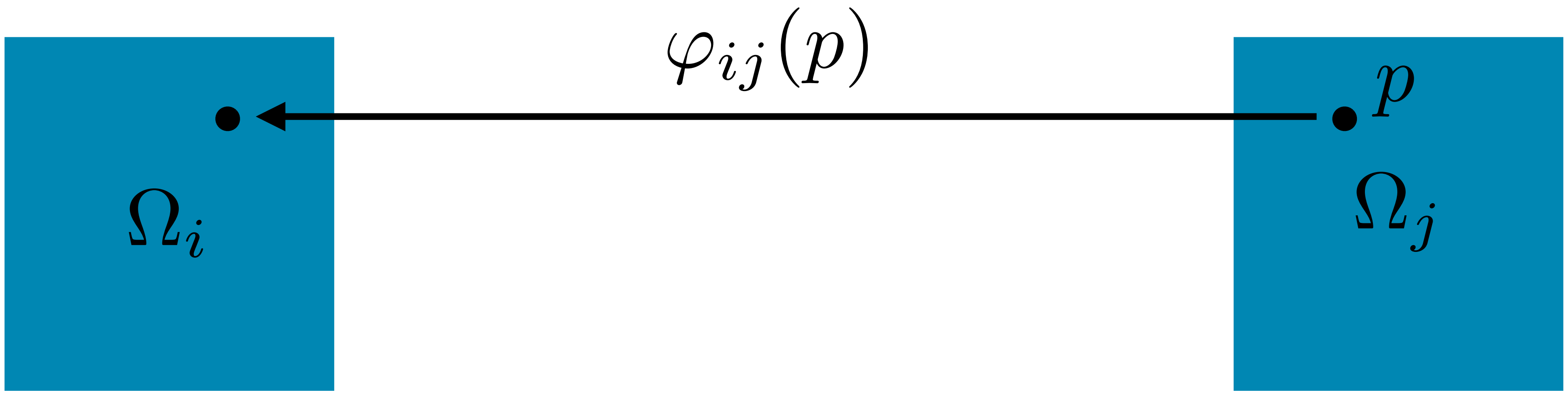
(b) $\varphi_{ij} = \varphi_{ji}^{-1}$, for all $(i, j) \in K$, and

Transition Functions

(b) $\varphi_{ij} = \varphi_{ji}^{-1}$, for all $(i,j) \in K$, and

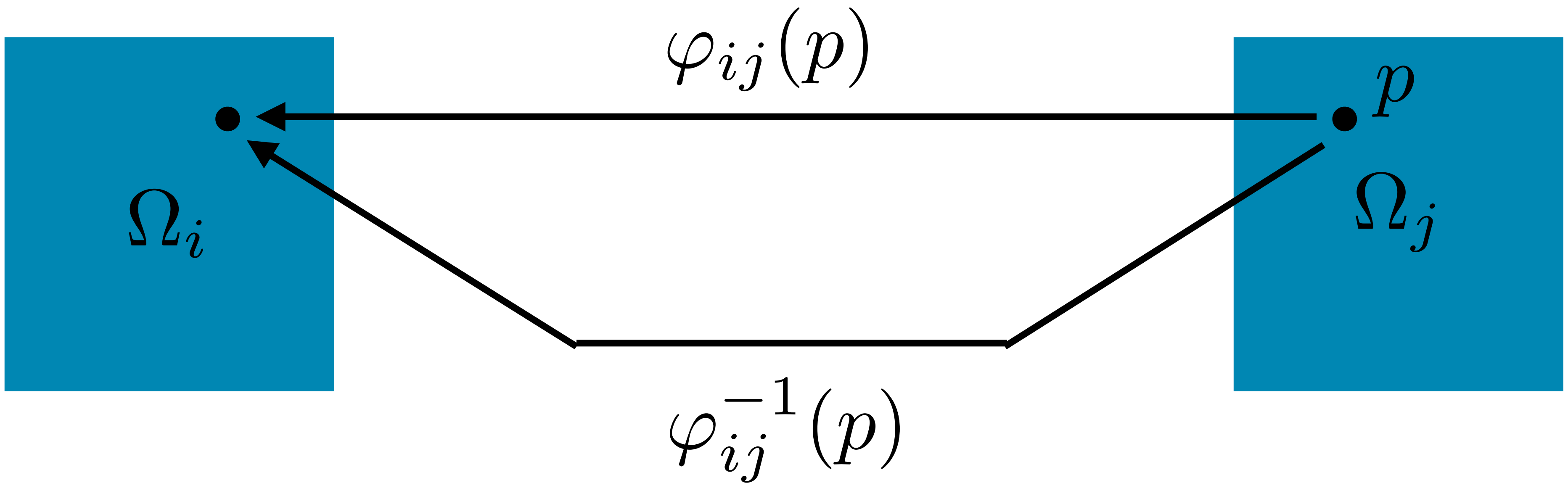
Transition Functions

(b) $\varphi_{ij} = \varphi_{ji}^{-1}$, for all $(i, j) \in K$, and



Transition Functions

(b) $\varphi_{ij} = \varphi_{ji}^{-1}$, for all $(i, j) \in K$, and



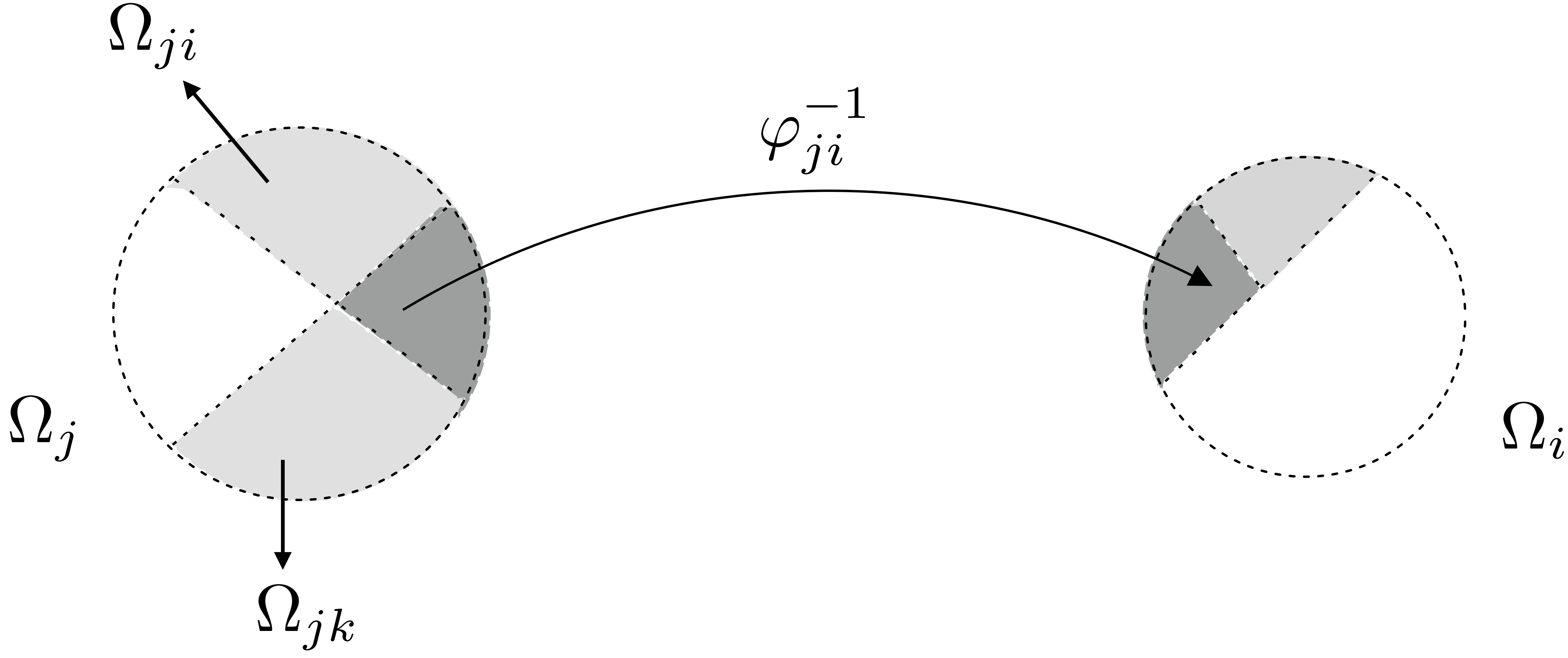
Transition Functions

Transition Functions

(c) for all i, j, and k, if $\Omega_{ji} \cap \Omega_{jk} \neq \emptyset$ then $\varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk}) \subseteq \Omega_{ik}$ and $\varphi_{ki}(x) = \varphi_{kj} \circ \varphi_{ji}(x)$, for all $x \in \varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk})$.

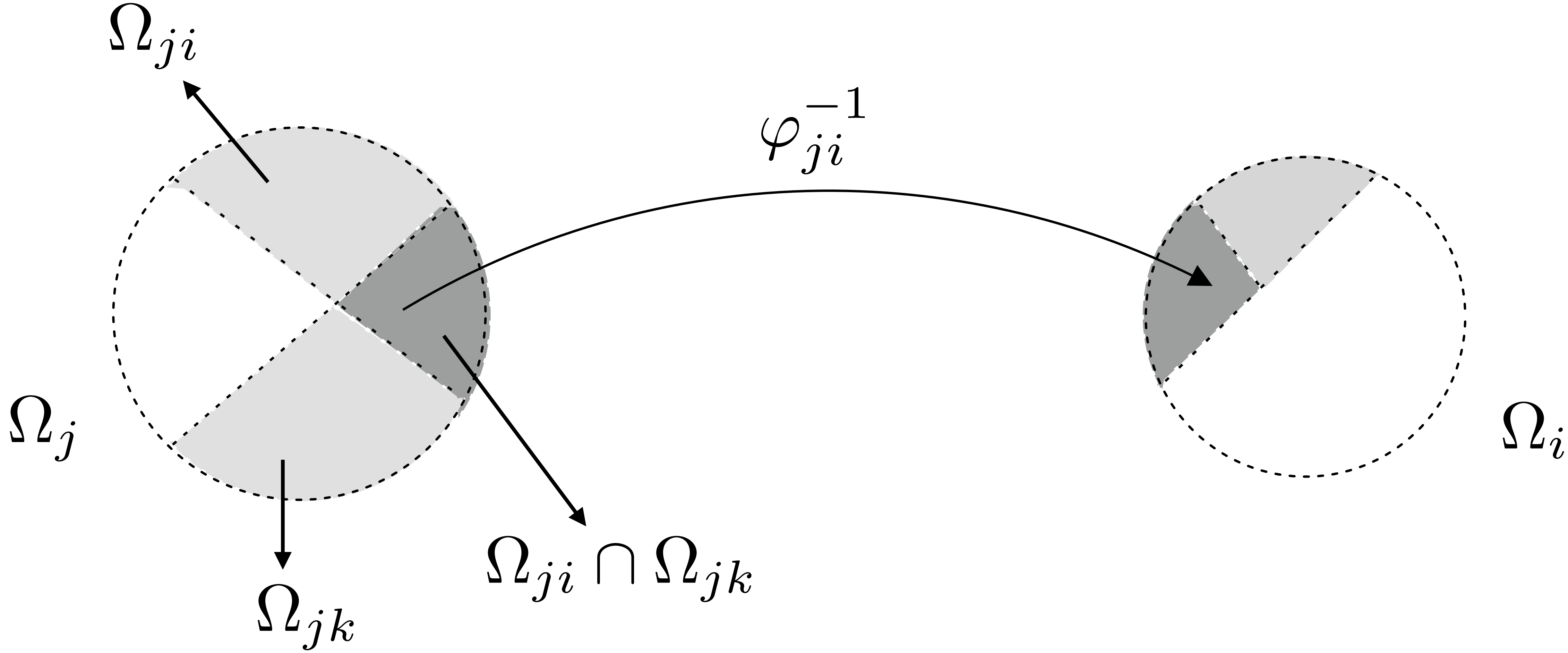
Transition Functions

(c) for all i, j, and k, if $\Omega_{ji} \cap \Omega_{jk} \neq \emptyset$ then $\varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk}) \subseteq \Omega_{ik}$ and $\varphi_{ki}(x) = \varphi_{kj} \circ \varphi_{ji}(x)$, for all $x \in \varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk})$.



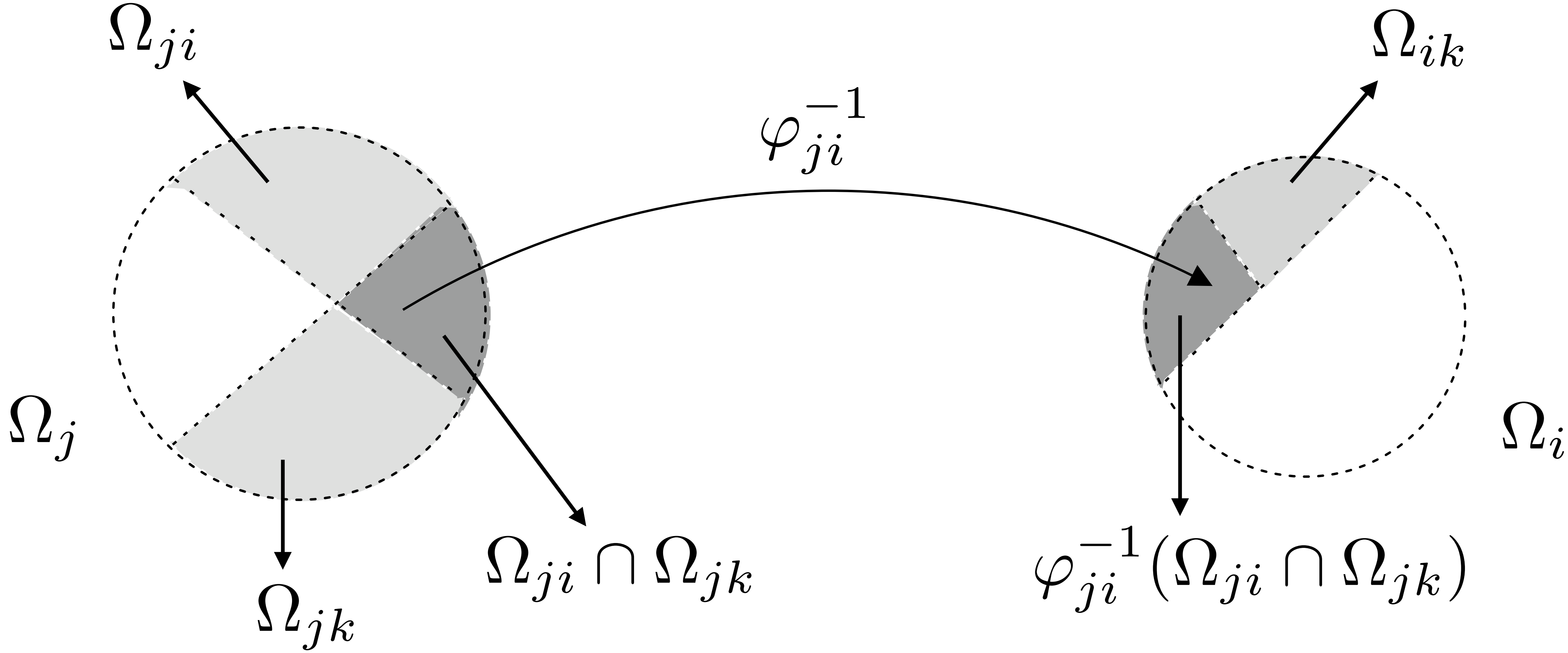
Transition Functions

(c) for all i, j, and k, if $\Omega_{ji} \cap \Omega_{jk} \neq \emptyset$ then $\varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk}) \subseteq \Omega_{ik}$ and $\varphi_{ki}(x) = \varphi_{kj} \circ \varphi_{ji}(x)$, for all $x \in \varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk})$.



Transition Functions

(c) for all i, j, and k, if $\Omega_{ji} \cap \Omega_{jk} \neq \emptyset$ then $\varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk}) \subseteq \Omega_{ik}$ and $\varphi_{ki}(x) = \varphi_{kj} \circ \varphi_{ji}(x)$, for all $x \in \varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk})$.



The Cocycle Condition

The Cocycle Condition

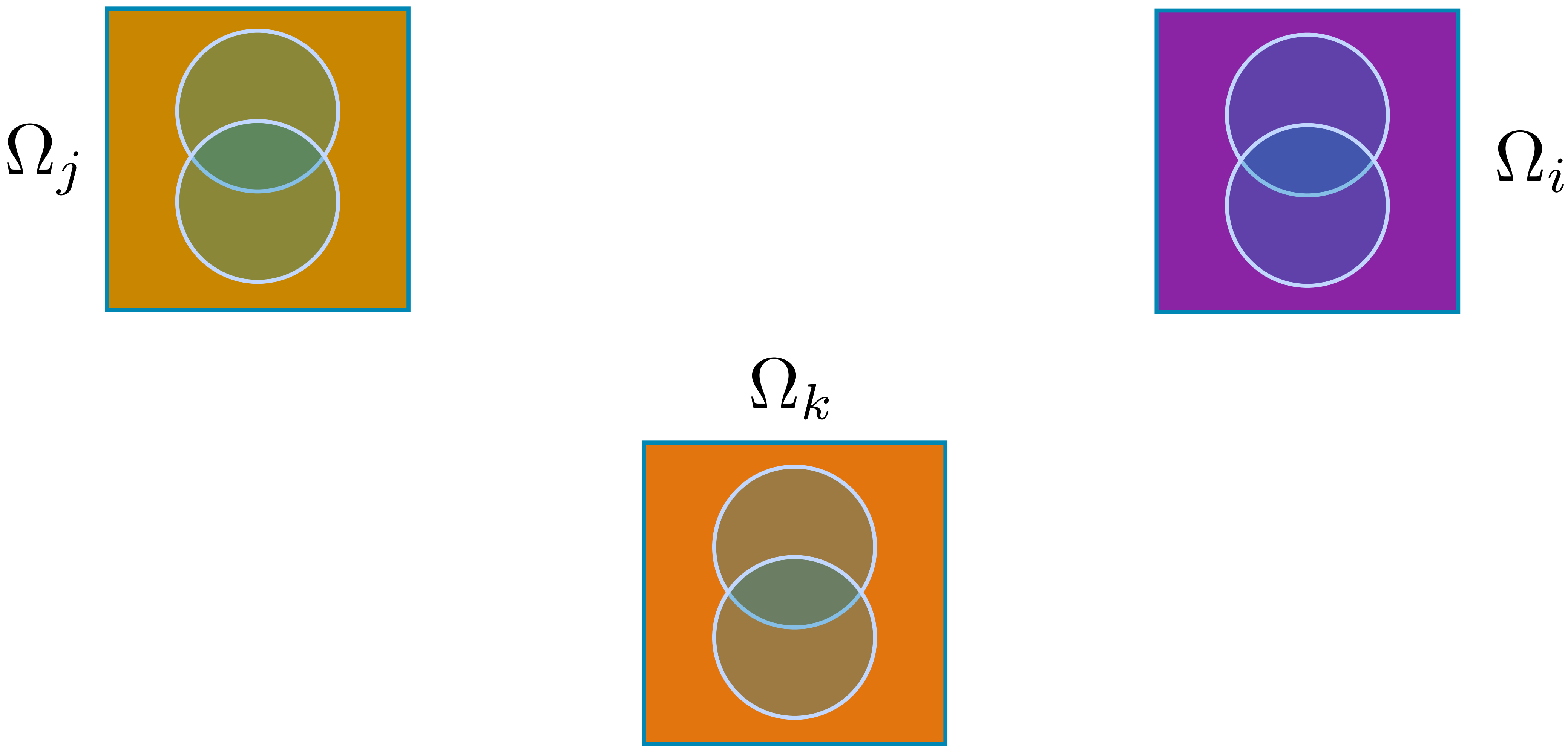
The "evil" cocycle condition

$\varphi_{ki}(x) = \varphi_{kj} \circ \varphi_{ji}(x)$, for all $x \in \varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk})$.

The Cocycle Condition

The “evil” cocycle condition

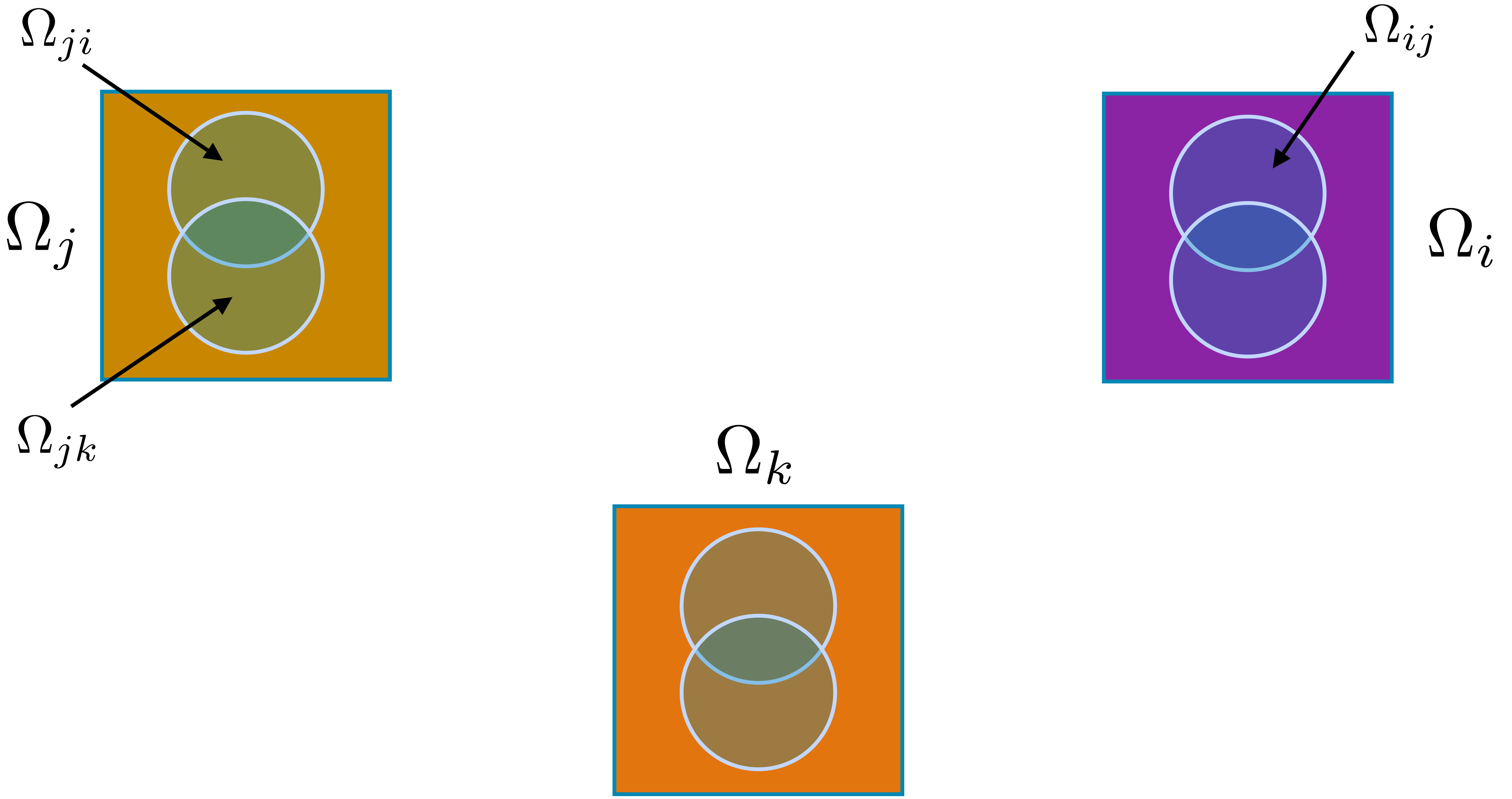
$$\varphi_{ki}(x) = \varphi_{kj} \circ \varphi_{ji}(x), \text{ for all } x \in \varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk}).$$



The Cocycle Condition

The "evil" cocycle condition

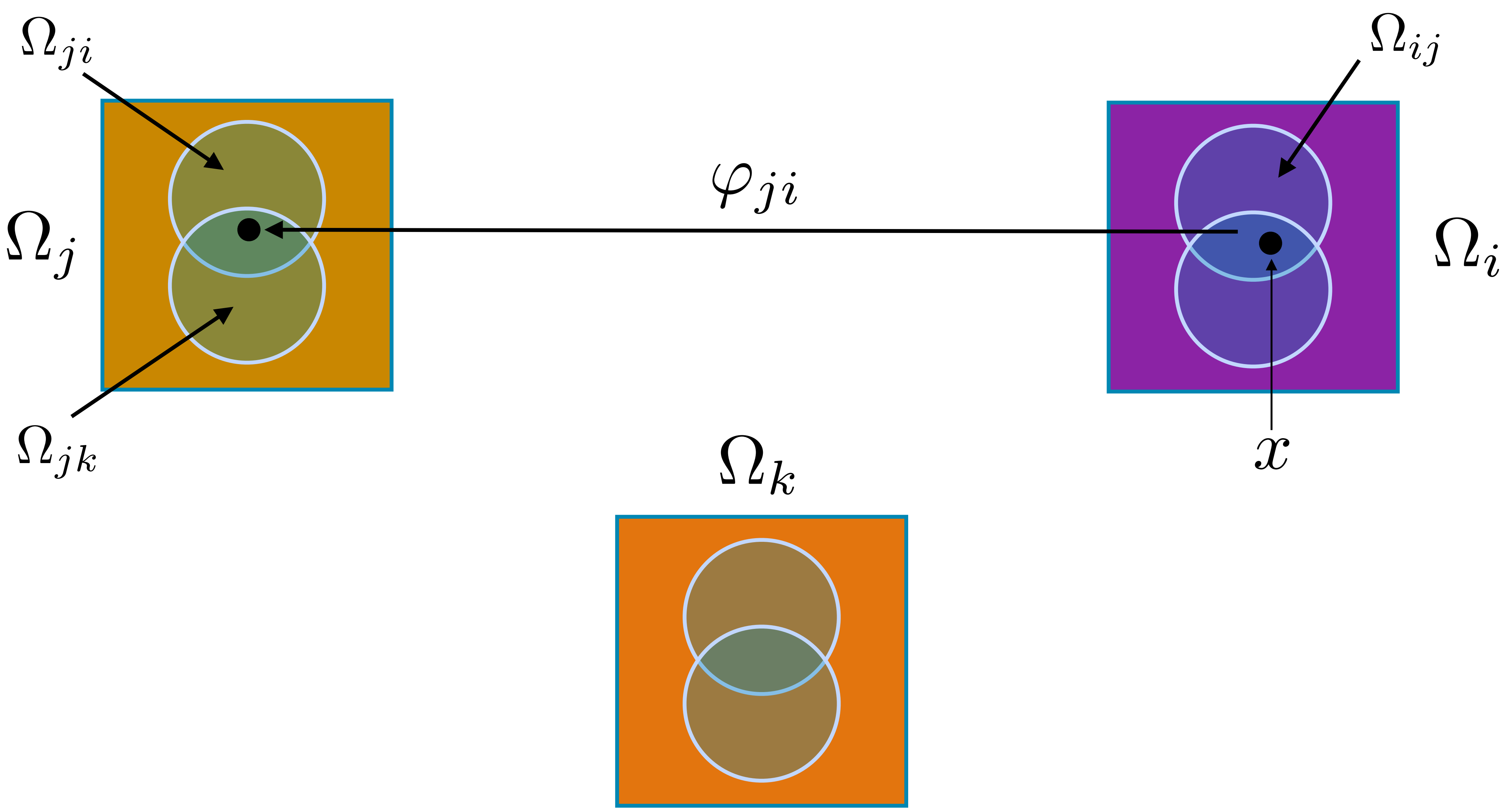
$$\varphi_{ki}(x) = \varphi_{kj} \circ \varphi_{ji}(x), \text{ for all } x \in \varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk}).$$



The Cocycle Condition

The "evil" cocycle condition

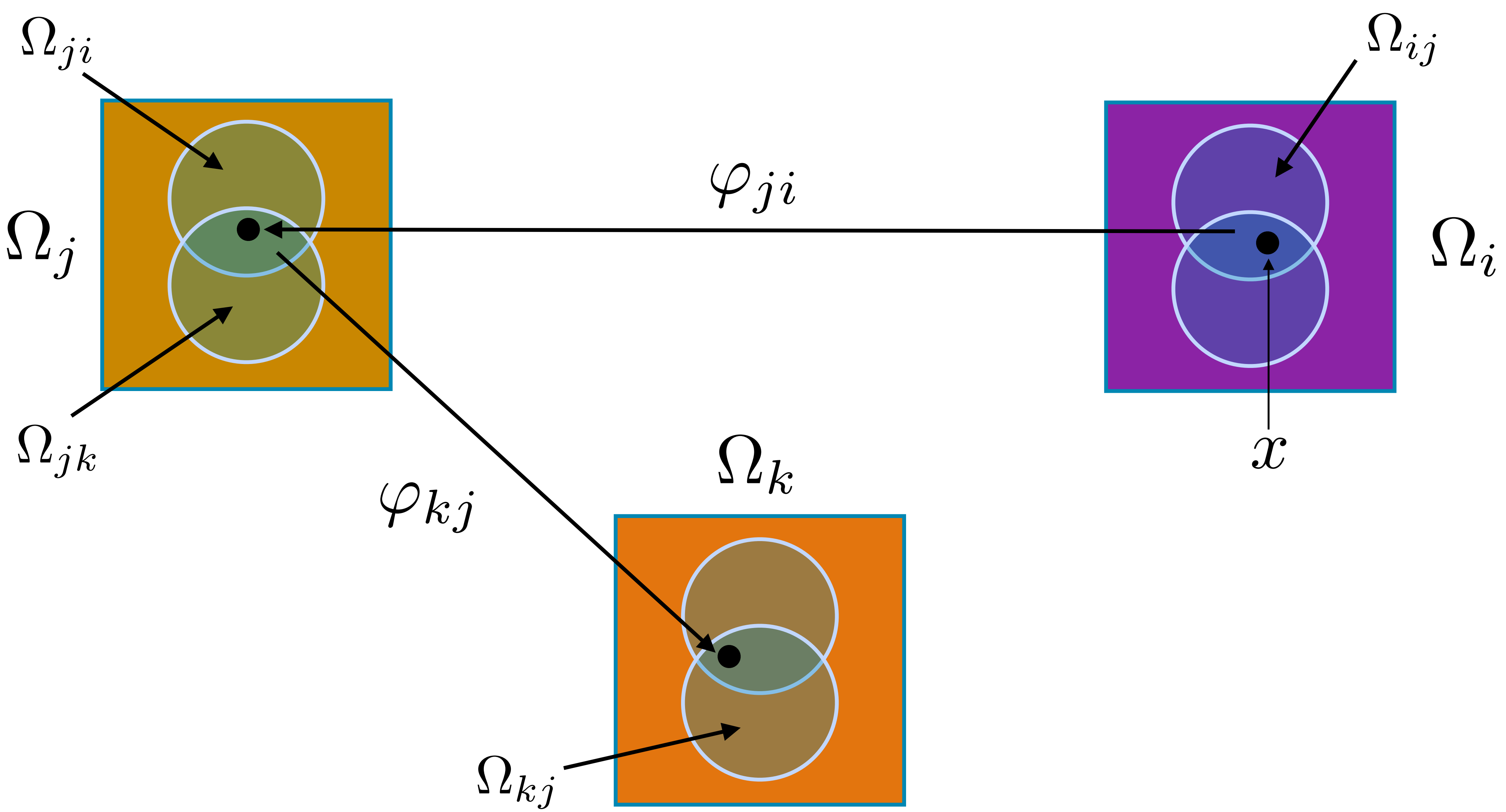
$$\varphi_{ki}(x) = \varphi_{kj} \circ \varphi_{ji}(x), \text{ for all } x \in \varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk}).$$



The Cocycle Condition

The "evil" cocycle condition

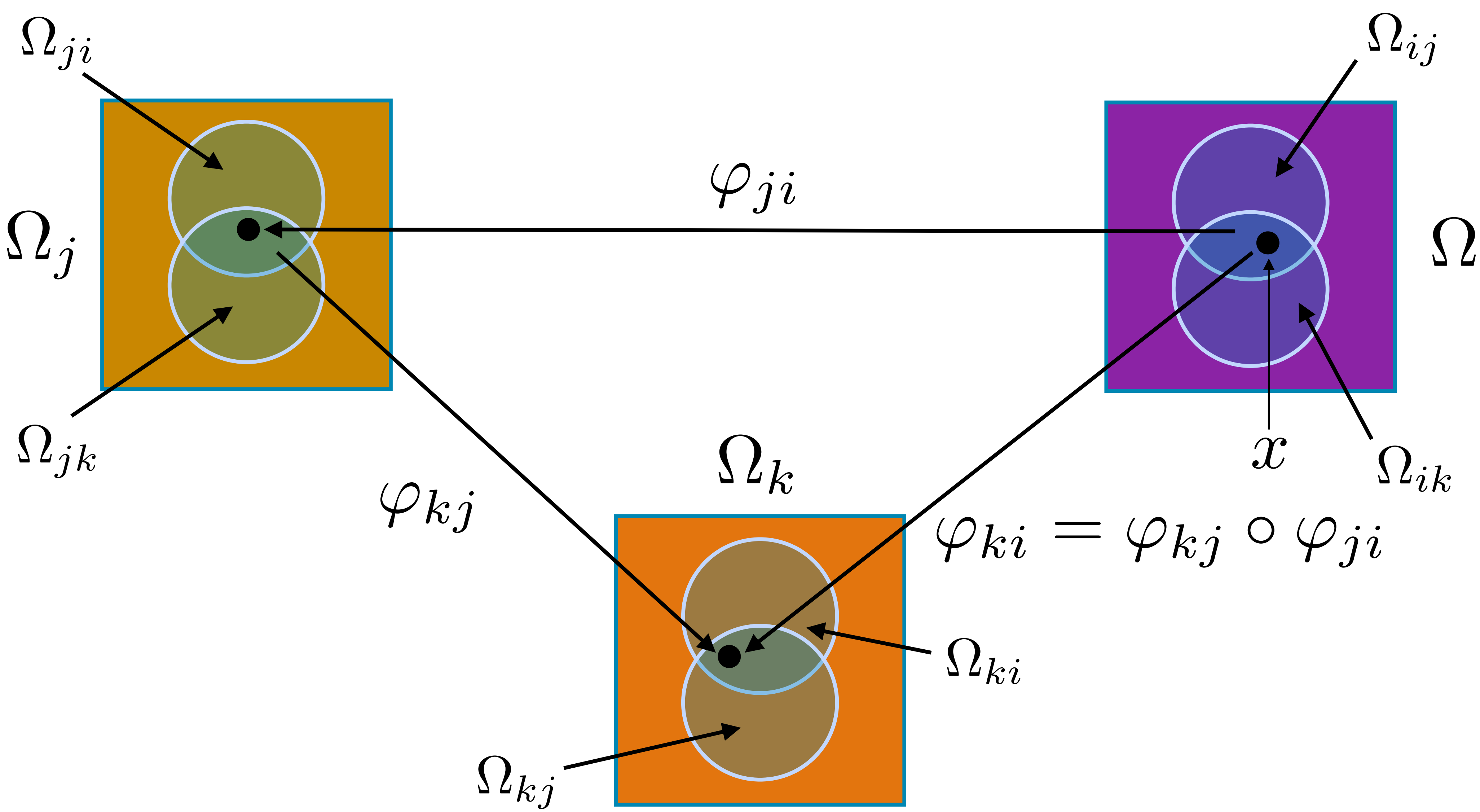
$$\varphi_{ki}(x) = \varphi_{kj} \circ \varphi_{ji}(x), \text{ for all } x \in \varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk}).$$



The Cocycle Condition

The "evil" cocycle condition

$$\varphi_{ki}(x) = \varphi_{kj} \circ \varphi_{ji}(x), \text{ for all } x \in \varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk}).$$



The Cocycle Condition

The Cocycle Condition

- The cocycle condition implies conditions (a) and (b).

The Cocycle Condition

- The cocycle condition implies conditions (a) and (b).

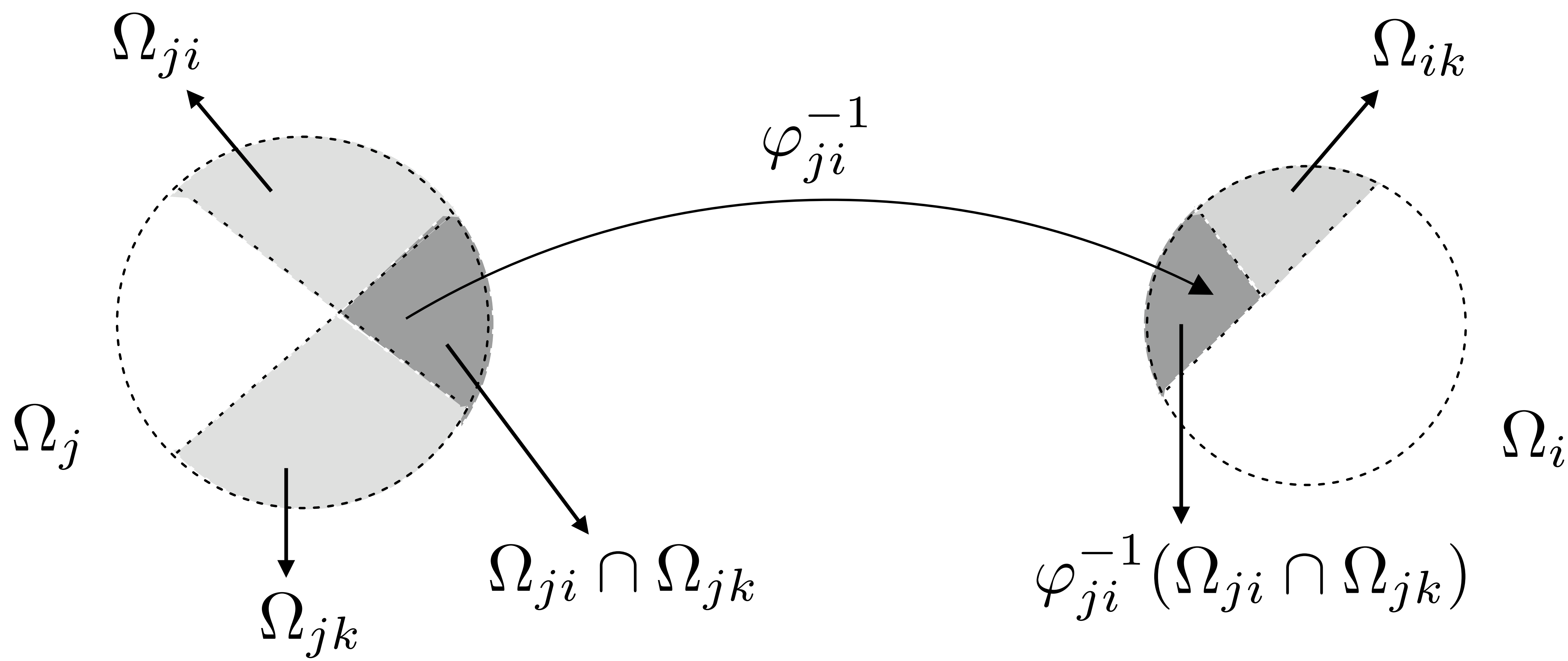
- Previous versions found in the literature are often incorrect.

The Cocycle Condition

- This is because the transition maps are only partial functions!

The Cocycle Condition

- This is because the transition maps are only partial functions!



Parametric Pseudo-Manifolds

Parametric Pseudo-Manifolds

- The question now becomes:

Parametric Pseudo-Manifolds

- The question now becomes:

Given a set of gluing data, $\mathcal{G}$, can we build a manifold from it?

Parametric Pseudo-Manifolds

- The question now becomes:

Given a set of gluing data, $\mathcal{G}$, can we build a manifold from it?

- Indeed, such a manifold is built by a **quotient construction**.

Parametric Pseudo-Manifolds

- The question now becomes:

Given a set of gluing data, $\mathcal{G}$, can we build a manifold from it?

- Indeed, such a manifold is built by a **quotient construction**.

- We form the disjoint union of the Ω_i and we identify Ω_{ij} with Ω_{ji} using φ_{ji}, an equivalence relation, $\sim$. We form the quotient

$$M_{\mathcal{G}} = \left(\coprod_i \Omega_i \right) / \sim, \ .$$

Parametric Pseudo-Manifolds

Parametric Pseudo-Manifolds

Theorem 1 [Gallier, Siqueira, and Xu, 2008]

For every set of gluing data,

$$\mathcal{G} = \big((\Omega_i)_{i\in I}, (\Omega_{ij})_{(i,j)\in I\times I}, (\varphi_{ji})_{(i,j)\in K\times K}\big) \ ,$$

there is an n-dimensional C^k manifold, $M_{\mathcal{G}}$, whose transition functions are the φ_{ji}'s.

Parametric Pseudo-Manifolds

Parametric Pseudo-Manifolds

REMARK:

A condition on the gluing data is needed to make sure that $M_{\mathcal{G}}$ is Hausdorff. Since it is quite technical, we will not show it here.

Parametric Pseudo-Manifolds

Parametric Pseudo-Manifolds

Theorem 1 is very nice, but . . .

Parametric Pseudo-Manifolds

Theorem 1 is very nice, but . . .

- Our proof is not constructive;

Parametric Pseudo-Manifolds

Theorem 1 is very nice, but . . .

- Our proof is not constructive;

- $M_{\mathcal{G}}$ is an abstract entity, which may not even be compact, orientable, etc.

Parametric Pseudo-Manifolds

Theorem 1 is very nice, but . . .

- Our proof is not constructive;

- $M_{\mathcal{G}}$ is an abstract entity, which may not even be compact, orientable, etc.

So, the question that remains is **how** to build a *concrete* manifold.

Parametric Pseudo-Manifolds

Theorem 1 is very nice, but ...

- Our proof is not constructive;
- $M_{\mathcal{G}}$ is an abstract entity, which may not even be compact, orientable, etc.

So, the question that remains is **how** to build a *concrete* manifold.

Let us first formalize our notion of "concreteness".

Parametric Pseudo-Manifolds

Parametric Pseudo-Manifolds

Let n, m, and k be integers, with $m > n \geq 1$ and $k \geq 1$ or $k = \infty$.

Parametric Pseudo-Manifolds

Let n, m, and k be integers, with $m > n \geq 1$ and $k \geq 1$ or $k = \infty$.

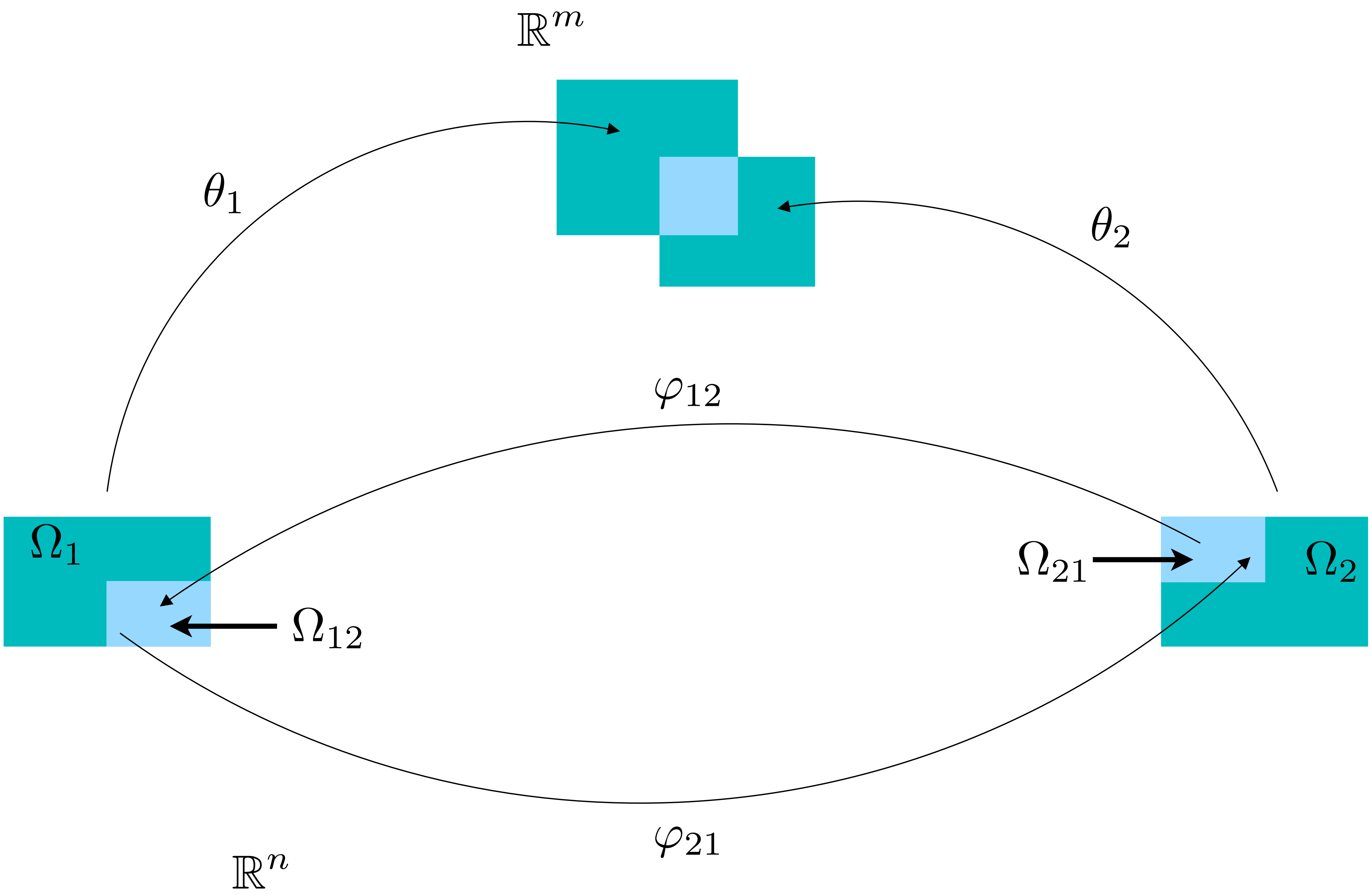
A **parametric** C^k **pseudo-manifold of dimension** n **in** $\mathbb{R}^m$ is a pair,

$$\mathcal{M} = (\mathcal{G}, (\theta_i)_{i \in I}) ,$$

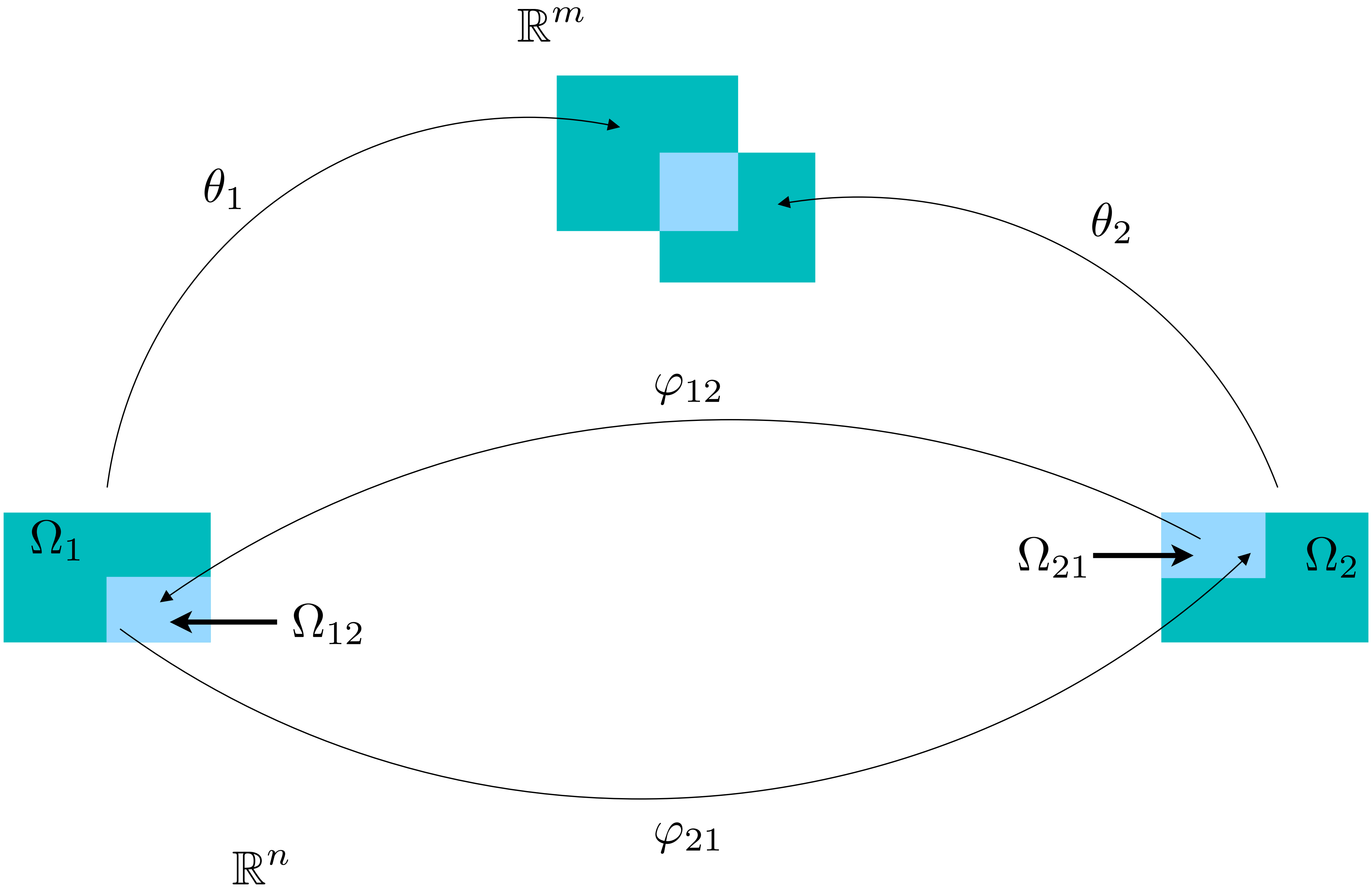
such that $\mathcal{G} = \big((\Omega_i)_{i \in I}, (\Omega_{ij})_{(i,j) \in I \times I}, (\varphi_{ij})_{(i,j) \in K \times K}\big)$ is a set of gluing data, for some finite I, and each θ_i is a C^k function, $\theta_i : \Omega_i \rightarrow \mathbb{R}^m$, called a **parametrization** such that the following holds:

Parametric Pseudo-Manifolds

Parametric Pseudo-Manifolds



Parametric Pseudo-Manifolds



- When $m = 3$ and $n = 2$, we say that $\mathcal{M}$ is a **parametric pseudo-surface**.

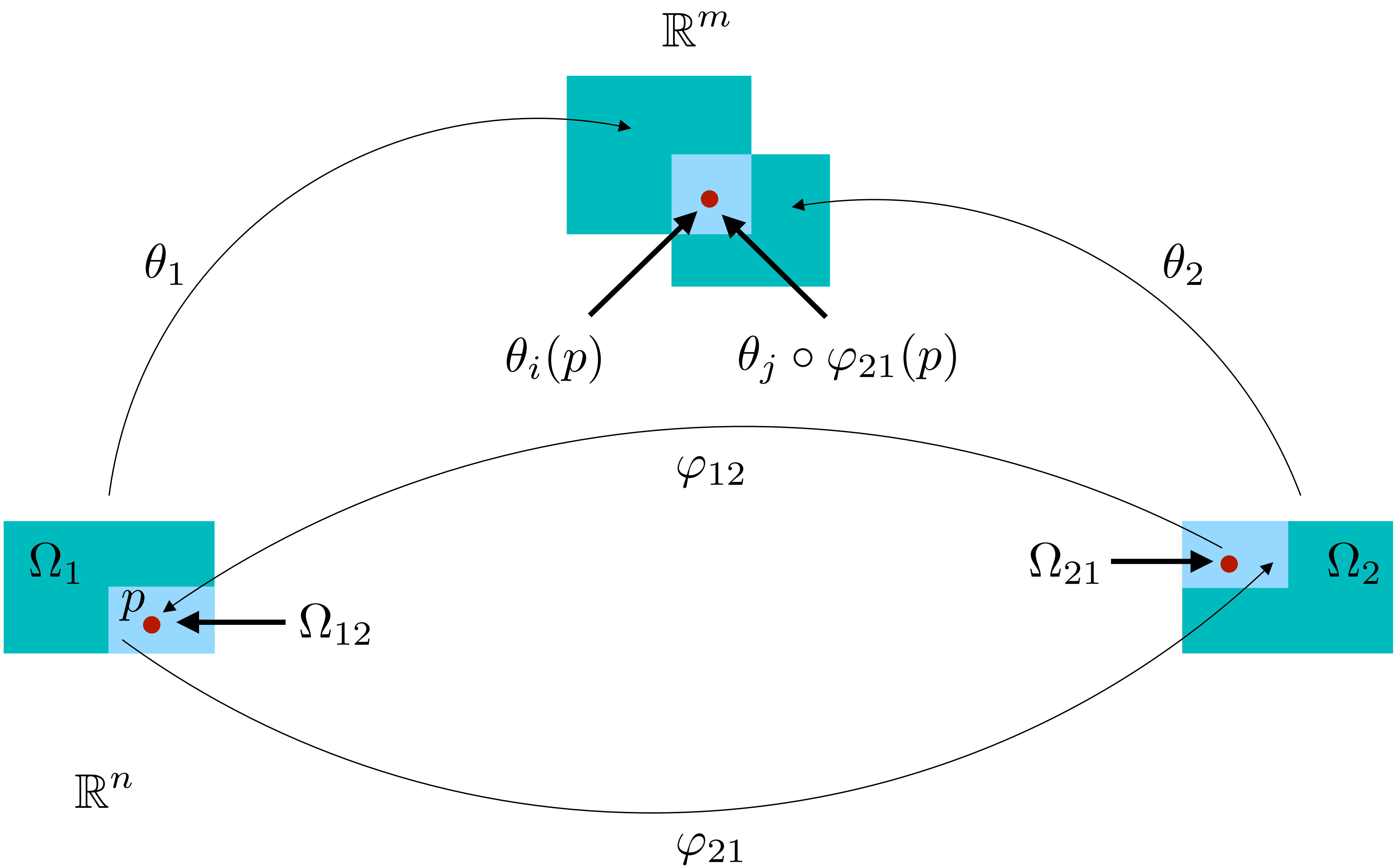
Parametric Pseudo-Manifolds

Parametric Pseudo-Manifolds

(C) For all $(i, j) \in K$, we have $\theta_i = \theta_j \circ \varphi_{ji}$.

Parametric Pseudo-Manifolds

(C) For all $(i, j) \in K$, we have $\theta_i = \theta_j \circ \varphi_{ji}$.



Parametric Pseudo-Manifolds

Parametric Pseudo-Manifolds

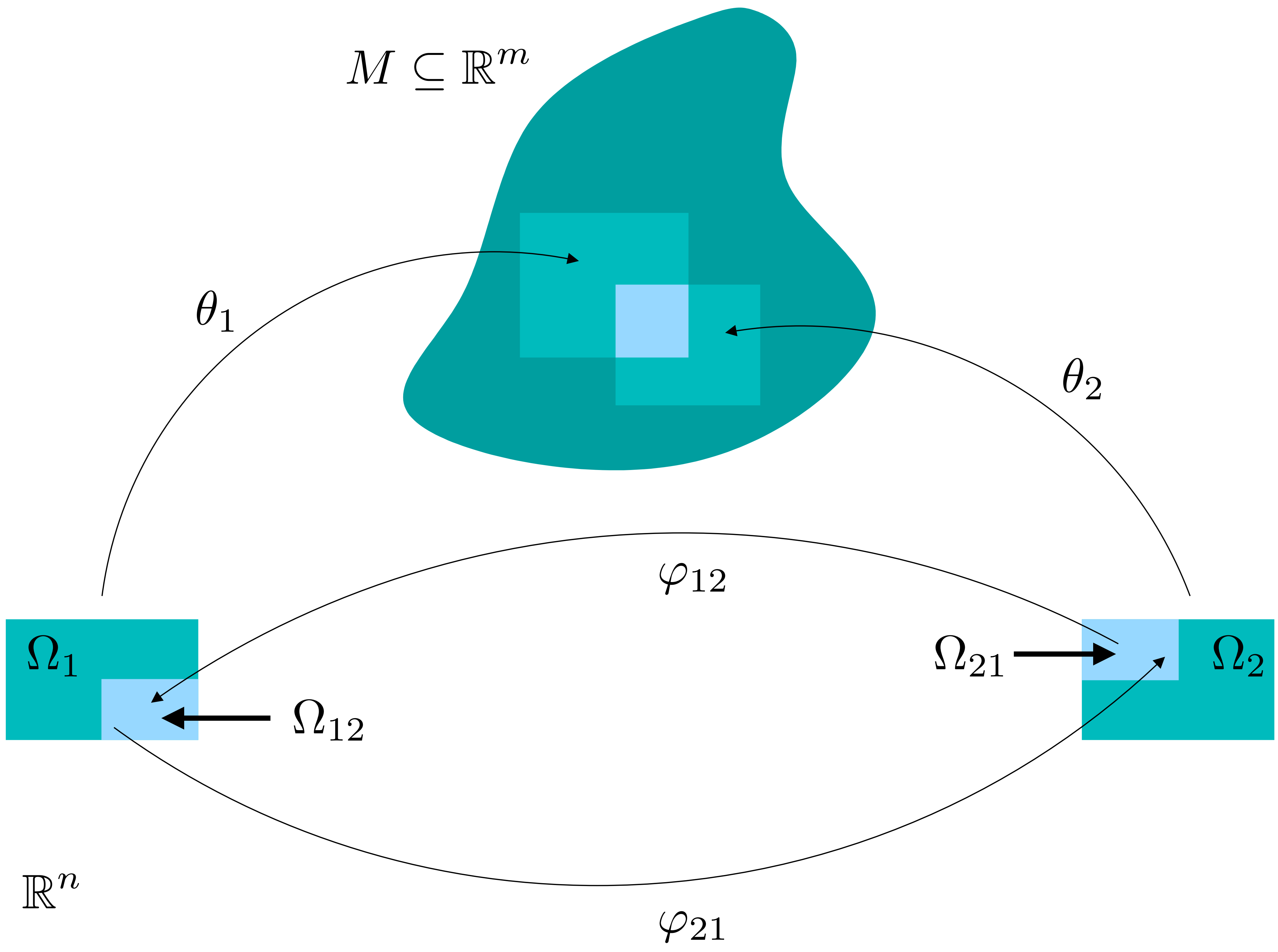
- The subset

$$M = \bigcup_{i \in I} \theta_i(\Omega_i)$$

of $\mathbb{R}^m$ is called the **image** of the parametric pseudo-manifold.

Parametric Pseudo-Manifolds

Parametric Pseudo-Manifolds



Parametric Pseudo-Manifolds

Parametric Pseudo-Manifolds

REMARK:

There is a (unique) surjective map:

$$\Theta : M_{\mathcal{G}} \longrightarrow M\,.$$

Parametric Pseudo-Manifolds

Parametric Pseudo-Manifolds

We proved that M can be given a manifold structure if we require the θ_i's to be bijective and to satisfy the following conditions:

Parametric Pseudo-Manifolds

We proved that M can be given a manifold structure if we require the θ_i's to be bijective and to satisfy the following conditions:

(C') For all $(i, j) \in K$,

$$\theta_i(\Omega_i) \cap \theta_j(\Omega_j) = \theta_i(\Omega_{ij}) = \theta_j(\Omega_{ji}) \,.$$

Parametric Pseudo-Manifolds

We proved that M can be given a manifold structure if we require the θ_i's to be bijective and to satisfy the following conditions:

(C') For all $(i, j) \in K$,

$$\theta_i(\Omega_i) \cap \theta_j(\Omega_j) = \theta_i(\Omega_{ij}) = \theta_j(\Omega_{ji}).$$

(C'') For all $(i, j) \notin K$,

$$\theta_i(\Omega_i) \cap \theta_j(\Omega_j) = \emptyset.$$

Conclusions

Conclusions

- We can *build* a parametric pseudo-manifold (PPM) from a set of gluing data and, *under certain conditions*, the image of a PPM can be given the structure of a manifold.

Conclusions

- We can *build* a parametric pseudo-manifold (PPM) from a set of gluing data and, *under certain conditions*, the image of a PPM can be given the structure of a manifold.

- In the following lecture, we describe a new constructive approach to define a set of gluing data from a triangle mesh.

Conclusions

- We can *build* a parametric pseudo-manifold (PPM) from a set of gluing data and, *under certain conditions*, the image of a PPM can be given the structure of a manifold.
- In the following lecture, we describe a new constructive approach to define a set of gluing data from a triangle mesh.
- We also describe how to build a parametric C^∞ pseudo-surface from the set of gluing data. The image of this parametric pseudo-surface approximates the vertices of the mesh.

Fitting Surfaces to Polygonal Meshes (Part I)

Marcelo Siqueira
UFMS

Outline

- The Surface Fitting Problem
- Building a Set of Gluing Data

The Surface Fitting Problem

The Surface Fitting Problem

Given a mesh S_T in $\mathbb{R}^3$, a positive integer k, and a positive real number ϵ, our goal here is to fit a C^k surface, S, in $\mathbb{R}^3$ to S_T.

The Surface Fitting Problem

Given a mesh S_T in $\mathbb{R}^3$, a positive integer k, and a positive real number ϵ, our goal here is to fit a C^k surface, S, in $\mathbb{R}^3$ to S_T.

The Manifold-Based Approach:

We solve the fitting problem by defining a C^k parametric pseudo-surface, $\mathcal{M}$, such that S is the image, M, of $\mathcal{M}$ in $\mathbb{R}^3$.

The Surface Fitting Problem

The Surface Fitting Problem

Key Idea:

The Surface Fitting Problem

<u>**Key Idea**</u>:

- Use S_T to define the set of gluing data, $\mathcal{G}$, of $\mathcal{M}$.

The Surface Fitting Problem

Key Idea:

- Use S_T to define the set of gluing data, $\mathcal{G}$, of $\mathcal{M}$.

TOPOLOGY

The Surface Fitting Problem

Key Idea:

- Use S_T to define the set of gluing data, $\mathcal{G}$, of $\mathcal{M}$.

TOPOLOGY

- Use $|S_T|$ to define the set of parametrizations, $(\theta_i)_{i \in I}$, of $\mathcal{M}$.

The Surface Fitting Problem

Key Idea:

- Use S_T to define the set of gluing data, $\mathcal{G}$, of $\mathcal{M}$.

TOPOLOGY

- Use $|S_T|$ to define the set of parametrizations, $(\theta_i)_{i \in I}$, of $\mathcal{M}$.

GEOMETRY

Building a Set of Gluing Data

Building a Set of Gluing Data

To define $\mathcal{G}$, we must

Building a Set of Gluing Data

To define $\mathcal{G}$, we must

- define the p-domains, $(\Omega_i)_{i \in I}$,

Building a Set of Gluing Data

To define $\mathcal{G}$, we must

- define the p-domains, $(\Omega_i)_{i \in I}$,

- define the gluing domains, $(\Omega_{ij})_{(i,j) \in I \times I}$,

Building a Set of Gluing Data

To define $\mathcal{G}$, we must

- define the p-domains, $(\Omega_i)_{i \in I}$,

- define the gluing domains, $(\Omega_{ij})_{(i,j) \in I \times I}$,

- define the transition functions, $(\varphi_{i,j})_{(i,j) \in K \times K}$.

Building a Set of Gluing Data

To define $\mathcal{G}$, we must

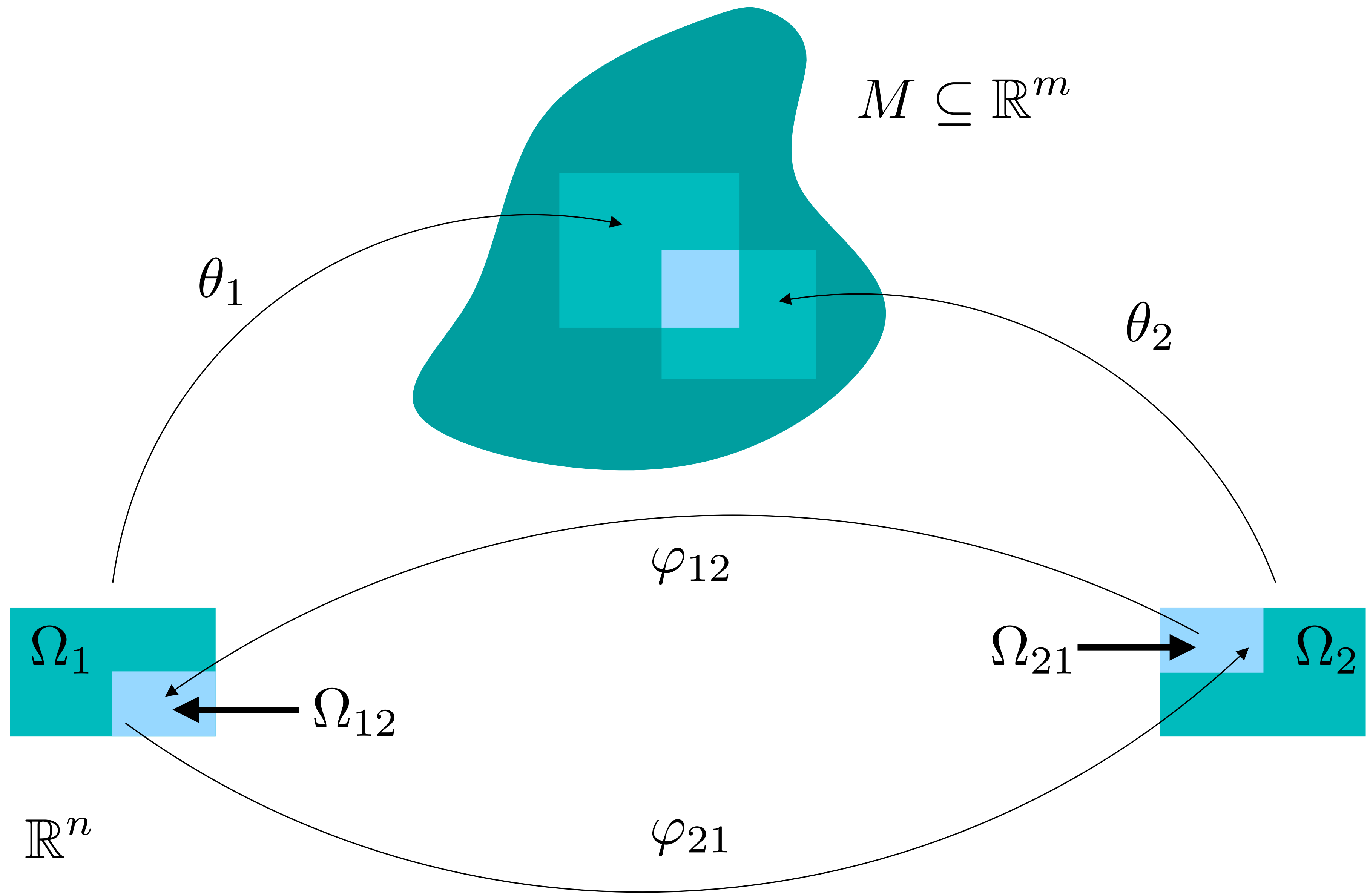
- define the p-domains, $(\Omega_i)_{i\in I}$,
- define the gluing domains, $(\Omega_{ij})_{(i,j)\in I\times I}$,
- define the transition functions, $(\varphi_{i,j})_{(i,j)\in K\times K}$.

$$\mathcal{G} = \big((\Omega)_{i\in I}, (\Omega_{i,j})_{(i,j)\in I\times I}, (\varphi_{i,j})_{(i,j)\in K\times K}\big)$$

Building a Set of Gluing Data

Building a Set of Gluing Data

The BIG PICTURE



Building a Set of Gluing Data

Building a Set of Gluing Data

p-Domains

Building a Set of Gluing Data

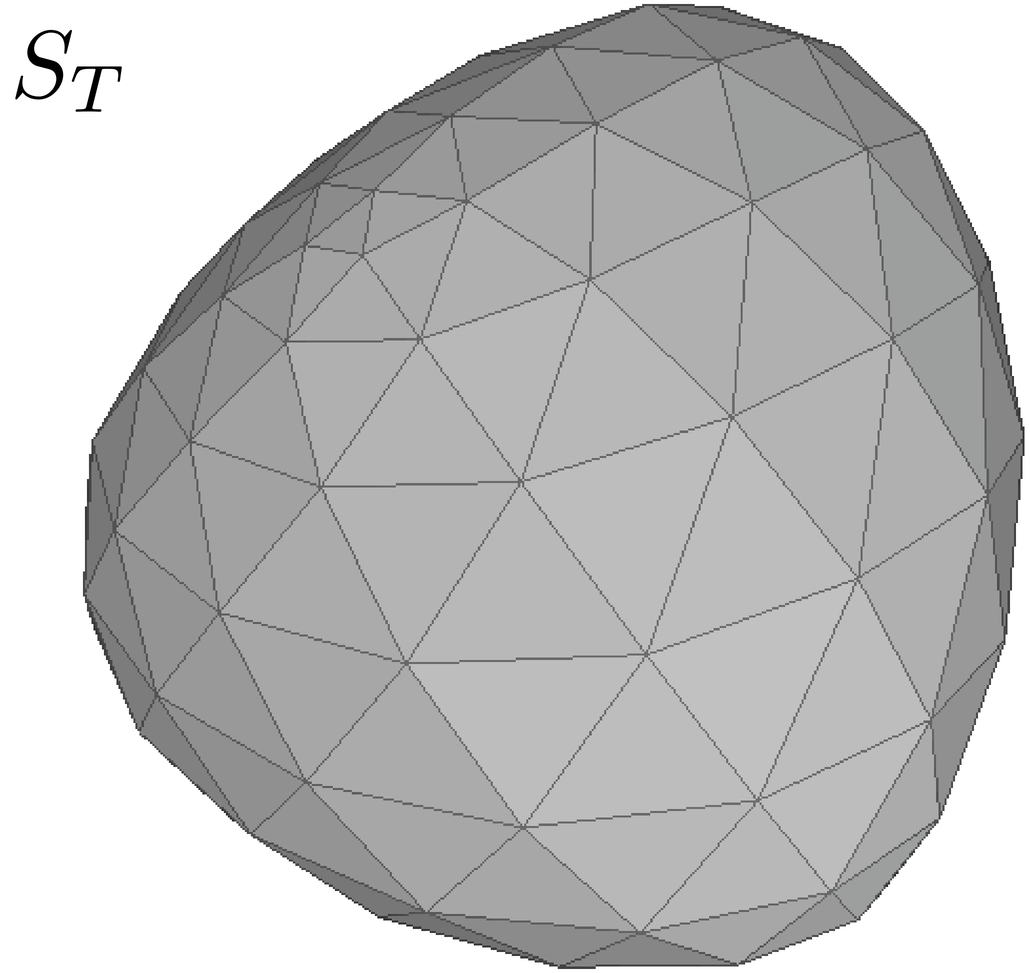
p-**Domains**

Assume that S_T is a **triangle** mesh (i.e., a simplicial surface).

Building a Set of Gluing Data

p-**Domains**

Assume that S_T is a **triangle** mesh (i.e., a simplicial surface).



Building a Set of Gluing Data

Building a Set of Gluing Data

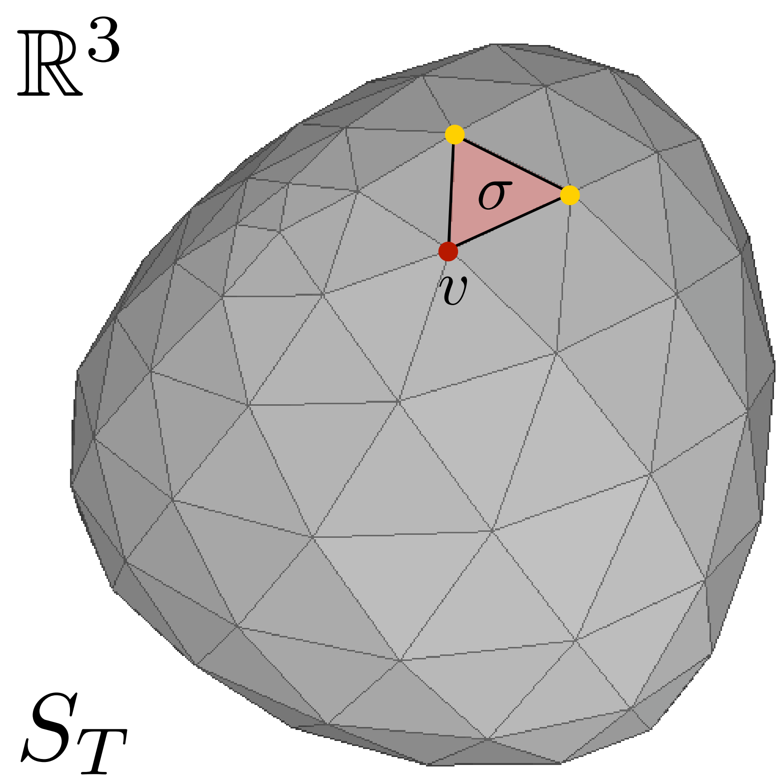
Let

$$I = \{(\sigma, v) \mid \sigma \text{ is a triangle of } S_T \text{ and } v \text{ is a vertex of } \sigma\}.$$

Building a Set of Gluing Data

Let

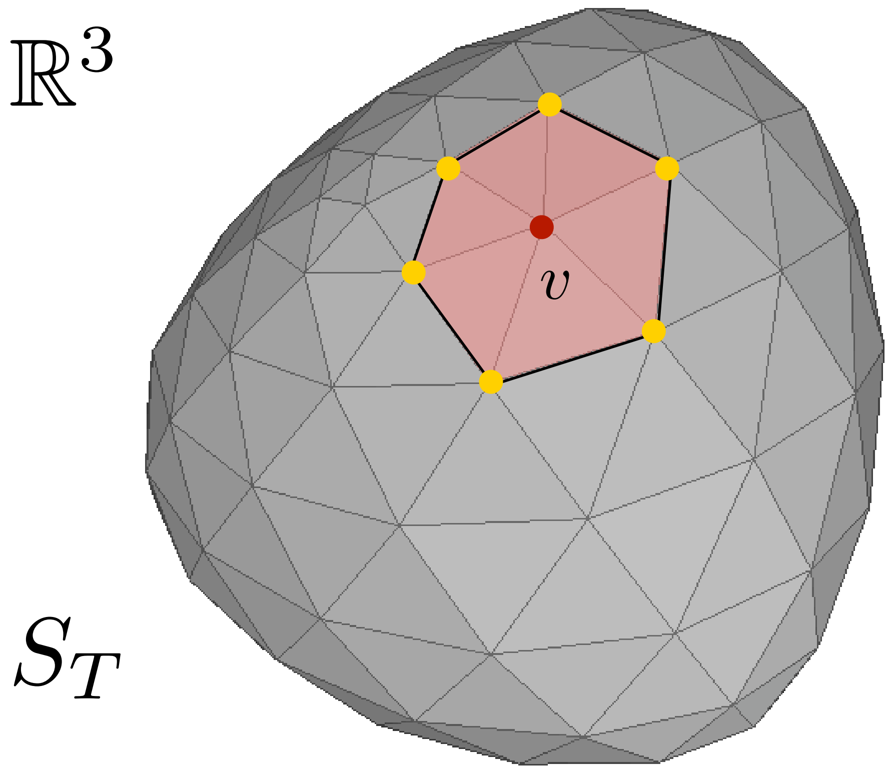
$$I = \{(\sigma, v) \mid \sigma \text{ is a triangle of } S_T \text{ and } v \text{ is a vertex of } \sigma\}\,.$$



Building a Set of Gluing Data

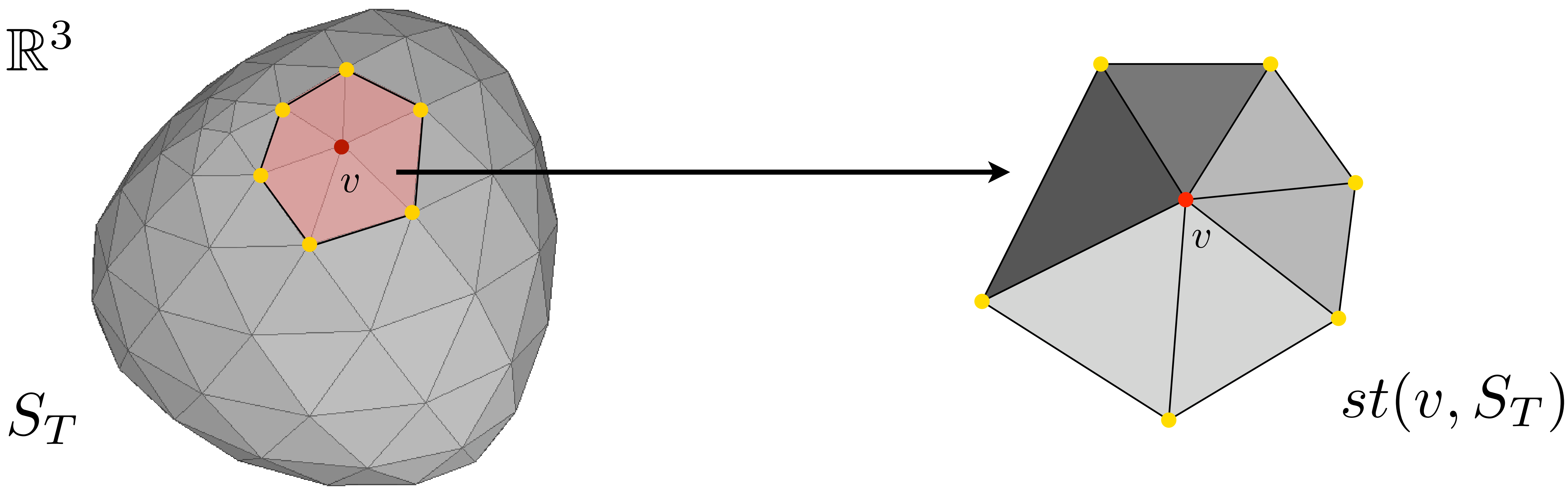
Building a Set of Gluing Data

For every vertex, v, of S_T, consider its **star**, $st(v, S_T)$:



Building a Set of Gluing Data

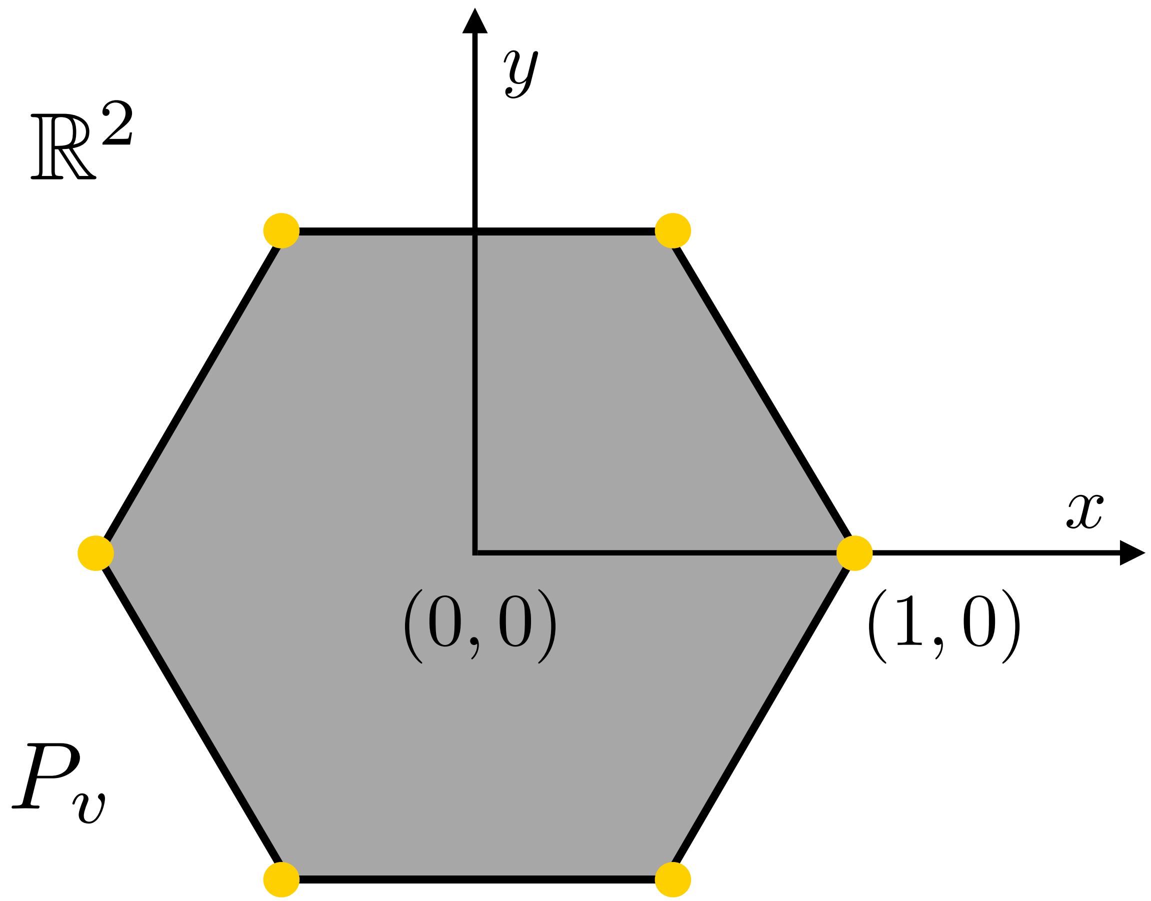
For every vertex, v, of S_T, consider its **star**, $st(v, S_T)$:



Building a Set of Gluing Data

Building a Set of Gluing Data

Define the **P-polygon, P_v, associated with** v as the m_v-gon inscribed in the circle of radius 1 and centered at the origin in $\mathbb{R}^2$:



m_v is the **degree** of v in S_T.

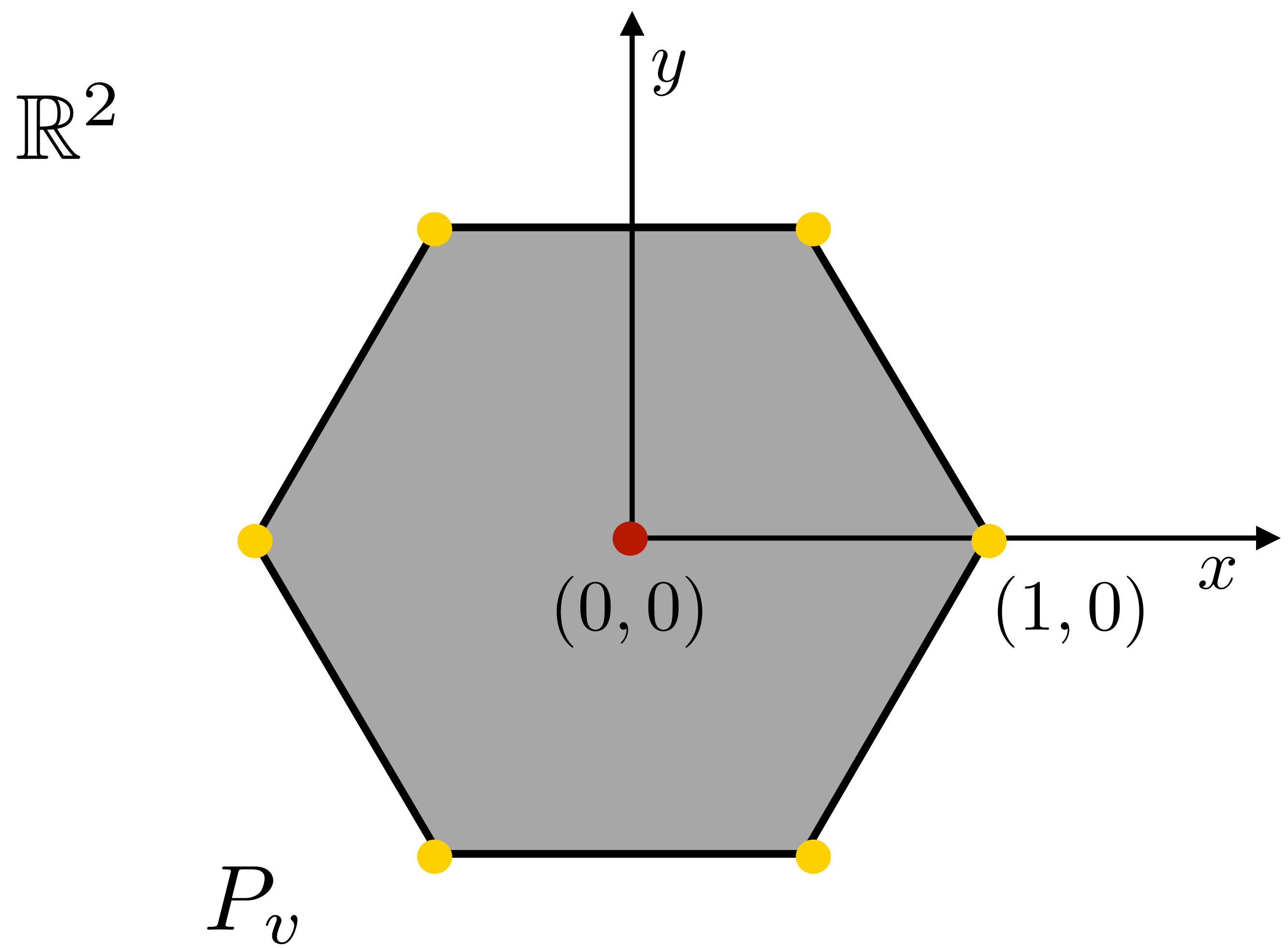
Building a Set of Gluing Data

Building a Set of Gluing Data

Define the **triangulation,** T_v**, associated with** v by adding straight edges (diagonals) connecting the barycenter of P_v to its vertices:

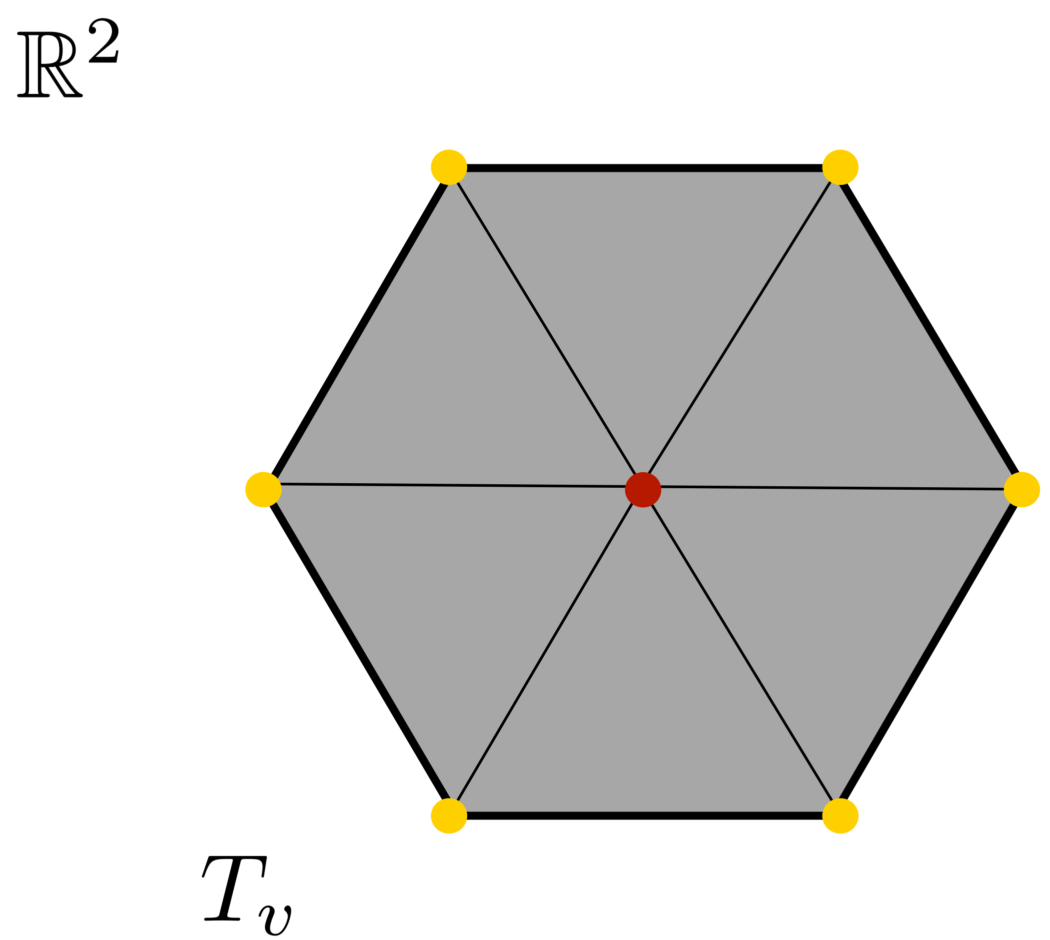
Building a Set of Gluing Data

Define the **triangulation, T_v, associated with** v by adding straight edges (diagonals) connecting the barycenter of P_v to its vertices:



Building a Set of Gluing Data

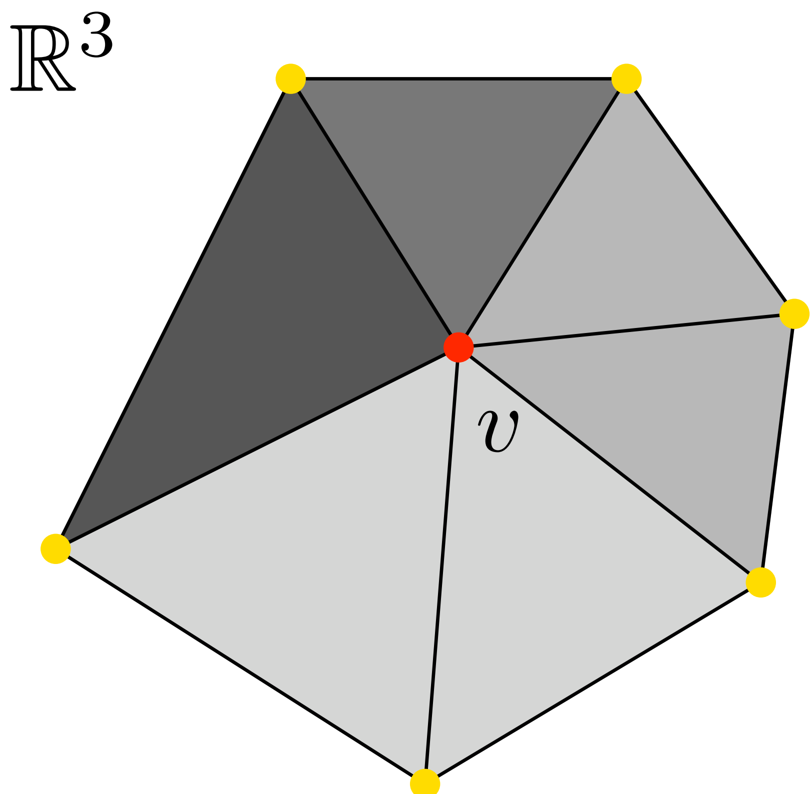
Define the **triangulation,** T_v**, associated with** v by adding straight edges (diagonals) connecting the barycenter of P_v to its vertices:



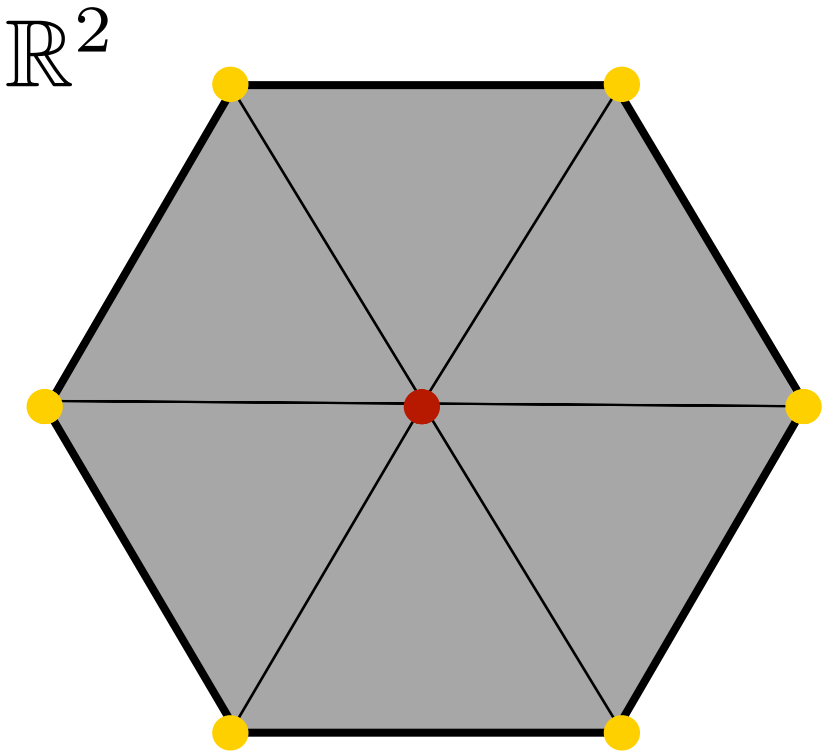
Building a Set of Gluing Data

Building a Set of Gluing Data

Remark: T_v is a parametrization of $st(v, S_T)$ in $\mathbb{R}^2$:



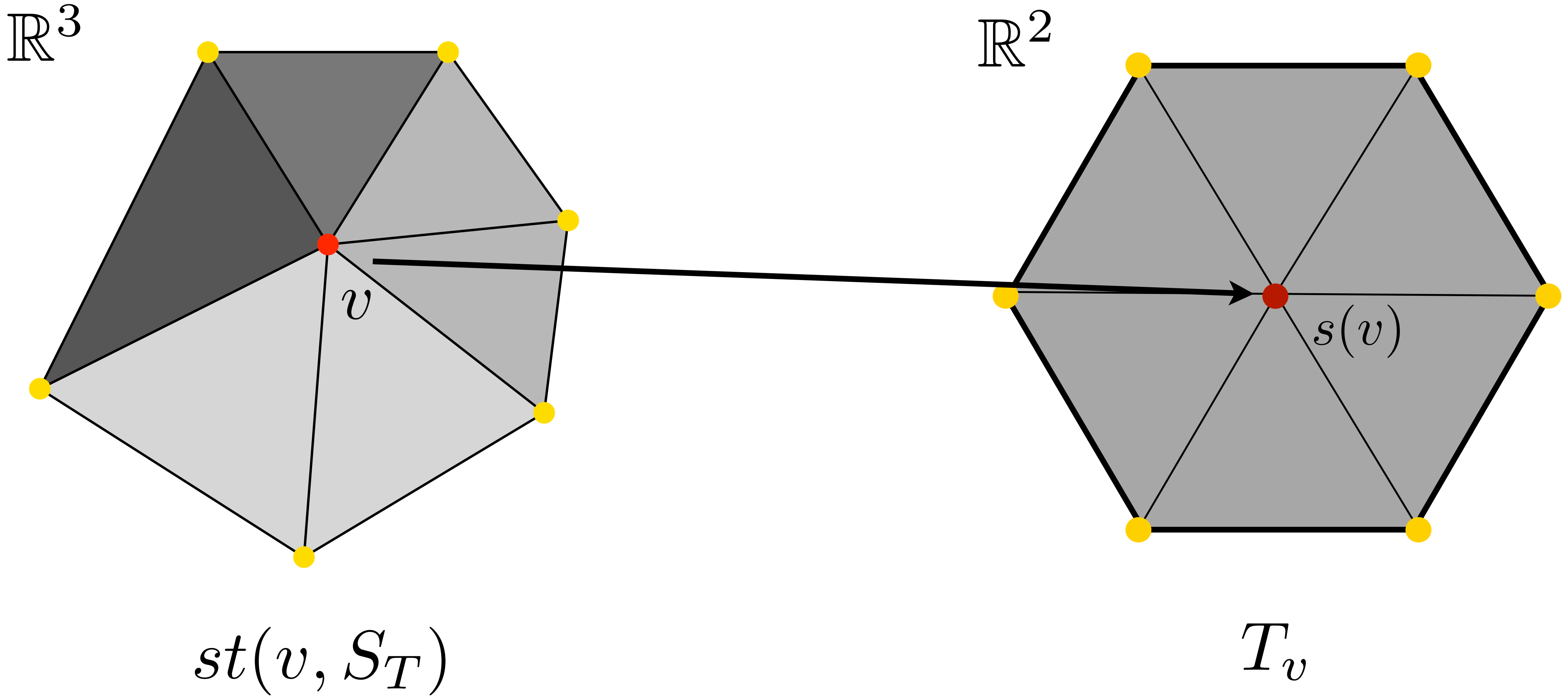
$st(v, S_T)$



T_v

Building a Set of Gluing Data

Remark: T_v is a parametrization of $st(v, S_T)$ in $\mathbb{R}^2$:



$$s : st(v, S_T) \to T_v$$

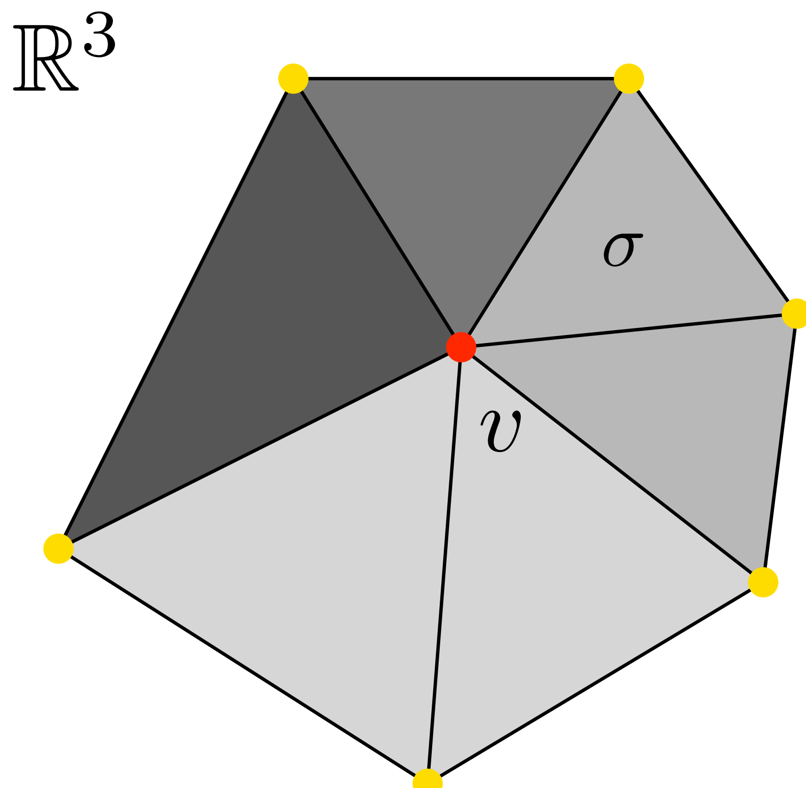
Building a Set of Gluing Data

Building a Set of Gluing Data

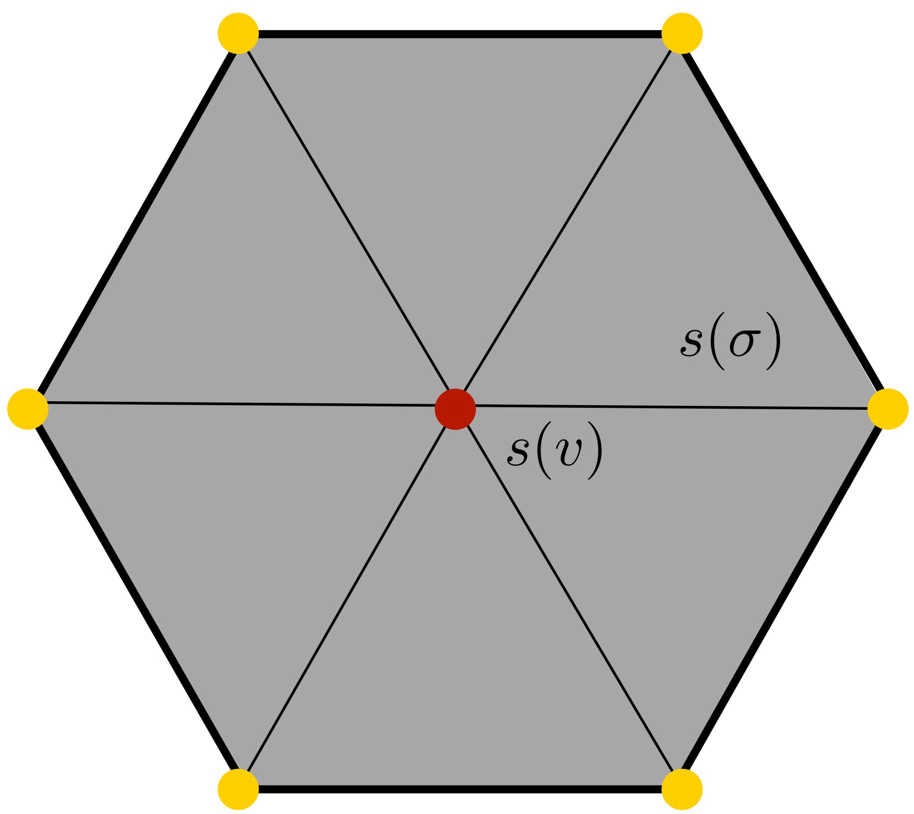
For each triangle σ of S_T and vertex v of σ, we define the **overlapping point**, $r_{v,\sigma}$, associated with $s(\sigma)$ in T_v, as follows:

Building a Set of Gluing Data

For each triangle σ of S_T and vertex v of σ, we define the **overlapping point**, $r_{v,\sigma}$, associated with $s(\sigma)$ in T_v, as follows:



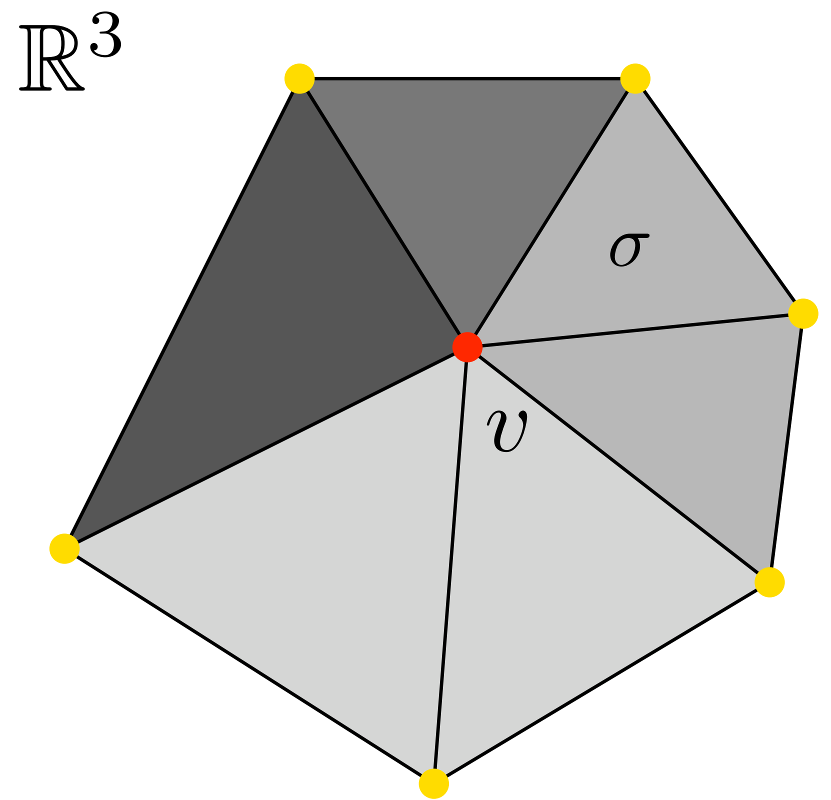
$st(v, S_T)$



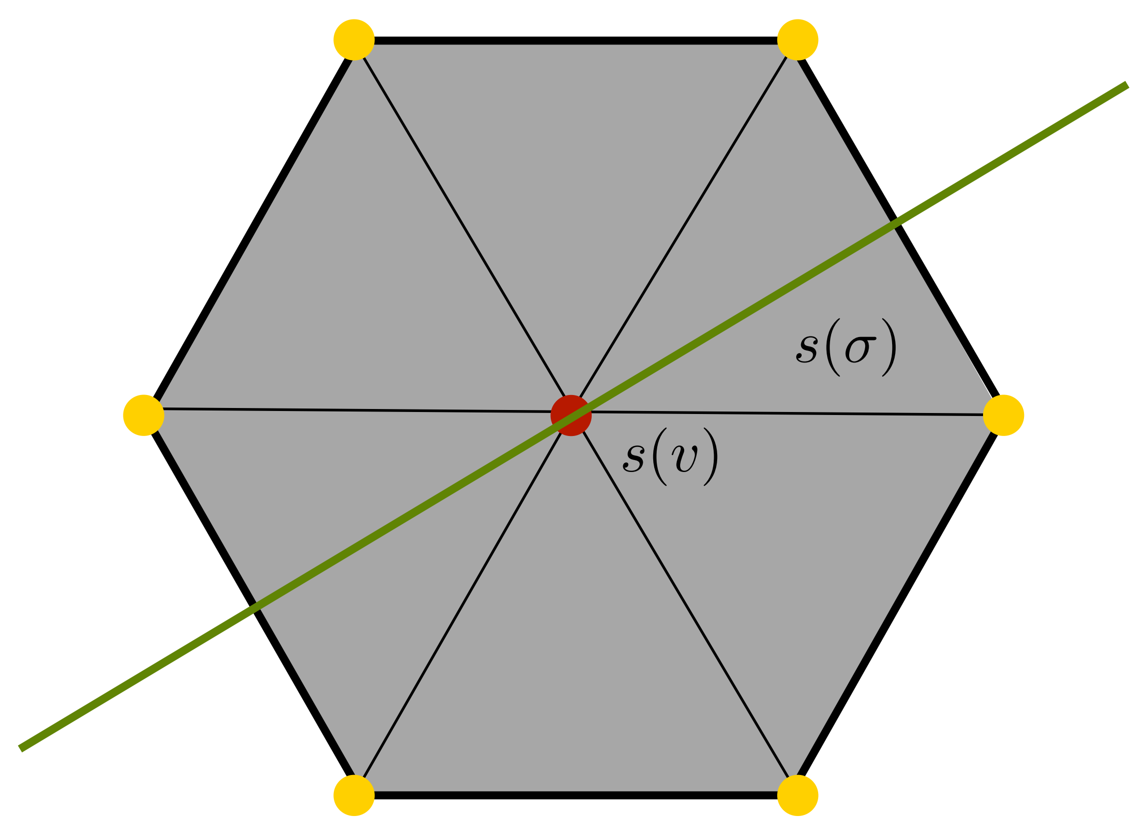
T_v

Building a Set of Gluing Data

For each triangle σ of S_T and vertex v of σ, we define the **overlapping point**, $r_{v,\sigma}$, associated with $s(\sigma)$ in T_v, as follows:



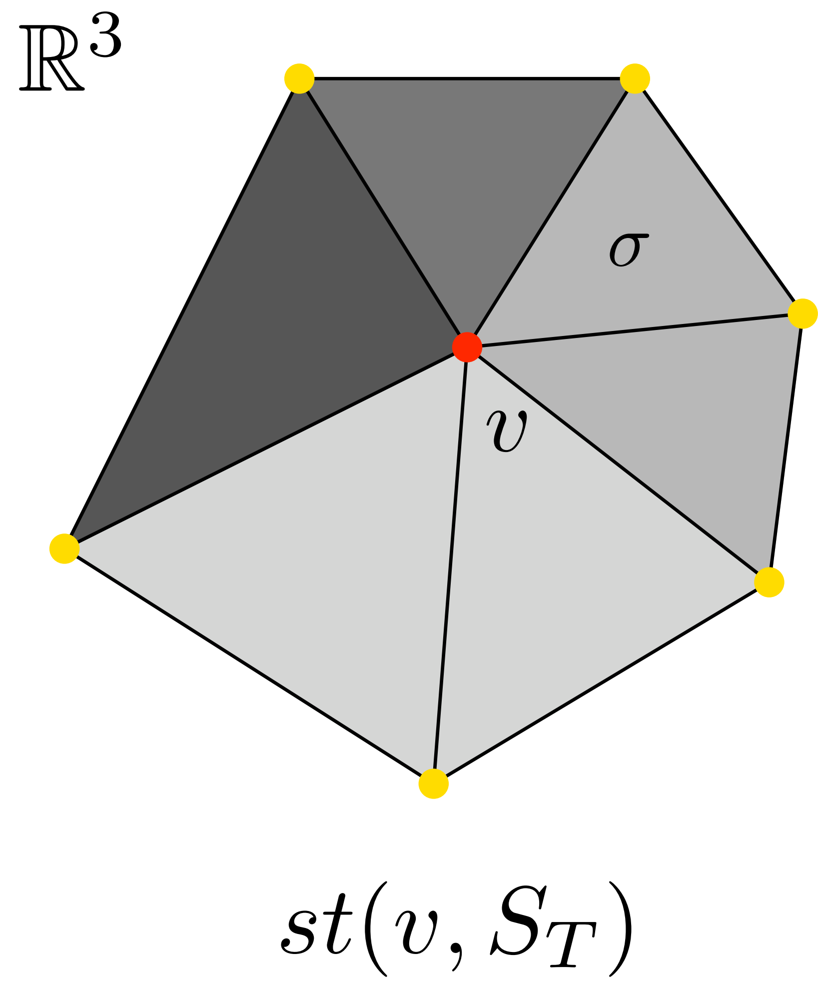
$st(v, S_T)$

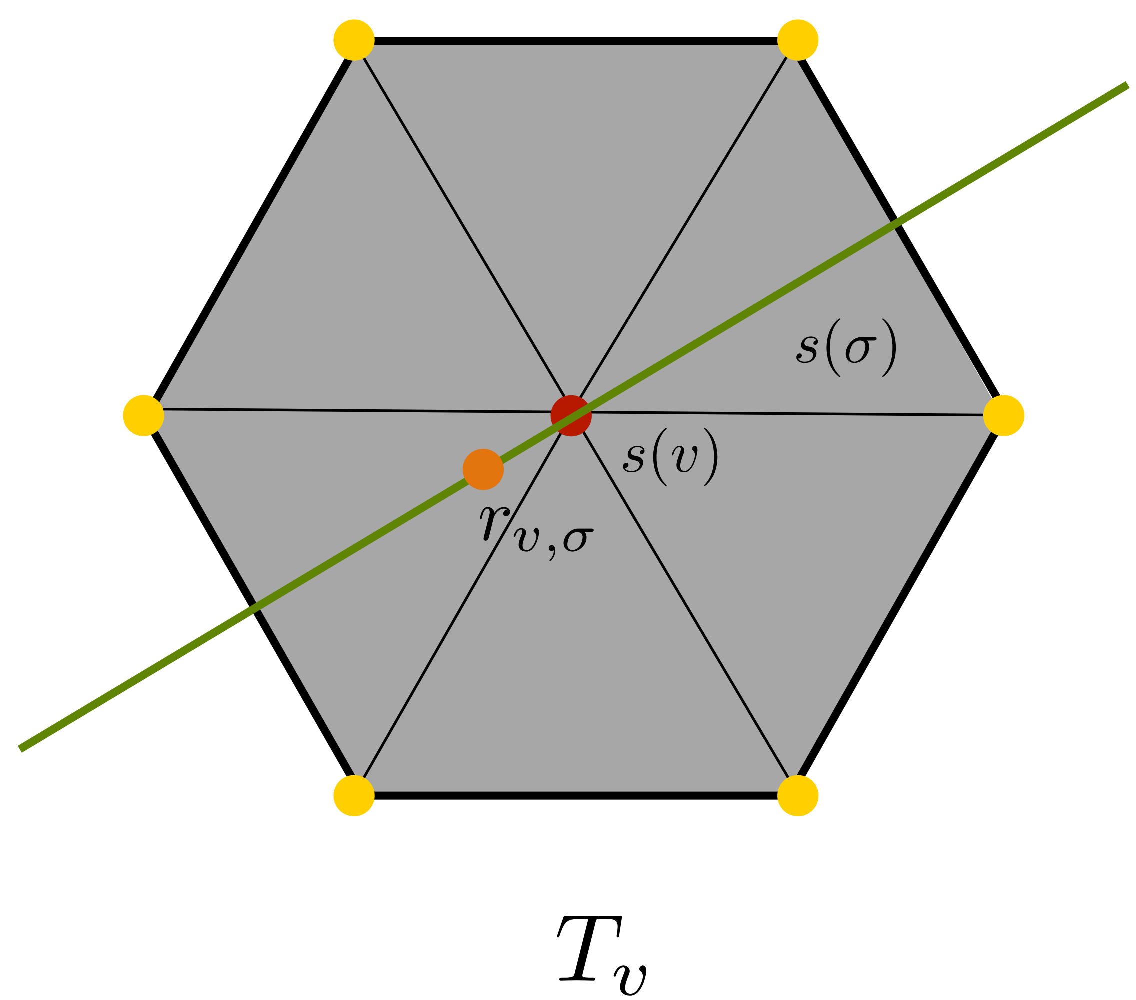


T_v

Building a Set of Gluing Data

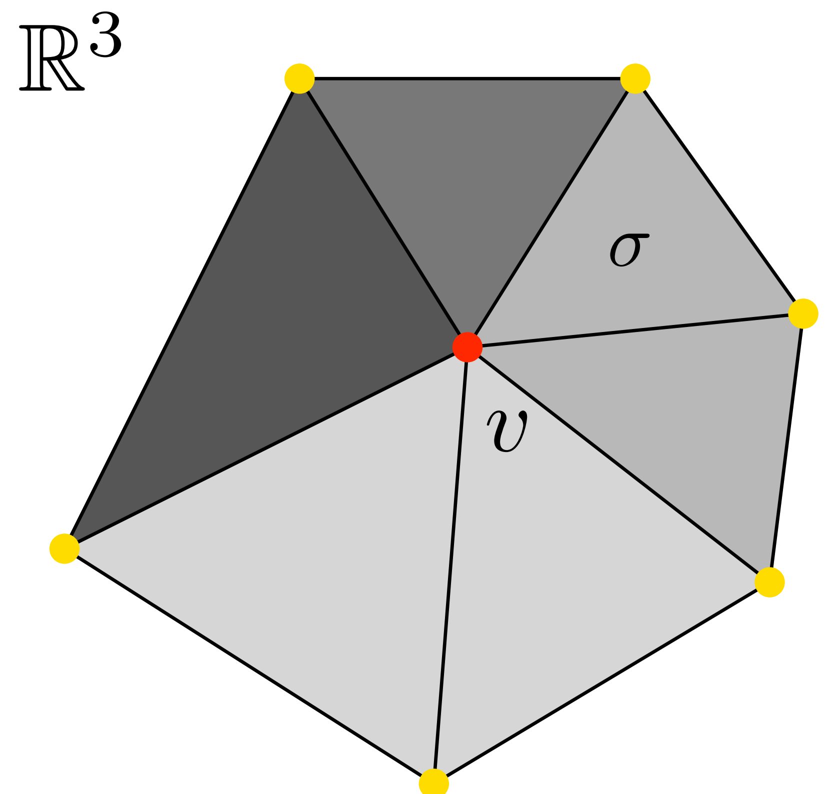
For each triangle σ of S_T and vertex v of σ, we define the **overlapping point**, $r_{v,\sigma}$, associated with $s(\sigma)$ in T_v, as follows:



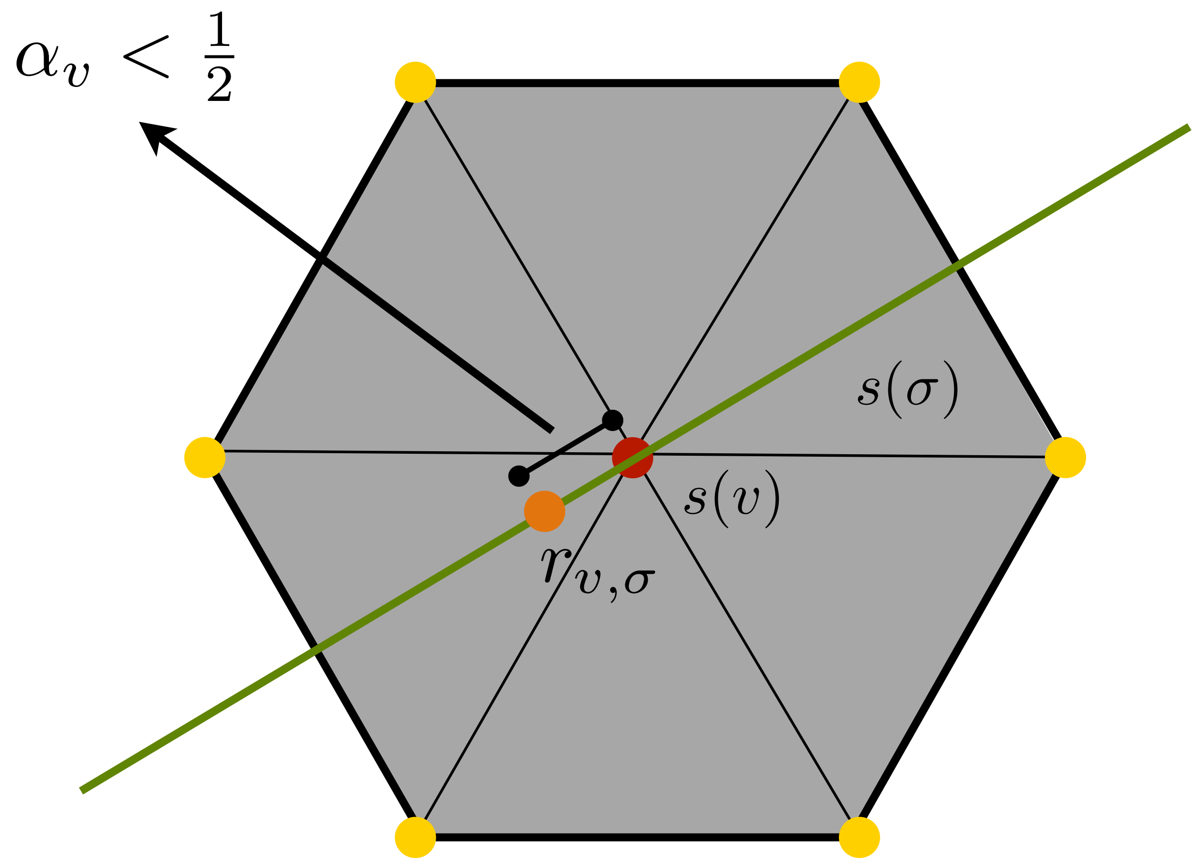


Building a Set of Gluing Data

For each triangle σ of S_T and vertex v of σ, we define the **overlapping point**, $r_{v,\sigma}$, associated with $s(\sigma)$ in T_v, as follows:



$st(v, S_T)$



T_v

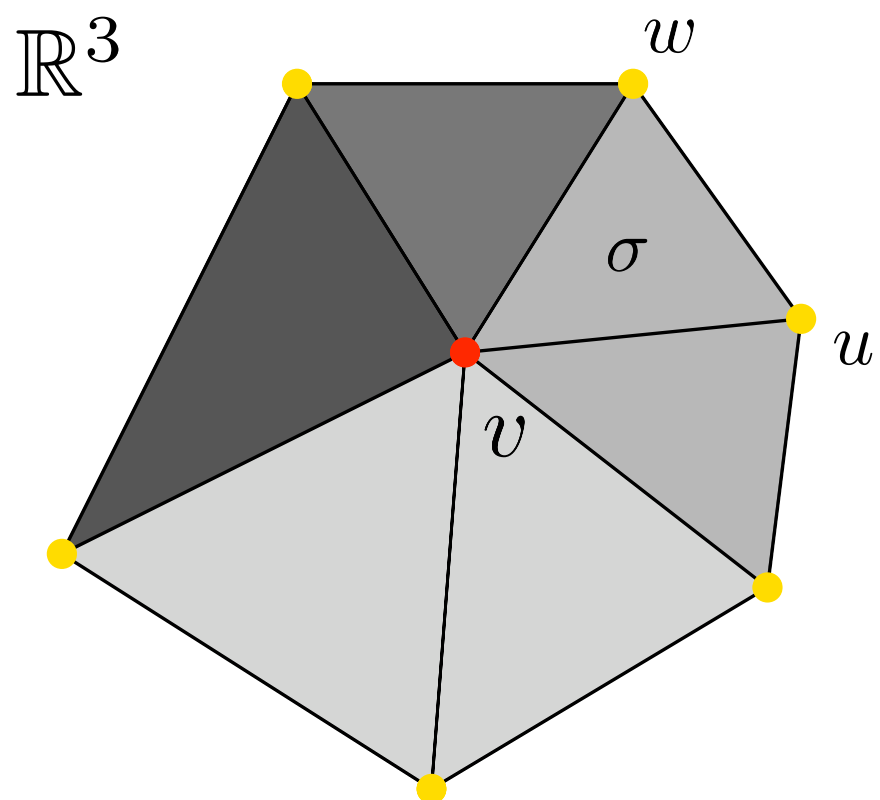
Building a Set of Gluing Data

Building a Set of Gluing Data

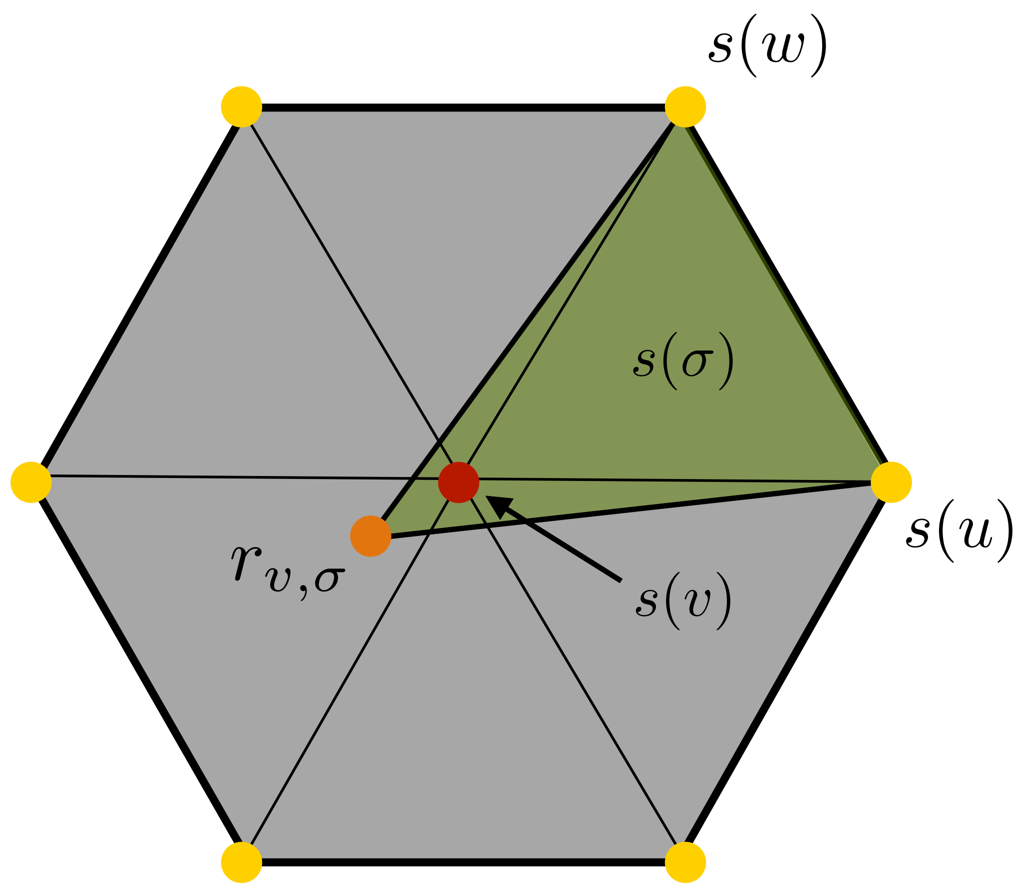
If $\sigma = [v, u, w]$ then consider the triangle $[r_{\sigma,v}, s(u), s(w)]$:

Building a Set of Gluing Data

If $\sigma = [v, u, w]$ then consider the triangle $[r_{\sigma,v}, s(u), s(w)]$:



$st(v, S_T)$



T_v

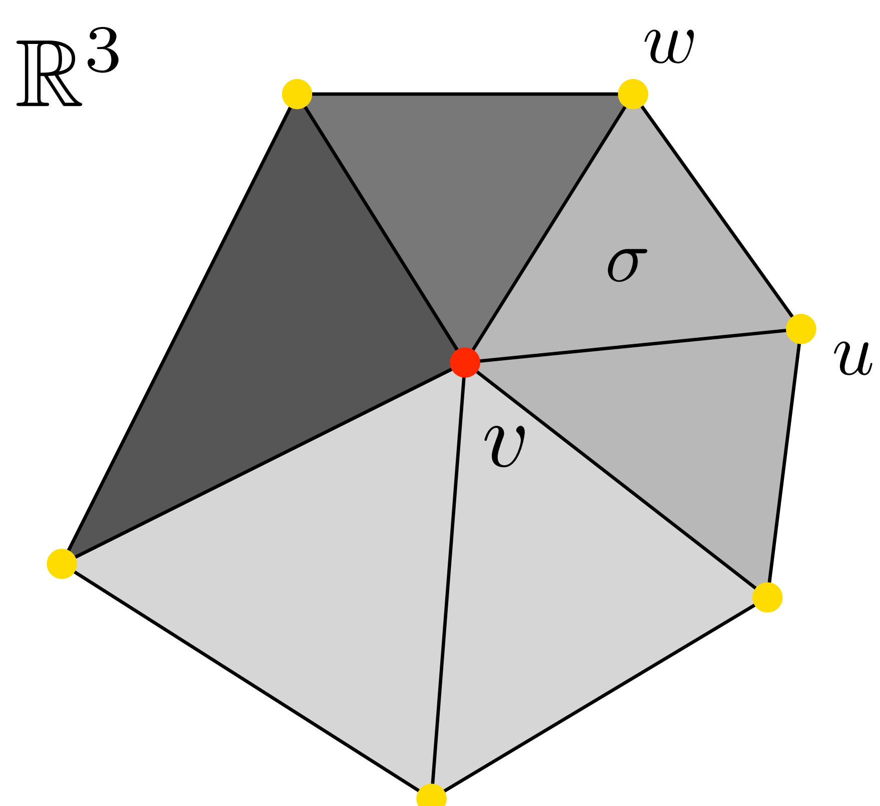
Building a Set of Gluing Data

Building a Set of Gluing Data

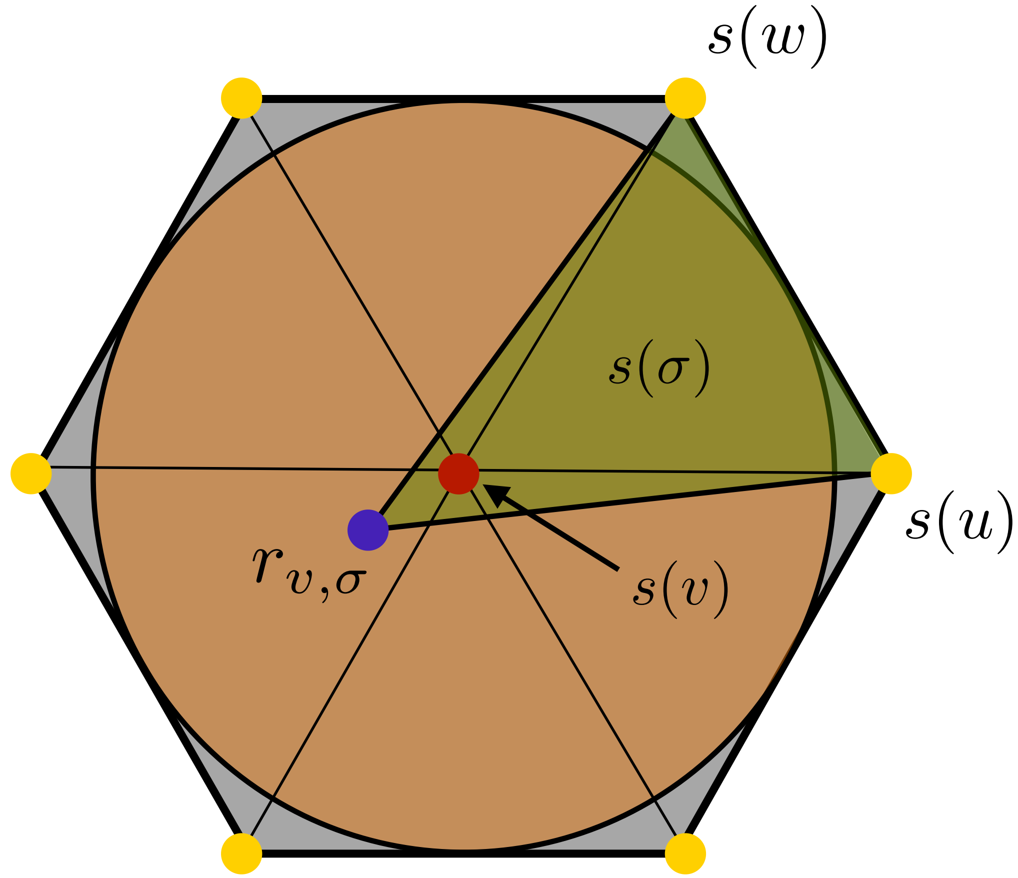
Consider the circle, C_v, inscribed in P_v:

Building a Set of Gluing Data

Consider the circle, C_v, inscribed in P_v:



$st(v, S_T)$



T_v

Building a Set of Gluing Data

Building a Set of Gluing Data

We let $\Omega_{(\sigma,v)}$ be

$$C_v \cap \mathrm{int}([r_{v,\sigma}, s(u), s(w)])\,,$$

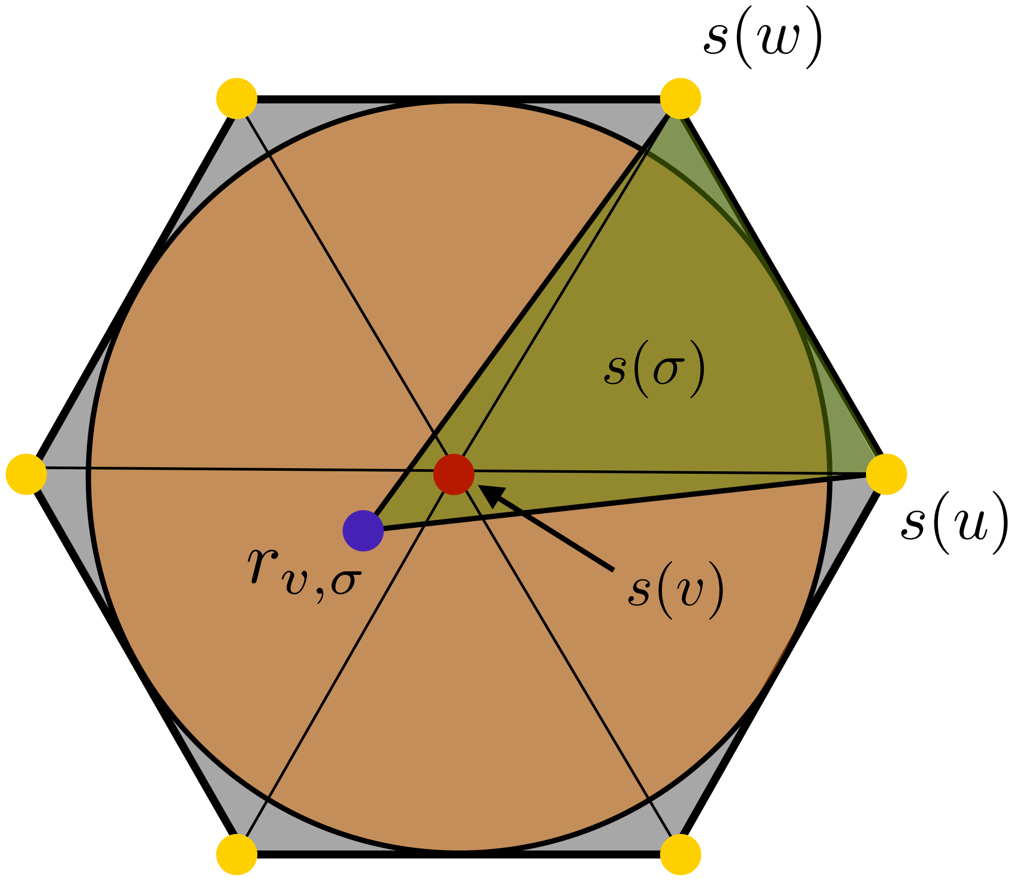
where $\mathrm{int}([r_{v,\sigma}, s(u), s(w)])$ is the interior of $[r_{v,\sigma}, s(u), s(w)]$.

Building a Set of Gluing Data

We let $\Omega_{(\sigma,v)}$ be

$$C_v \cap \mathrm{int}([r_{v,\sigma}, s(u), s(w)]) \, ,$$

where $\mathrm{int}([r_{v,\sigma}, s(u), s(w)])$ is the interior of $[r_{v,\sigma}, s(u), s(w)]$.

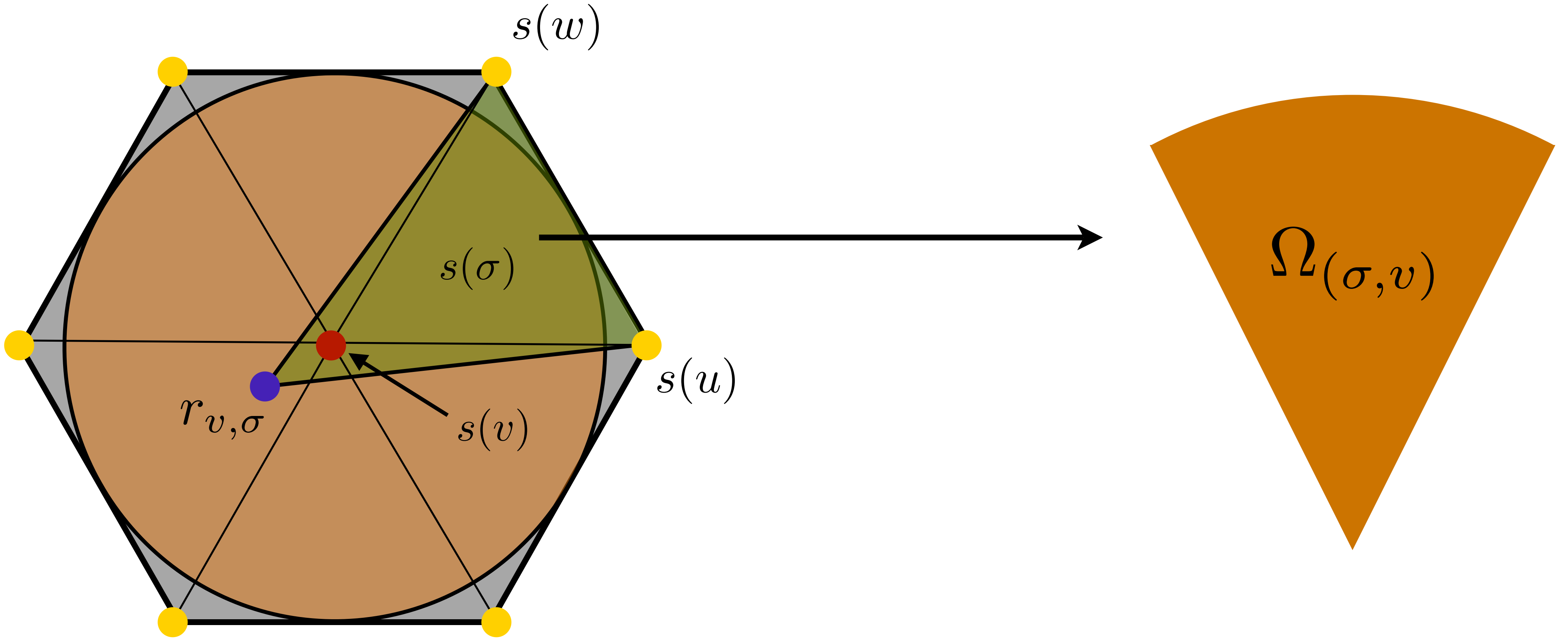


Building a Set of Gluing Data

We let $\Omega_{(\sigma,v)}$ be

$$C_v \cap \mathrm{int}([r_{v,\sigma}, s(u), s(w)])\,,$$

where $\mathrm{int}([r_{v,\sigma}, s(u), s(w)])$ is the interior of $[r_{v,\sigma}, s(u), s(w)]$.



Building a Set of Gluing Data

Remark:

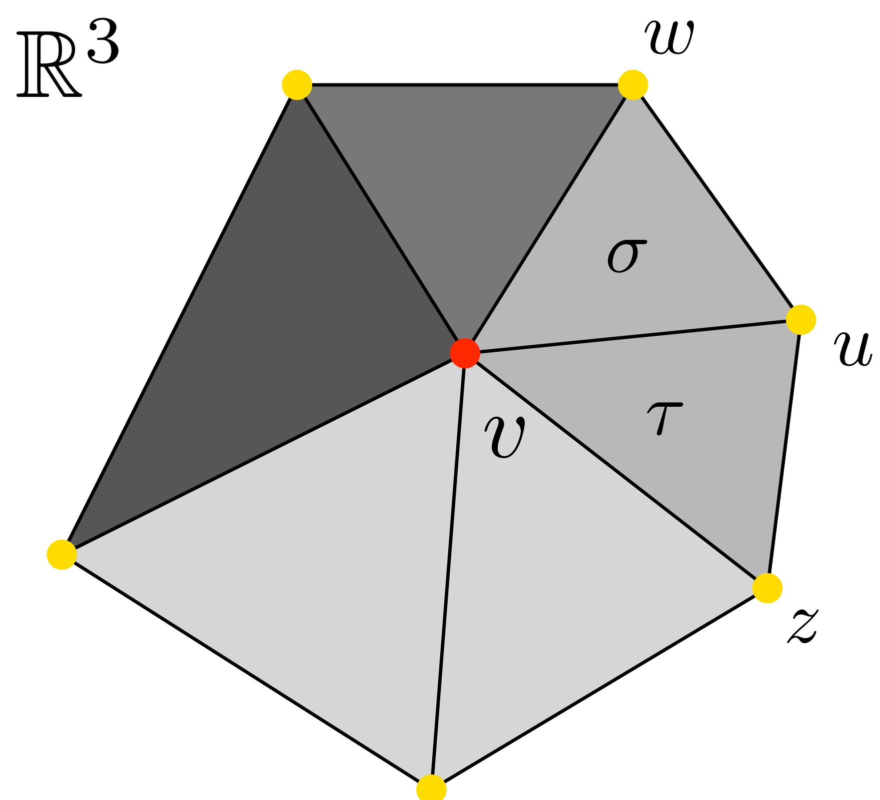
From Jean Gallier's lecture, we should have

$$\Omega_{(\sigma,v)} \cap \Omega_{(\tau,u)} = \emptyset \, ,$$

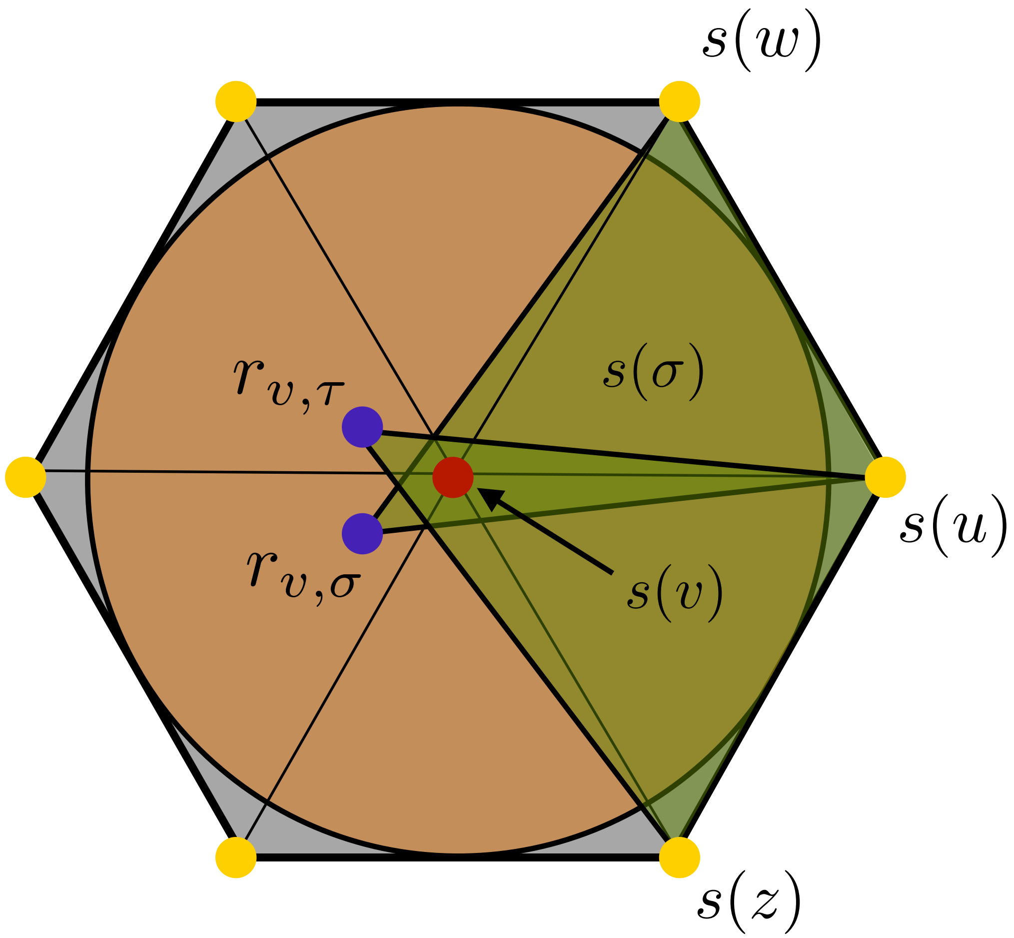
for any two pairs, (σ, v) and (τ, u), in I. Did I make it right?

Building a Set of Gluing Data

Building a Set of Gluing Data

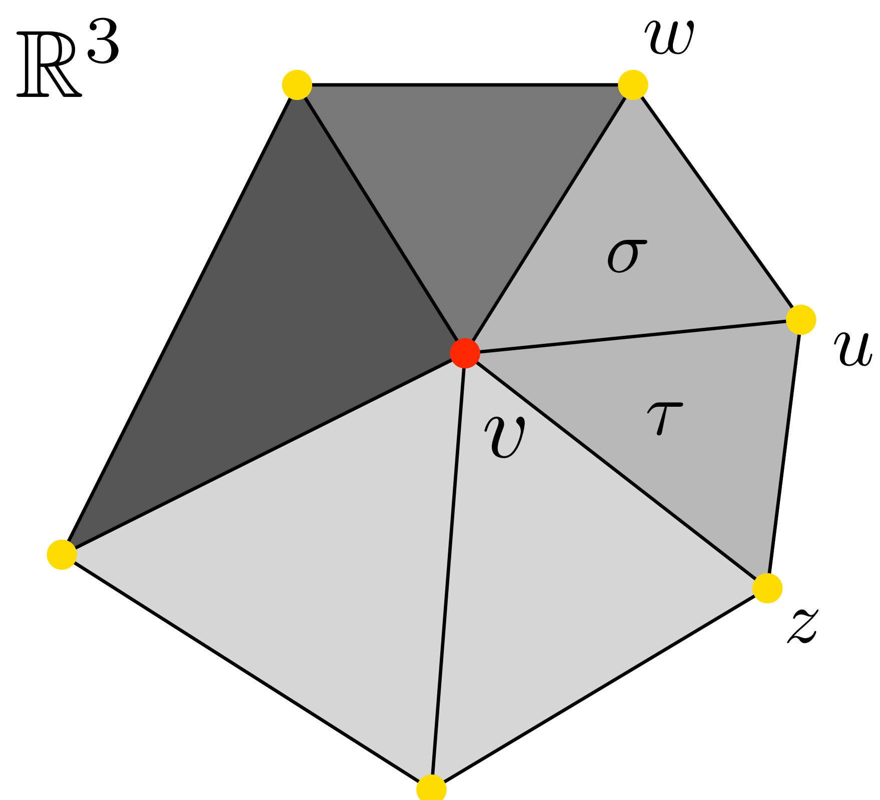


$st(v, S_T)$

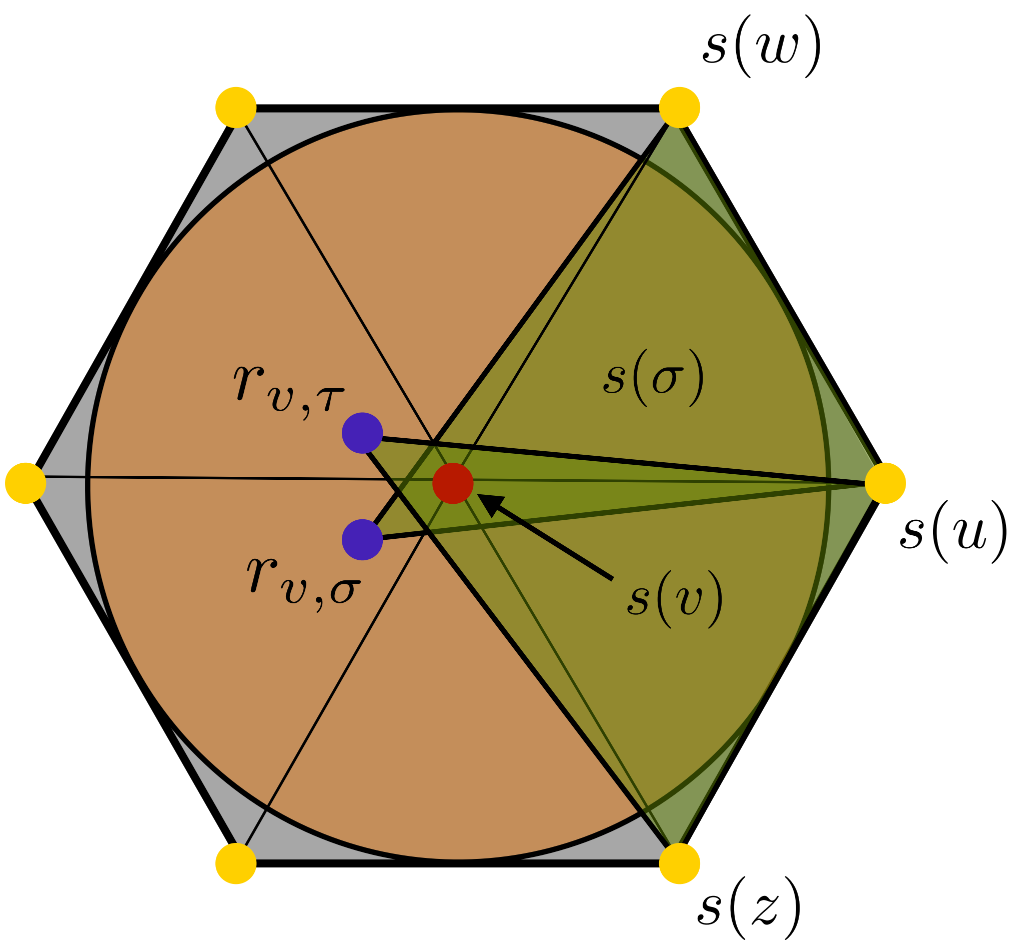


T_v

Building a Set of Gluing Data



$st(v, S_T)$



T_v

Clearly, $\Omega_{(\sigma,v)} \cap \Omega_{(\tau,v)} \neq \emptyset$.

Building a Set of Gluing Data

Building a Set of Gluing Data

So, I did NOT make it right.

Building a Set of Gluing Data

So, I did NOT make it right.

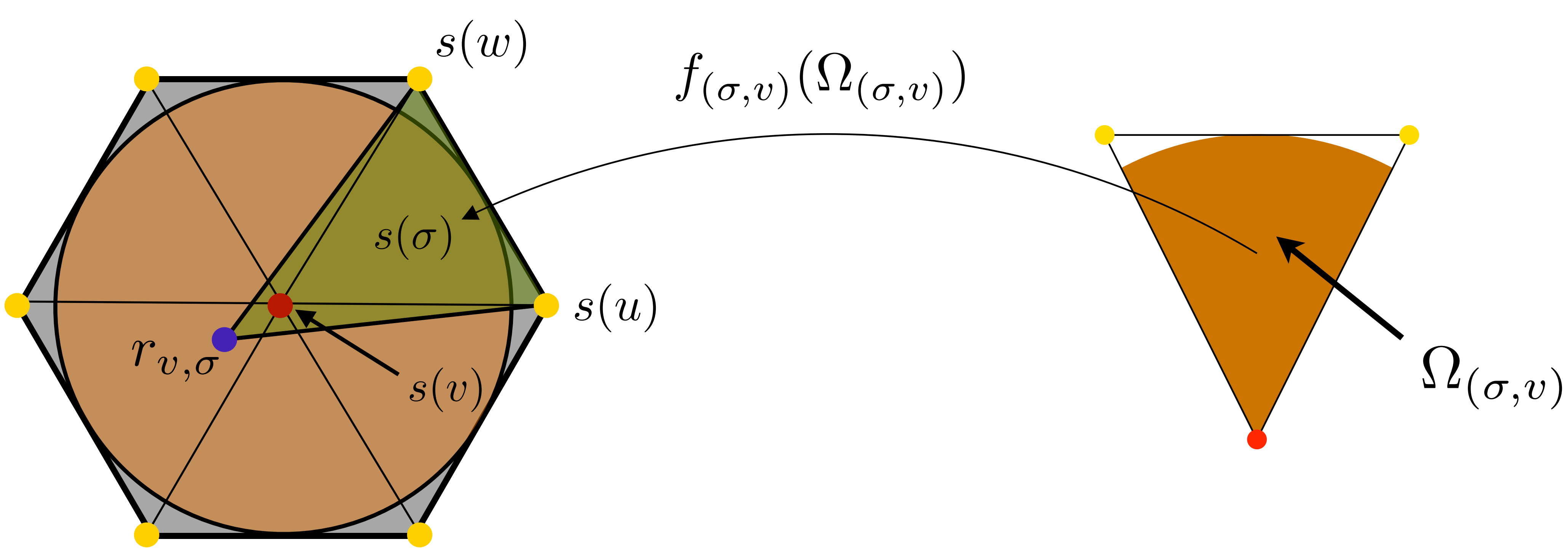
What now?

Building a Set of Gluing Data

So, I did NOT make it right.

What now?

We can *fix* that by letting $\Omega_{(\sigma,v)}$ be a set inside a triangle which is the inverse image of $[r_{v,\sigma}, s(u), s(w)]$ under a rigid transformation!



Building a Set of Gluing Data

Building a Set of Gluing Data

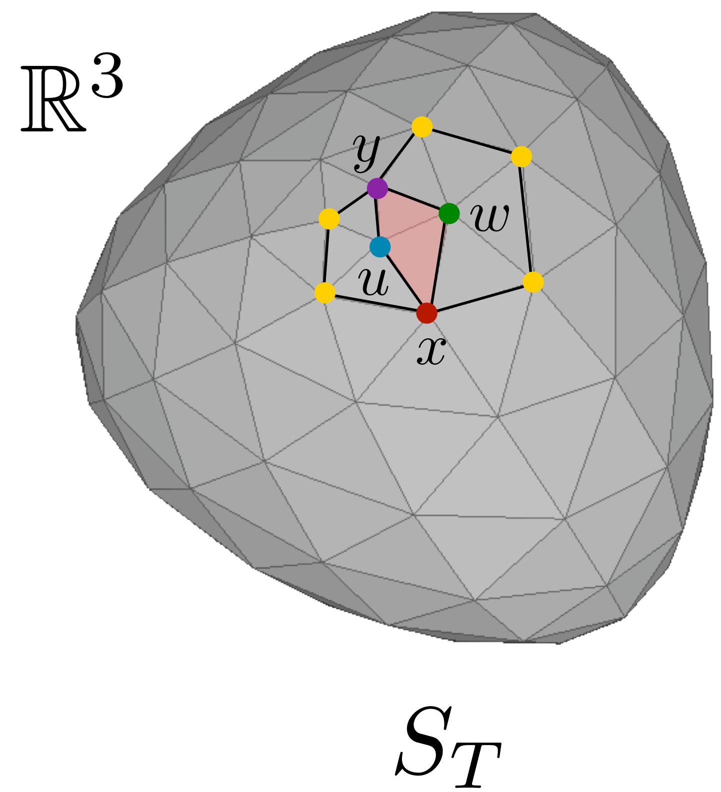
Since I is a finite set and the "enclosing" triangles are compact, we can certainly separate each p-domain from the others in $\mathbb{R}^2$.

Building a Set of Gluing Data

Gluing domains

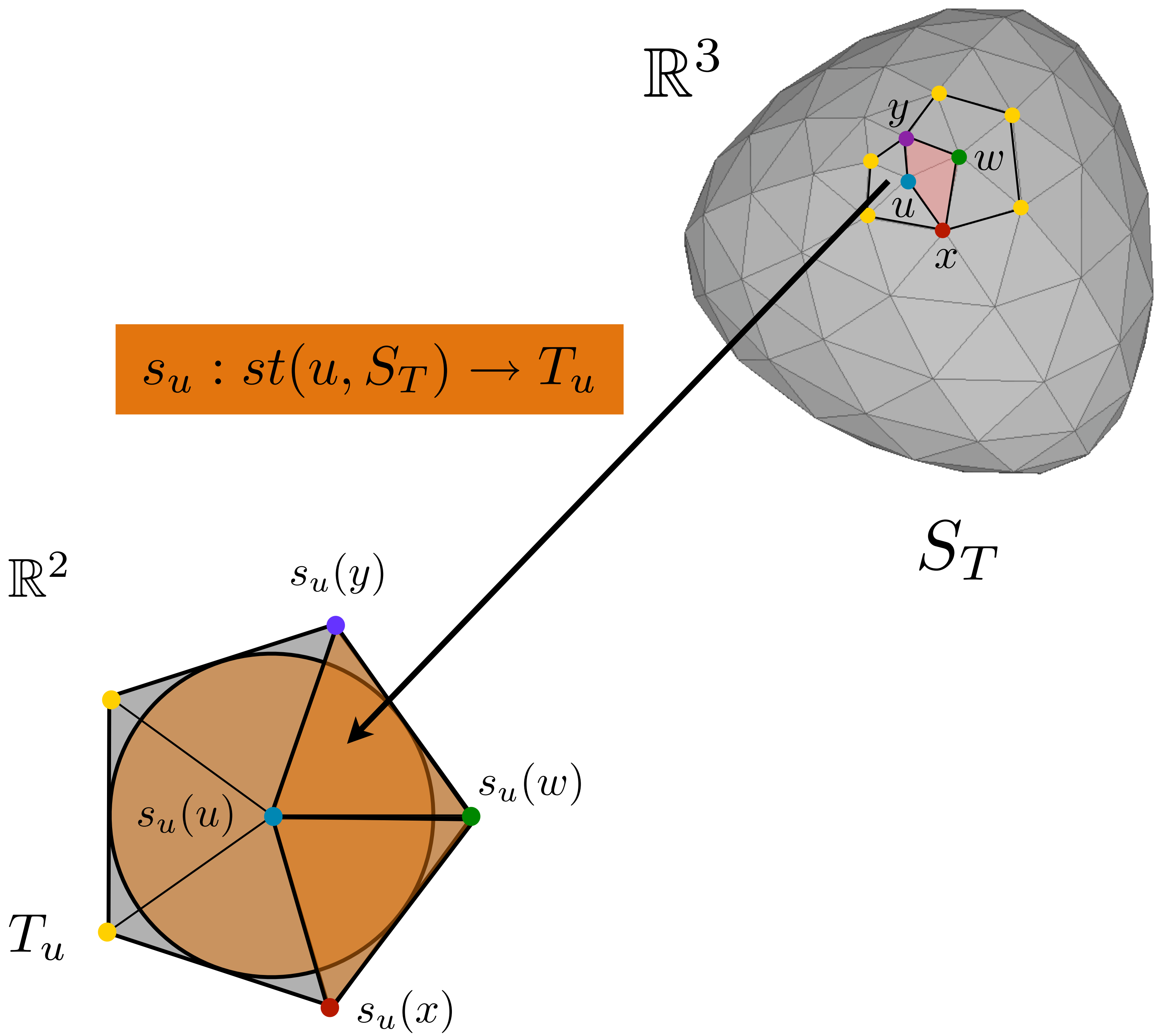
Building a Set of Gluing Data

Gluing domains



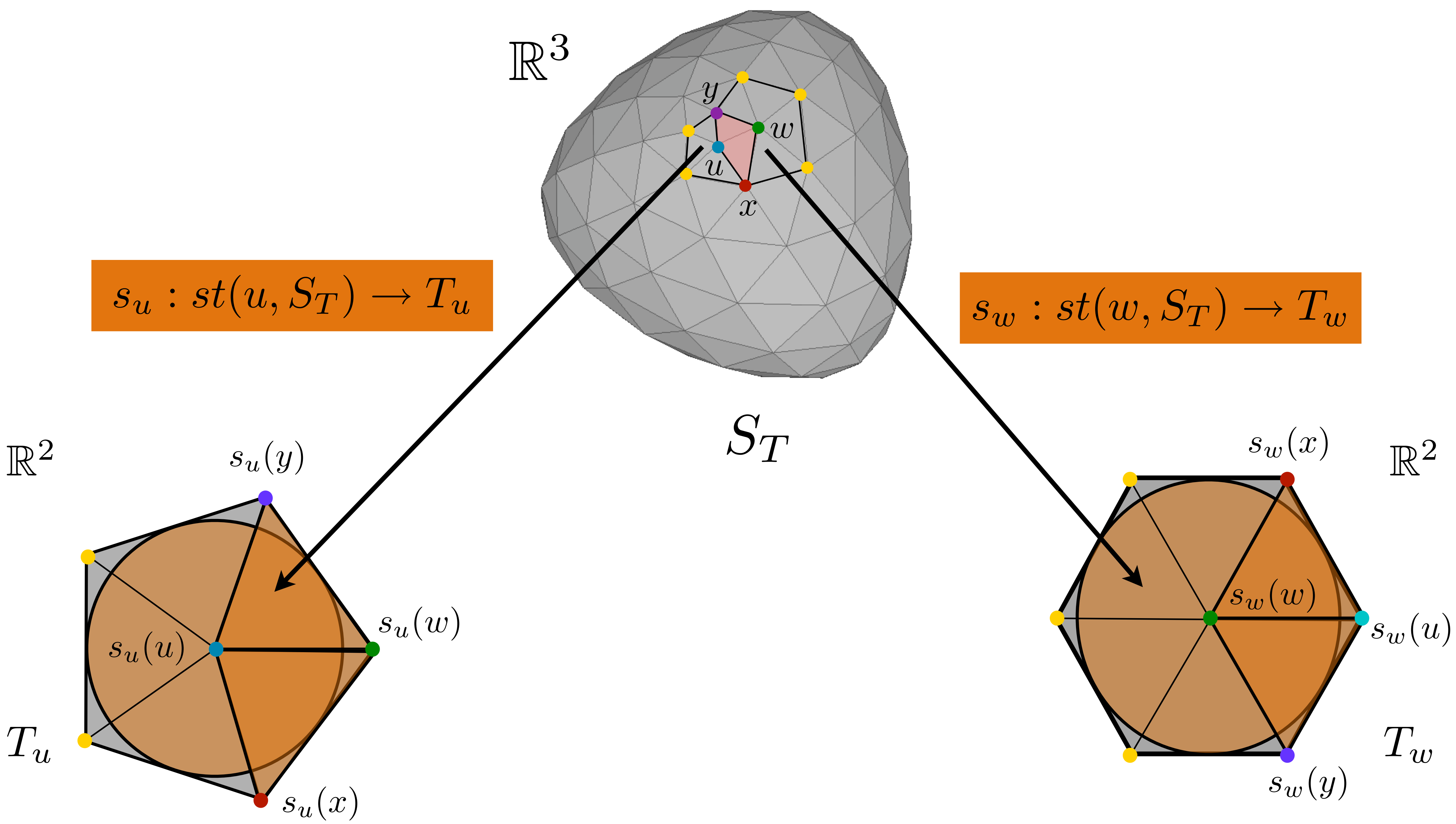
Building a Set of Gluing Data

Gluing domains



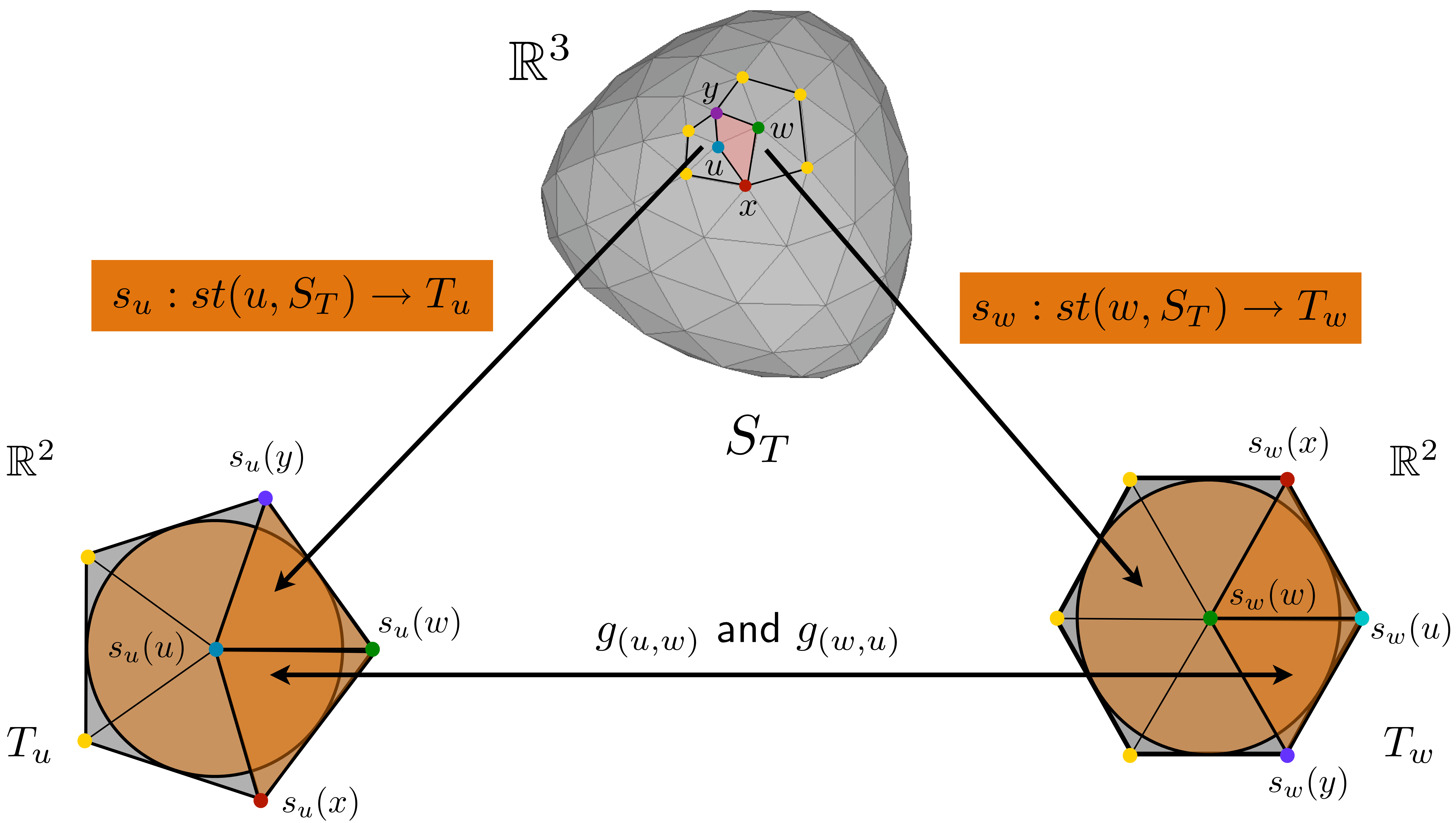
Building a Set of Gluing Data

Gluing domains



Building a Set of Gluing Data

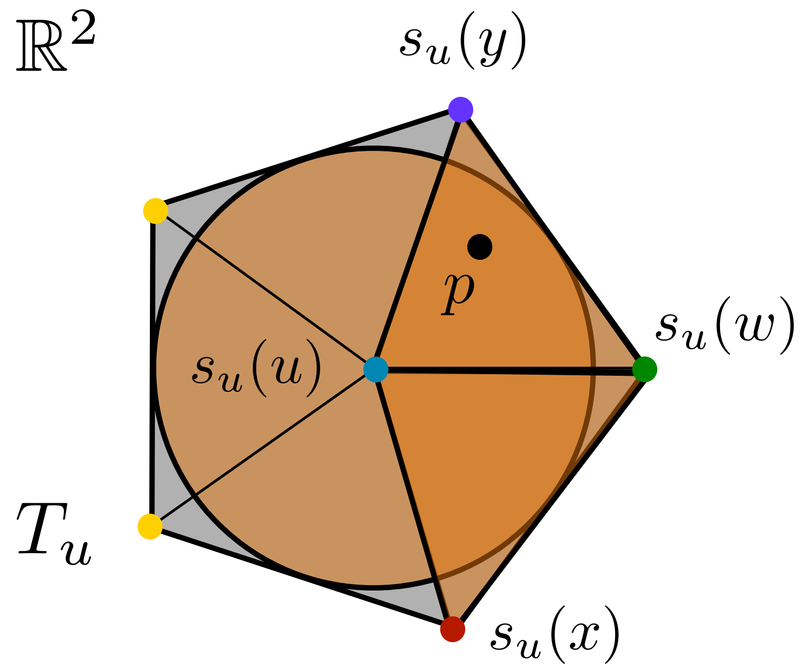
Gluing domains



Building a Set of Gluing Data

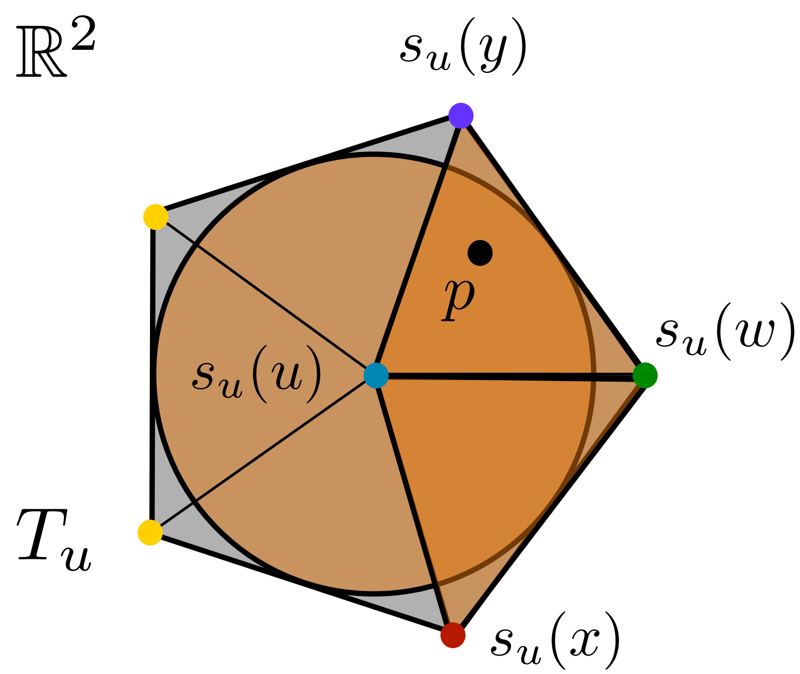
Building a Set of Gluing Data

Let p be a point in the region $C_u \cap [s_u(u), s_u(x), s_u(w), s_u(y)]$.



Building a Set of Gluing Data

Let p be a point in the region $C_u \cap [s_u(u), s_u(x), s_u(w), s_u(y)]$.



Let (θ, r) be the polar coordinates of point p with respect to the local coordinate system of P_u (i.e., origin at $s_u(u) = (0,0)$).

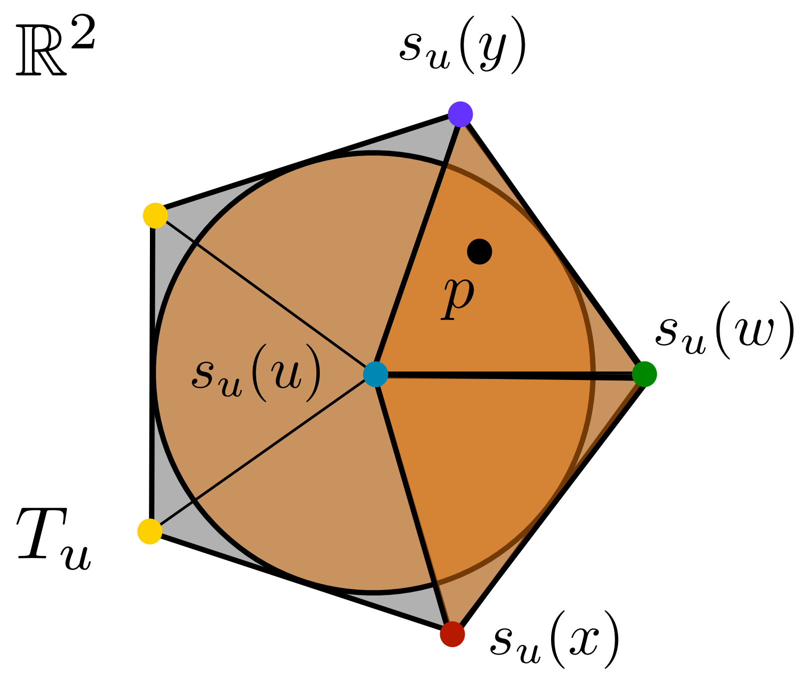
Building a Set of Gluing Data

Building a Set of Gluing Data

Let $g_u : [0, 2\pi) \times \mathbb{R}_+ \to [0, 2\pi) \times \mathbb{R}_+$ be the map

$$g_u(p) = g_u((\theta, r)) = \left(\frac{6}{m_u} \cdot \theta \, , \, \frac{\cos(\frac{\pi}{6})}{\cos(\frac{\pi}{m_u})} \cdot r \right) ,$$

where m_u is the degree of u.

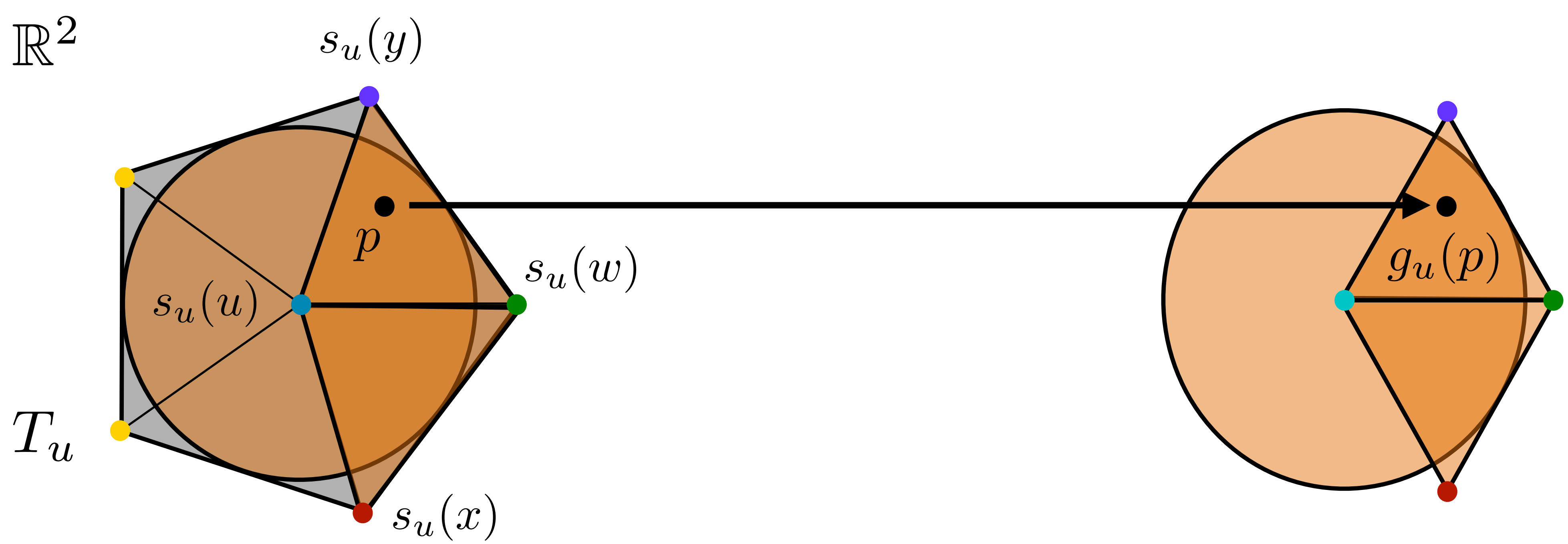


Building a Set of Gluing Data

Let $g_u : [0, 2\pi) \times \mathbb{R}_+ \to [0, 2\pi) \times \mathbb{R}_+$ be the map

$$g_u(p) = g_u((\theta, r)) = \left(\frac{6}{m_u} \cdot \theta \, , \, \frac{\cos(\frac{\pi}{6})}{\cos(\frac{\pi}{m_u})} \cdot r \right) ,$$

where m_u is the degree of u.



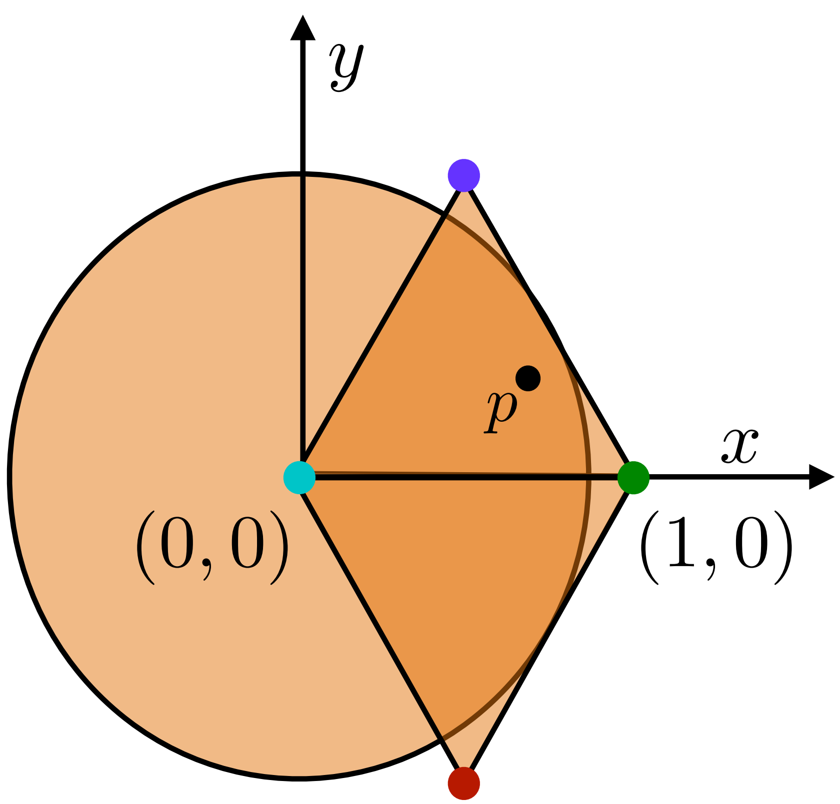
Building a Set of Gluing Data

Building a Set of Gluing Data

Let $h : \mathbb{R}^2 \to \mathbb{R}^2$ be the map $h(p) = h((x, y)) = (1 - x, -y)$:

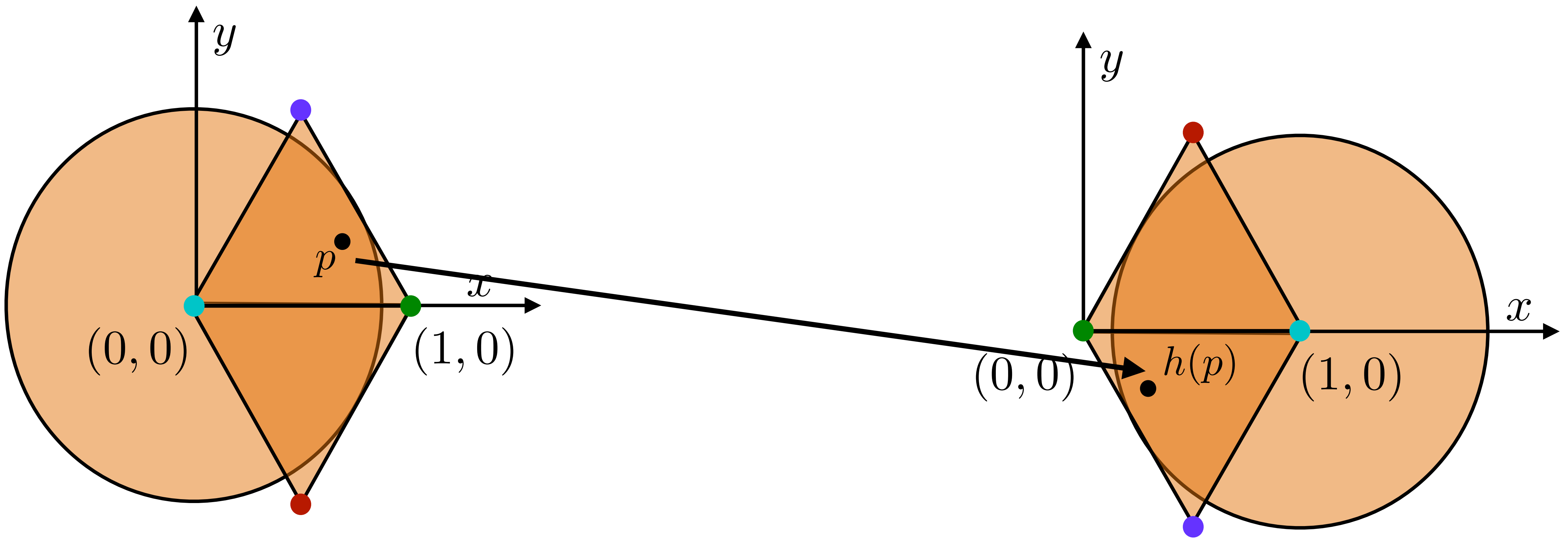
Building a Set of Gluing Data

Let $h : \mathbb{R}^2 \to \mathbb{R}^2$ be the map $h(p) = h((x, y)) = (1 - x, -y)$:



Building a Set of Gluing Data

Let $h : \mathbb{R}^2 \to \mathbb{R}^2$ be the map $h(p) = h((x, y)) = (1 - x, -y)$:



Building a Set of Gluing Data

Building a Set of Gluing Data

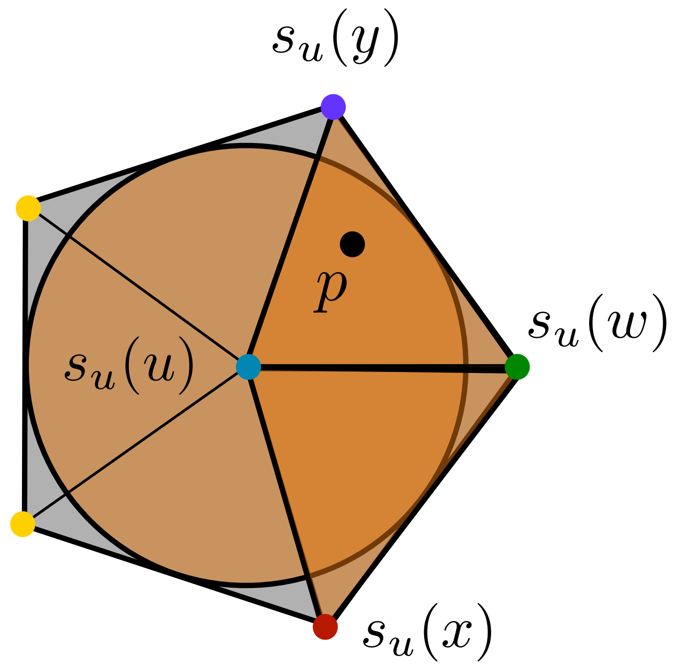
Finally, we define $g_{(u,w)} : [0, 2\pi) \times \mathbb{R}_+ \to [0, 2\pi) \times \mathbb{R}_+$ as

$$g_{(u,w)}(p) = g_{(u,w)}((\theta, r)) = g_w^{-1} \circ h \circ g_u((\theta, r)) \, .$$

Building a Set of Gluing Data

Finally, we define $g_{(u,w)} : [0, 2\pi) \times \mathbb{R}_+ \to [0, 2\pi) \times \mathbb{R}_+$ as

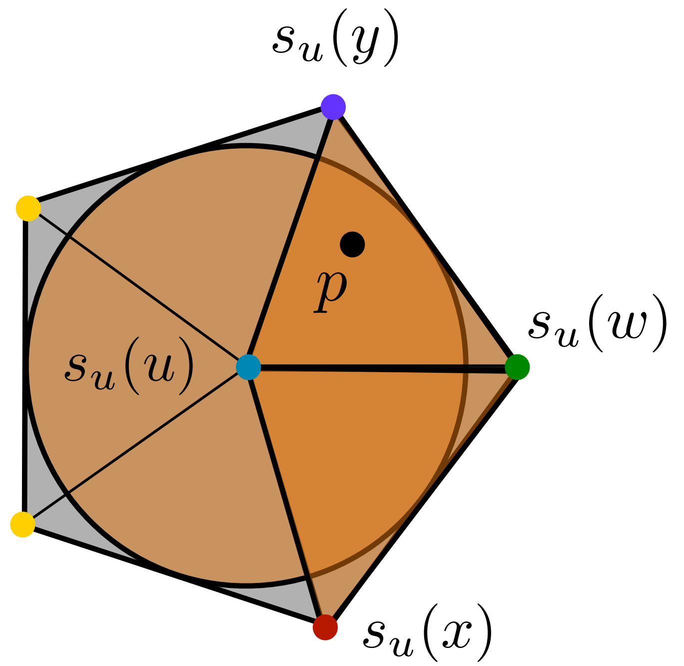
$$g_{(u,w)}(p) = g_{(u,w)}((\theta, r)) = g_w^{-1} \circ h \circ g_u((\theta, r)) \, .$$

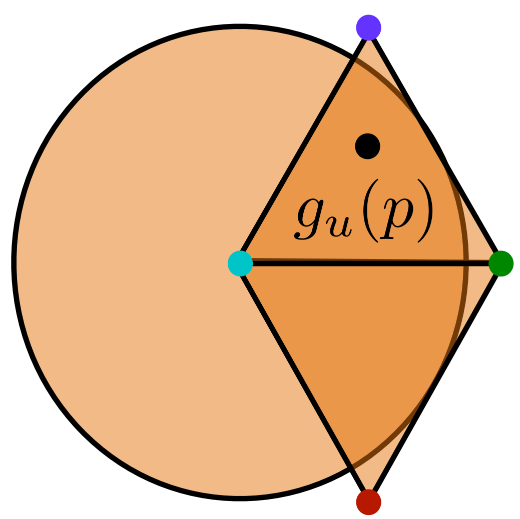


Building a Set of Gluing Data

Finally, we define $g_{(u,w)} : [0, 2\pi) \times \mathbb{R}_+ \to [0, 2\pi) \times \mathbb{R}_+$ as

$$g_{(u,w)}(p) = g_{(u,w)}((\theta, r)) = g_w^{-1} \circ h \circ g_u((\theta, r))\,.$$

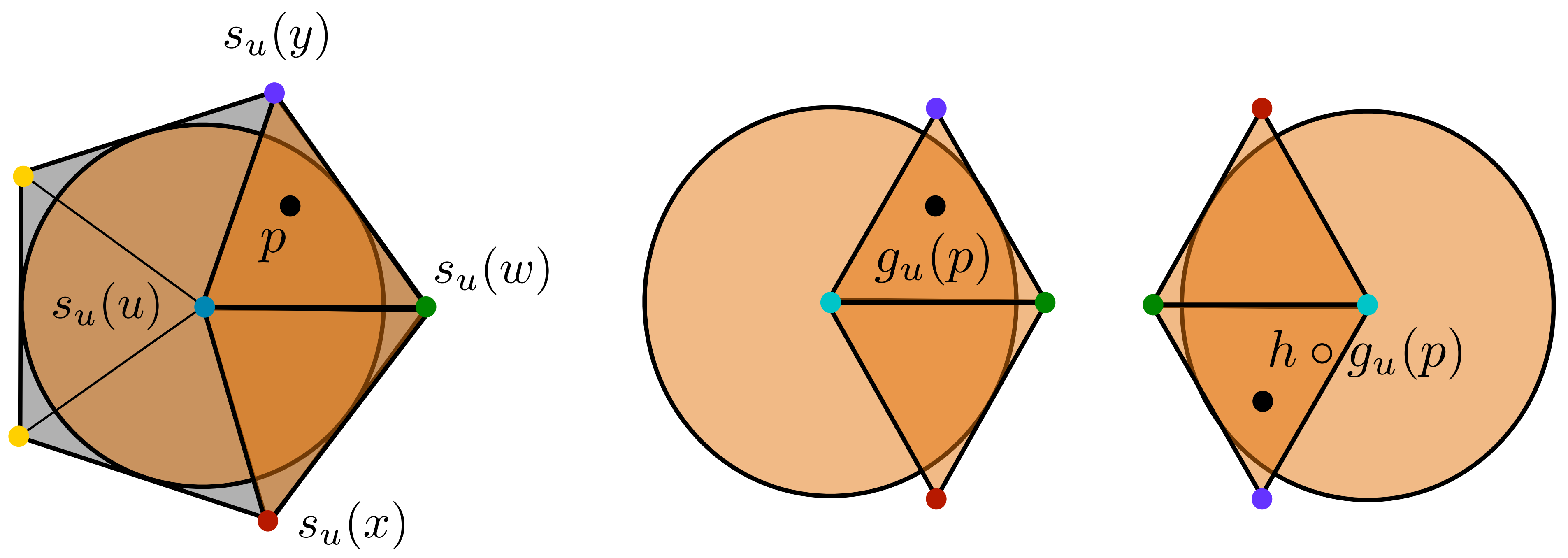




Building a Set of Gluing Data

Finally, we define $g_{(u,w)} : [0, 2\pi) \times \mathbb{R}_+ \to [0, 2\pi) \times \mathbb{R}_+$ as

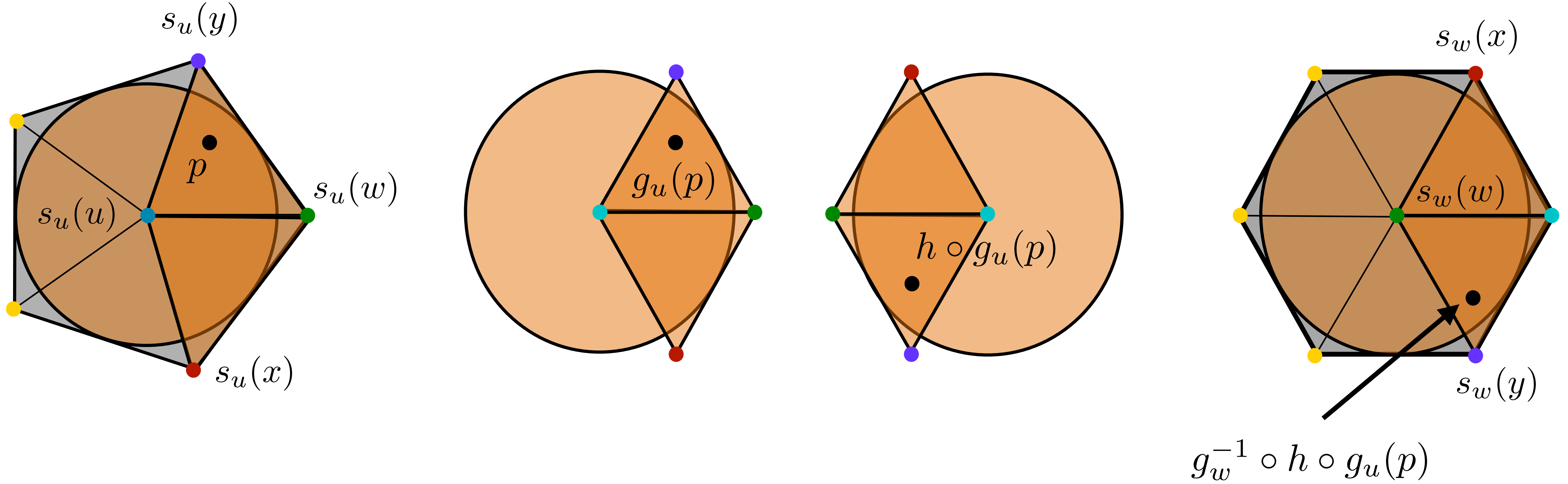
$$g_{(u,w)}(p) = g_{(u,w)}((\theta, r)) = g_w^{-1} \circ h \circ g_u((\theta, r)) \,.$$



Building a Set of Gluing Data

Finally, we define $g_{(u,w)} : [0, 2\pi) \times \mathbb{R}_+ \to [0, 2\pi) \times \mathbb{R}_+$ as

$$g_{(u,w)}(p) = g_{(u,w)}((\theta, r)) = g_w^{-1} \circ h \circ g_u((\theta, r)) \, .$$



Building a Set of Gluing Data

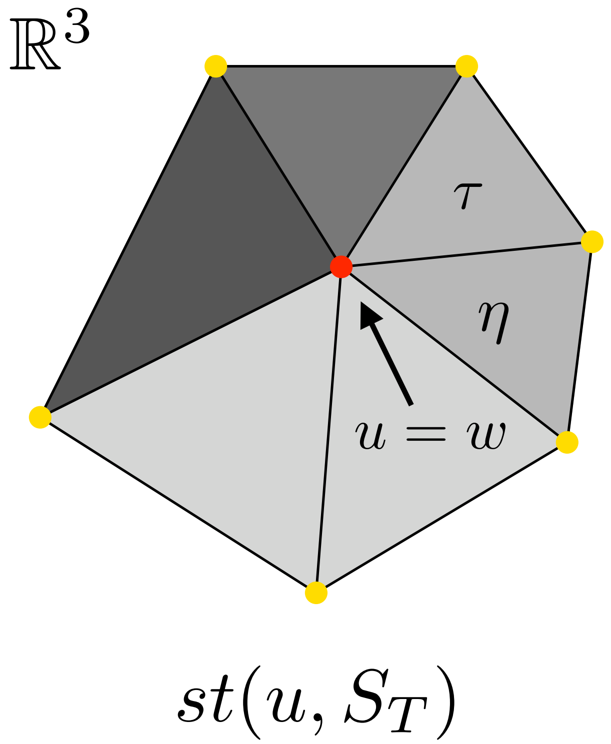
Building a Set of Gluing Data

For any two $(\tau, u), (\eta, w) \in I$, we define $\Omega_{(\tau,u)(\eta,w)}$ as follows:

Building a Set of Gluing Data

For any two $(\tau, u), (\eta, w) \in I$, we define $\Omega_{(\tau,u)(\eta,w)}$ as follows:

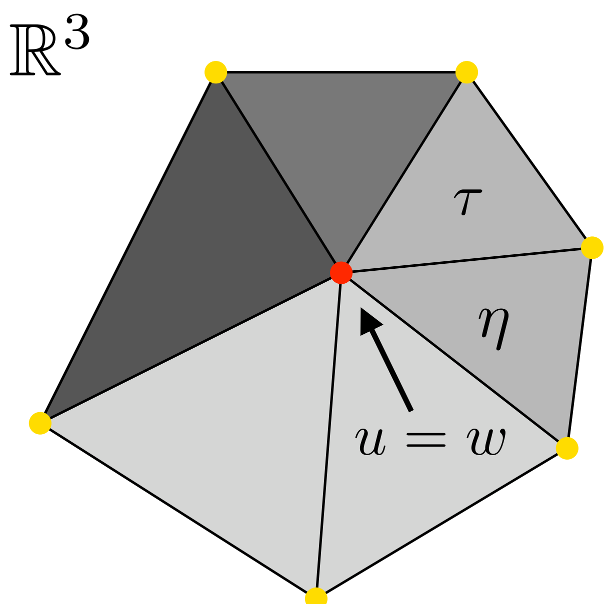
(1) $u = w$



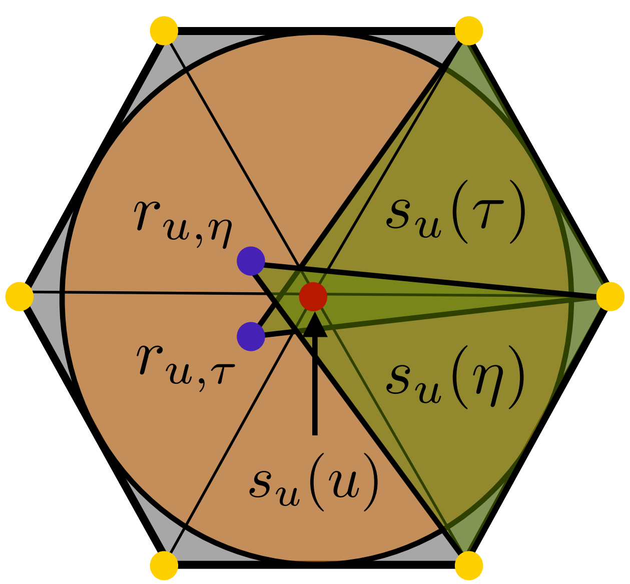
Building a Set of Gluing Data

For any two $(\tau, u), (\eta, w) \in I$, we define $\Omega_{(\tau,u)(\eta,w)}$ as follows:

(1) $u = w$



$st(u, S_T)$

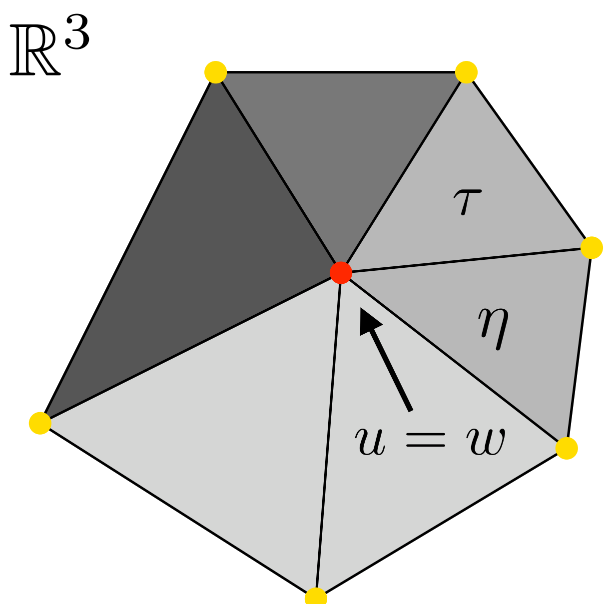


T_u

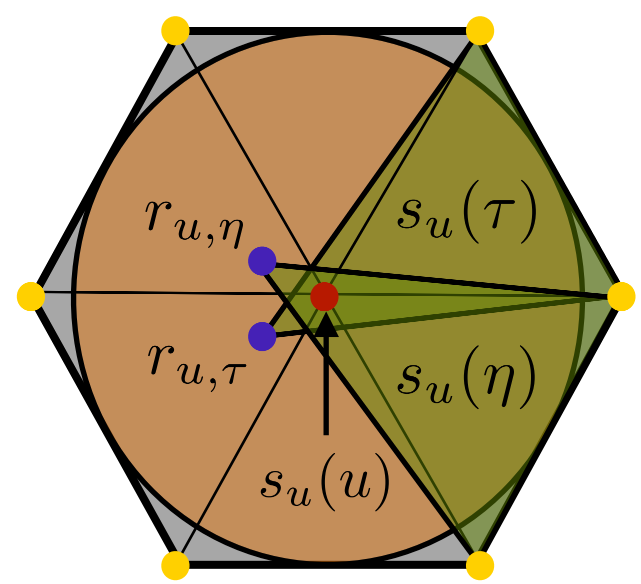
Building a Set of Gluing Data

For any two $(\tau, u), (\eta, w) \in I$, we define $\Omega_{(\tau,u)(\eta,w)}$ as follows:

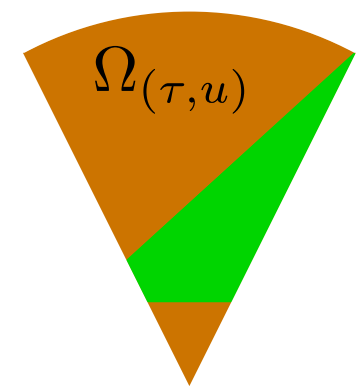
(1) $u = w$



$st(u, S_T)$



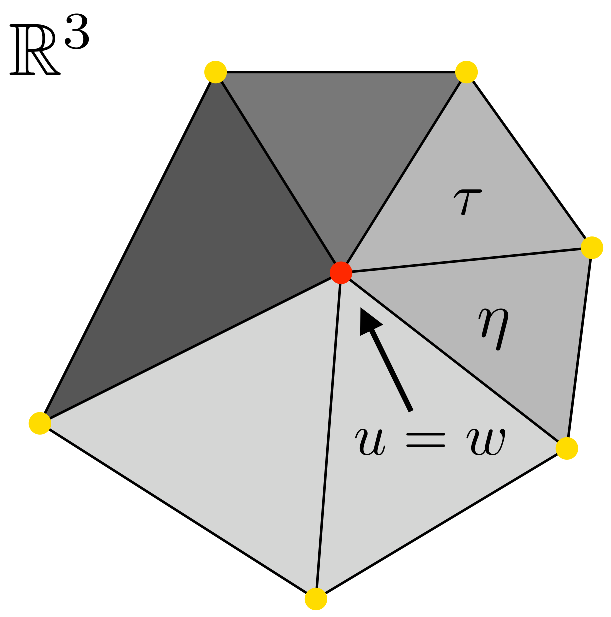
T_u



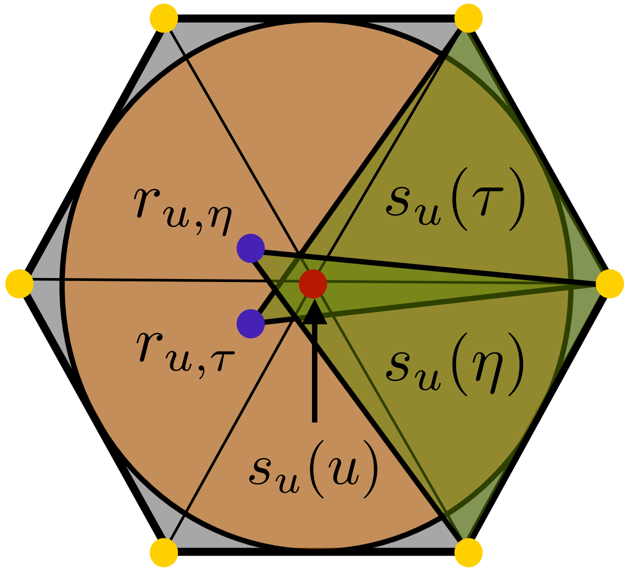
Building a Set of Gluing Data

For any two $(\tau, u), (\eta, w) \in I$, we define $\Omega_{(\tau,u)(\eta,w)}$ as follows:

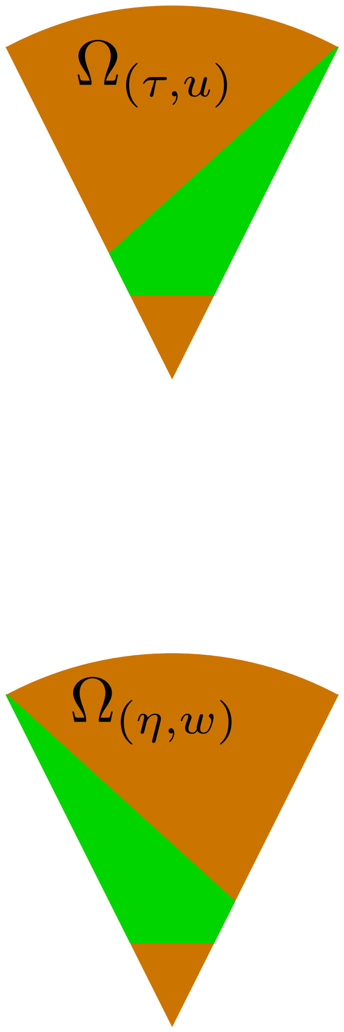
(1) $u = w$



$st(u, S_T)$



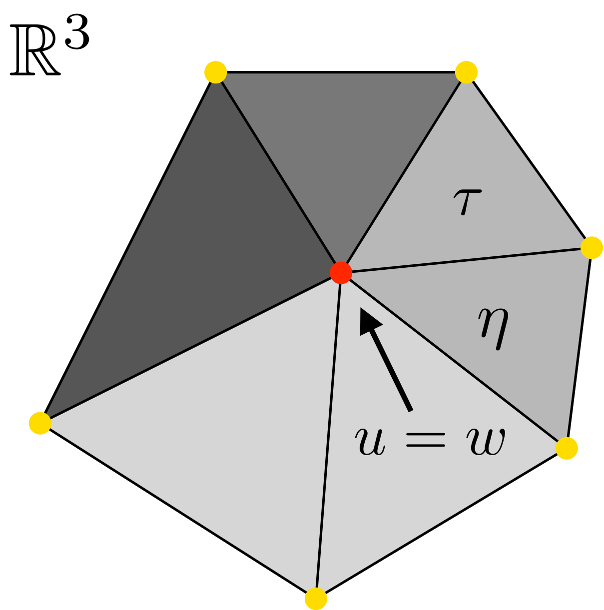
T_u



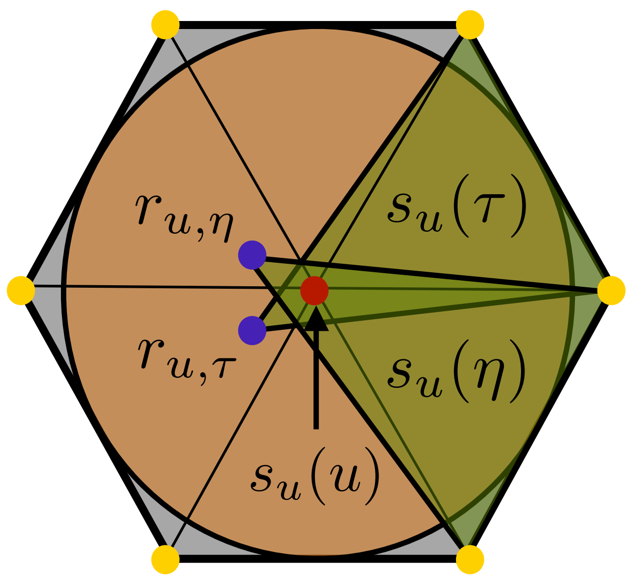
Building a Set of Gluing Data

For any two $(\tau, u), (\eta, w) \in I$, we define $\Omega_{(\tau,u)(\eta,w)}$ as follows:

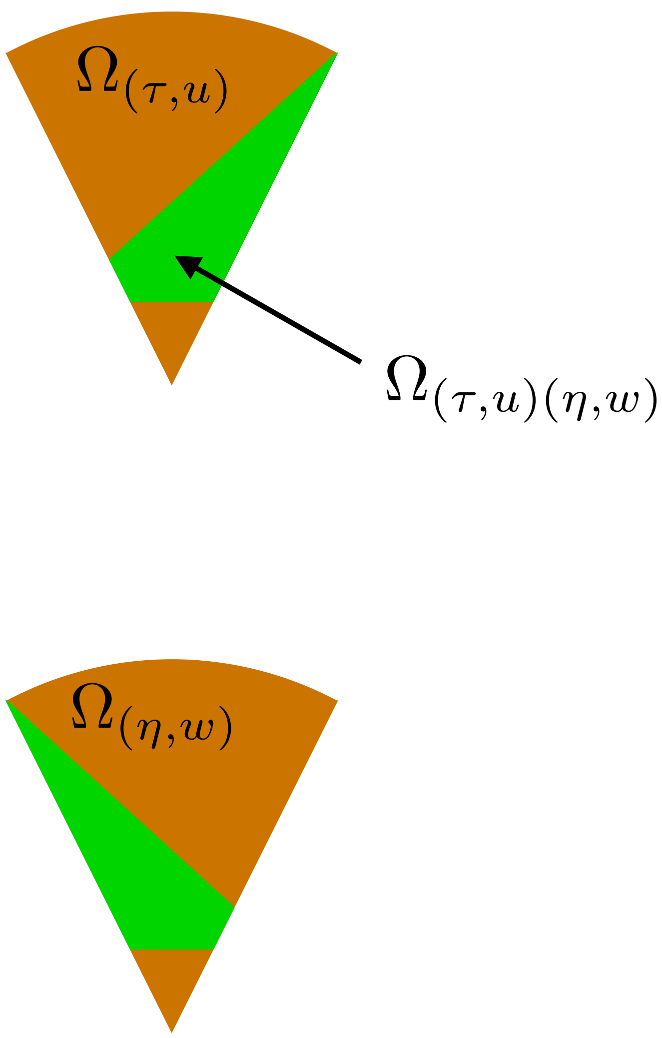
(1) $u = w$



$st(u, S_T)$



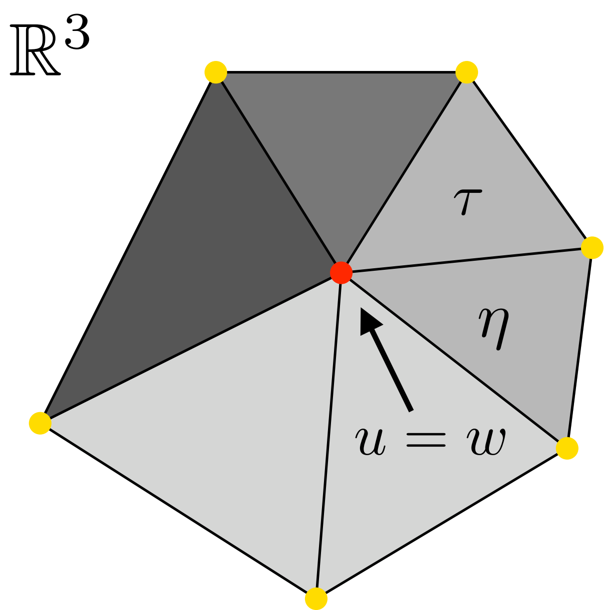
T_u



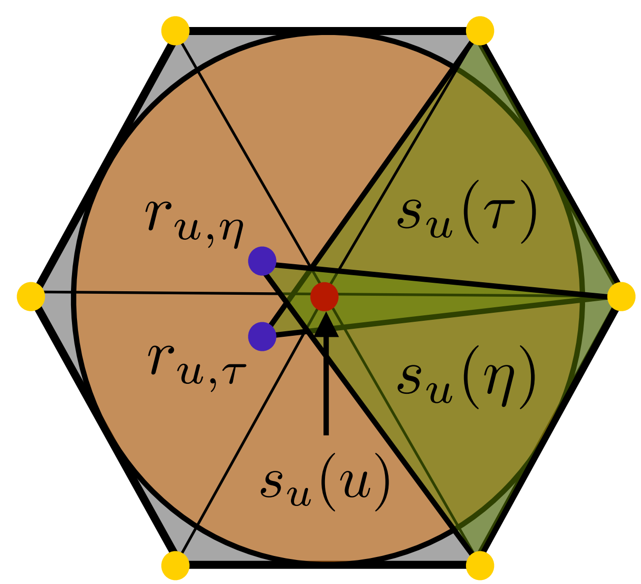
Building a Set of Gluing Data

For any two $(\tau, u), (\eta, w) \in I$, we define $\Omega_{(\tau,u)(\eta,w)}$ as follows:

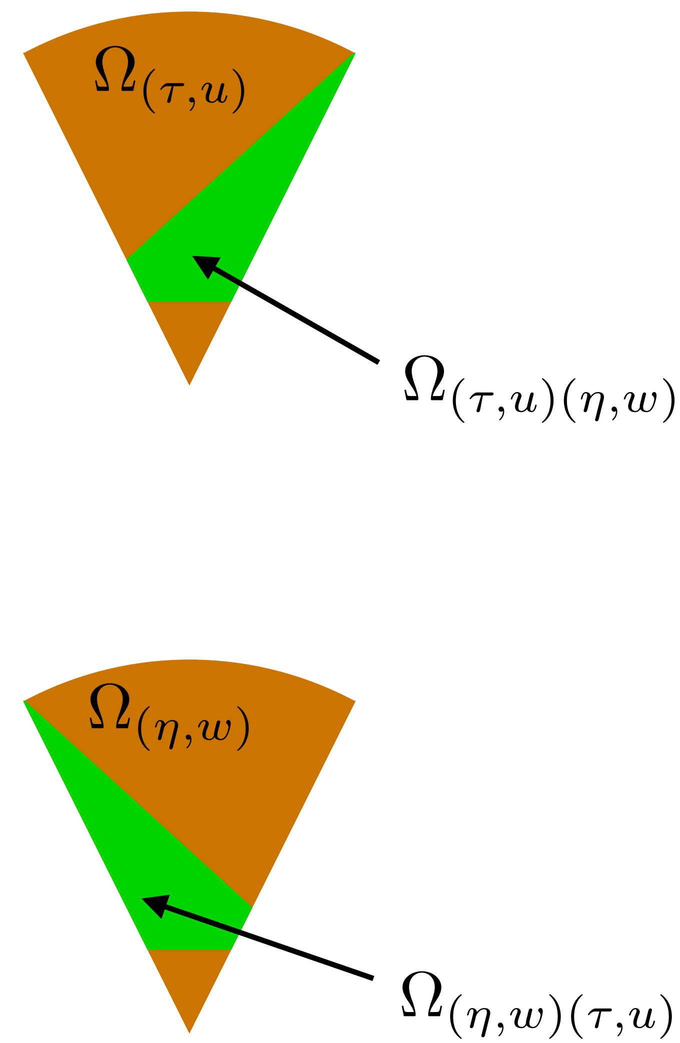
(1) $u = w$



$st(u, S_T)$



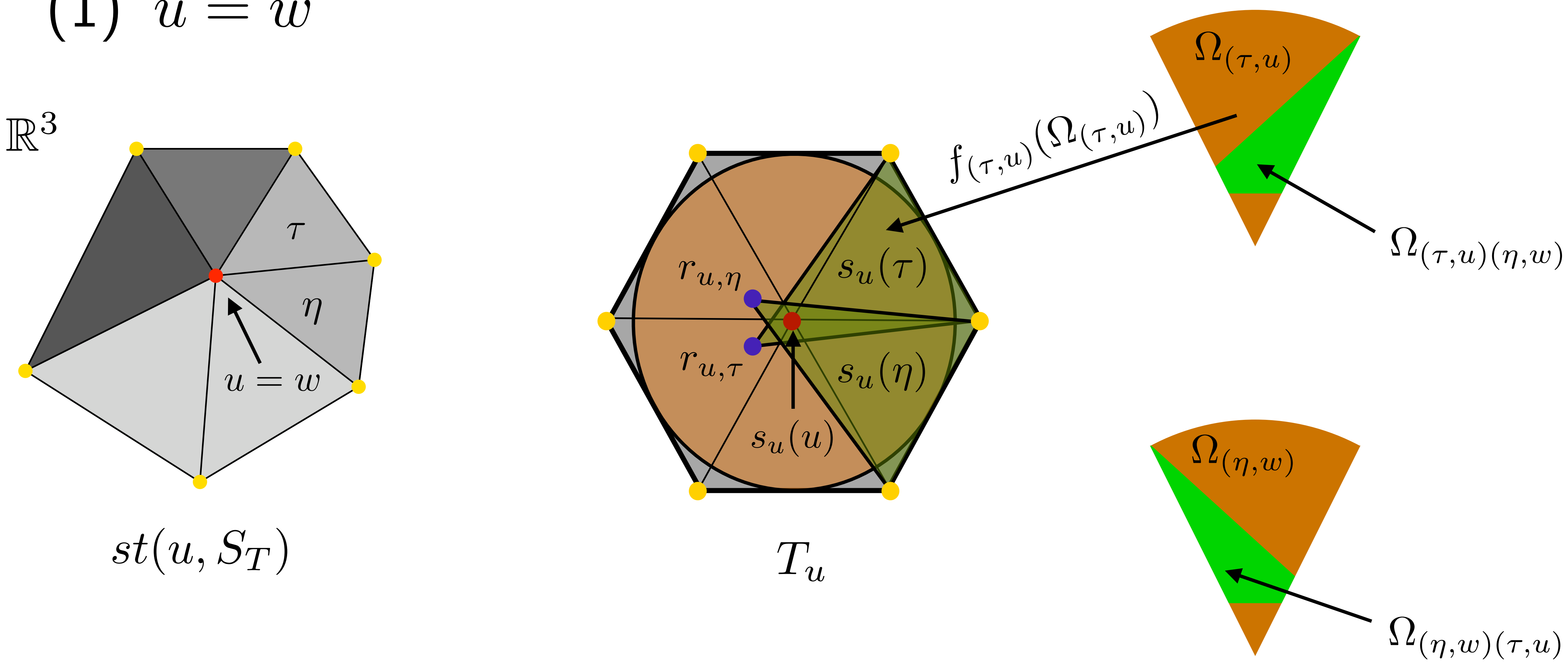
T_u



Building a Set of Gluing Data

For any two $(\tau, u), (\eta, w) \in I$, we define $\Omega_{(\tau,u)(\eta,w)}$ as follows:

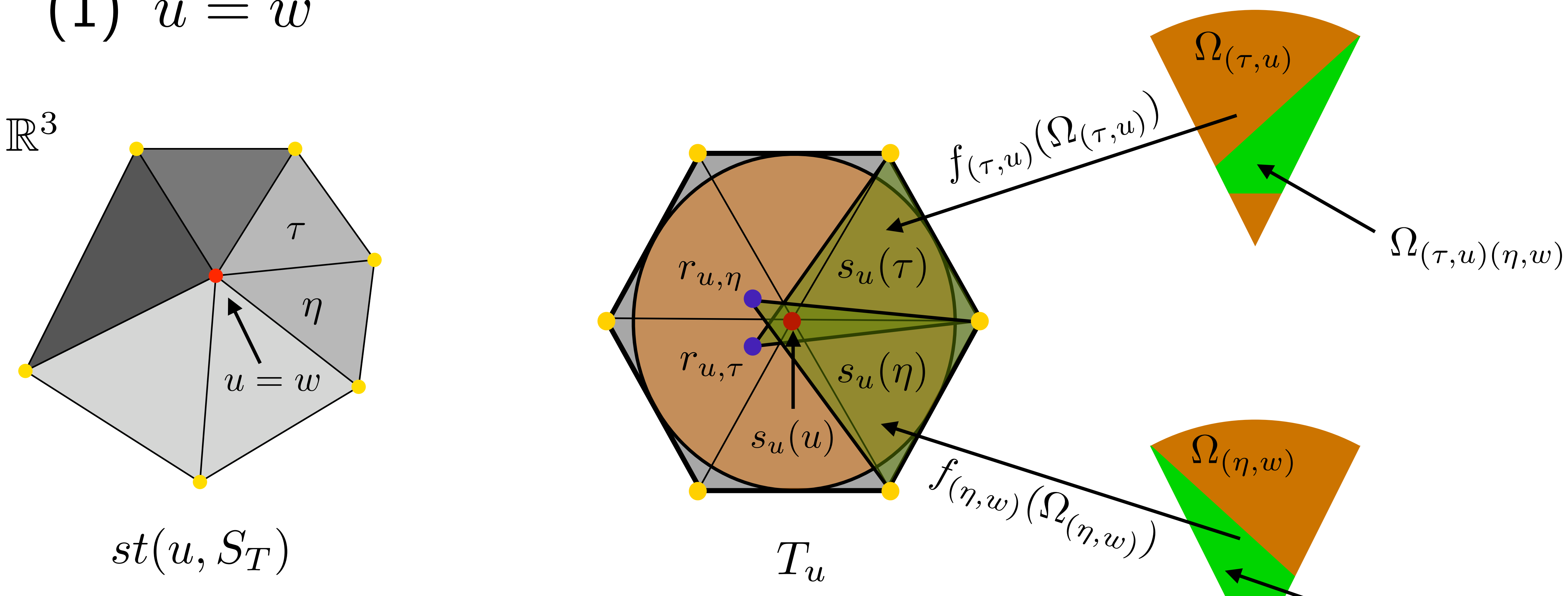
(1) $u = w$



Building a Set of Gluing Data

For any two $(\tau, u), (\eta, w) \in I$, we define $\Omega_{(\tau,u)(\eta,w)}$ as follows:

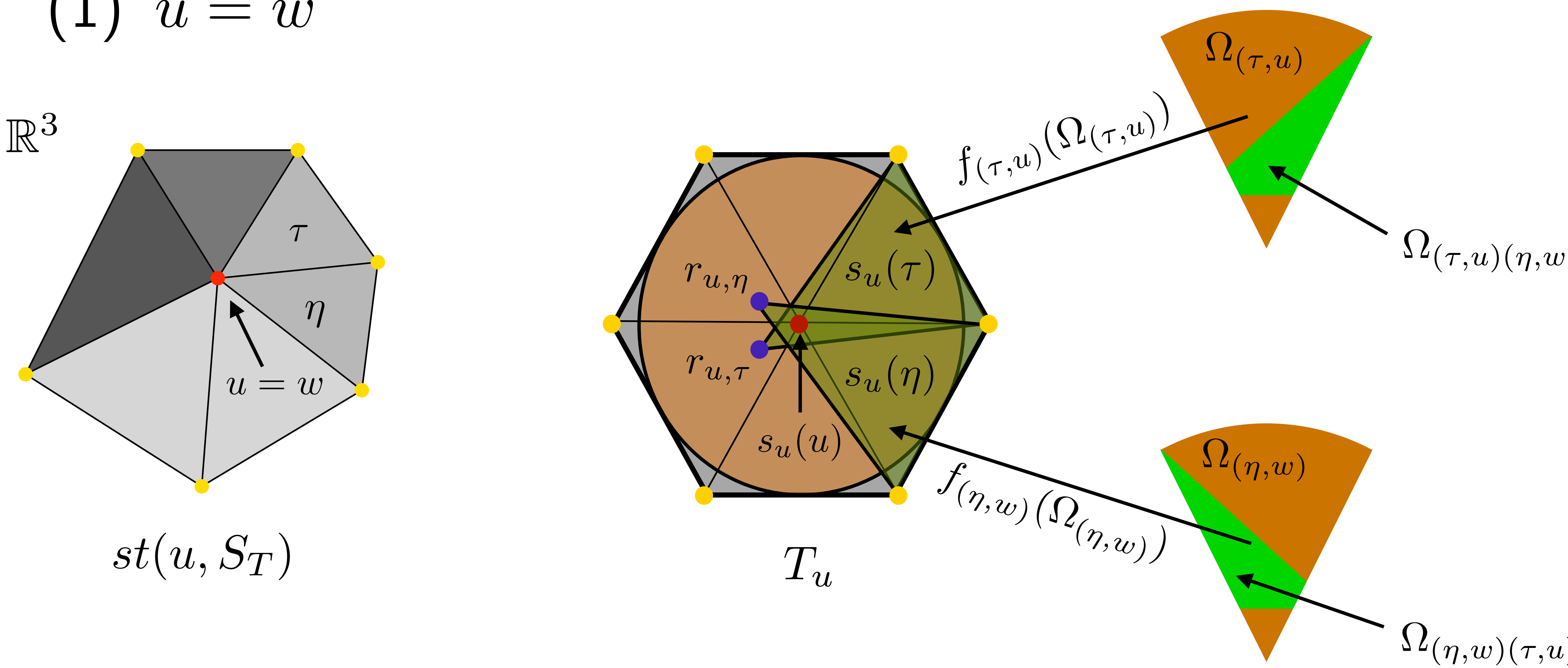
(1) $u = w$



Building a Set of Gluing Data

For any two $(\tau, u), (\eta, w) \in I$, we define $\Omega_{(\tau,u)(\eta,w)}$ as follows:

(1) $u = w$



$$\Omega_{(\tau,u)(\eta,w)} = f^{-1}_{(\tau,u)}(f_{(\tau,u)}(\Omega_{\tau,u}) \cap f_{(\eta,w)}(\Omega_{\eta,w}))$$

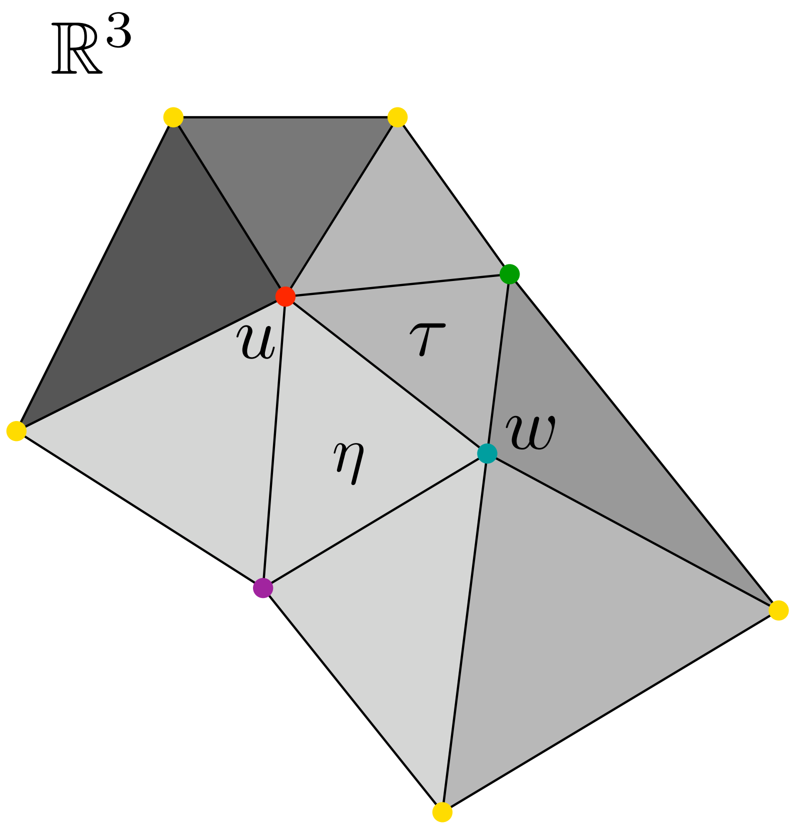
Building a Set of Gluing Data

Building a Set of Gluing Data

(2) $u \neq w$ and w is a vertex of τ or u is a vertex of η

Building a Set of Gluing Data

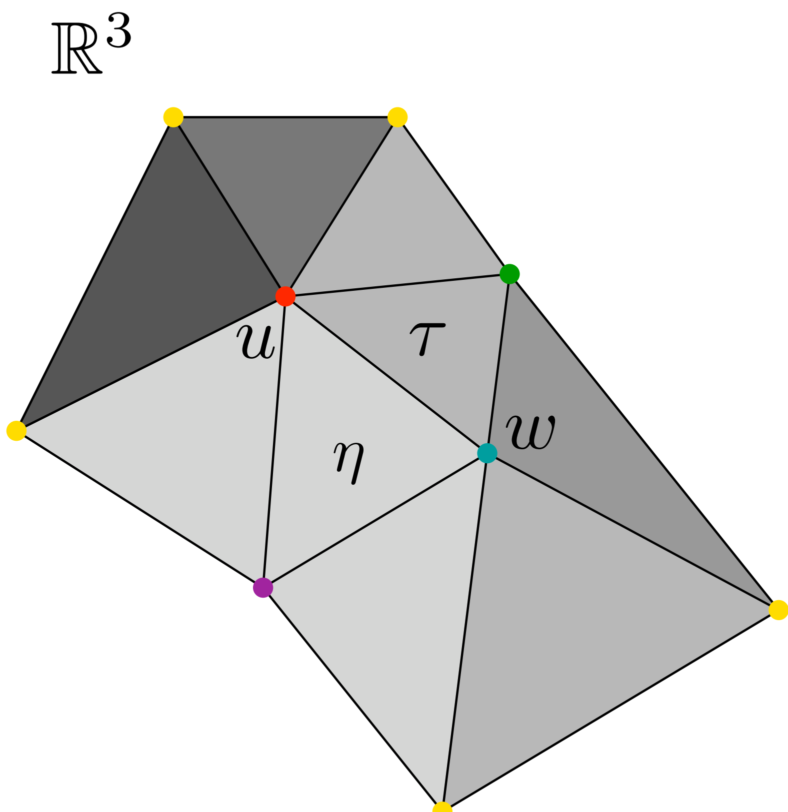
(2) $u \neq w$ and w is a vertex of τ or u is a vertex of η



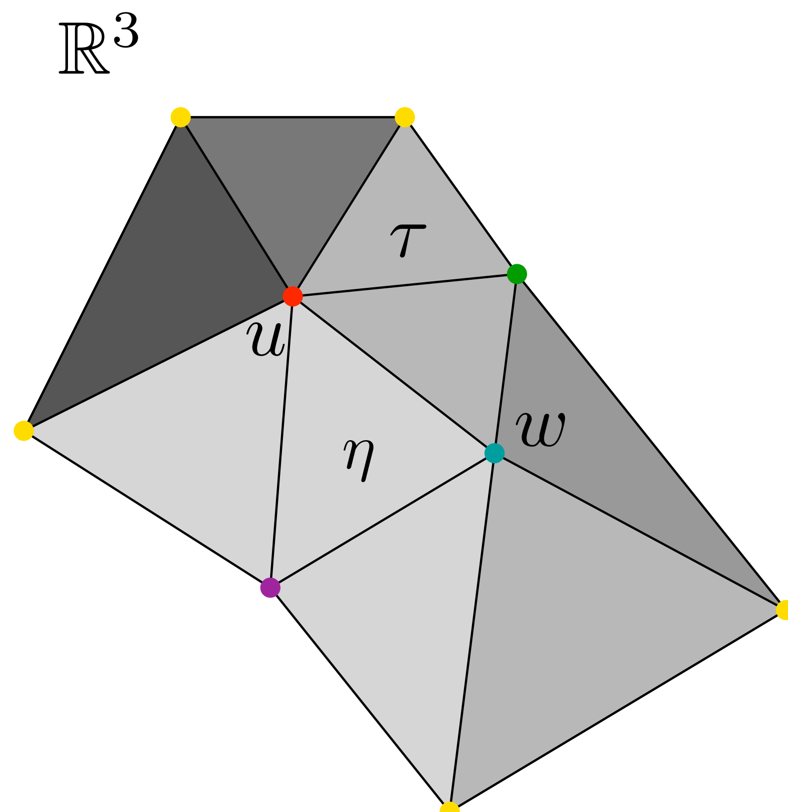
$st(u, S_T) \cup st(w, S_T)$

Building a Set of Gluing Data

(2) $u \neq w$ and w is a vertex of τ or u is a vertex of η



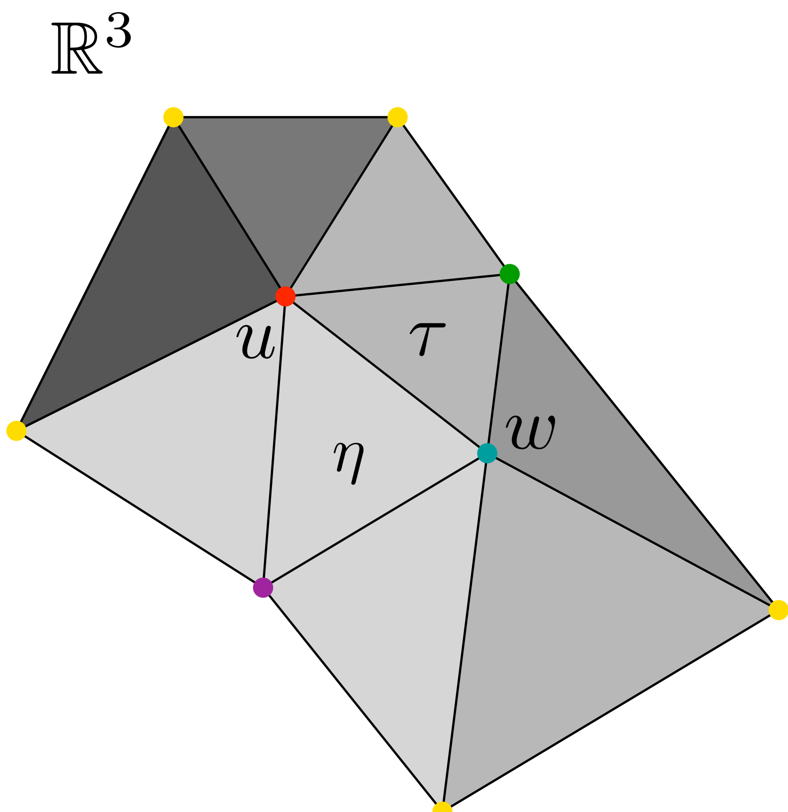
$st(u, S_T) \cup st(w, S_T)$



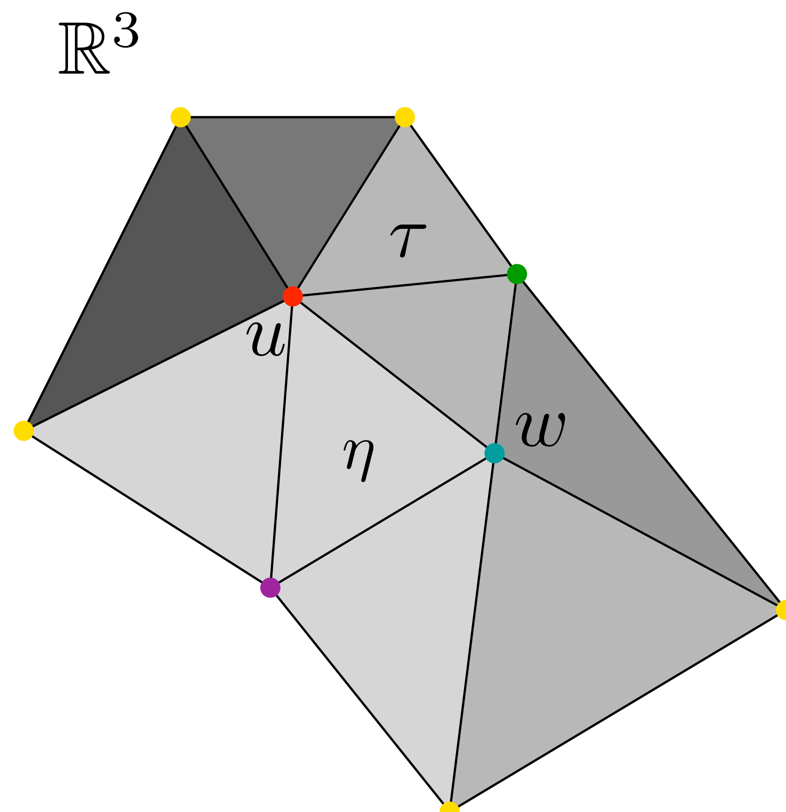
$st(u, S_T) \cup st(w, S_T)$

Building a Set of Gluing Data

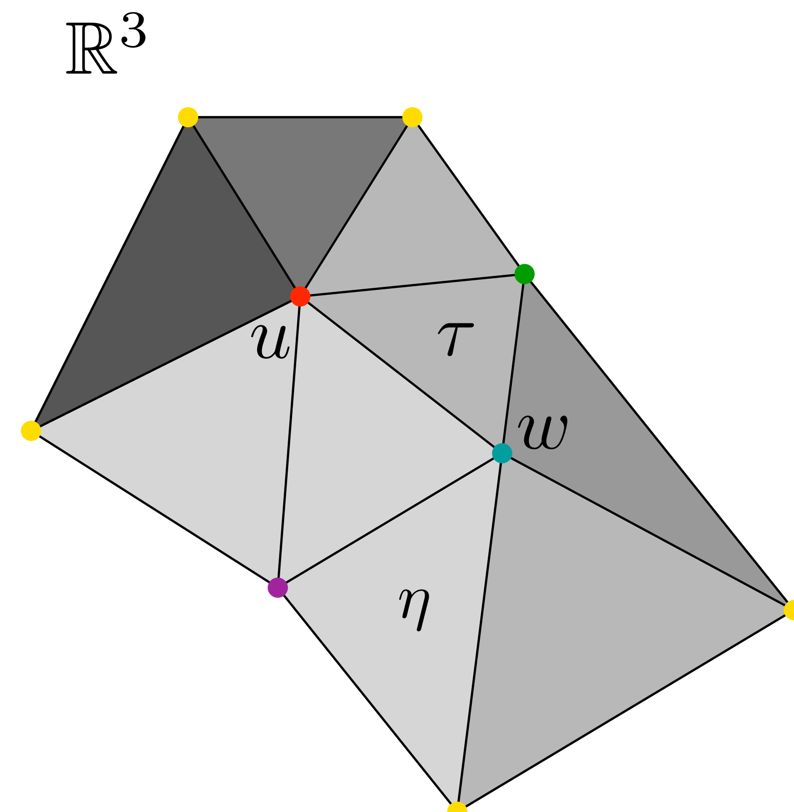
(2) $u \neq w$ and w is a vertex of τ or u is a vertex of η



$st(u, S_T) \cup st(w, S_T)$



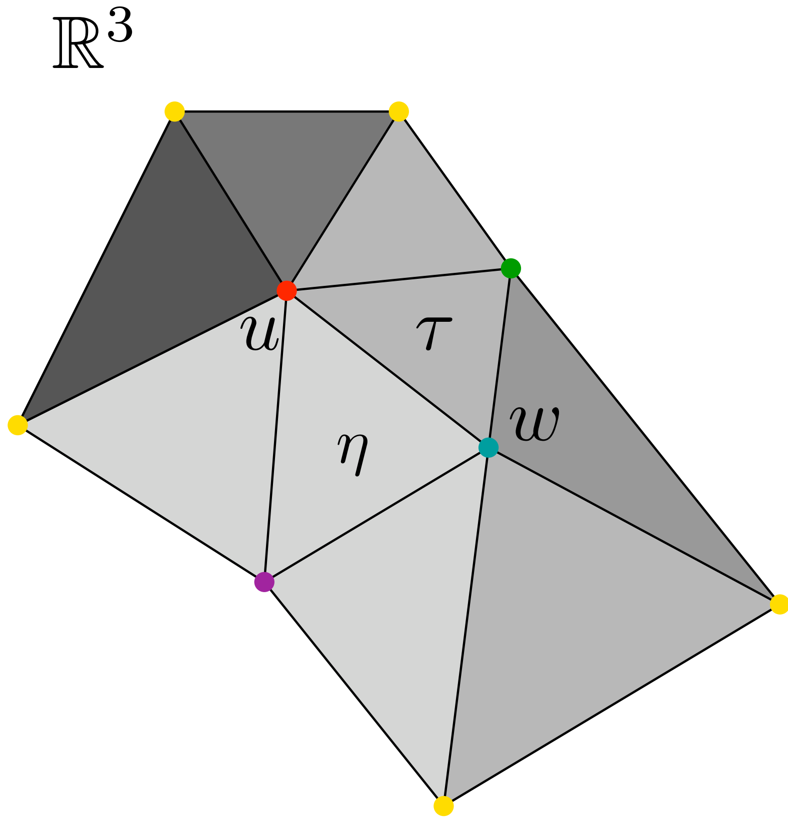
$st(u, S_T) \cup st(w, S_T)$



$st(u, S_T) \cup st(w, S_T)$

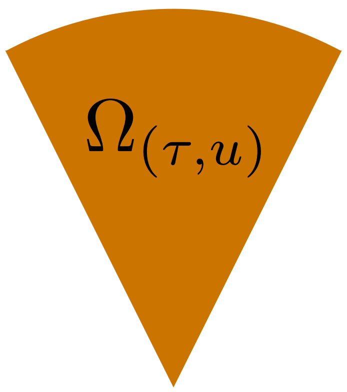
Building a Set of Gluing Data

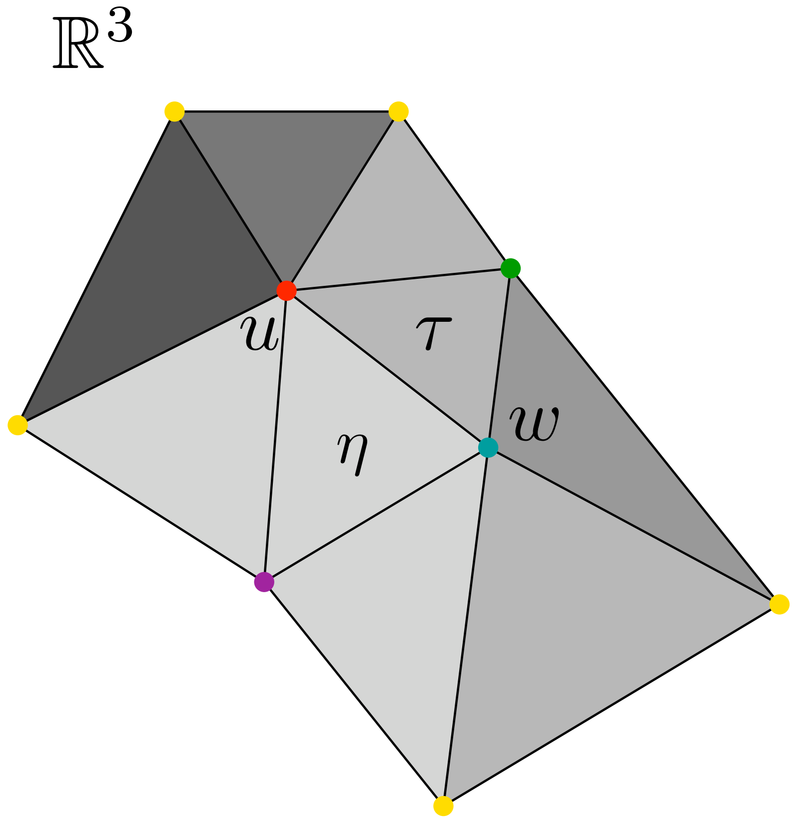
Building a Set of Gluing Data



$$st(u, S_T) \cup st(w, S_T)$$

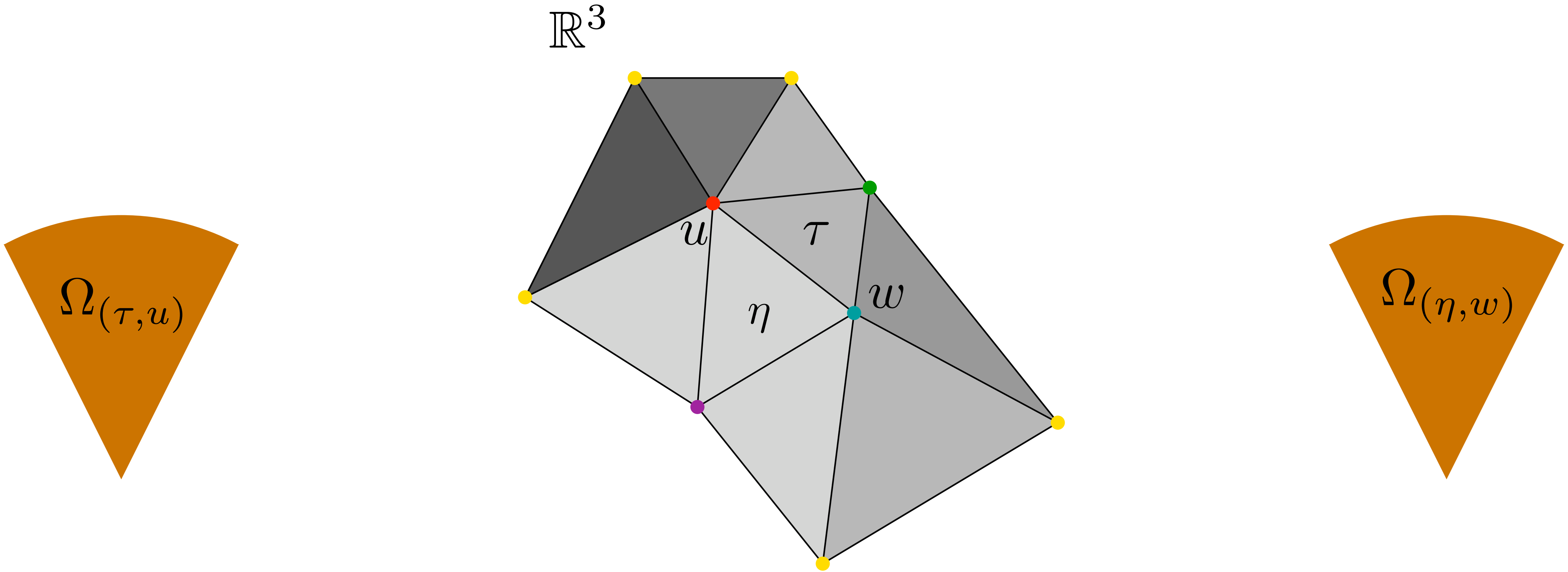
Building a Set of Gluing Data





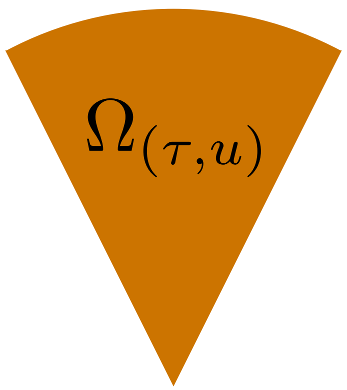
$$st(u, S_T) \cup st(w, S_T)$$

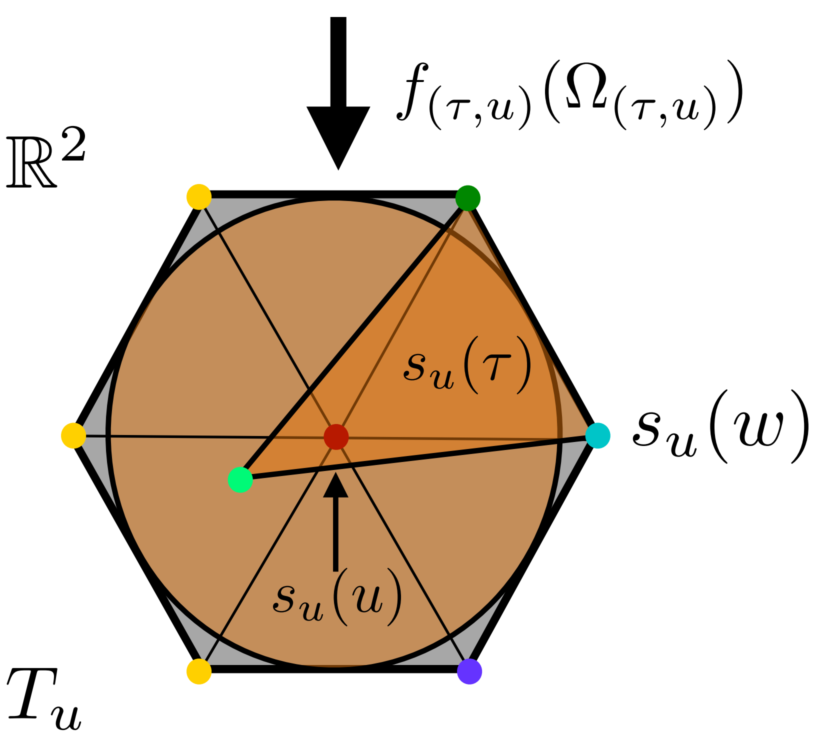
Building a Set of Gluing Data

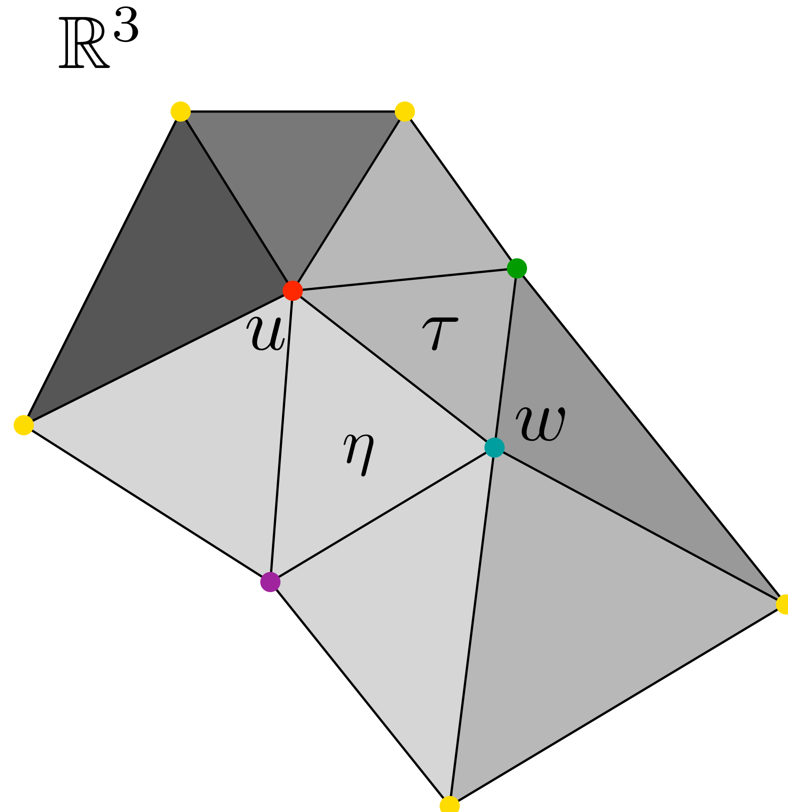


$$st(u, S_T) \cup st(w, S_T)$$

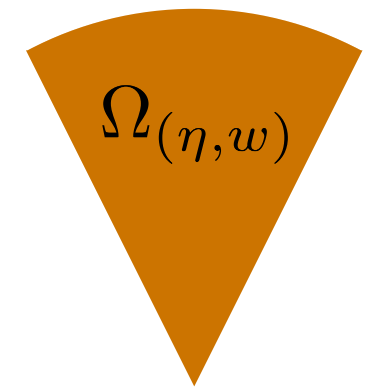
Building a Set of Gluing Data



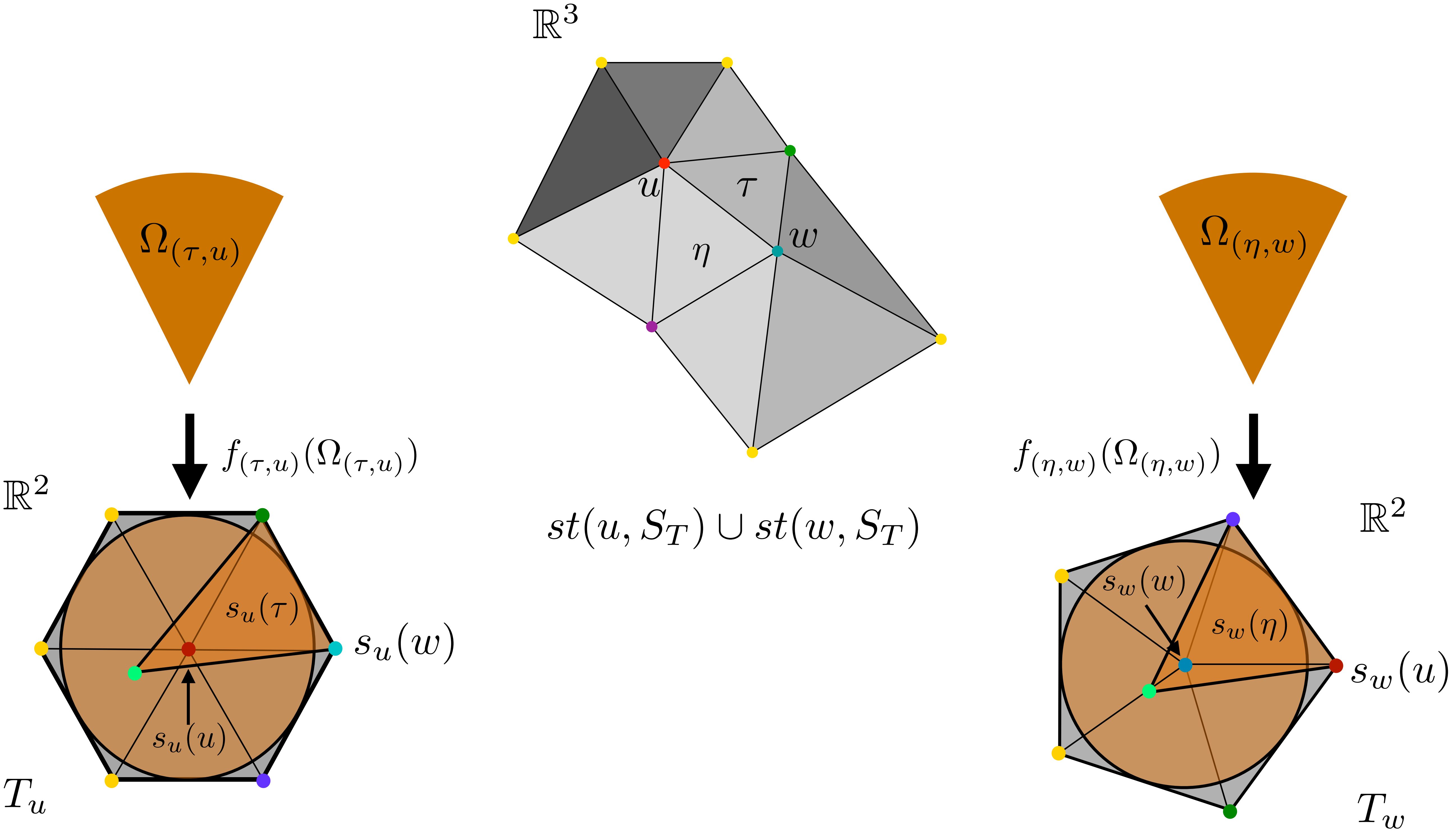




$st(u, S_T) \cup st(w, S_T)$

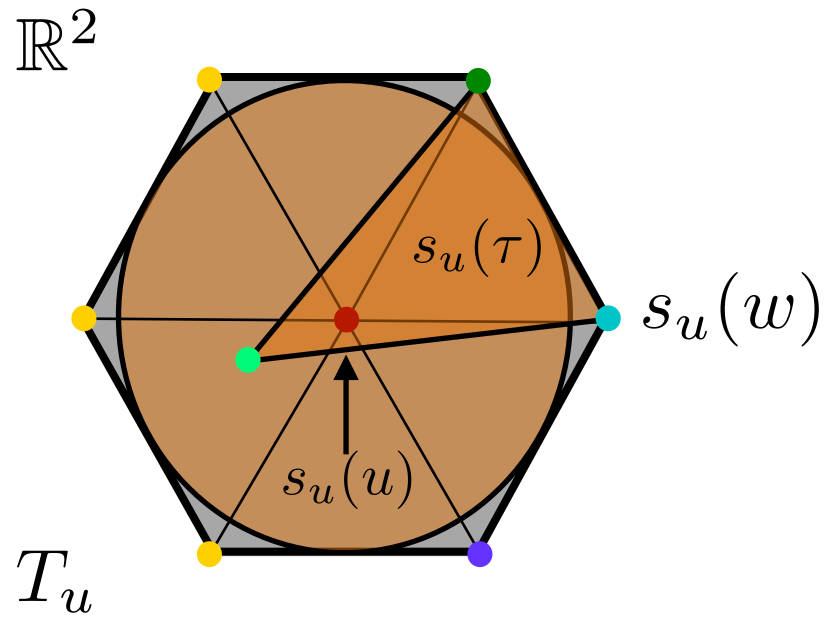


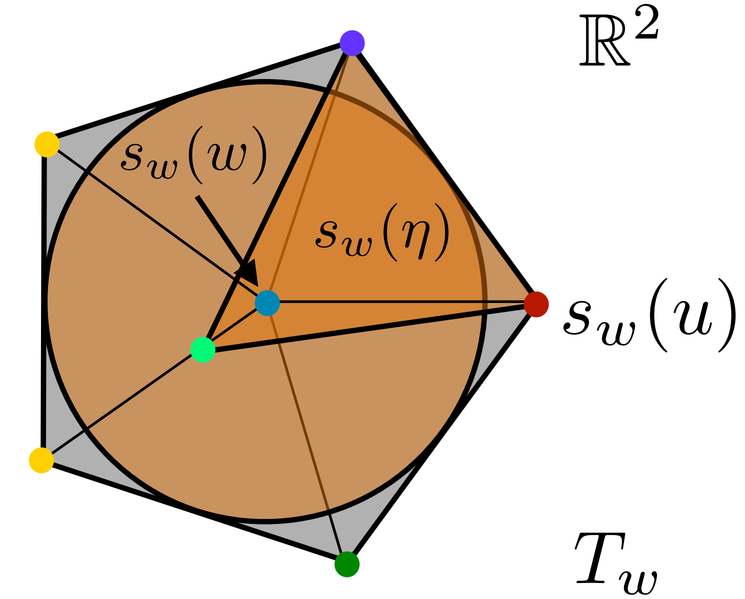
Building a Set of Gluing Data



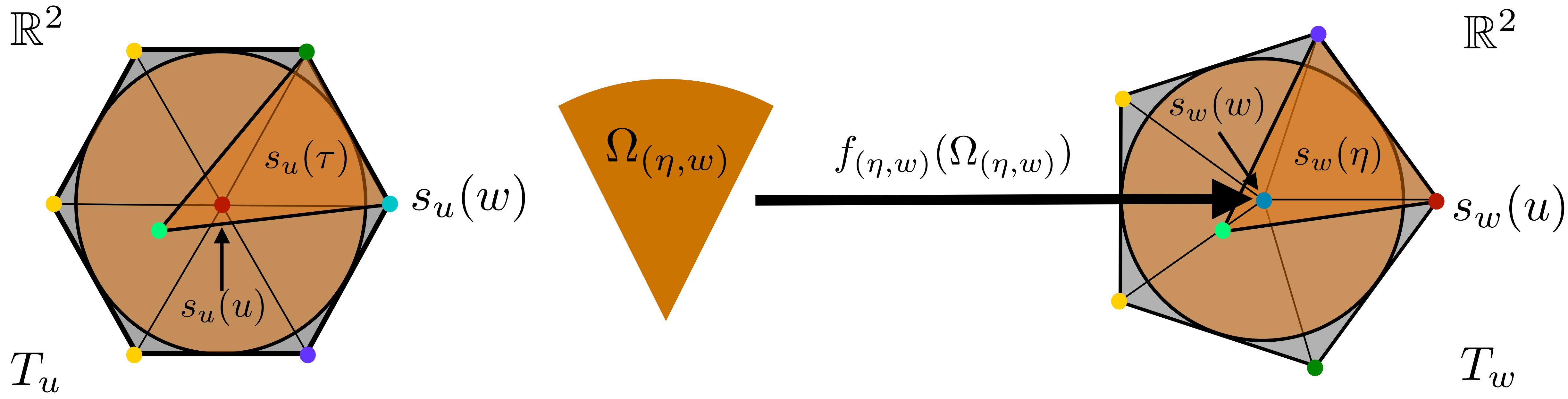
Building a Set of Gluing Data

Building a Set of Gluing Data

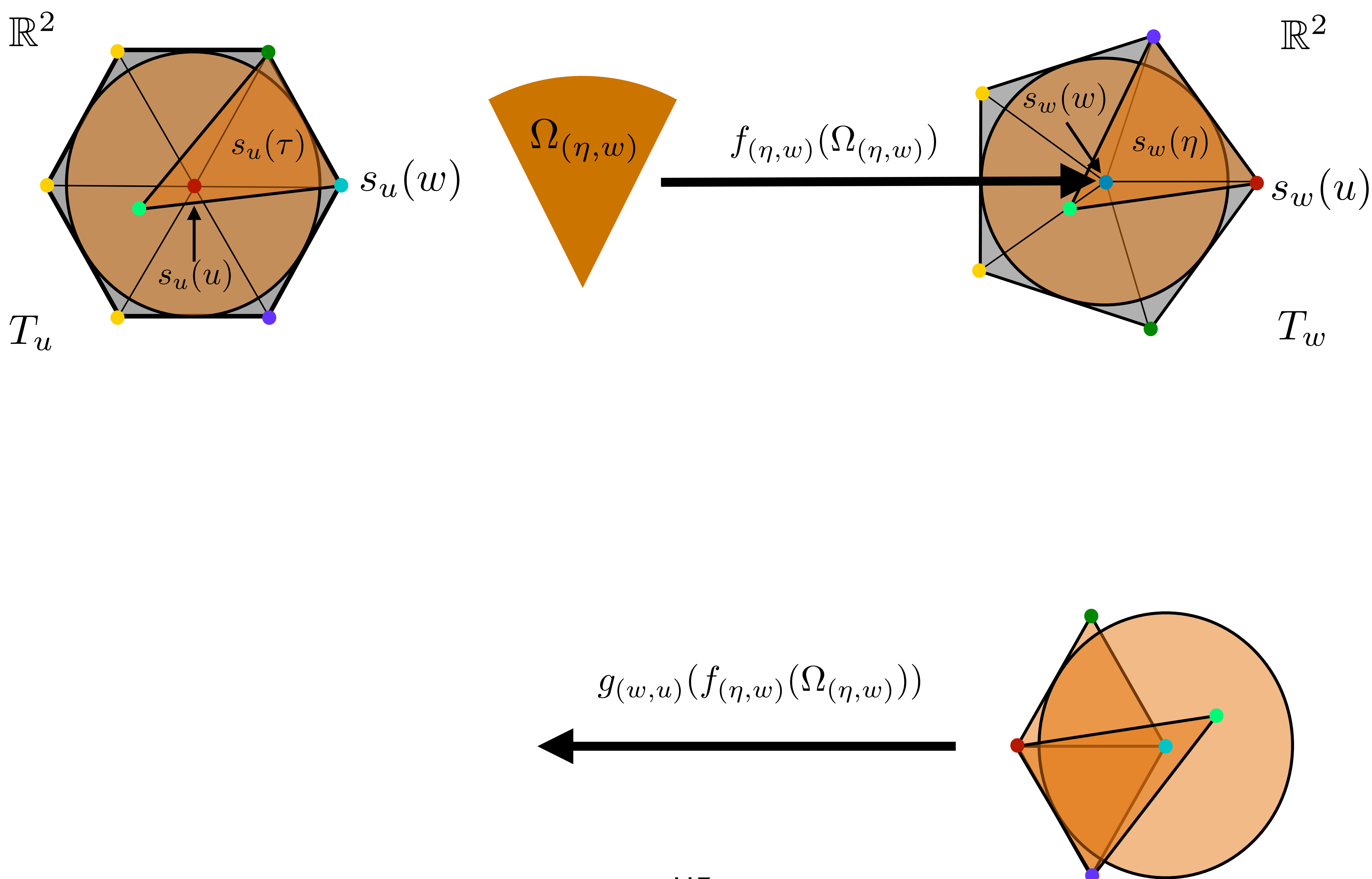




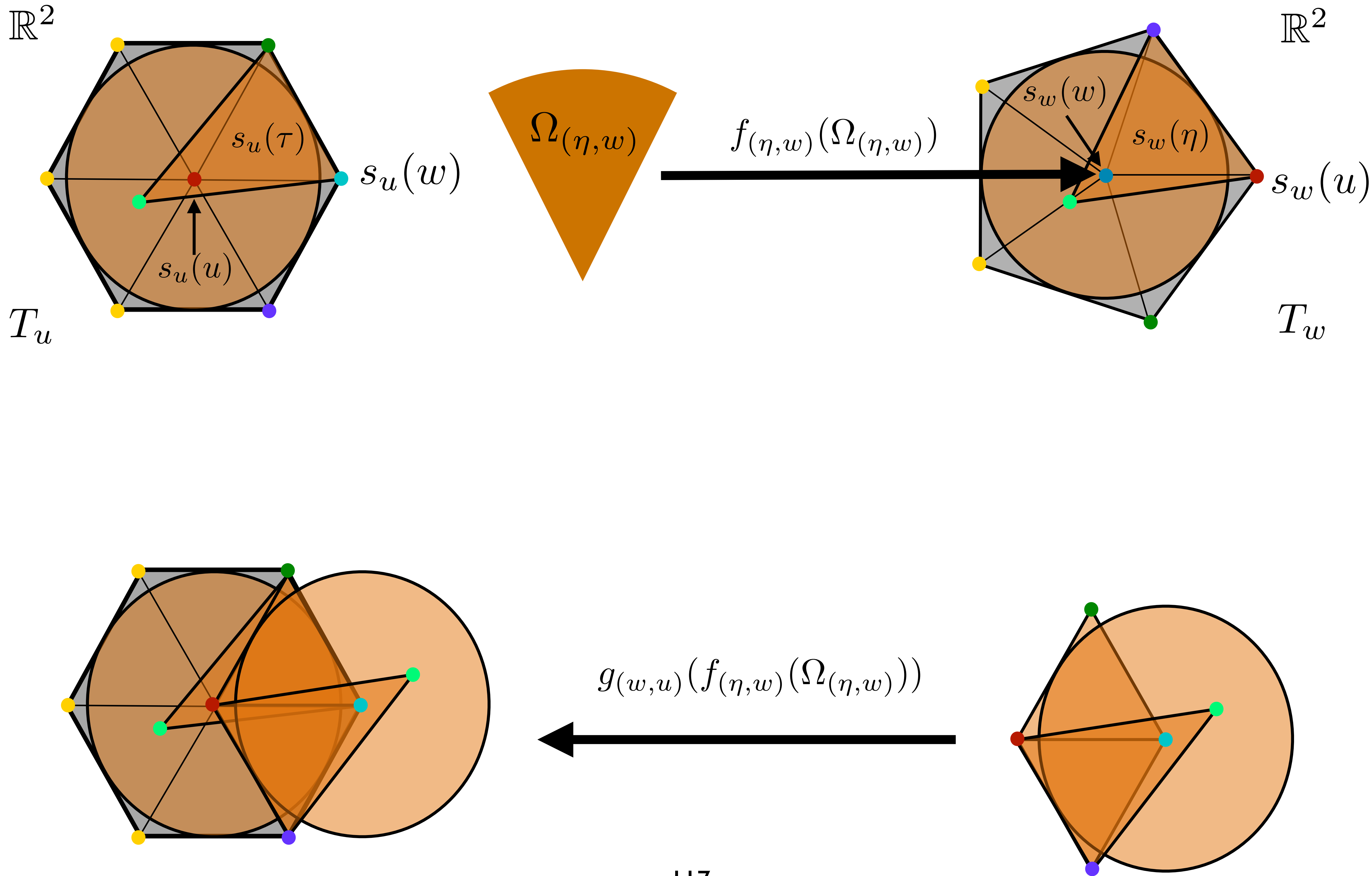
Building a Set of Gluing Data



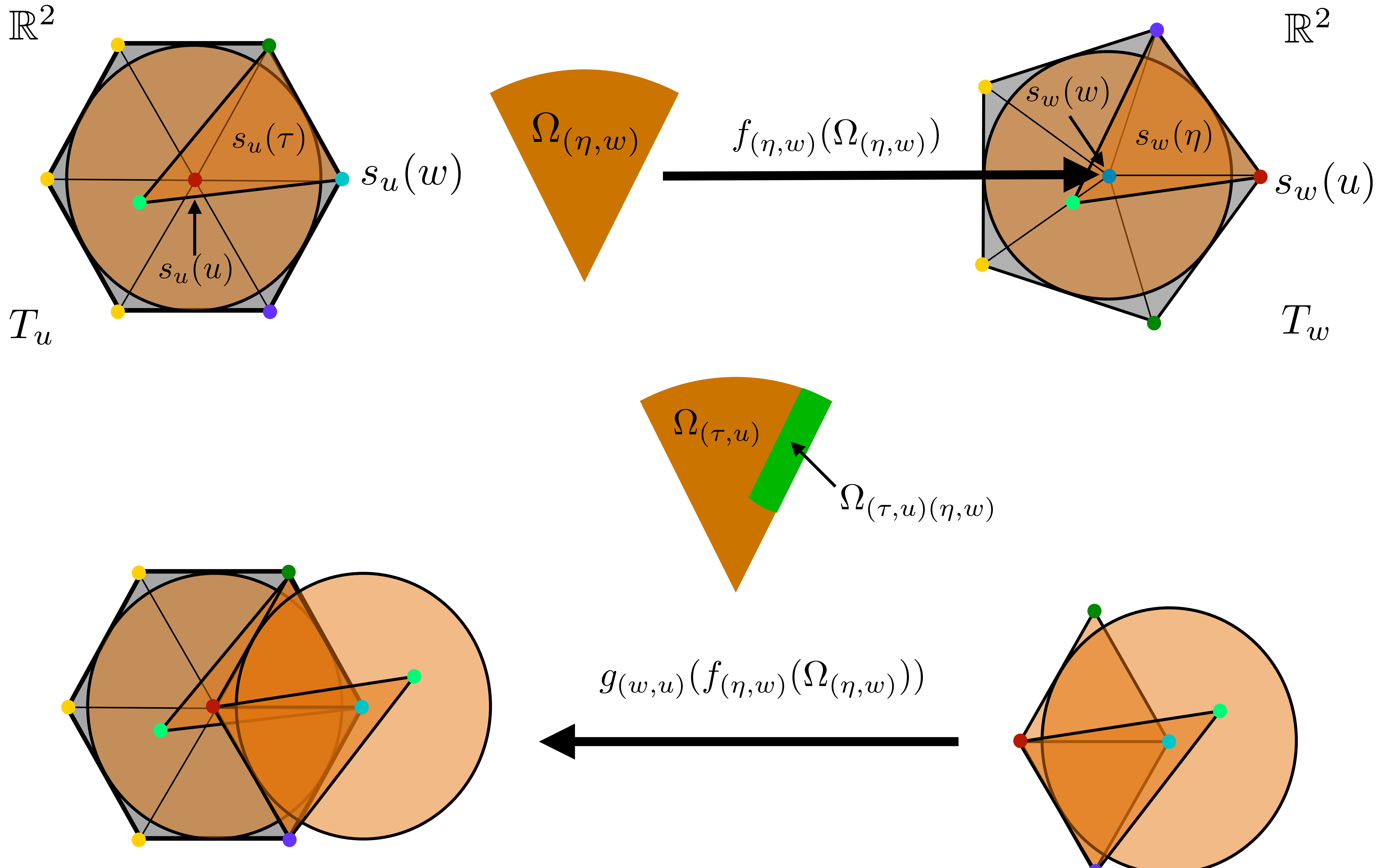
Building a Set of Gluing Data



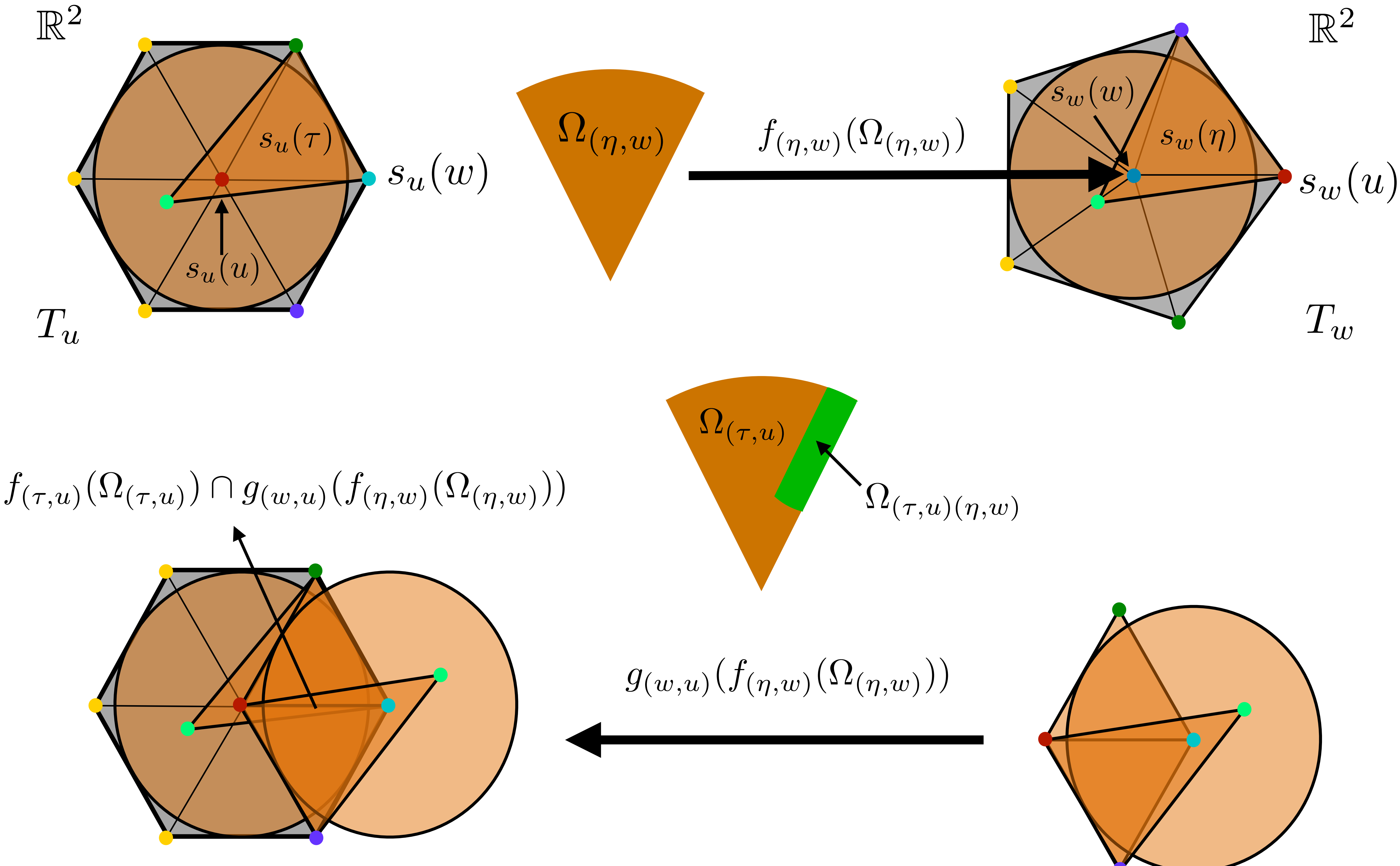
Building a Set of Gluing Data



Building a Set of Gluing Data

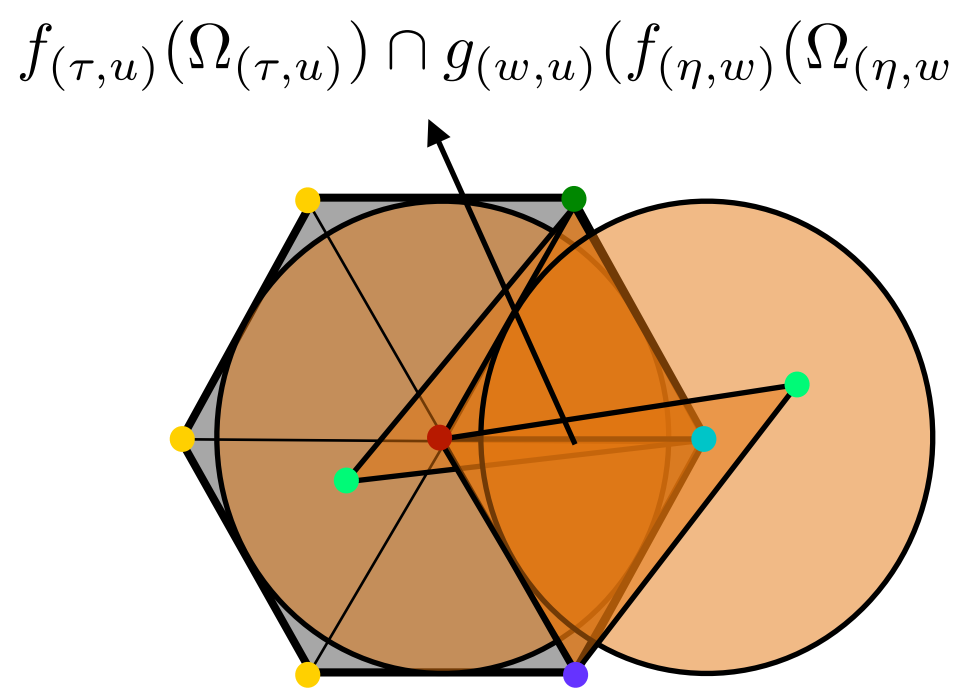


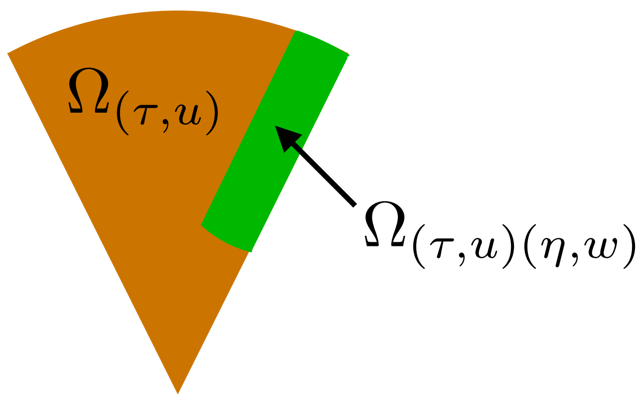
Building a Set of Gluing Data



Building a Set of Gluing Data

$$\Omega_{(\tau,u)(\eta,w)} = f^{-1}_{(\tau,u)}(f_{(\tau,u)}(\Omega_{(\tau,u)}) \cap g_{(w,u)}(f_{(\eta,w)}(\Omega_{(\eta,w)})))$$





Building a Set of Gluing Data

Building a Set of Gluing Data

(3) $u \neq w$ and w is not a vertex of τ nor u is a vertex of η

Building a Set of Gluing Data

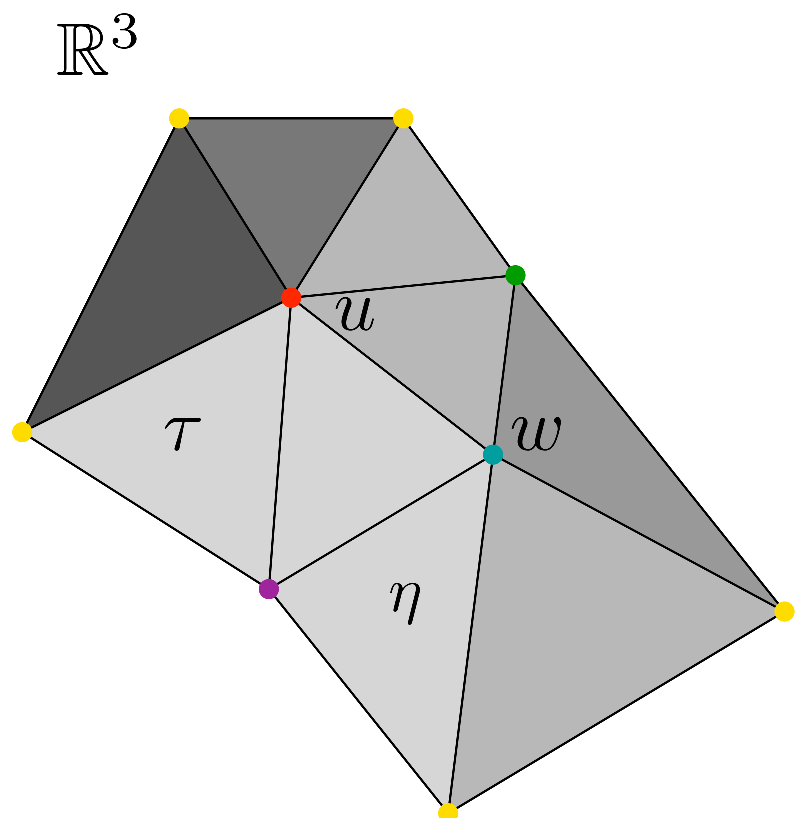
(3) $u \neq w$ and w is not a vertex of τ nor u is a vertex of η

$$\Omega_{(\tau,u)(\eta,w)} = \emptyset$$

Building a Set of Gluing Data

(3) $u \neq w$ and w is not a vertex of τ nor u is a vertex of η

$$\Omega_{(\tau,u)(\eta,w)} = \emptyset$$

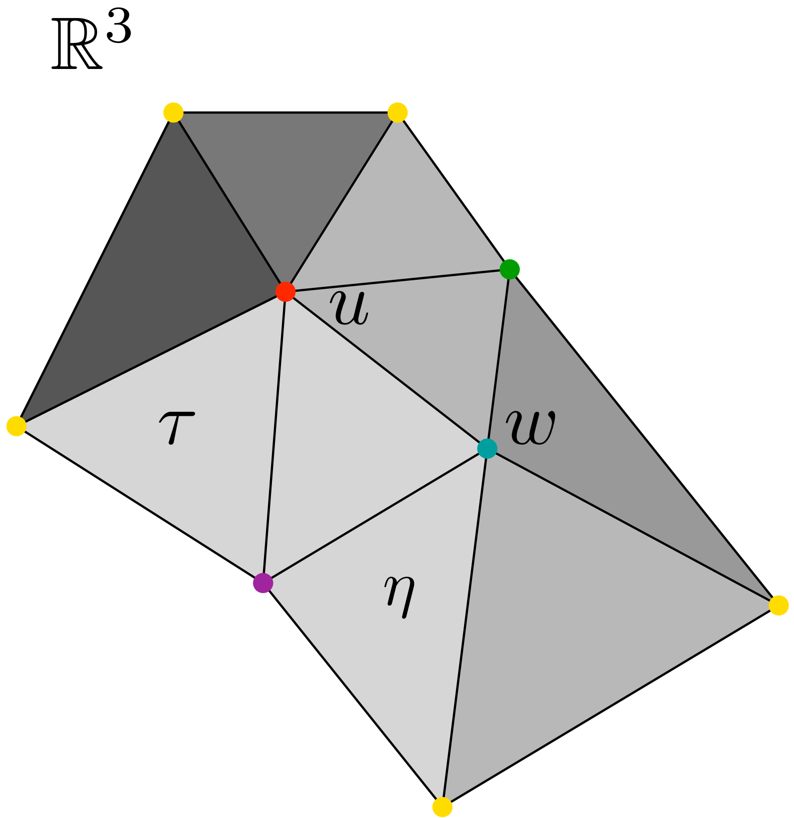


$st(u, S_T) \cup st(w, S_T)$

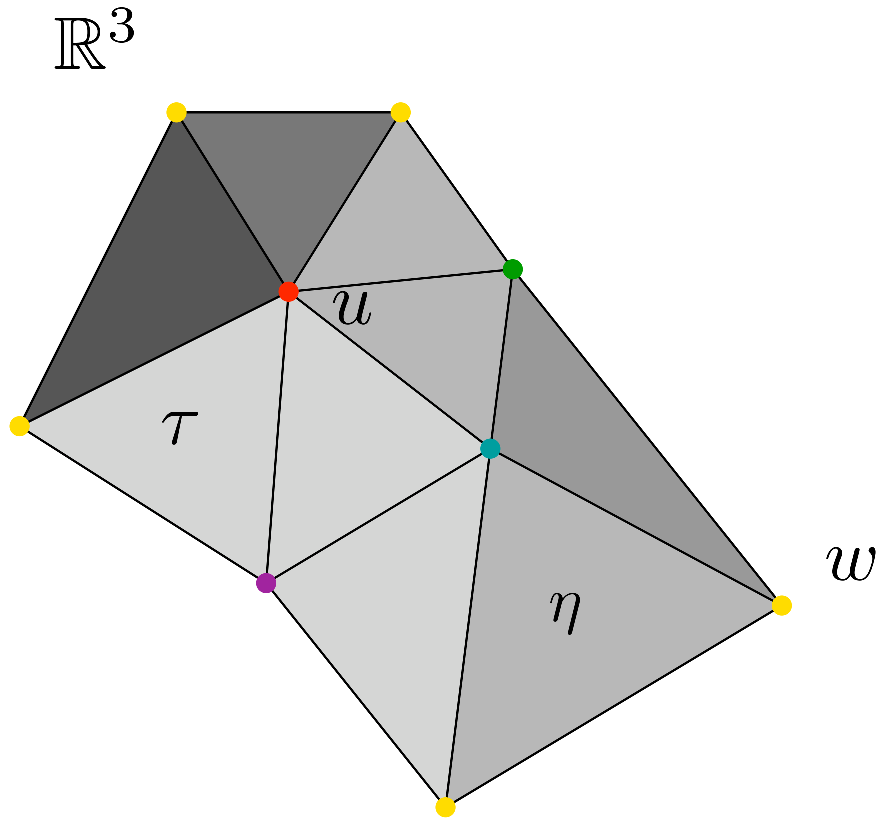
Building a Set of Gluing Data

(3) $u \neq w$ and w is not a vertex of τ nor u is a vertex of η

$$\Omega_{(\tau,u)(\eta,w)} = \emptyset$$



$st(u, S_T) \cup st(w, S_T)$



$st(u, S_T) \cup st(w, S_T)$

Building a Set of Gluing Data

Building a Set of Gluing Data

We can show that the above definition of gluing domain satisfies condition (2) of the definition of sets of gluing data we saw before:

Building a Set of Gluing Data

We can show that the above definition of gluing domain satisfies condition (2) of the definition of sets of gluing data we saw before:

(2) For every pair $(i, j) \in I \times I$, the set Ω_{ij} is an open subset of Ω_i. Furthermore, $\Omega_{ii} = \Omega_i$ and $\Omega_{ji} \neq \emptyset$ if and only if $\Omega_{ij} \neq \emptyset$.

Fitting Surfaces to Polygonal Meshes (Part II)

Marcelo Siqueira
UFMS

Outline

- Building a Set of Gluing Data
- The User's Perspective
- Building Parametrizations
- Results
- Conclusions

Building a Set of Gluing Data

Building a Set of Gluing Data

Transition functions

Building a Set of Gluing Data

Transition functions

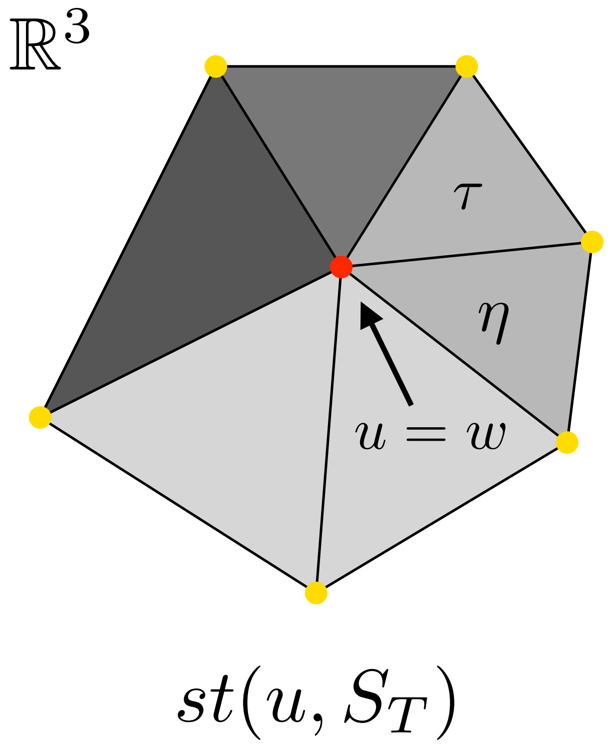
Let

$$K = \{((\tau, u), (\eta, w)) \in I \times I \mid \Omega_{(\tau,u),(\eta,w)} \neq \emptyset\}\,.$$

Building a Set of Gluing Data

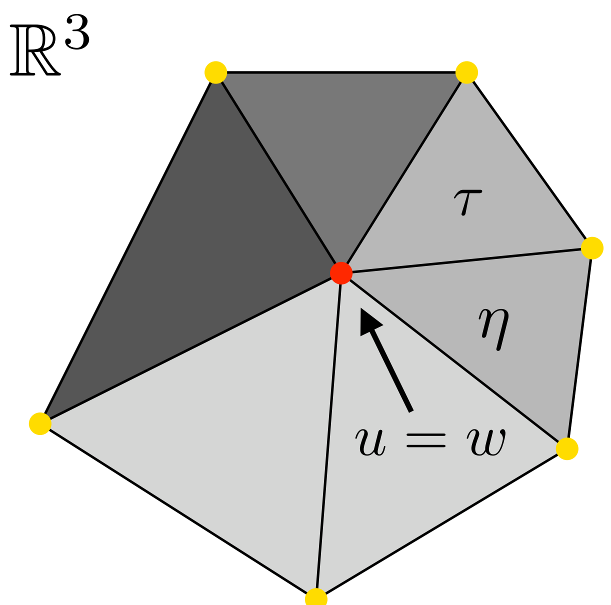
Building a Set of Gluing Data

$(1)\ \ u = w$

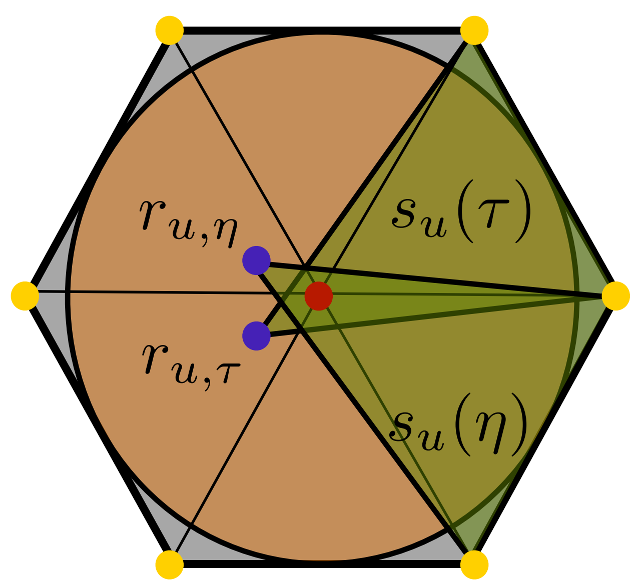


Building a Set of Gluing Data

$(1) \;\; u = w$



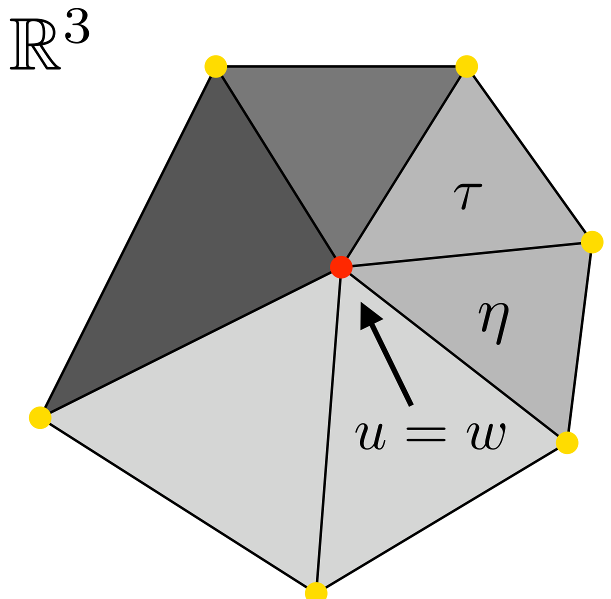
$st(u, S_T)$



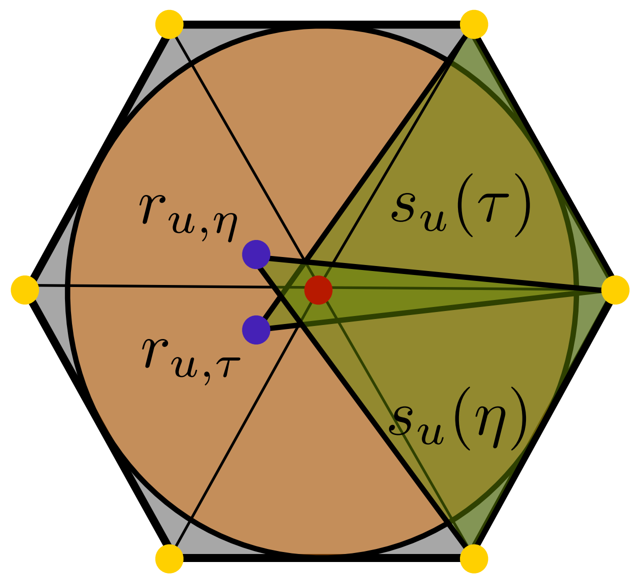
T_u

Building a Set of Gluing Data

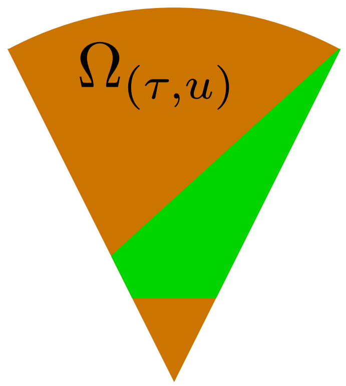
(1) $u = w$



$st(u, S_T)$

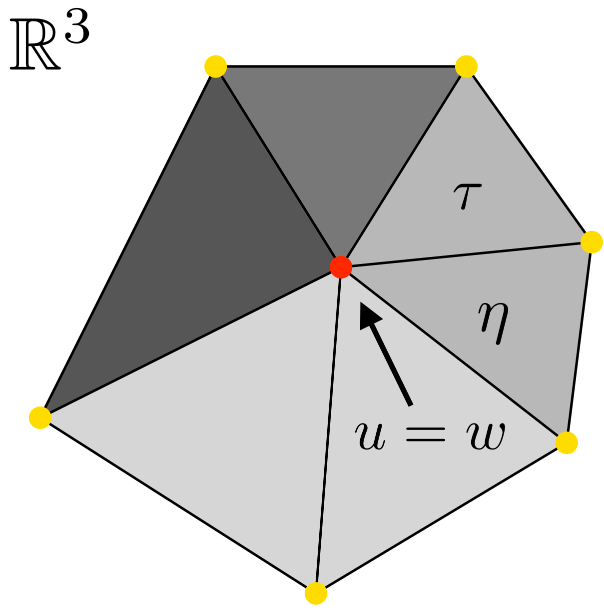


T_u

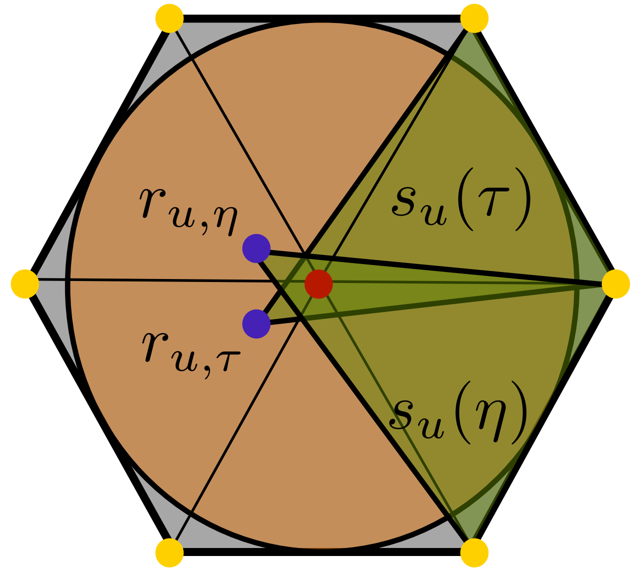


Building a Set of Gluing Data

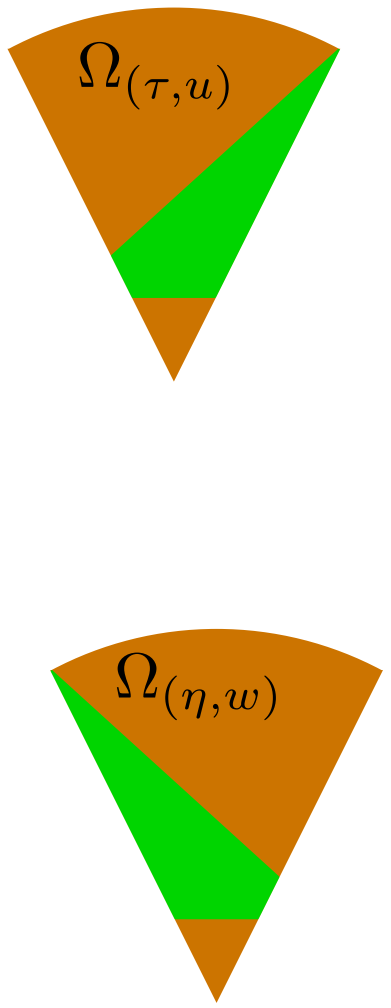
(1) $u = w$



$st(u, S_T)$

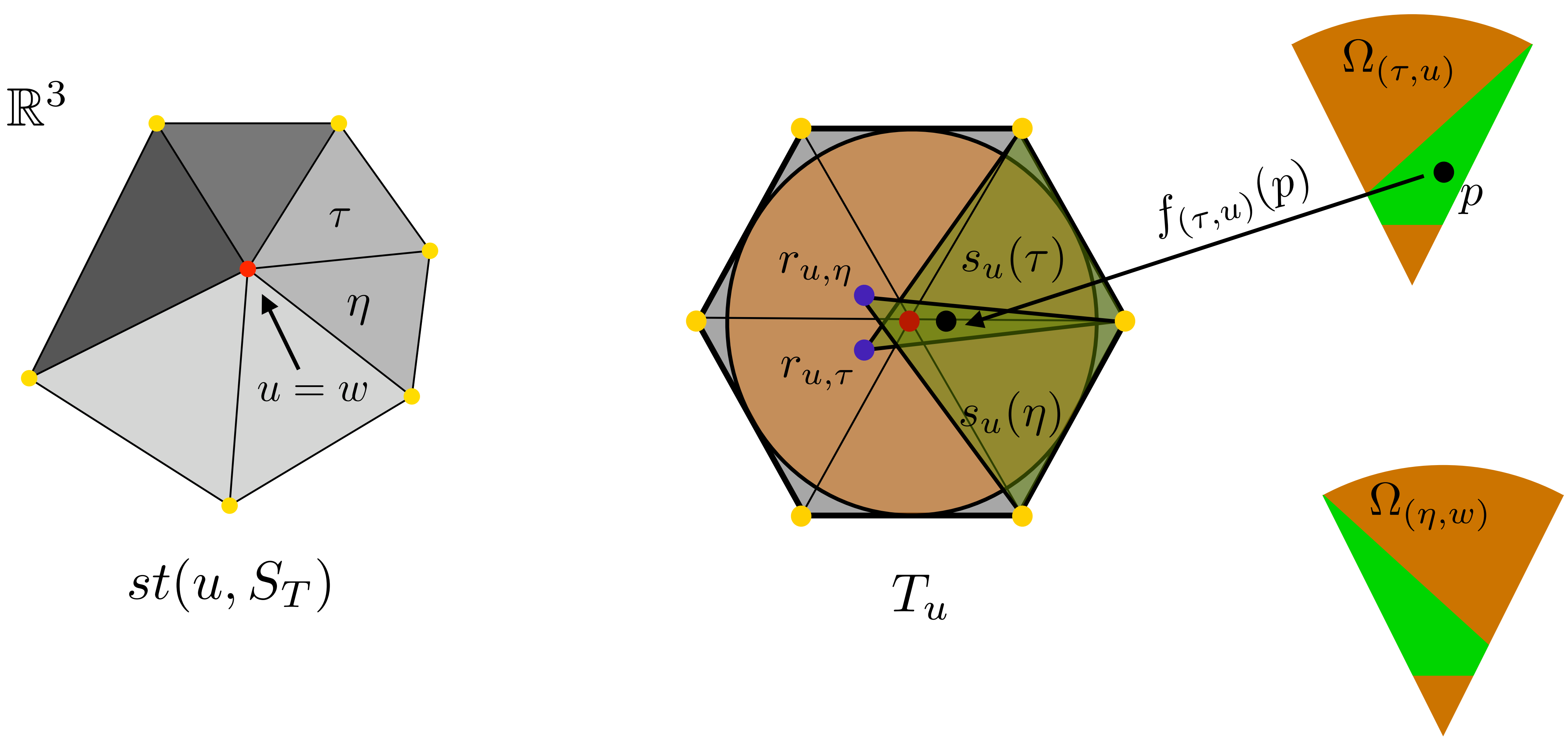


T_u



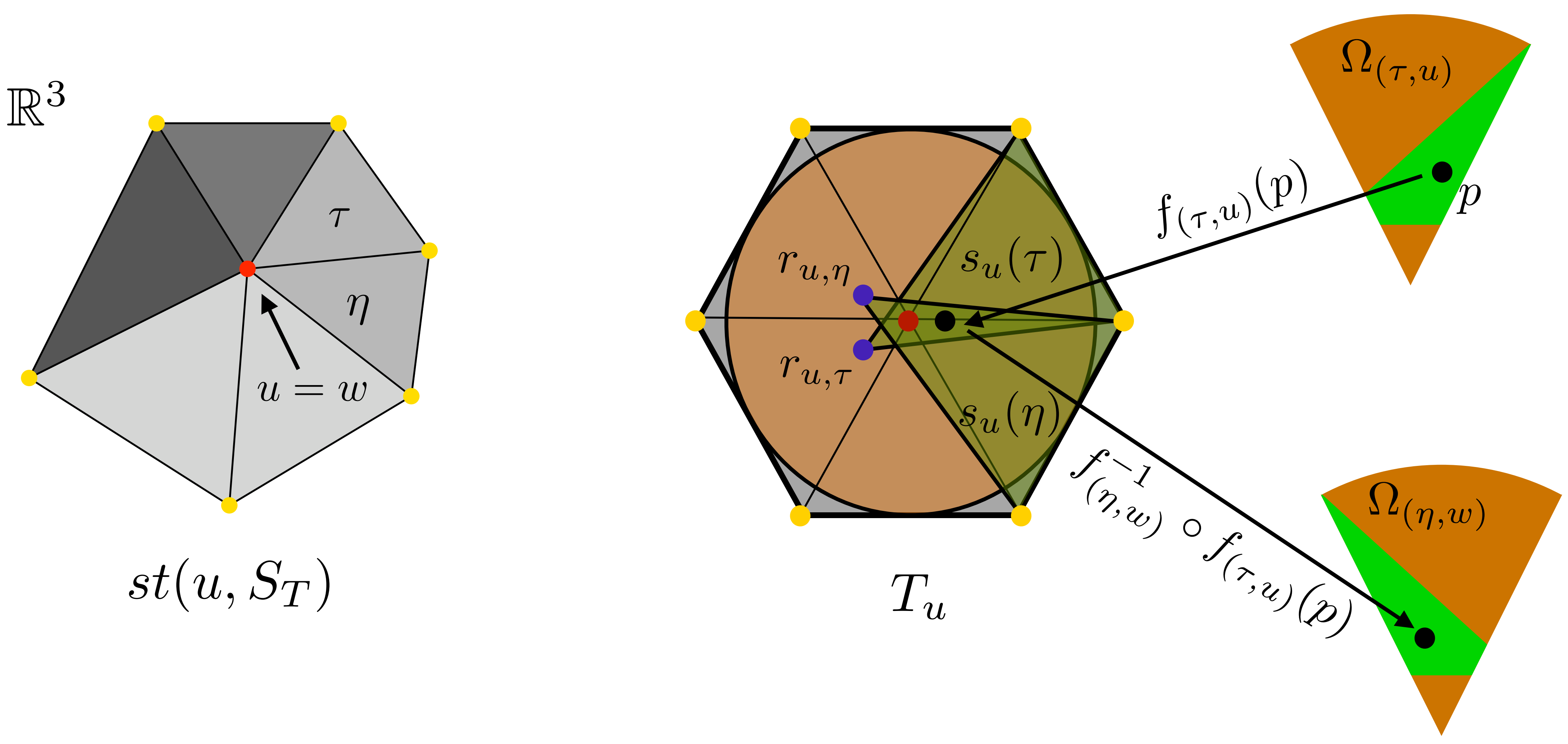
Building a Set of Gluing Data

(1) $u = w$



Building a Set of Gluing Data

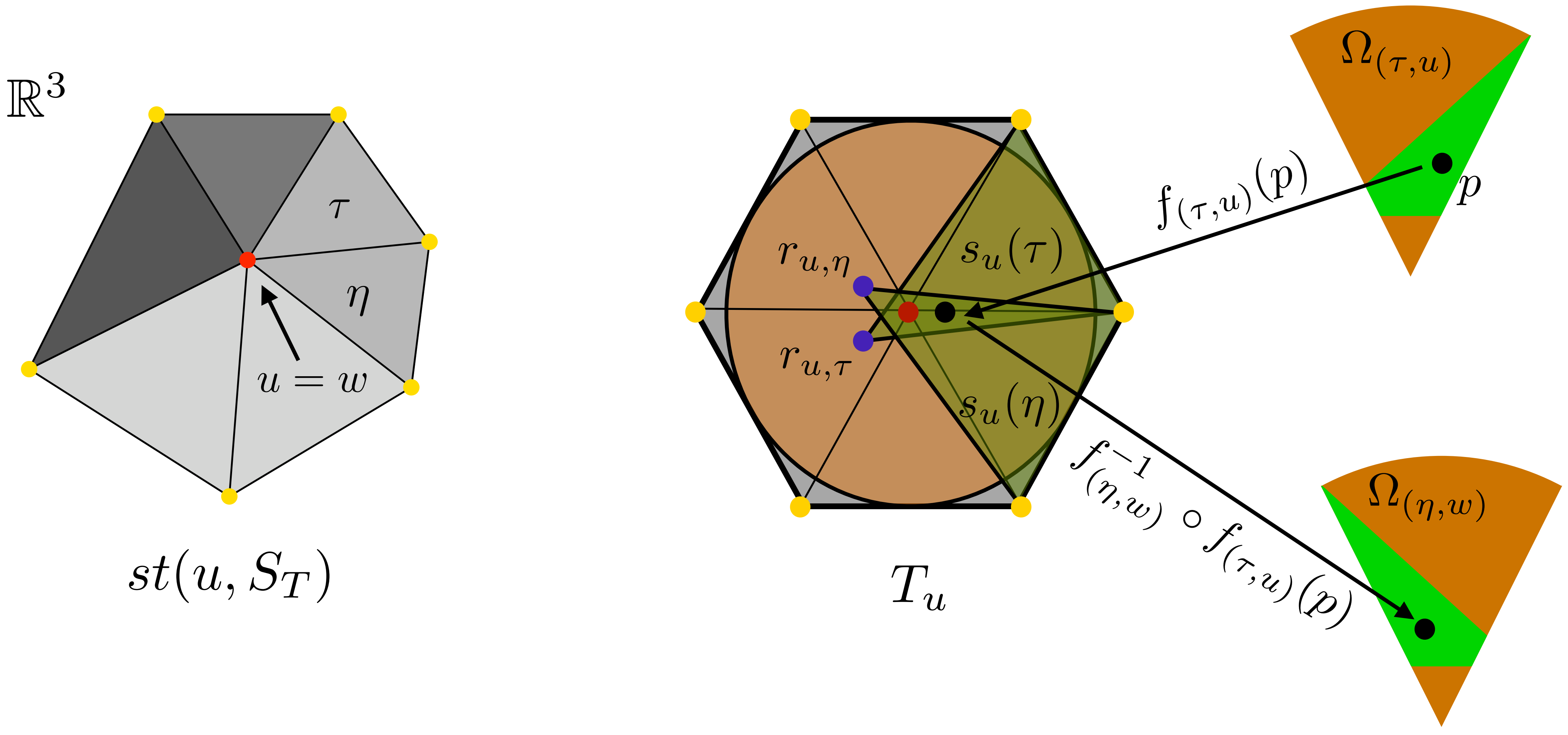
(1) $u = w$



Building a Set of Gluing Data

(1) $u = w$

$$f^{-1}_{(\eta,w)} \circ f_{(\tau,u)}(p)$$



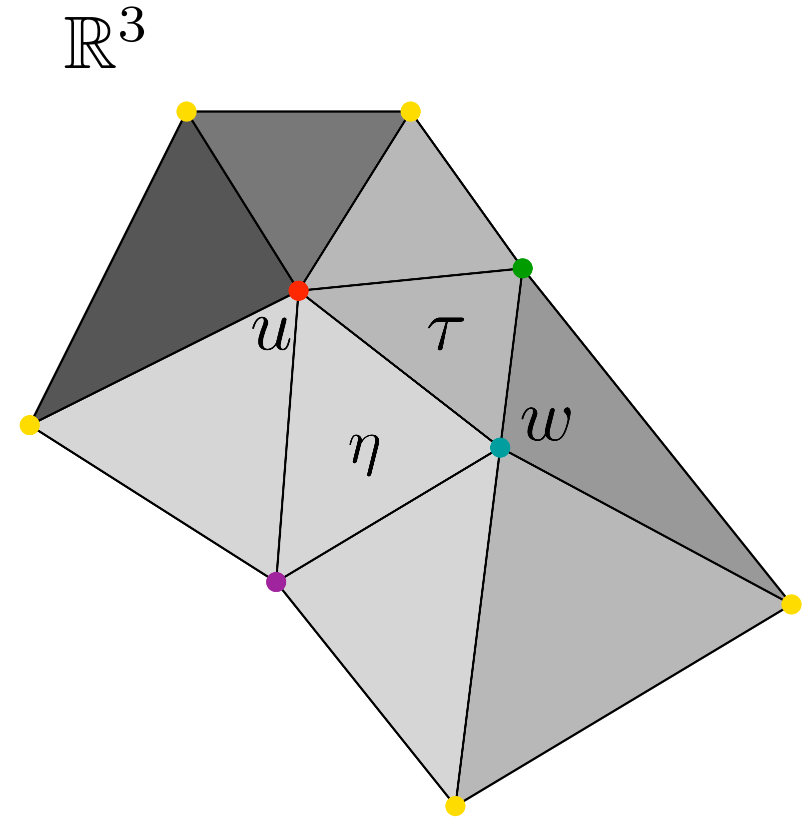
Building a Set of Gluing Data

Building a Set of Gluing Data

(2) otherwise

Building a Set of Gluing Data

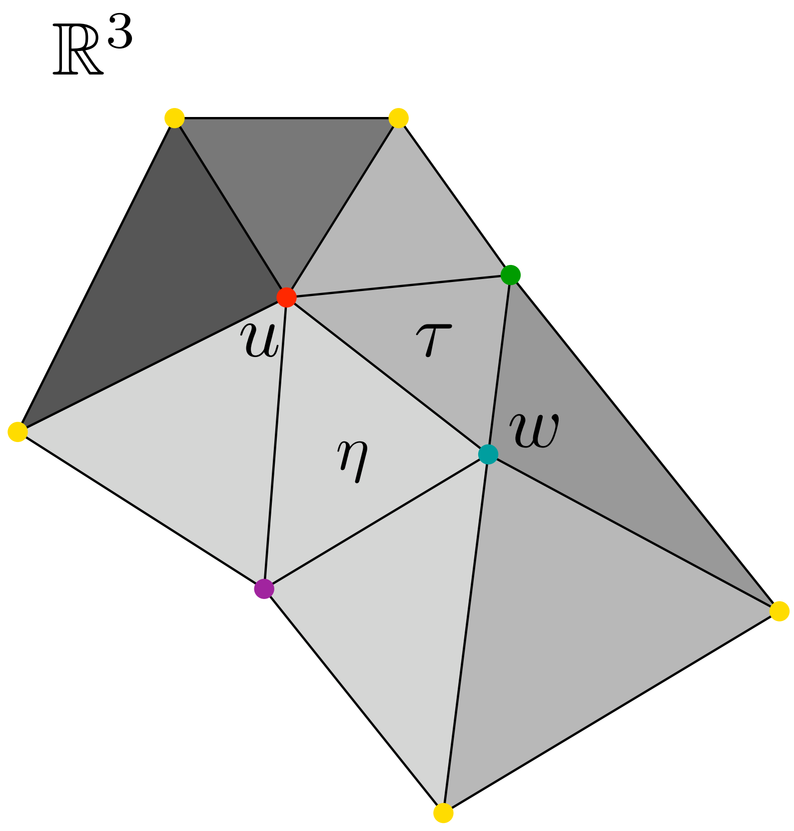
(2) otherwise

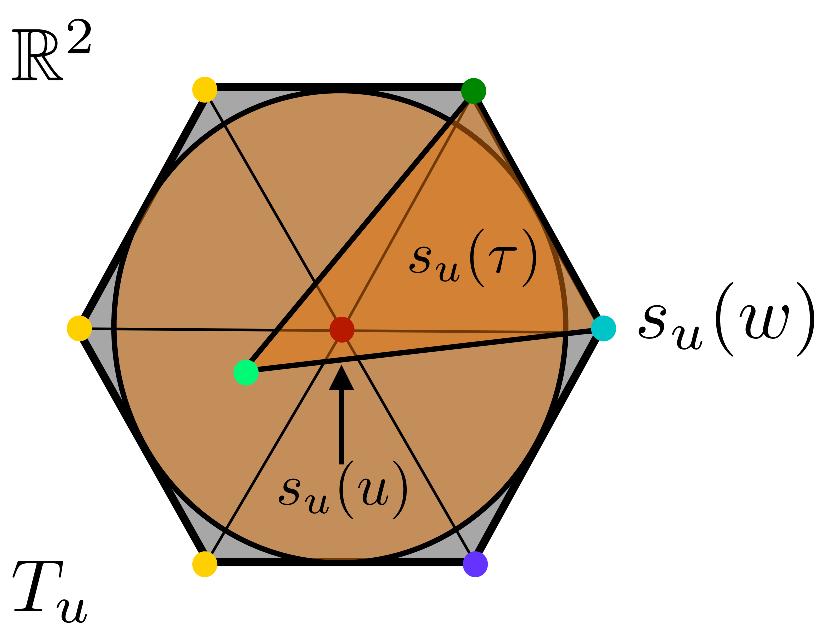


$st(u, S_T) \cup st(w, S_T)$

Building a Set of Gluing Data

(2) otherwise

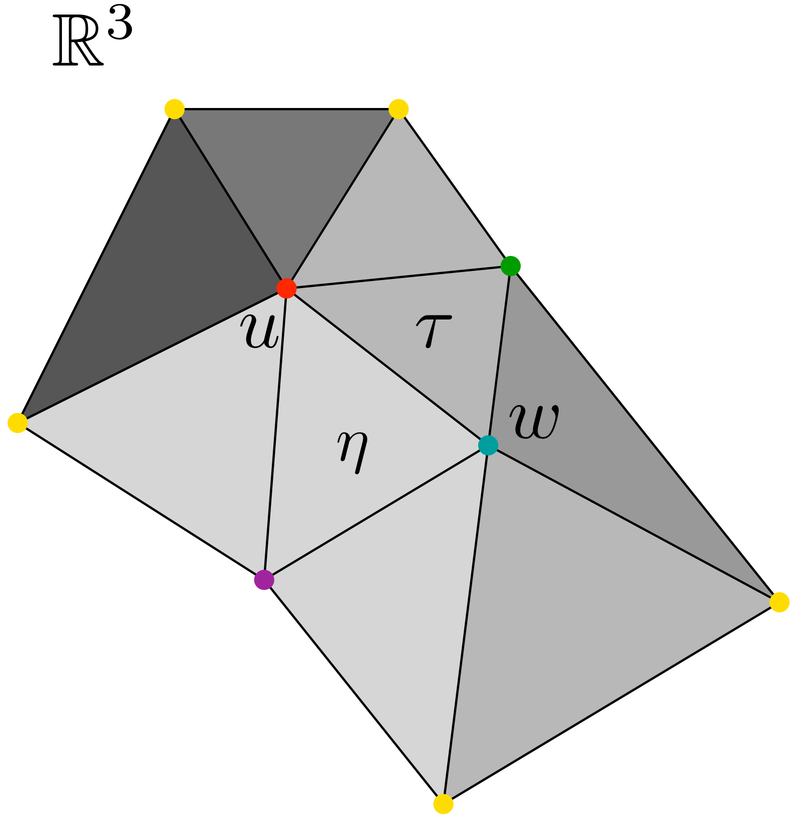




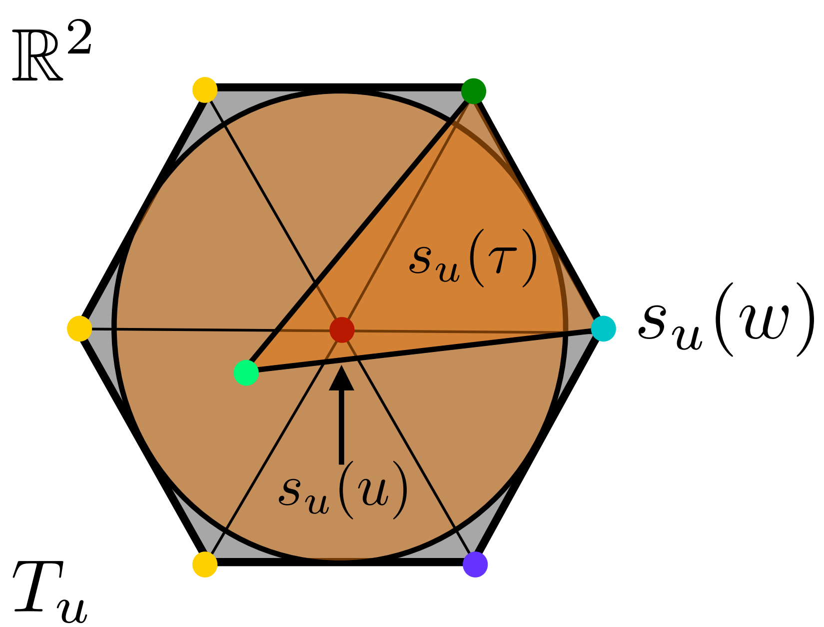
$st(u, S_T) \cup st(w, S_T)$

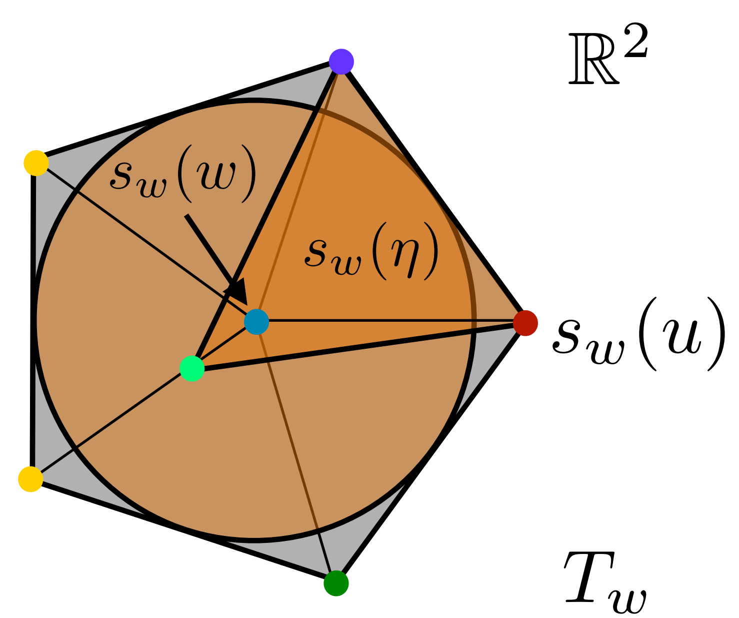
Building a Set of Gluing Data

(2) otherwise



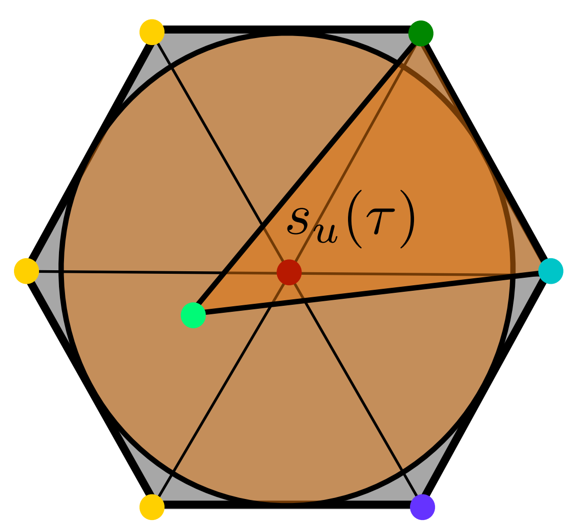
$st(u, S_T) \cup st(w, S_T)$

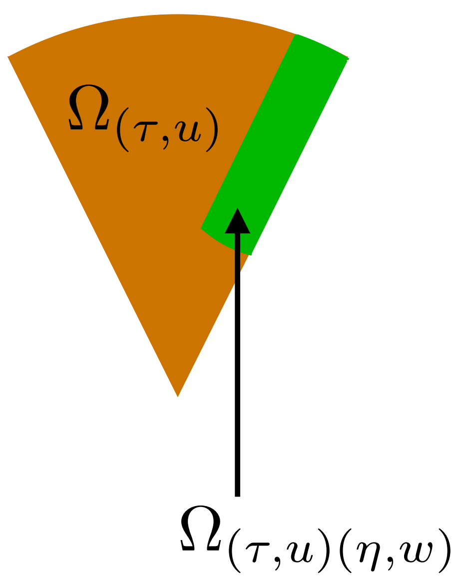




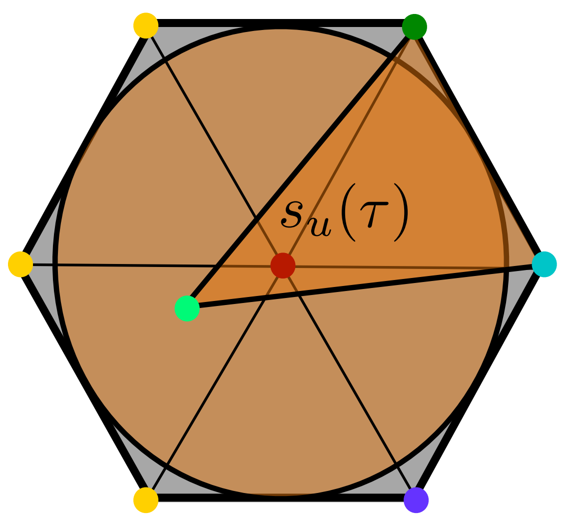
Building a Set of Gluing Data

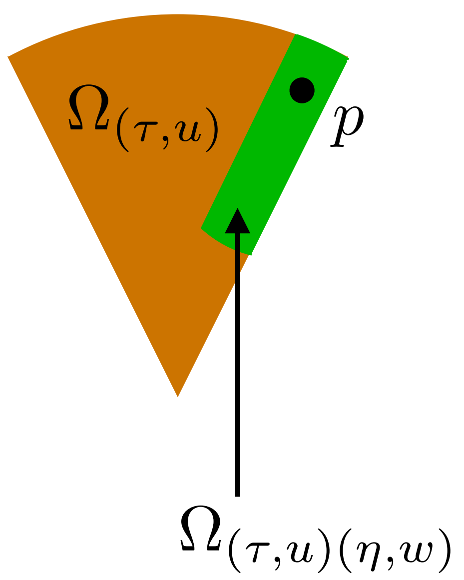
Building a Set of Gluing Data



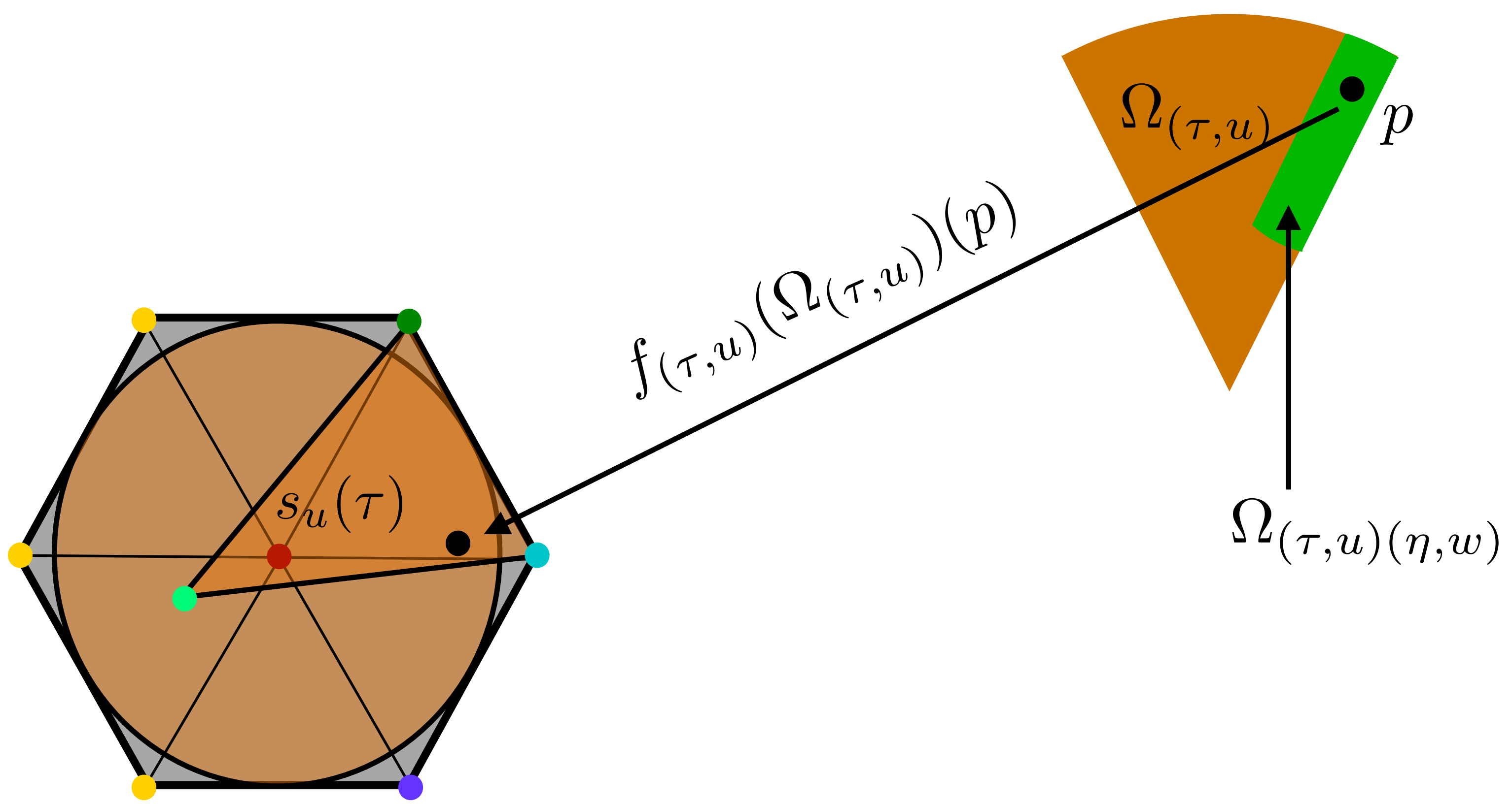


Building a Set of Gluing Data

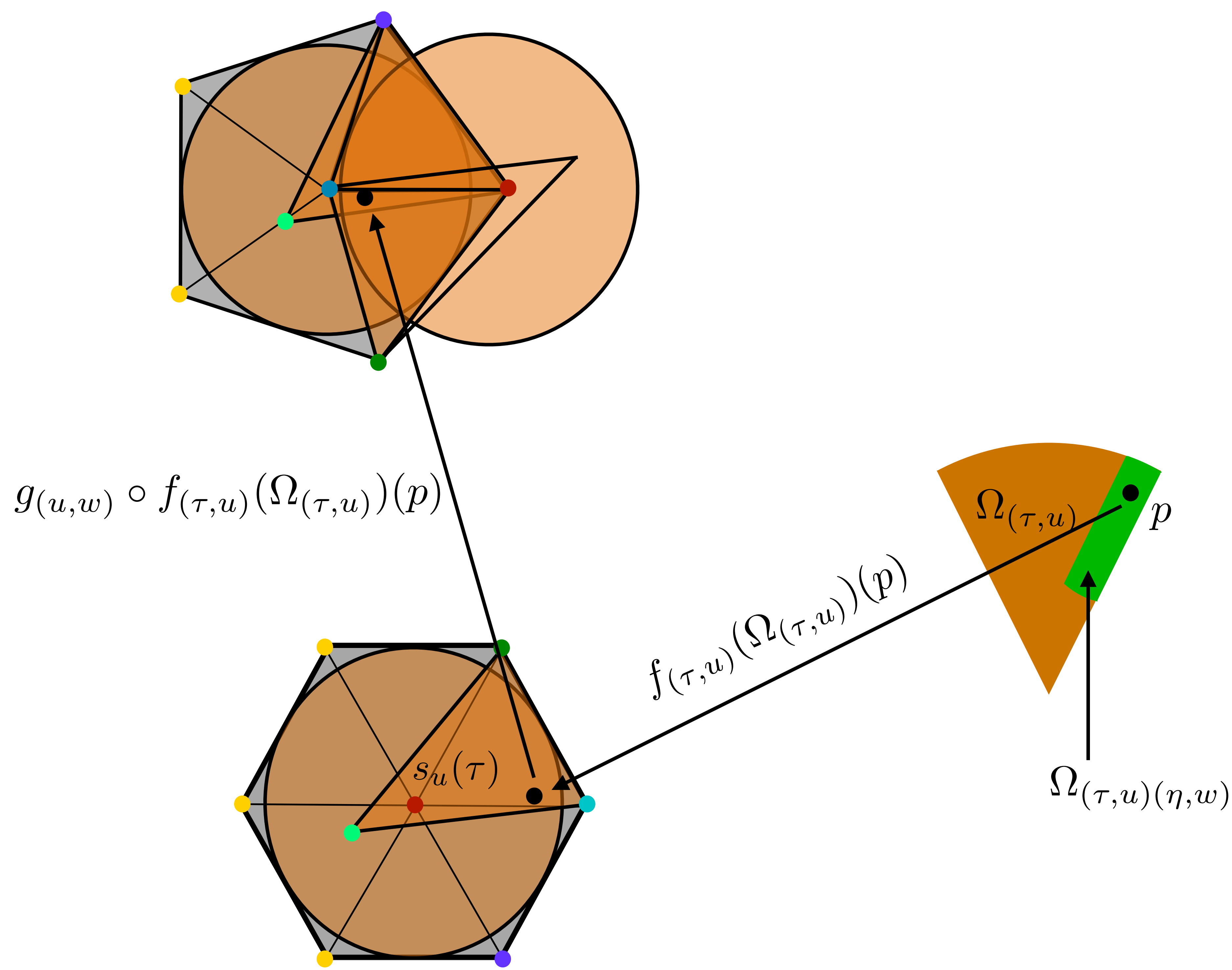




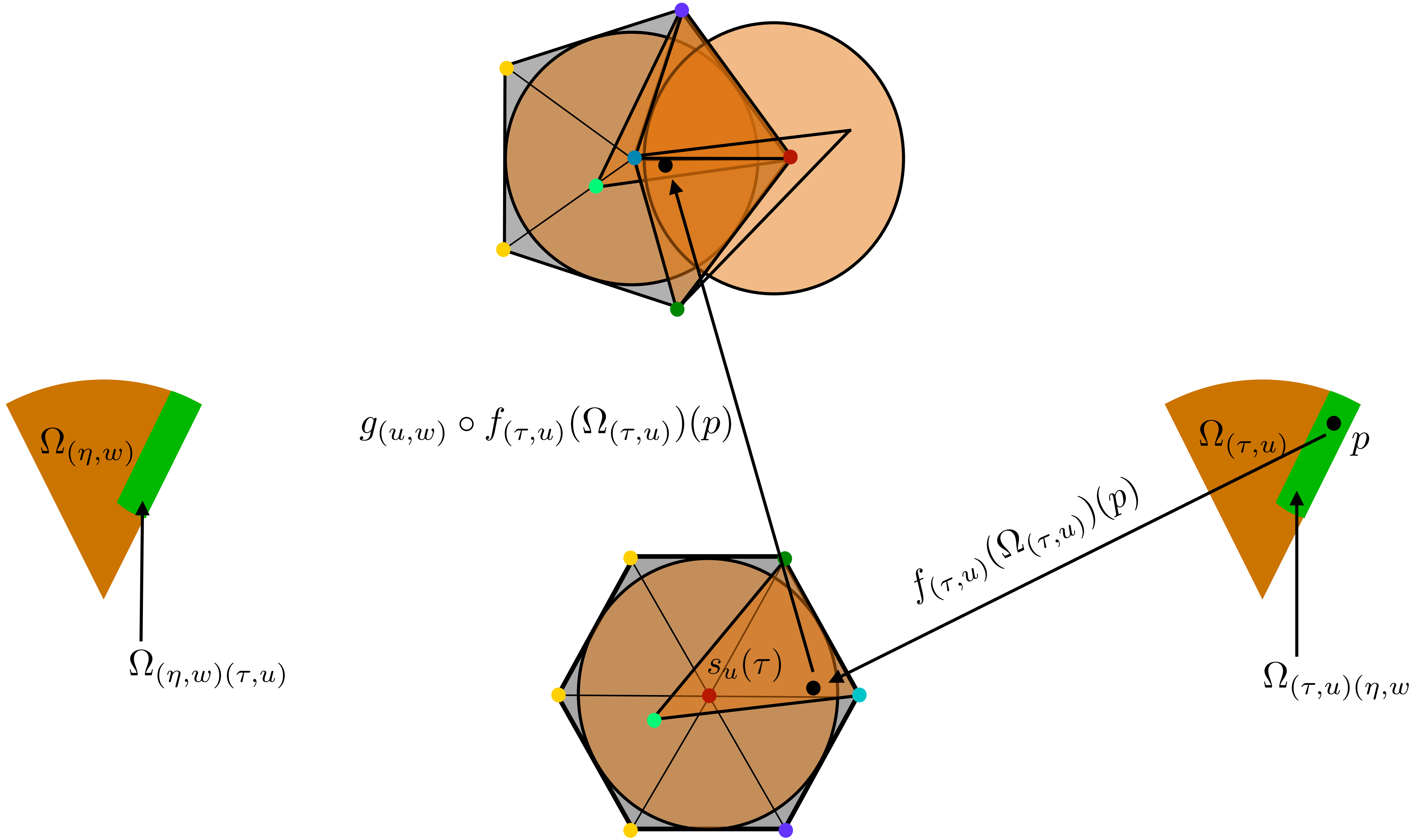
Building a Set of Gluing Data



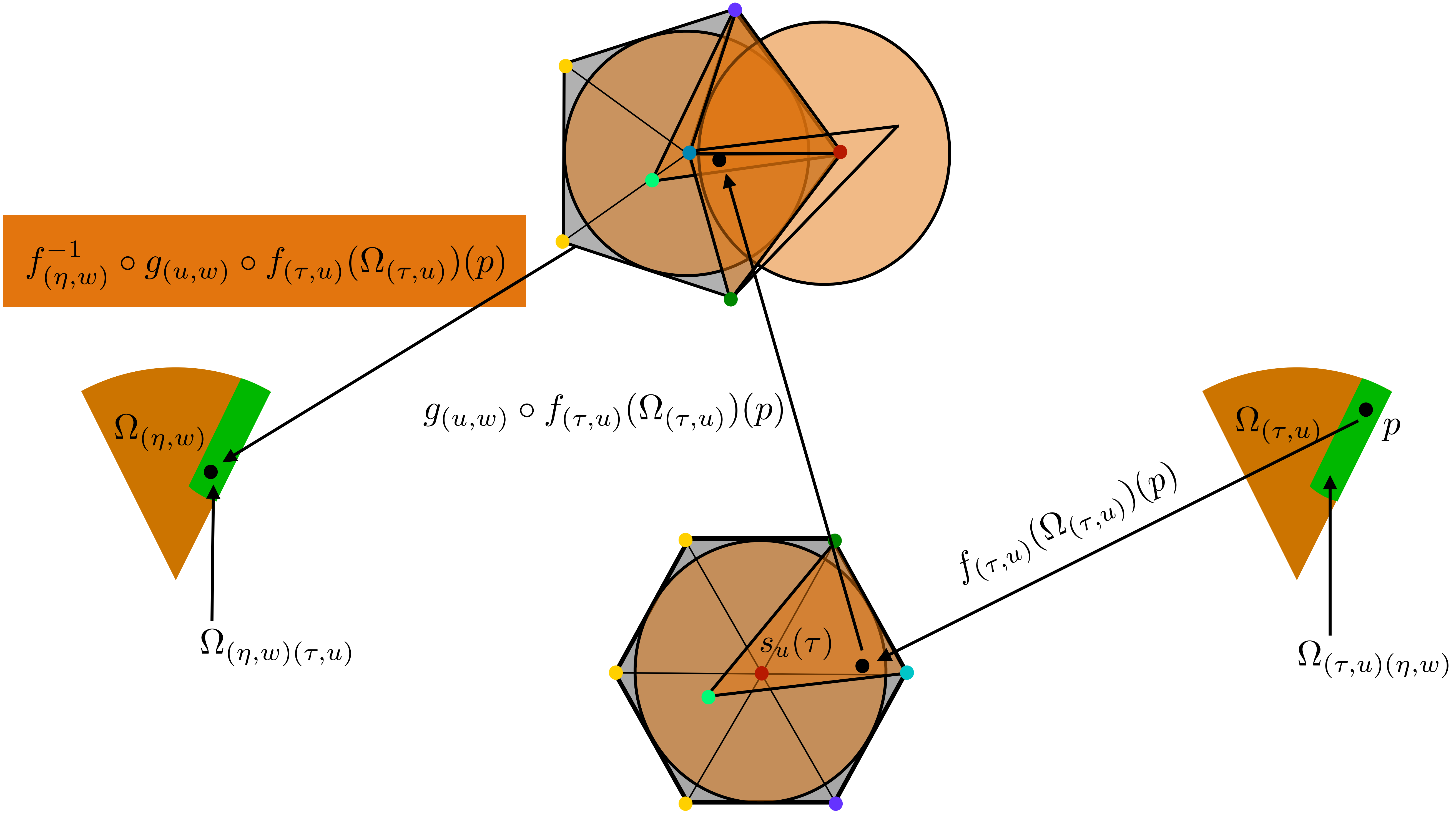
Building a Set of Gluing Data



Building a Set of Gluing Data



Building a Set of Gluing Data



Building a Set of Gluing Data

Building a Set of Gluing Data

For every $((\tau, u), (\eta, w)) \in K$, we define

$$\varphi_{(\eta,w)(\tau,u)} : \Omega_{(\tau,u),(\eta,w)} \rightarrow \varphi_{(\eta,w)(\tau,u)} \, ,$$

the transition function from $\Omega_{(\tau,u)}$ to $\Omega_{(\eta,w)}$, to be

Building a Set of Gluing Data

For every $((\tau, u), (\eta, w)) \in K$, we define

$$\varphi_{(\eta,w)(\tau,u)} : \Omega_{(\tau,u),(\eta,w)} \to \varphi_{(\eta,w)(\tau,u)} \,,$$

the transition function from $\Omega_{(\tau,u)}$ to $\Omega_{(\eta,w)}$, to be

$$\varphi_{(\eta,w)(\tau,u)}(p) = \begin{cases} f^{-1}_{(\eta,w)} \circ f_{(\tau,u)}(p) & \text{if } u = w \\ \\ f^{-1}_{(\eta,w)} \circ g_{(u,w)} \circ f_{(\tau,u)}(p) & \text{otherwise}\,, \end{cases}$$

for every $p \in \Omega_{(\tau,u)(\eta,w)}$.

Building a Set of Gluing Data

Building a Set of Gluing Data

We can show that the above definition of transition functions satisfies conditions (3)(a)-(c) of the definition of sets of gluing data:

Building a Set of Gluing Data

We can show that the above definition of transition functions satisfies conditions (3)(a)-(c) of the definition of sets of gluing data:

(a) $\varphi_{ii} = \mathrm{id}_{\Omega_i}$, for all $i \in I$,

(b) $\varphi_{ij} = \varphi_{ji}^{-1}$, for all $(i, j) \in K$, and

(c) for all i, j, and k, if $\Omega_{ji} \cap \Omega_{jk} \neq \emptyset$ then $\varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk}) \subseteq \Omega_{ik}$ and $\varphi_{ki}(x) = \varphi_{kj} \circ \varphi_{ji}(x)$, for all $x \in \varphi_{ji}^{-1}(\Omega_{ji} \cap \Omega_{jk})$.

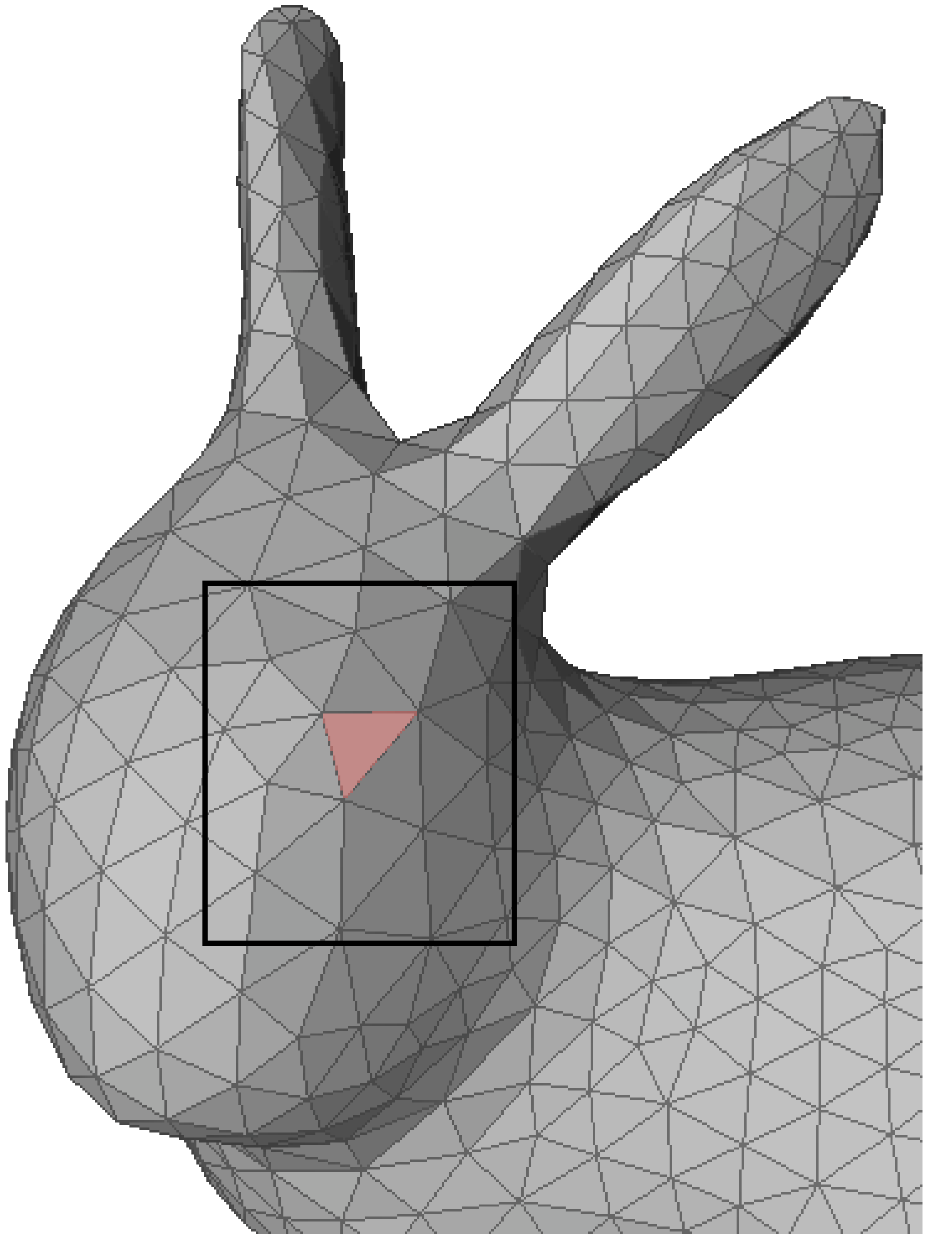
User's Perspective

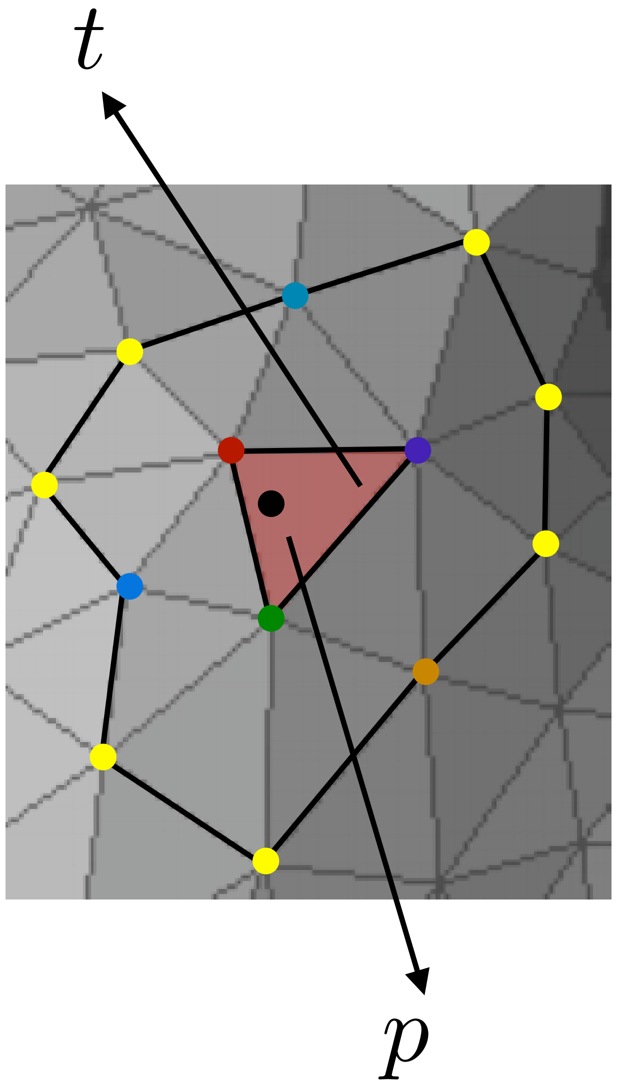
User's Perspective

Let t be a triangle in S_T and p be any point in t:

User's Perspective

Let t be a triangle in S_T and p be any point in t:

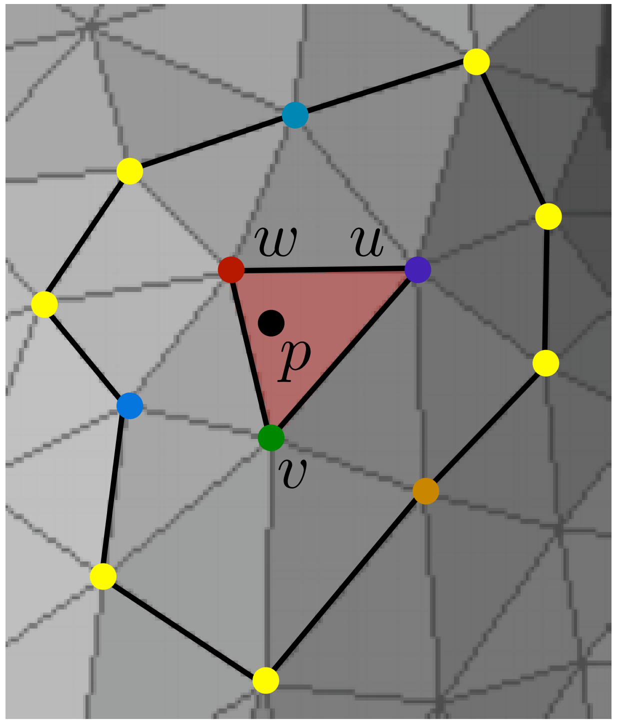




User’s Perspective

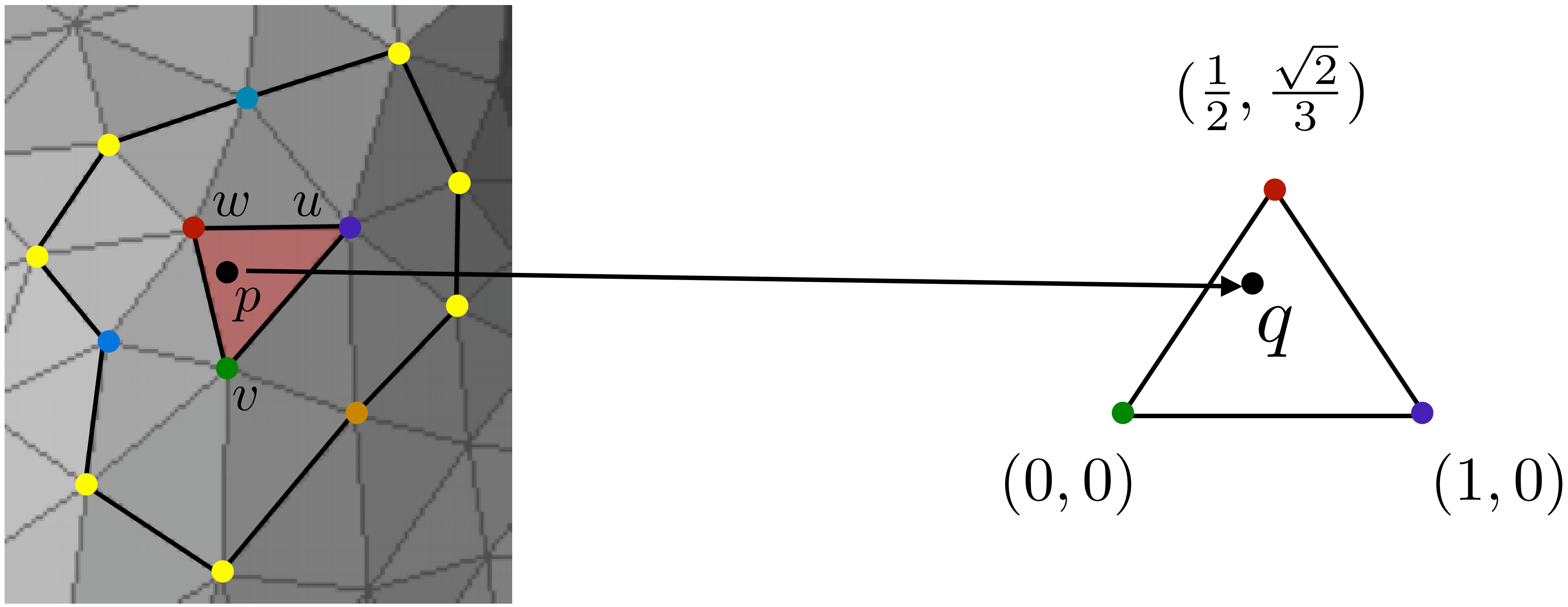
User's Perspective

Map p to an equilateral triangle in $\mathbb{R}^2$.



User's Perspective

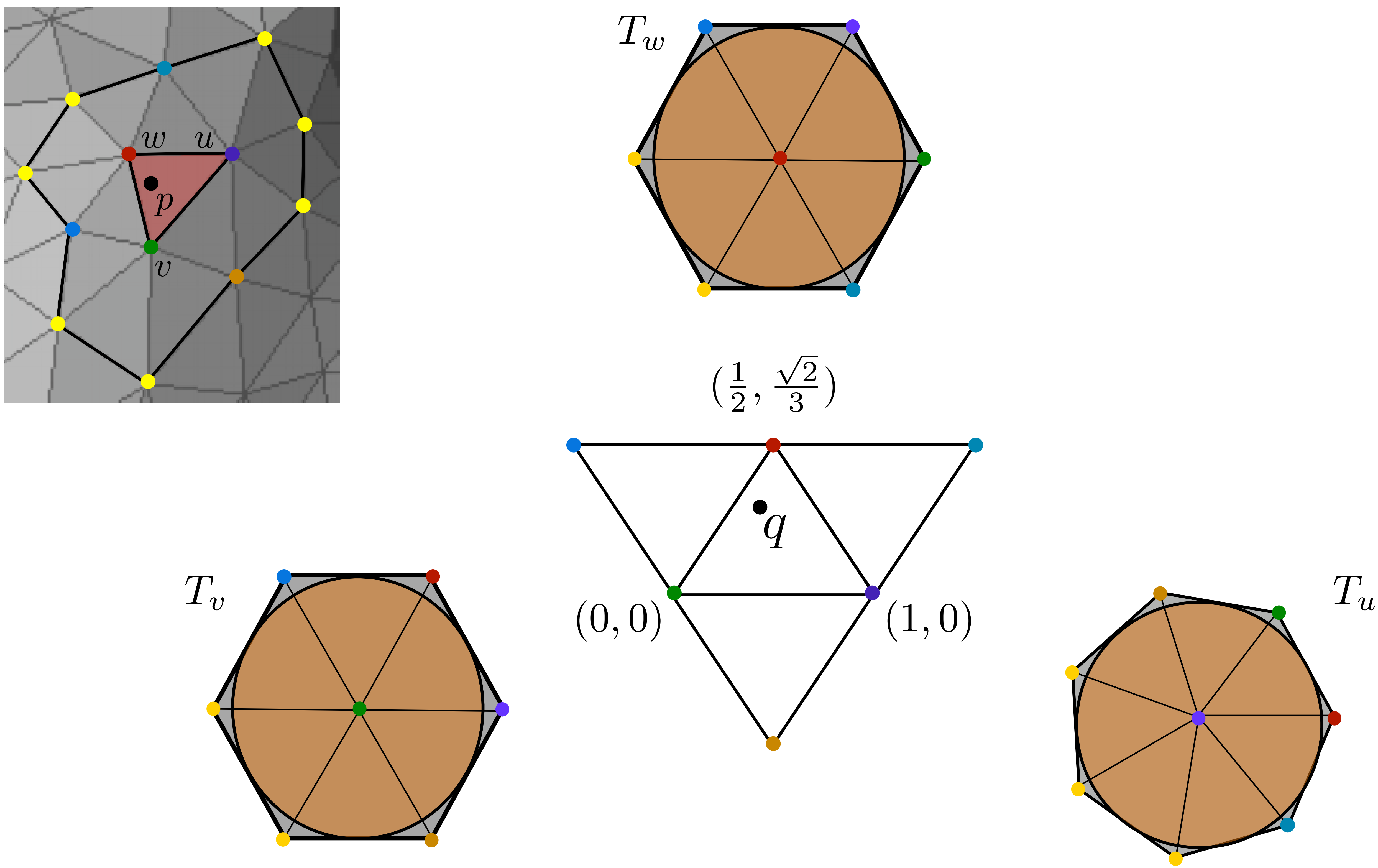
Map p to an equilateral triangle in $\mathbb{R}^2$.



We can do that by using barycentric coordinates.

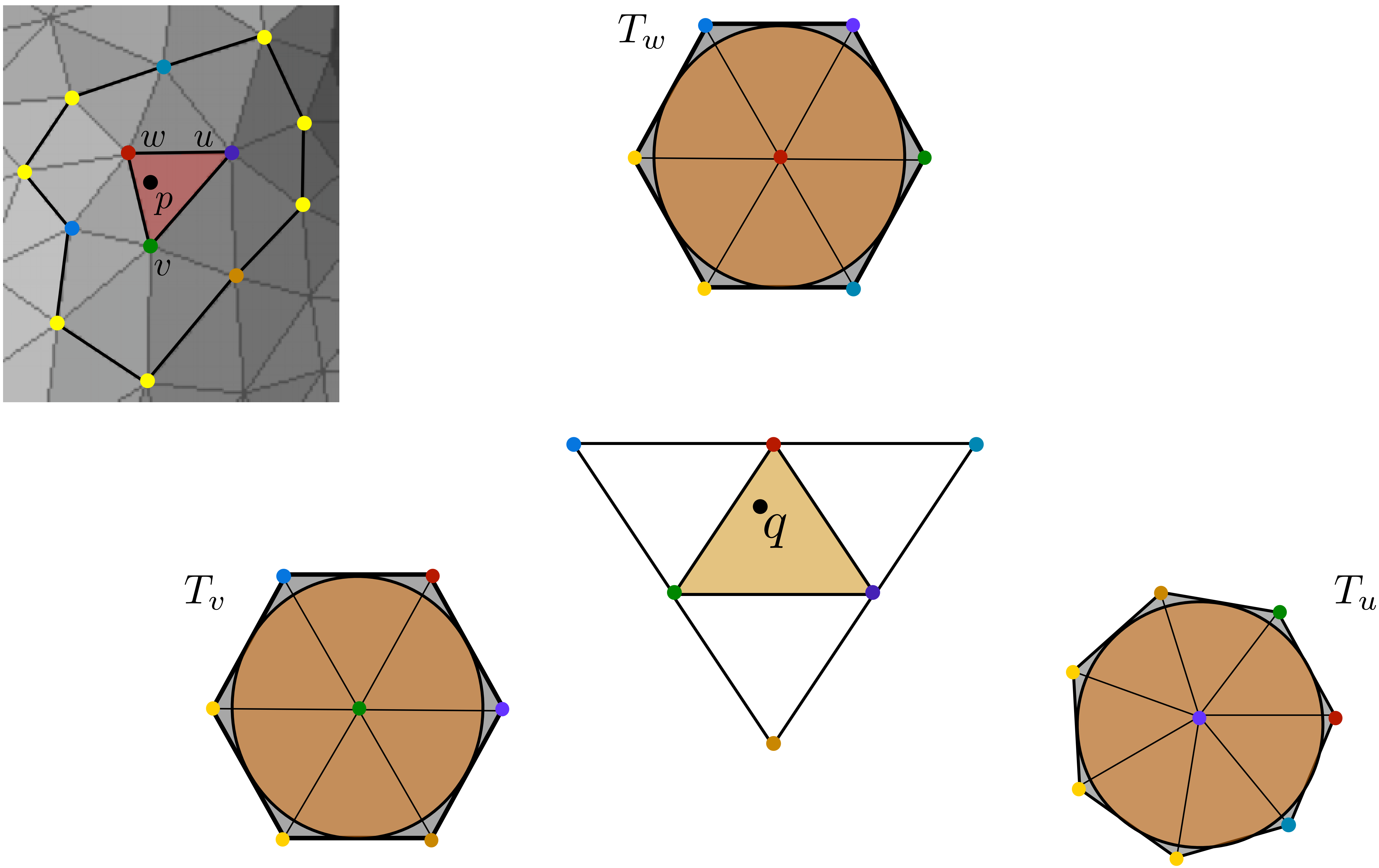
User’s Perspective

User's Perspective

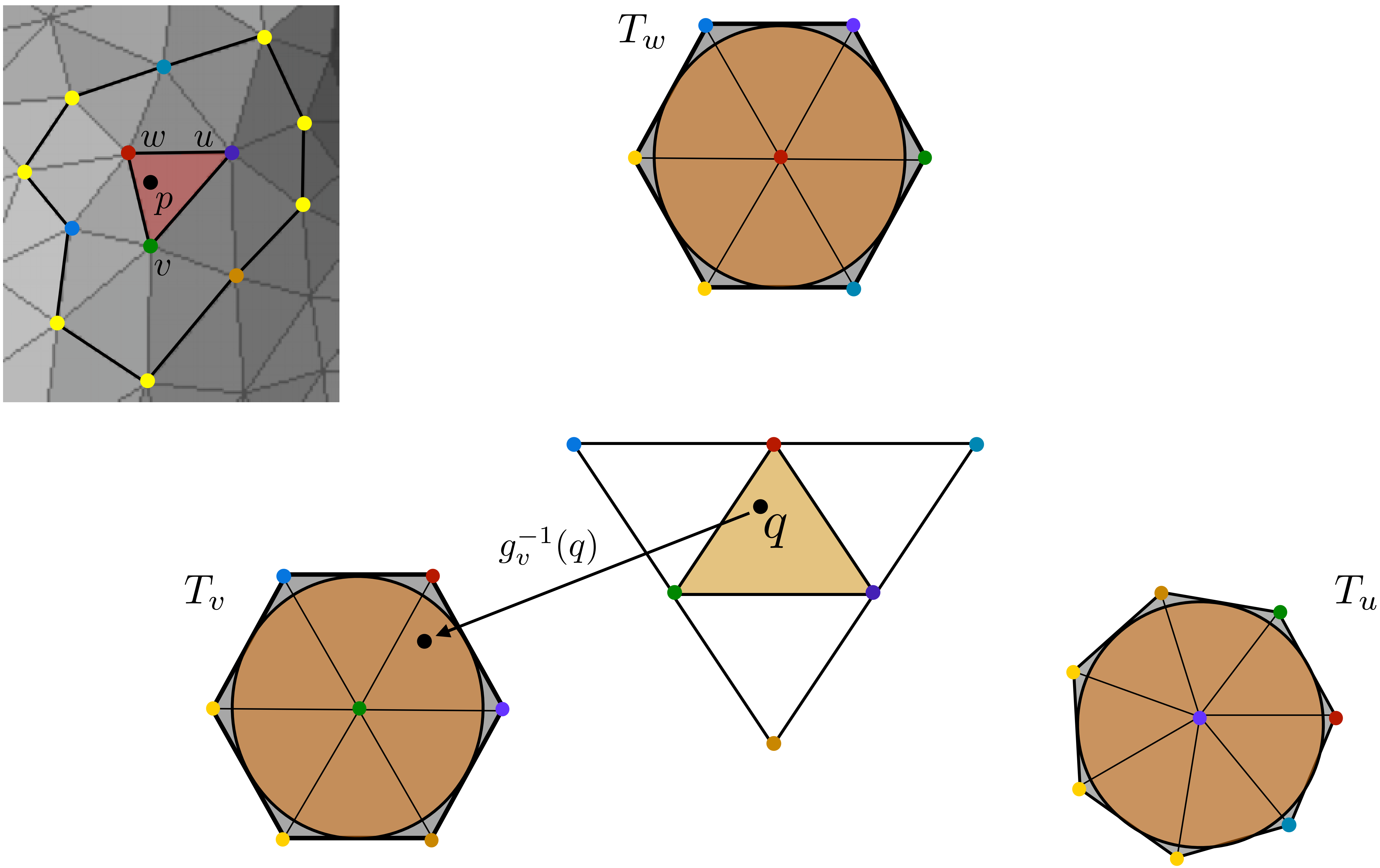


User’s Perspective

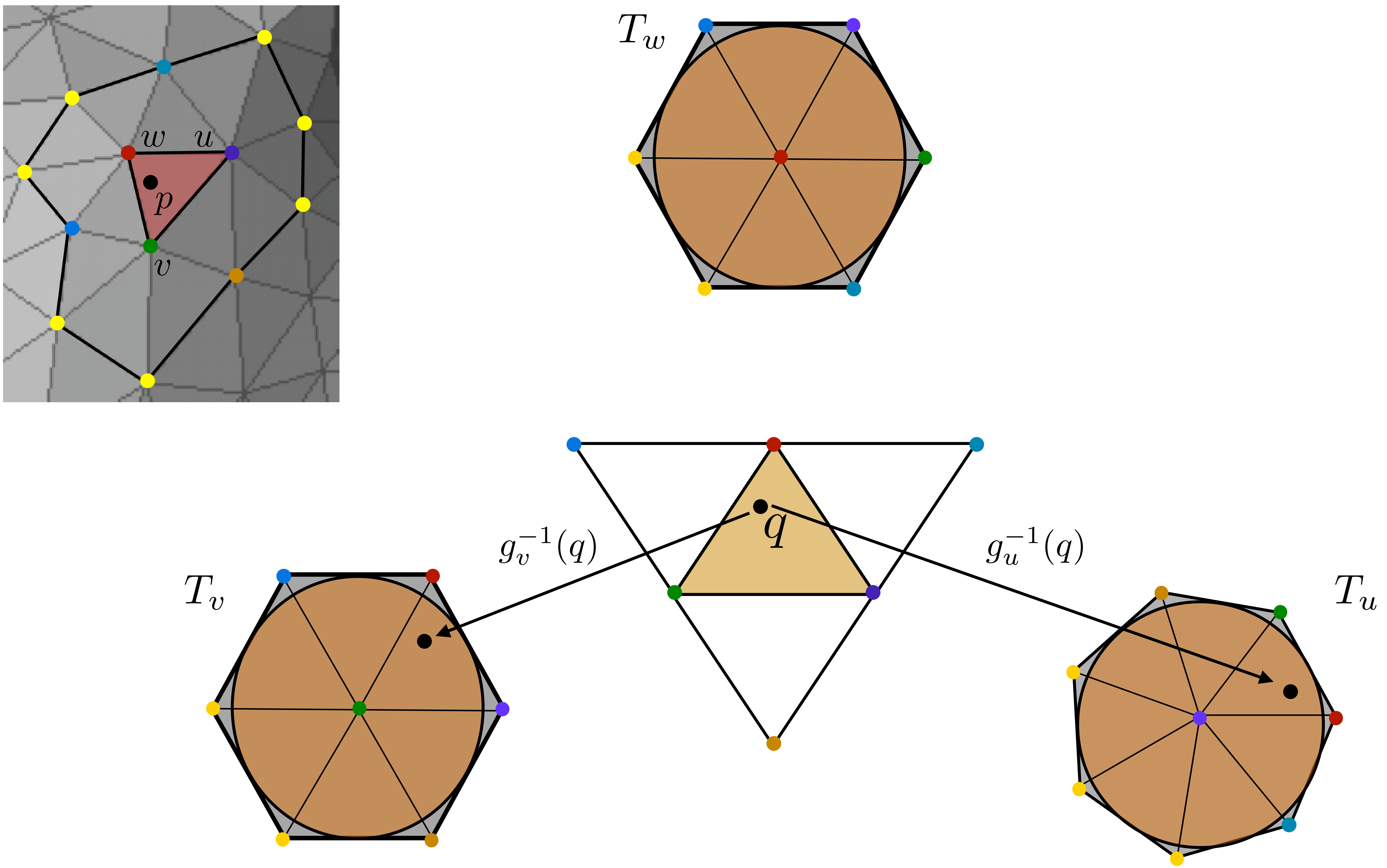
User's Perspective



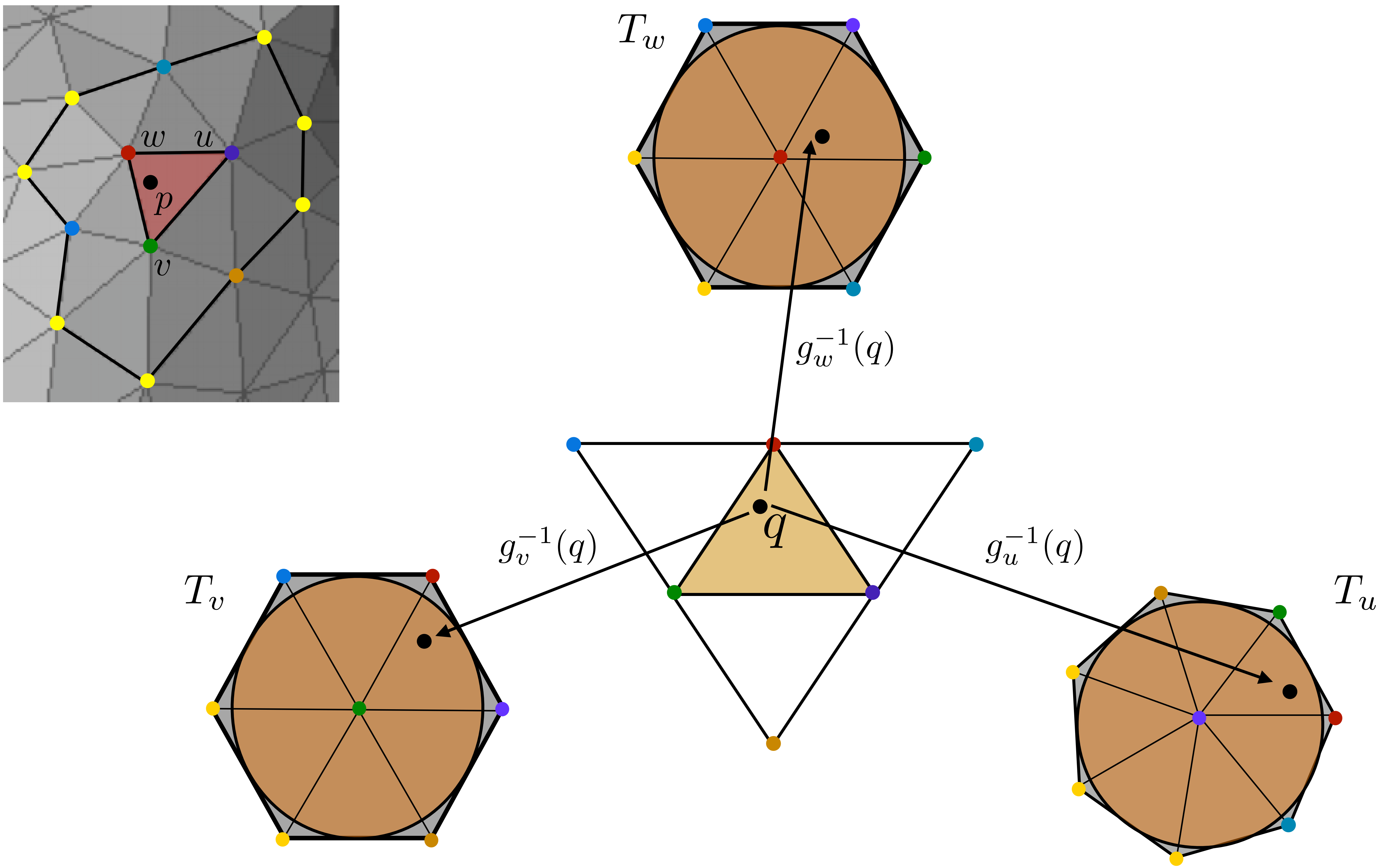
User's Perspective



User's Perspective

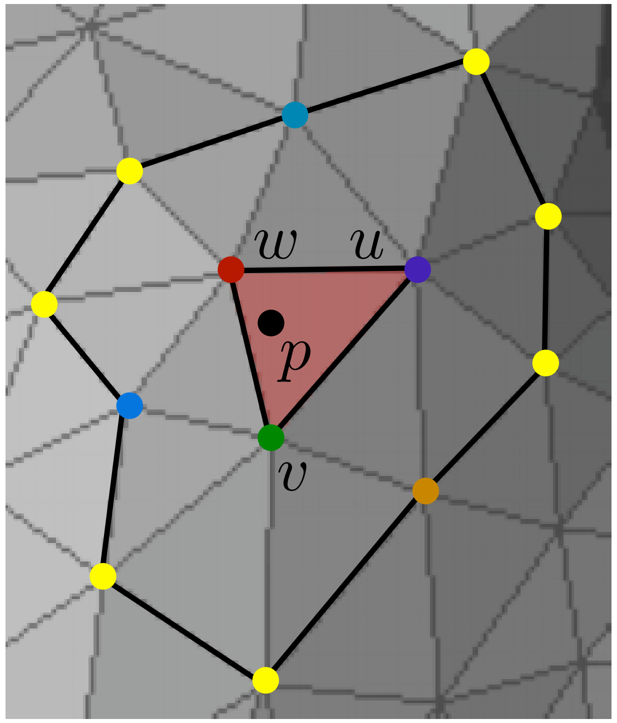


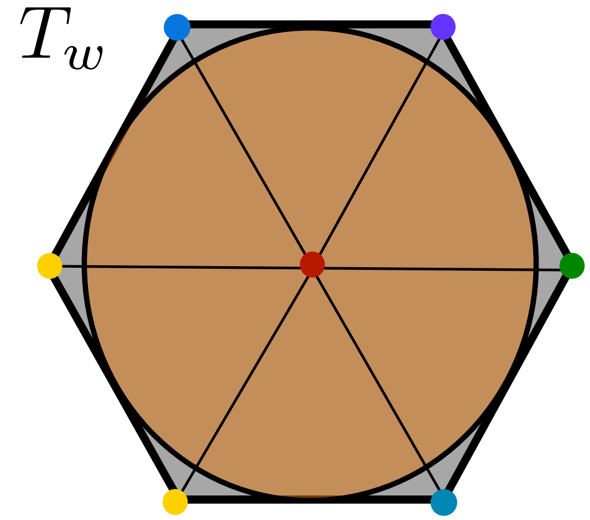
User's Perspective

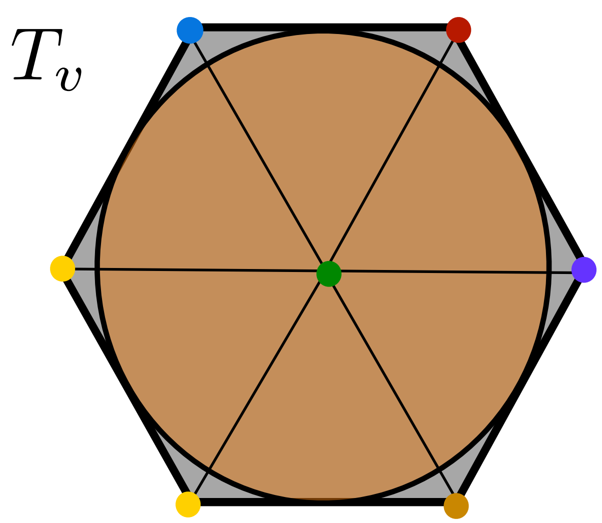


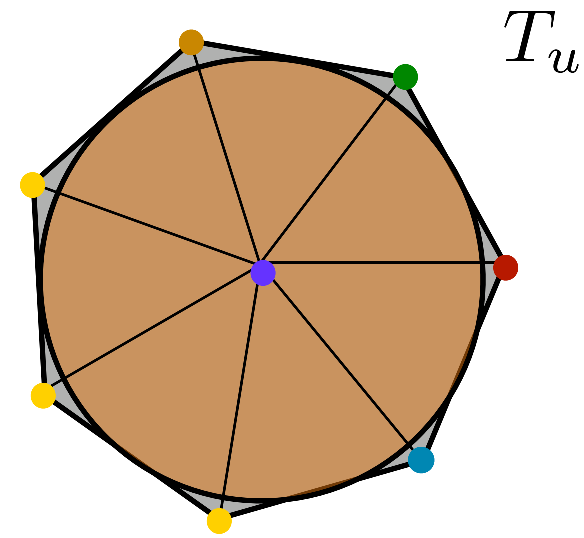
User’s Perspective

User's Perspective

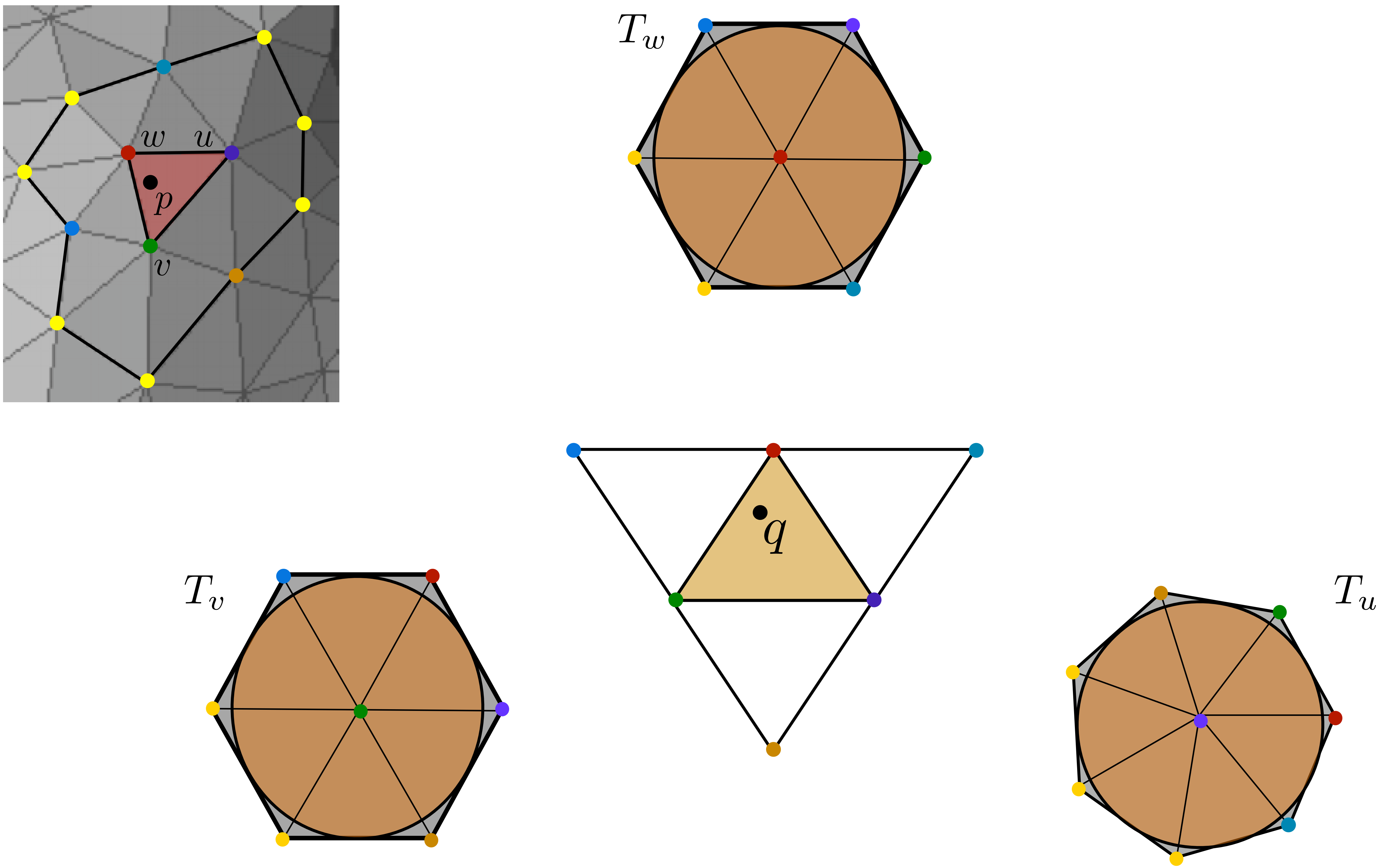




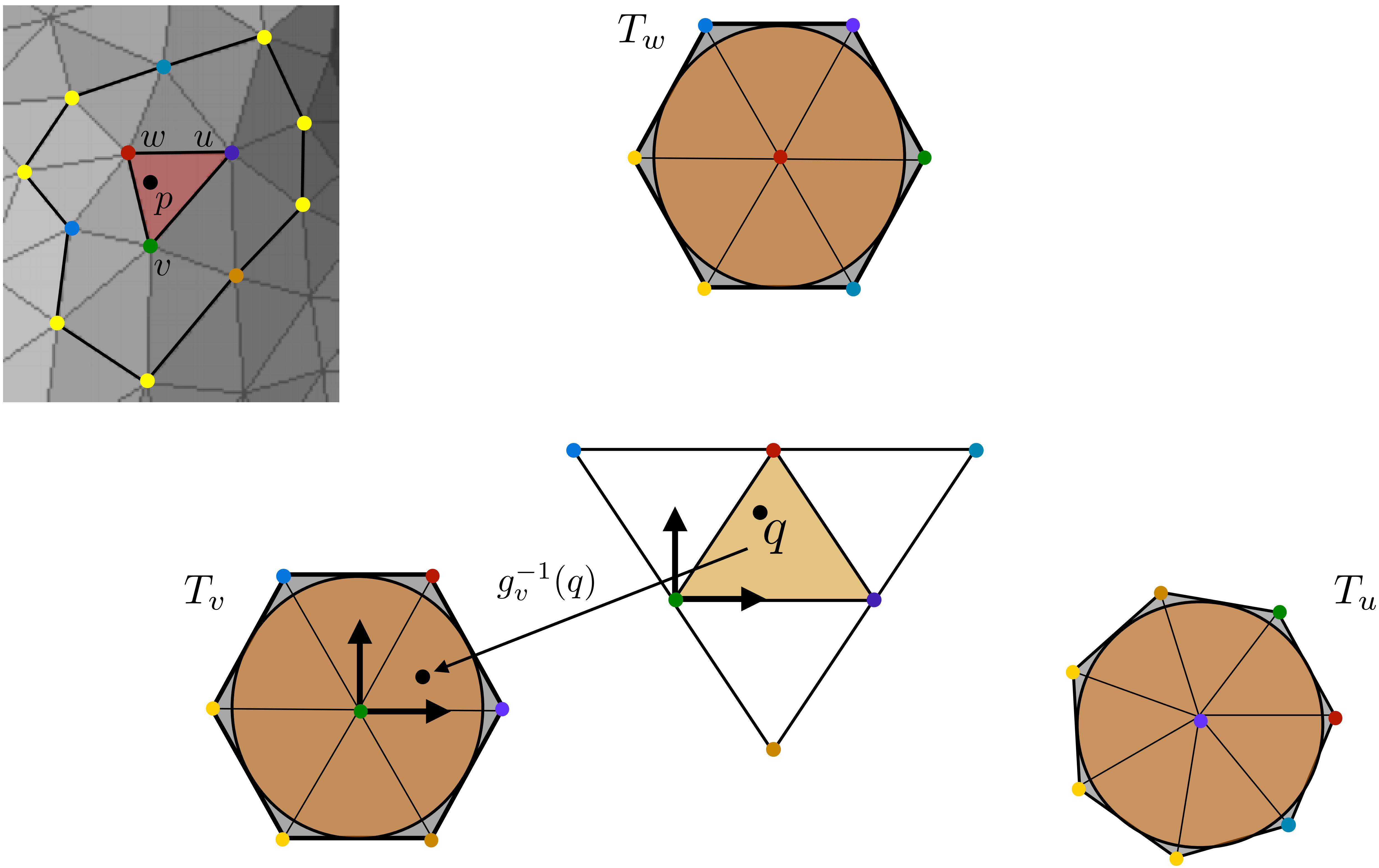




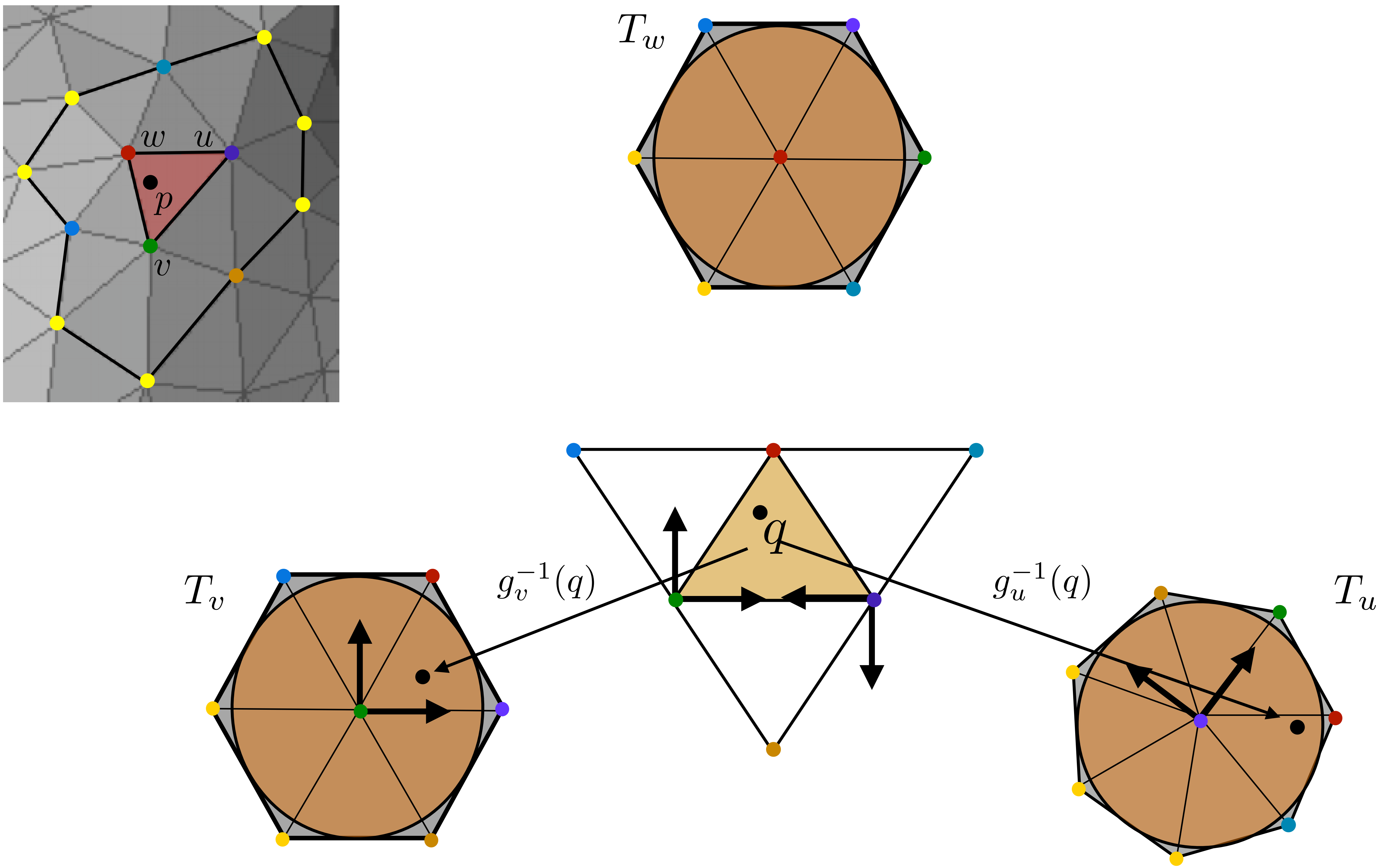
User's Perspective



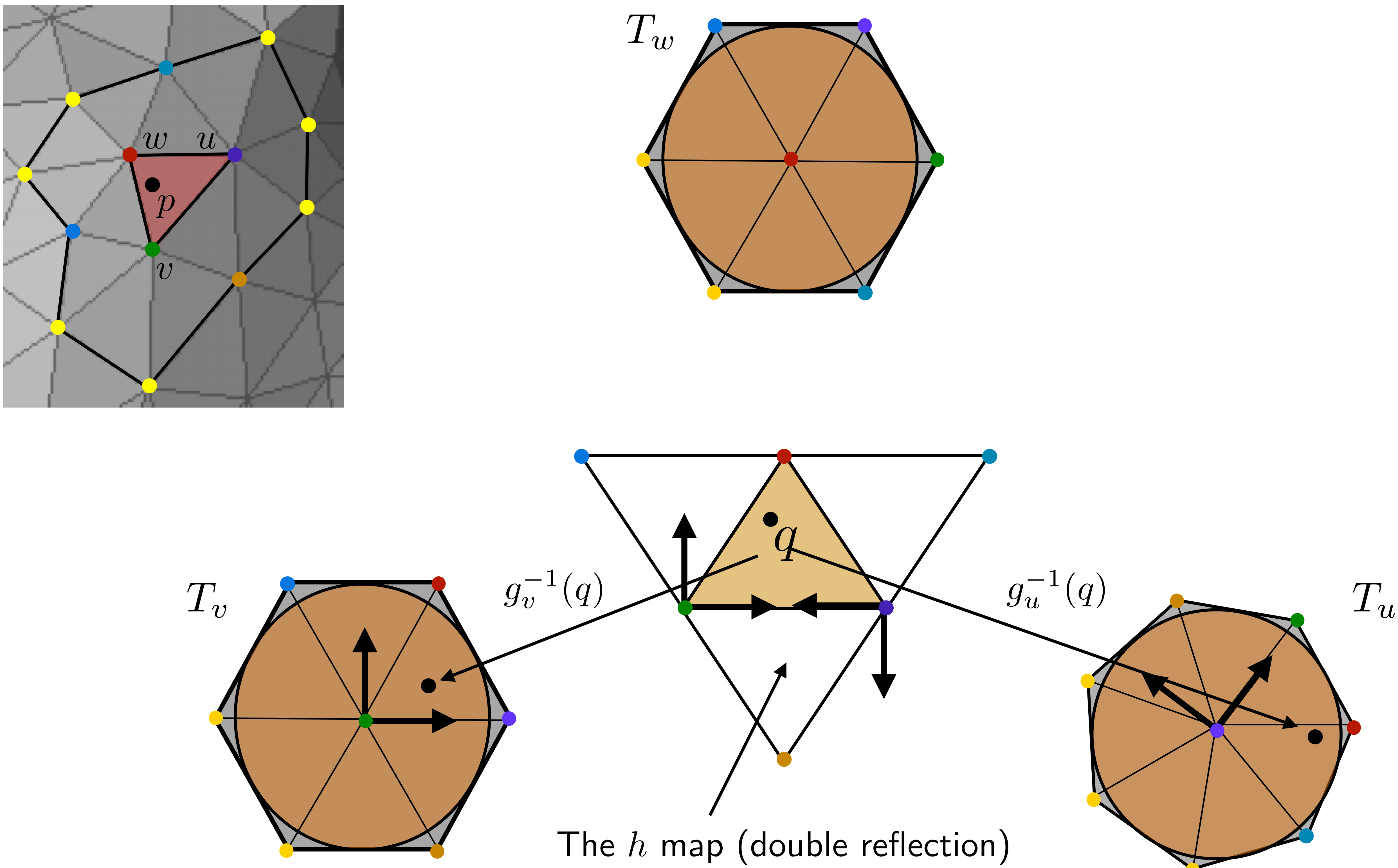
User's Perspective



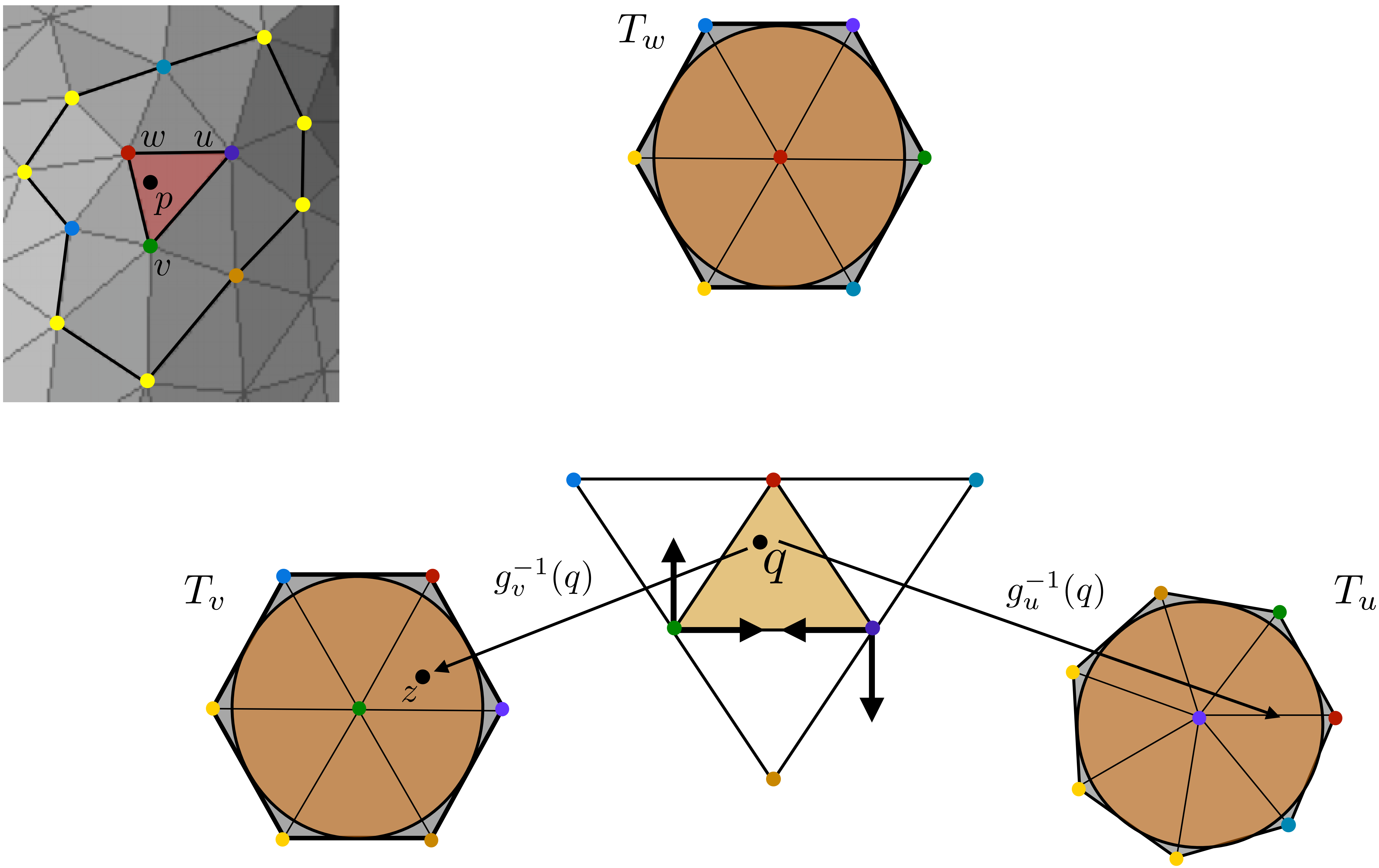
User's Perspective



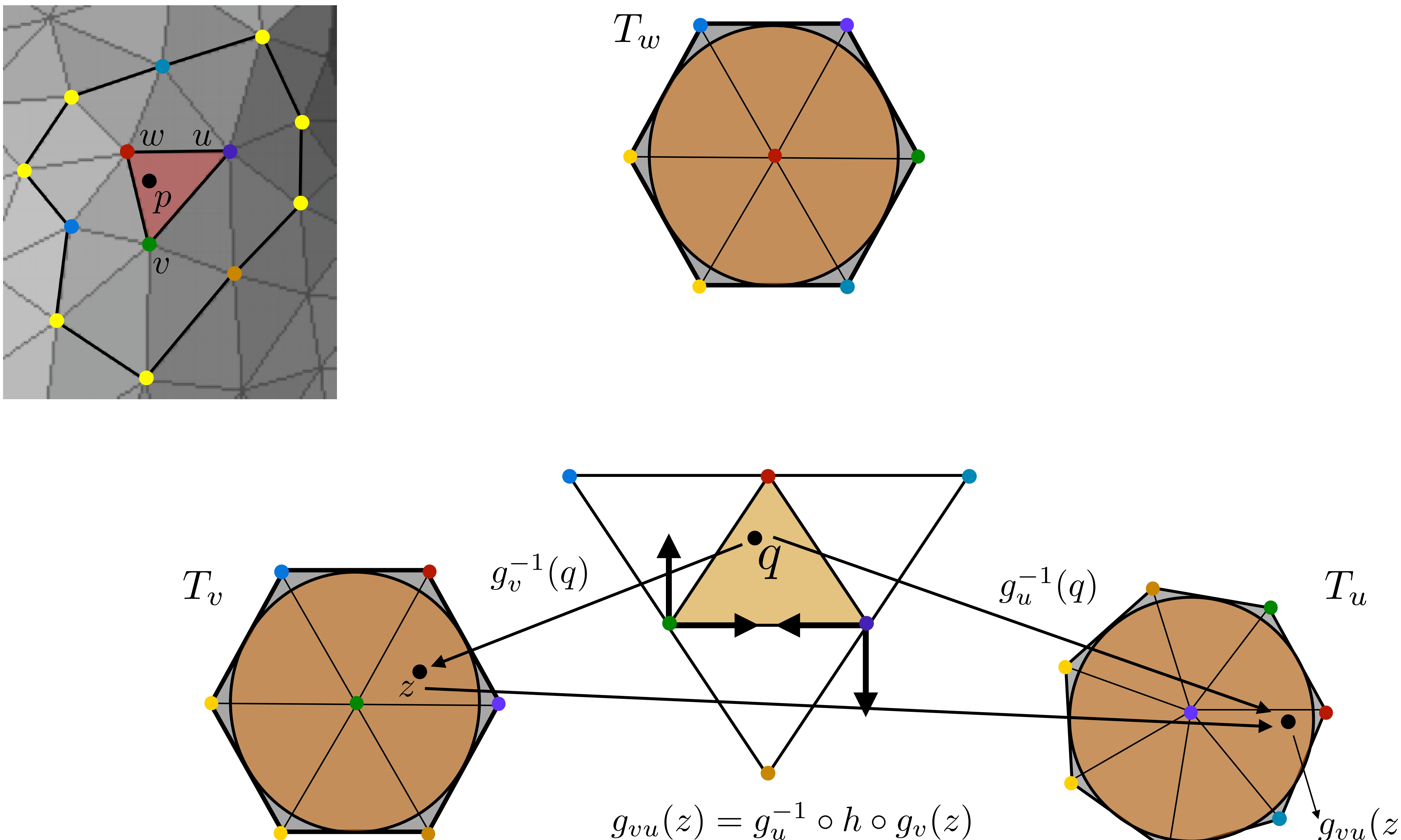
User's Perspective



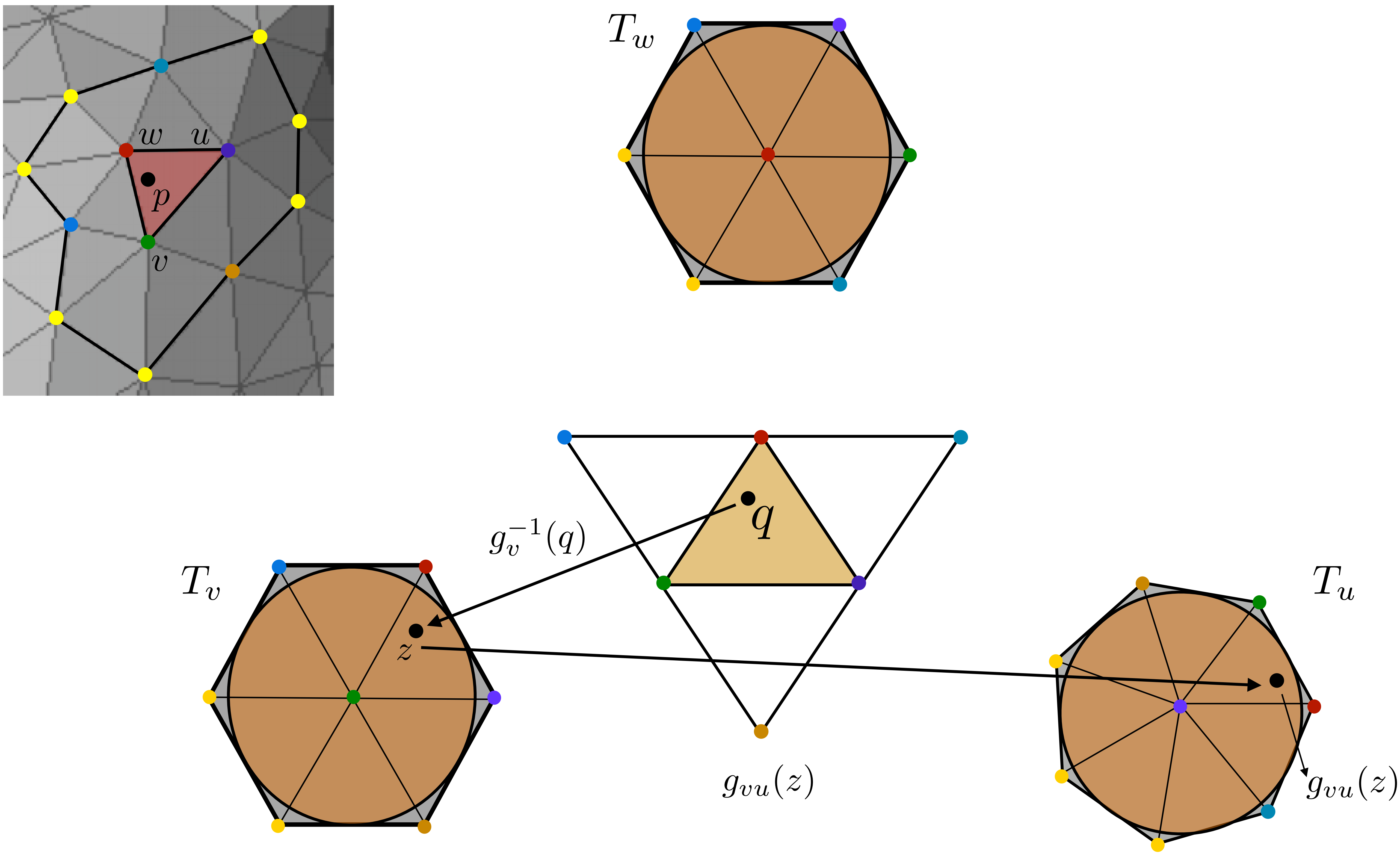
User's Perspective



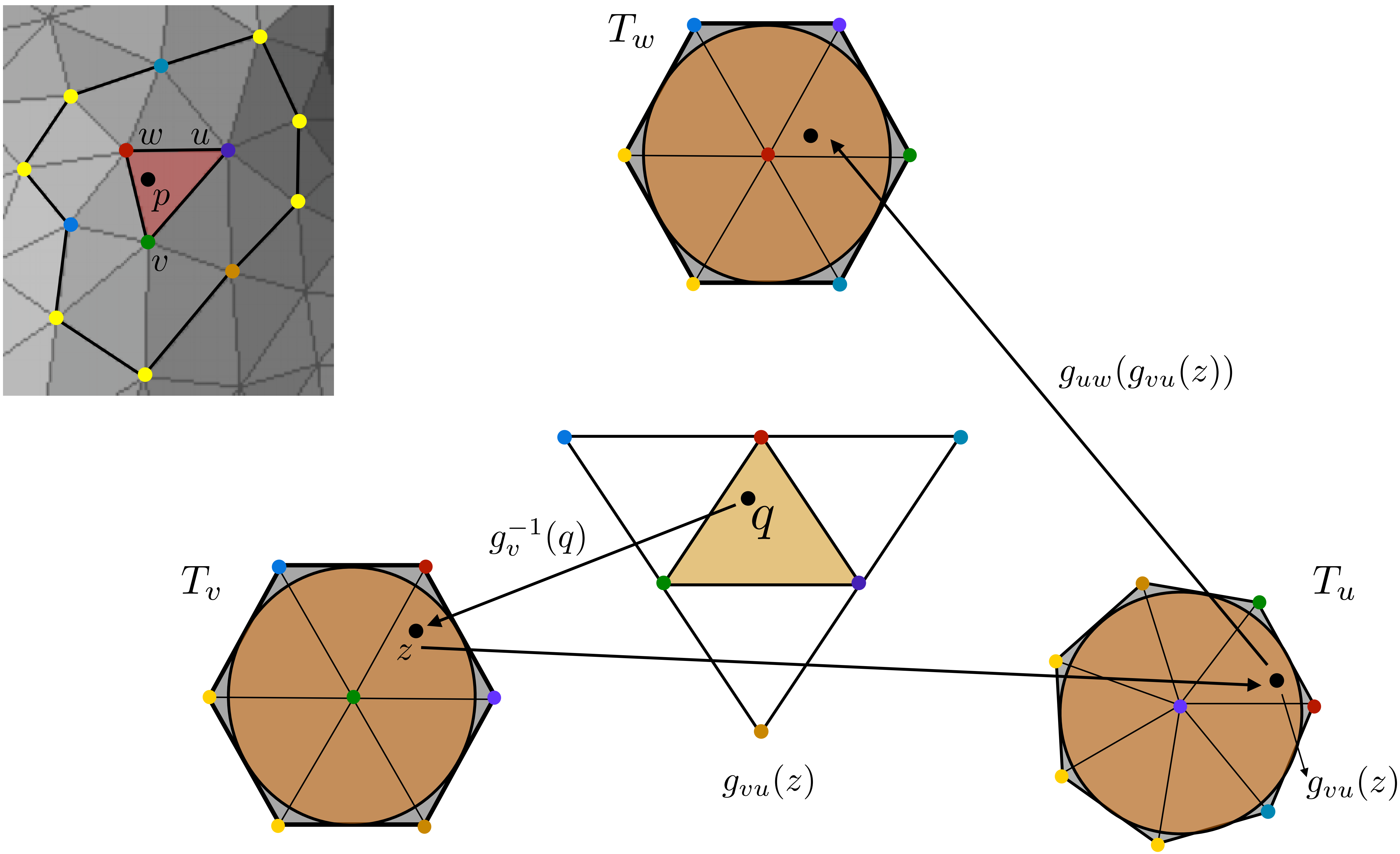
User's Perspective



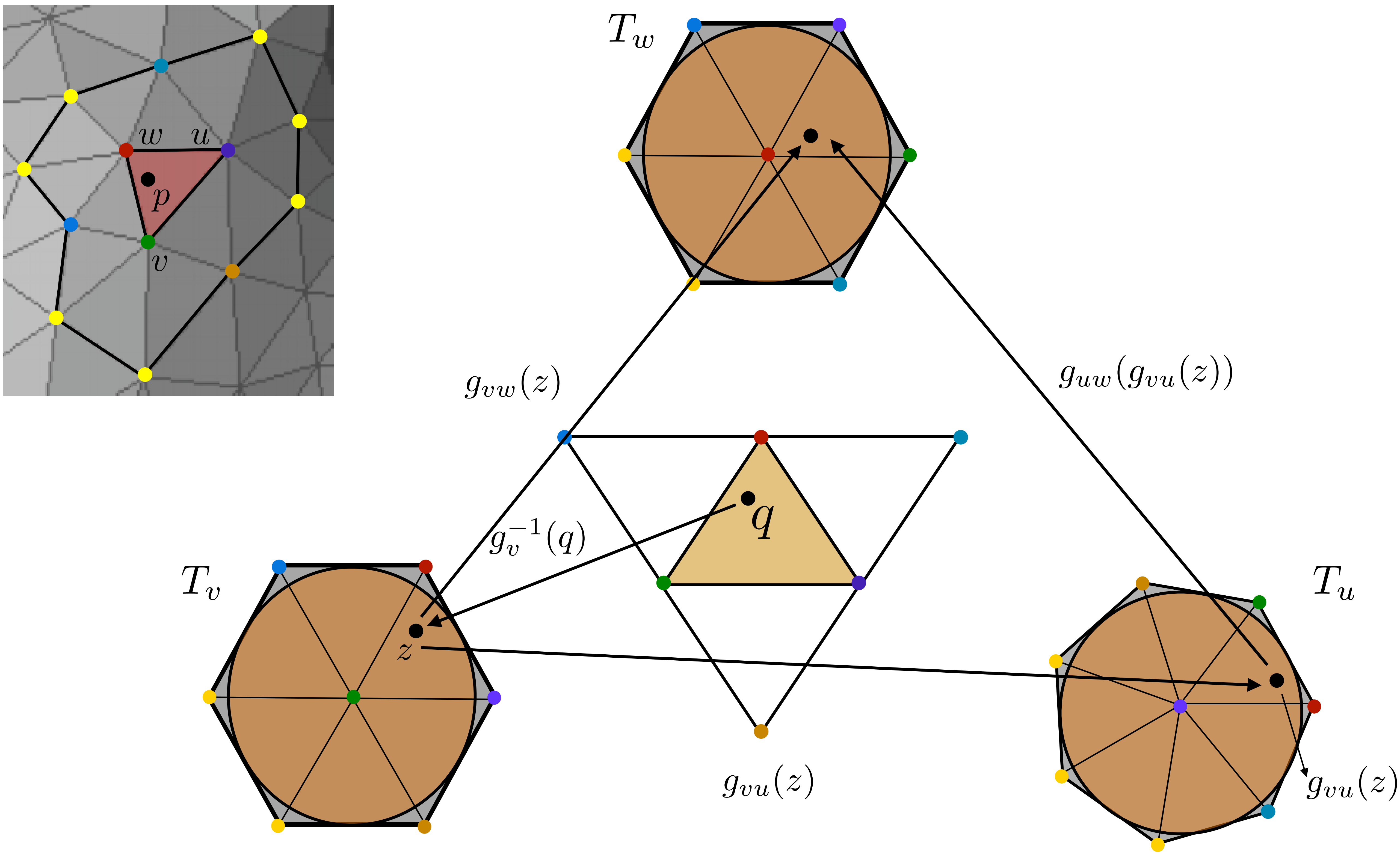
User's Perspective



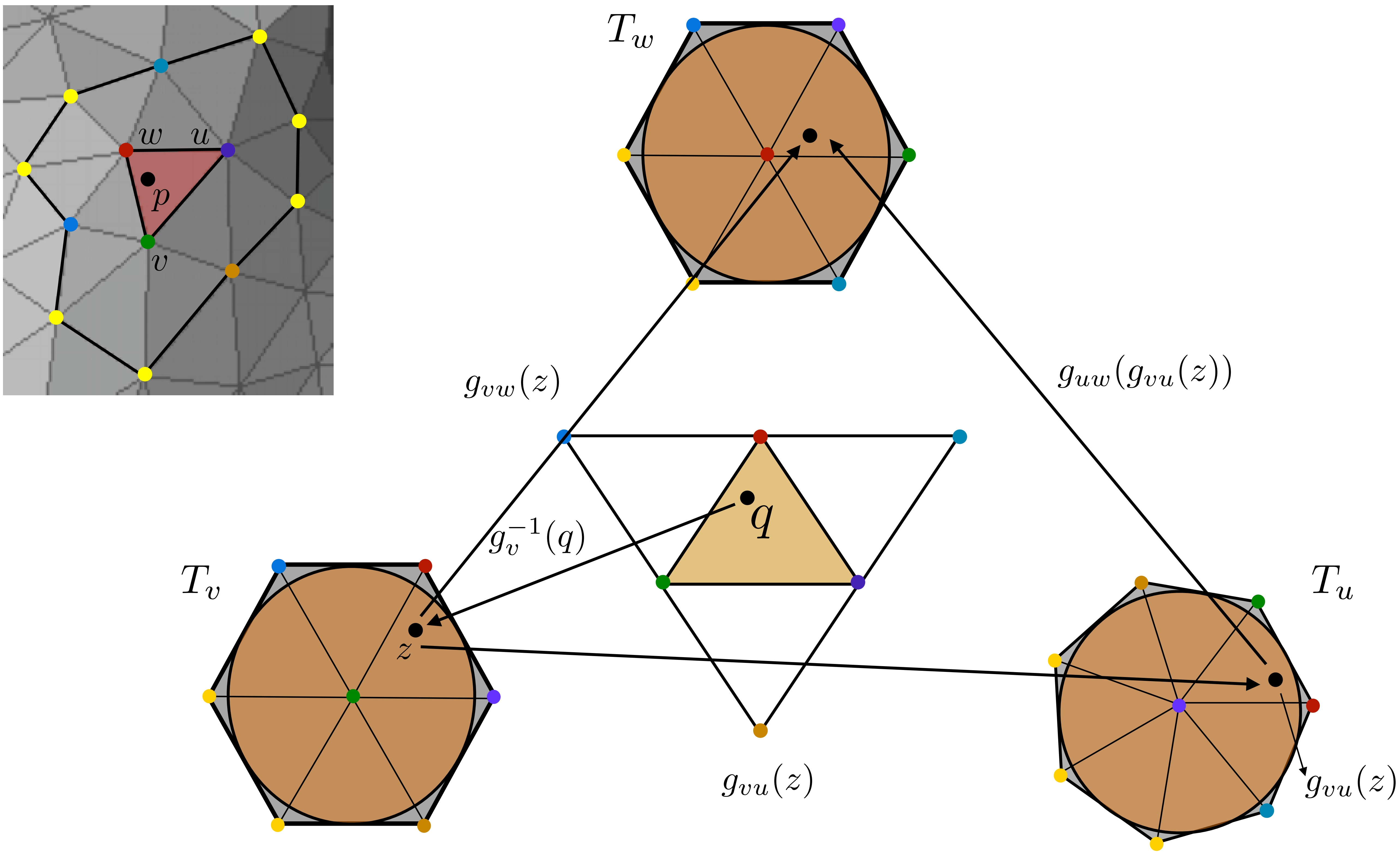
User's Perspective



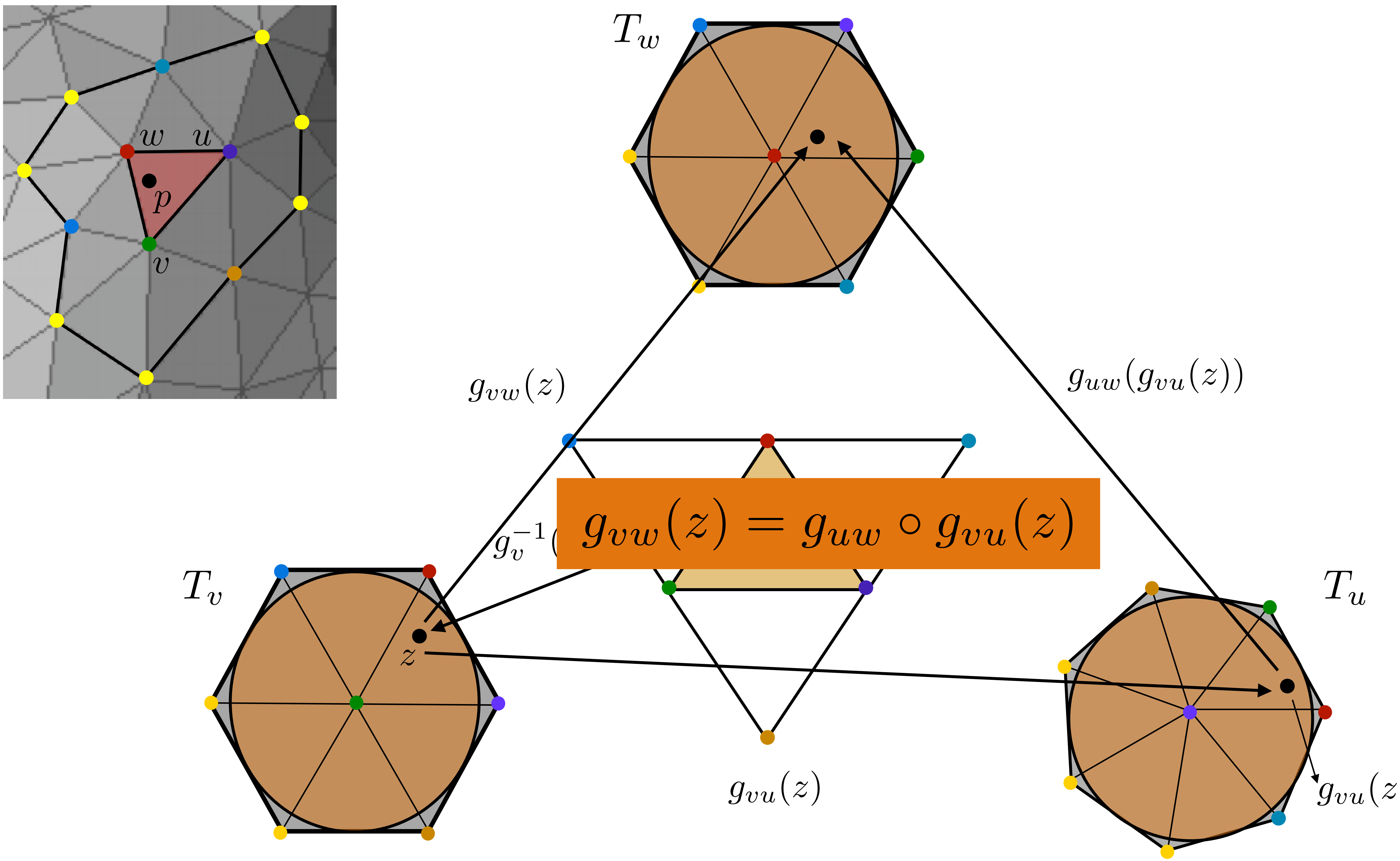
User's Perspective



User's Perspective



User’s Perspective



$$g_{vw}(z) = g_{uw} \circ g_{vu}(z)$$

Building Parametrizations

Building Parametrizations

For each $(\sigma, v) \in I$, we define a **weight function**,

$$\gamma_{(\sigma,v)} : \mathbb{R}^2 \to \mathbb{R}\,,$$

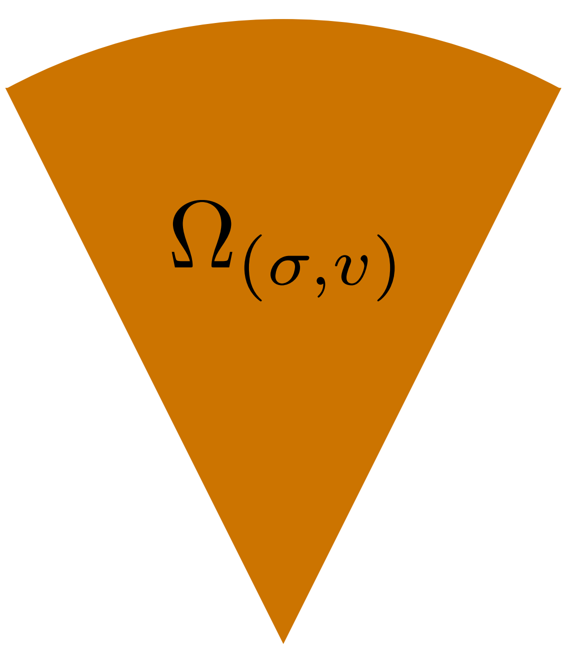
which is the product of two C^∞ curves (and therefore, C^∞ too).

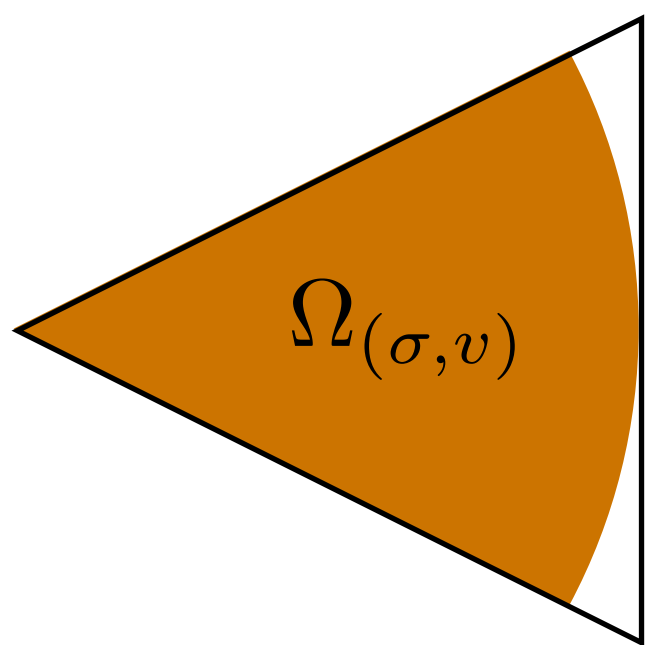
Building Parametrizations

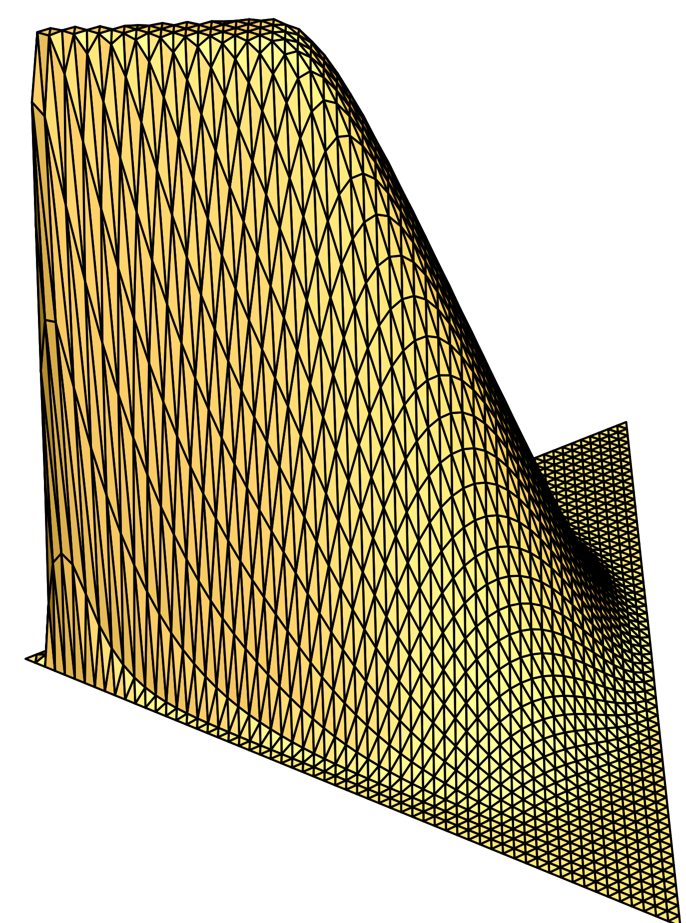
For each $(\sigma, v) \in I$, we define a **weight function**,

$$\gamma_{(\sigma,v)} : \mathbb{R}^2 \to \mathbb{R}\,,$$

which is the product of two C^∞ curves (and therefore, C^∞ too).

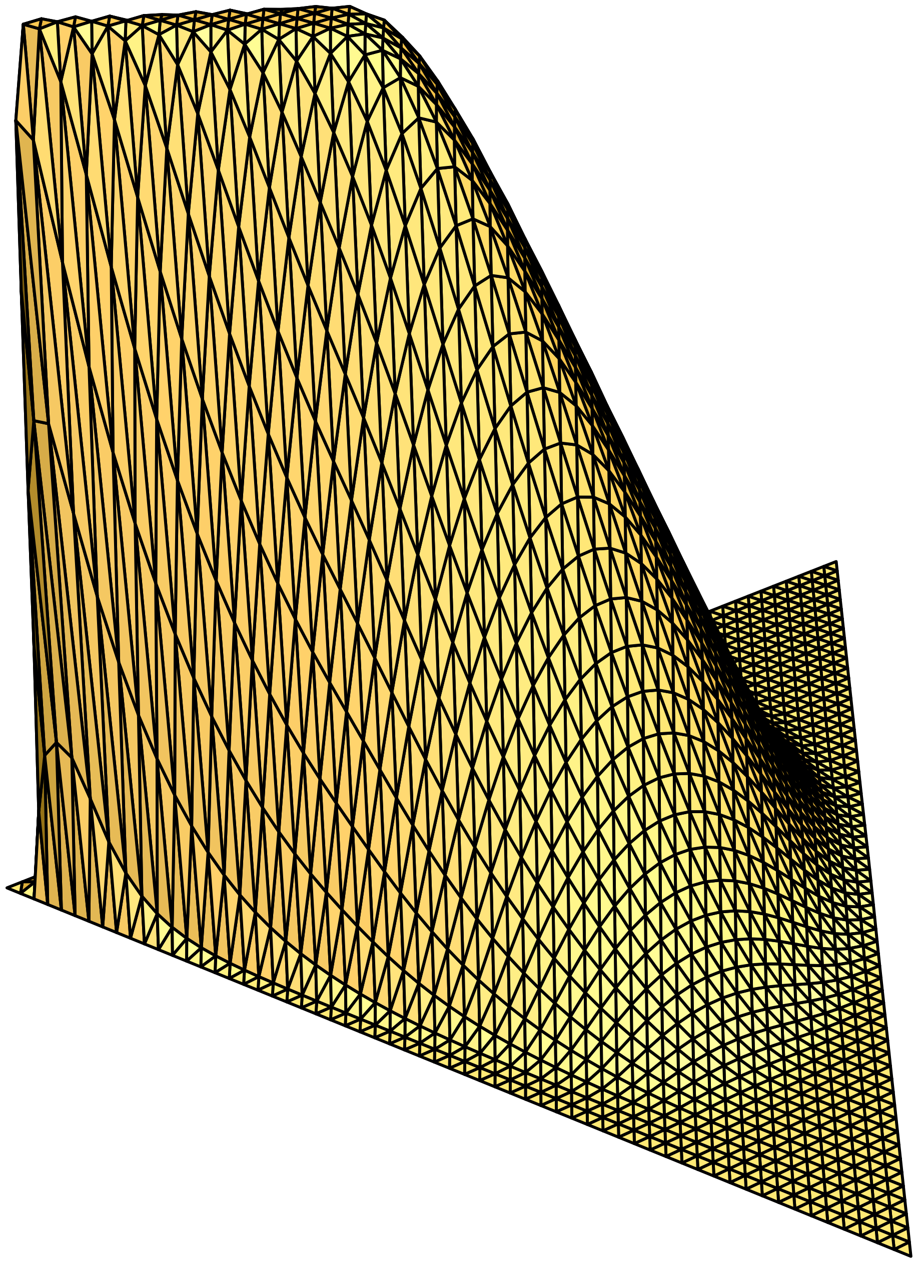




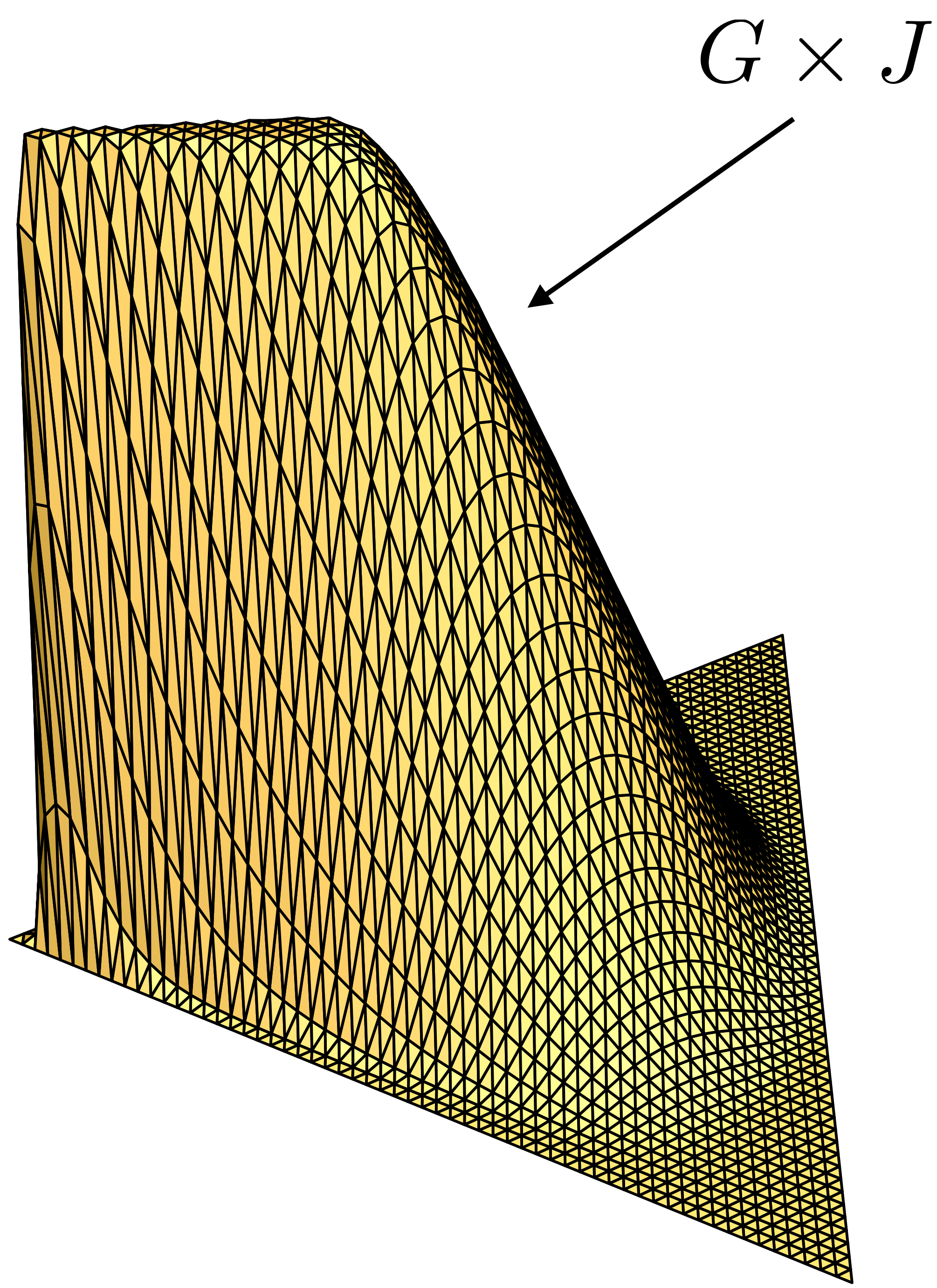


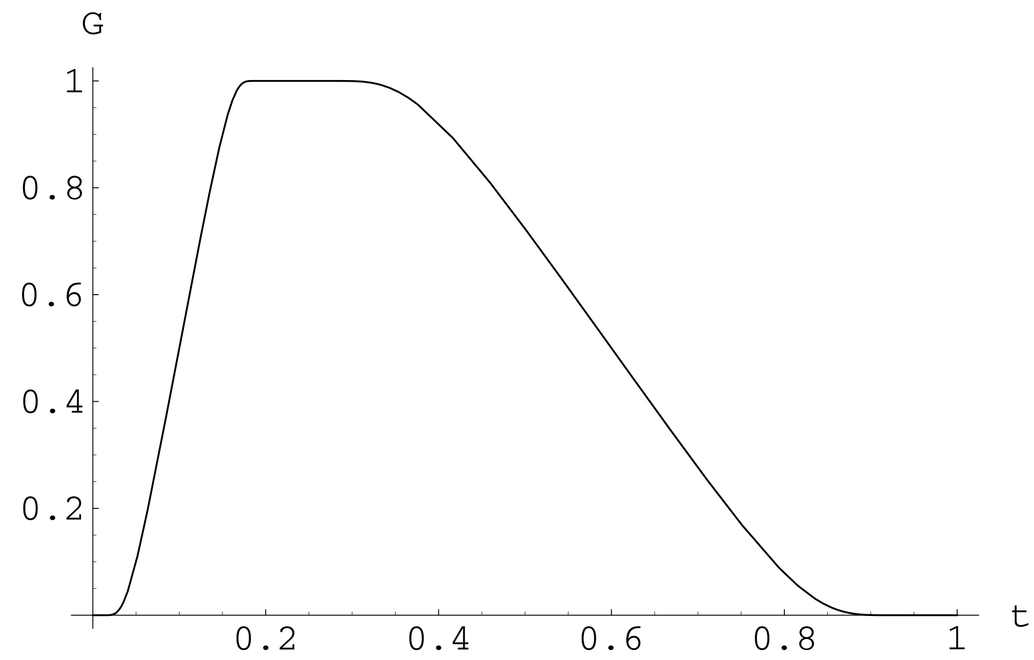
Building Parametrizations

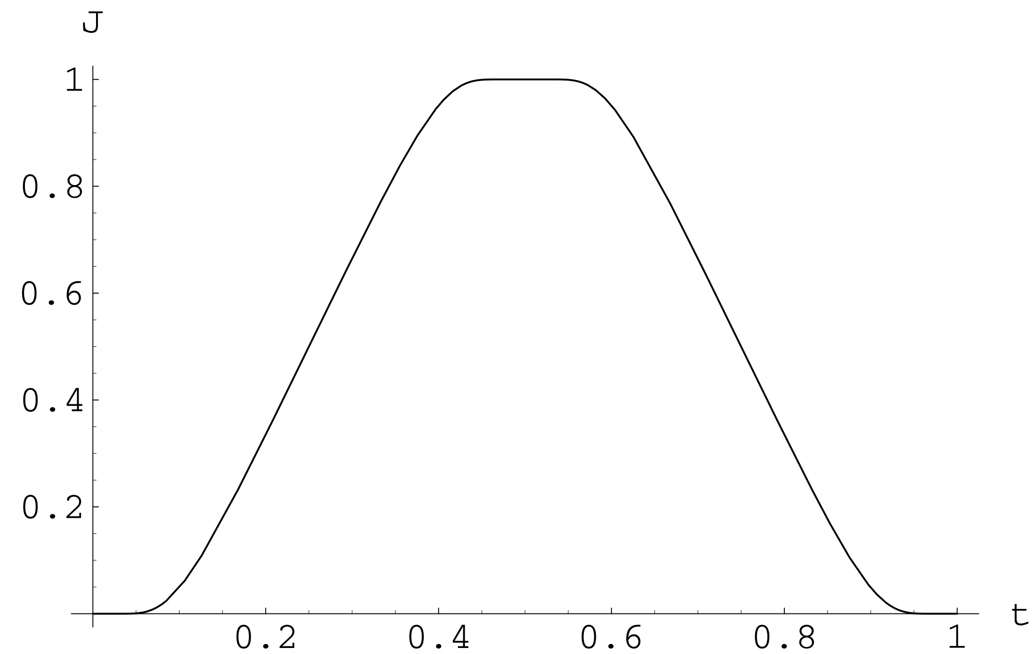
Building Parametrizations



Building Parametrizations







Building Parametrizations

Building Parametrizations

For each $(\sigma, v) \in I$, we define a **Bézier patch**,

$$\psi_{(\sigma,v)} : \mathbb{R}^2 \to \mathbb{R}^3 \,,$$

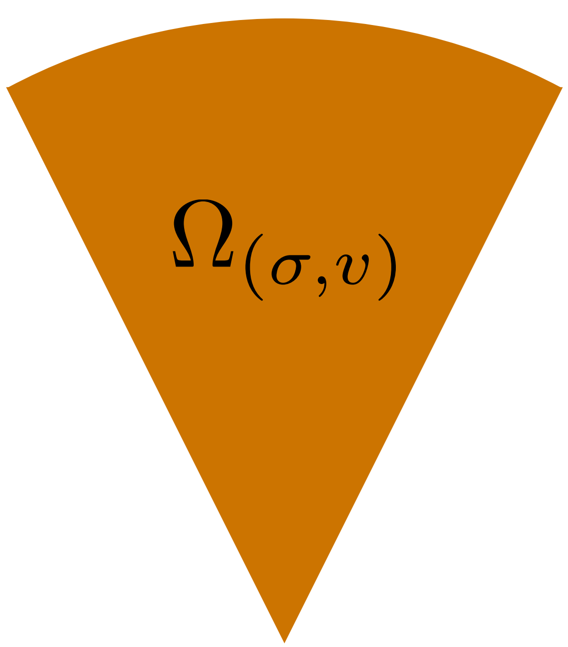
whose control points are defined in the "envelope" triangle of $\Omega_{(\sigma,v)}$.

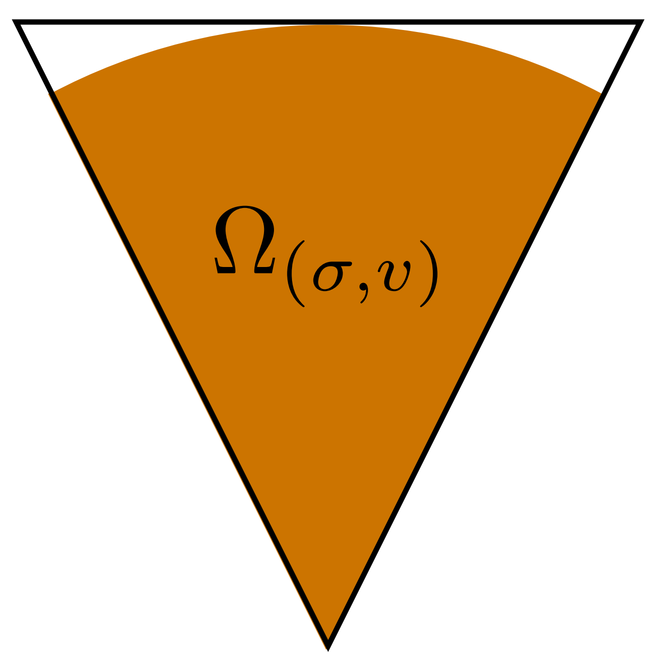
Building Parametrizations

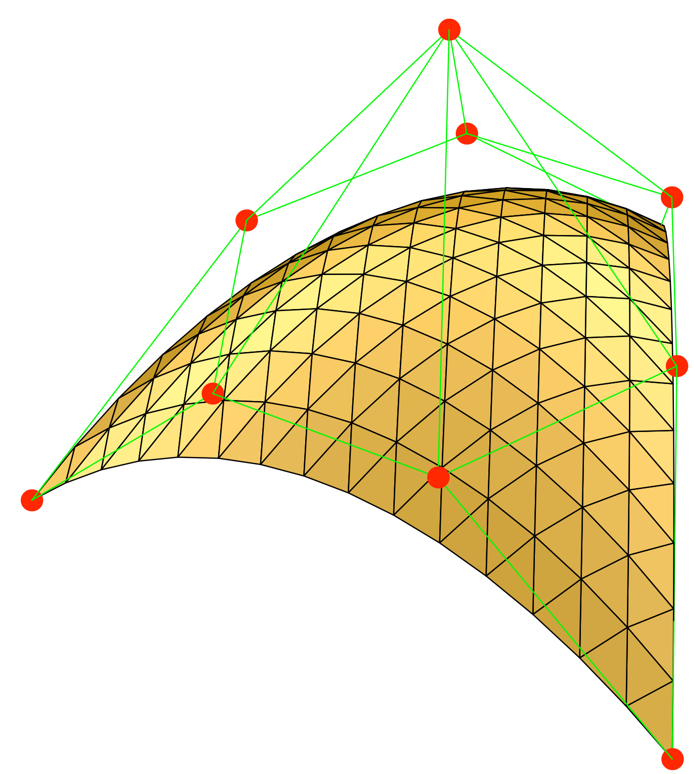
For each $(\sigma, v) \in I$, we define a **Bézier patch**,

$$\psi_{(\sigma,v)} : \mathbb{R}^2 \to \mathbb{R}^3 \,,$$

whose control points are defined in the "envelope" triangle of $\Omega_{(\sigma,v)}$.

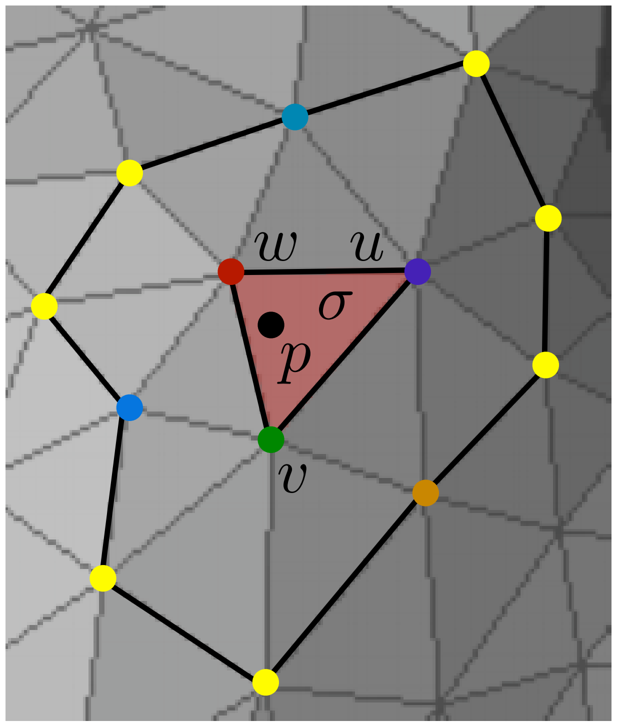




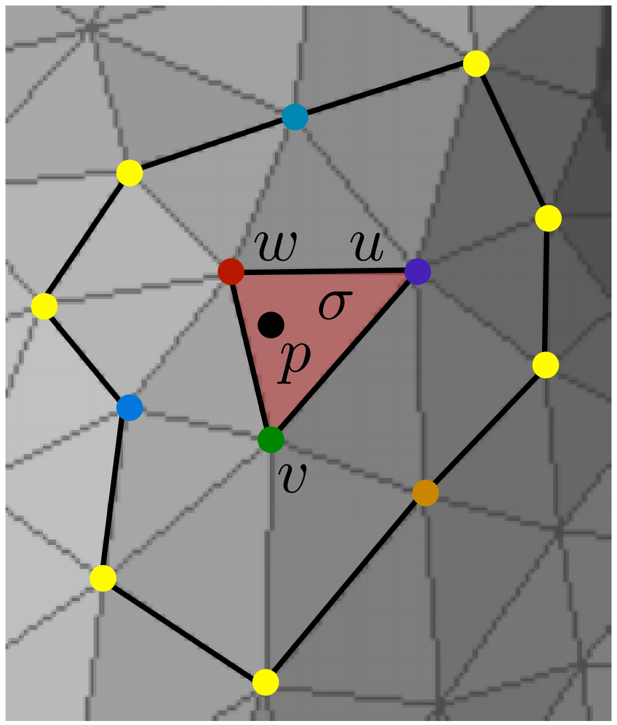


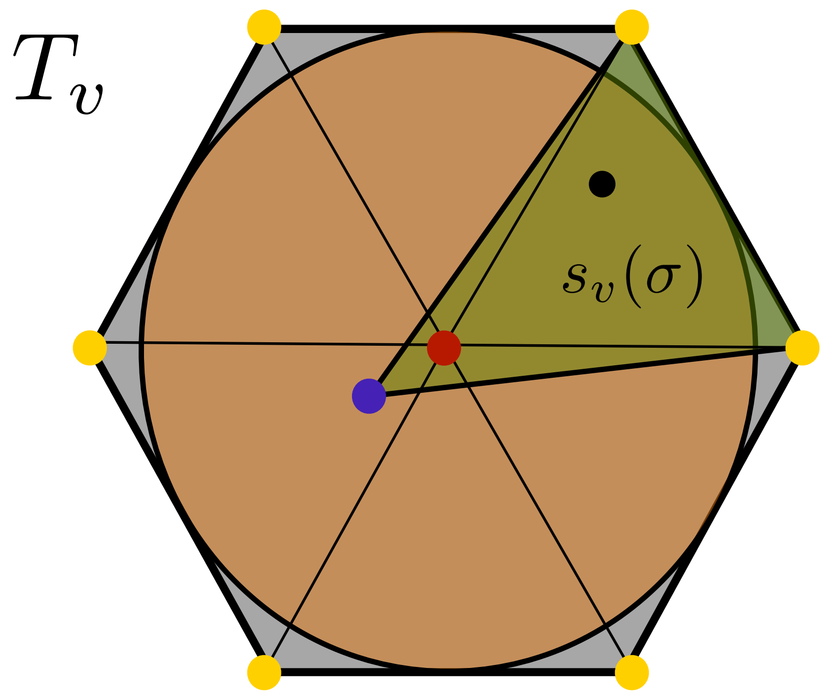
Building Parametrizations

Building Parametrizations

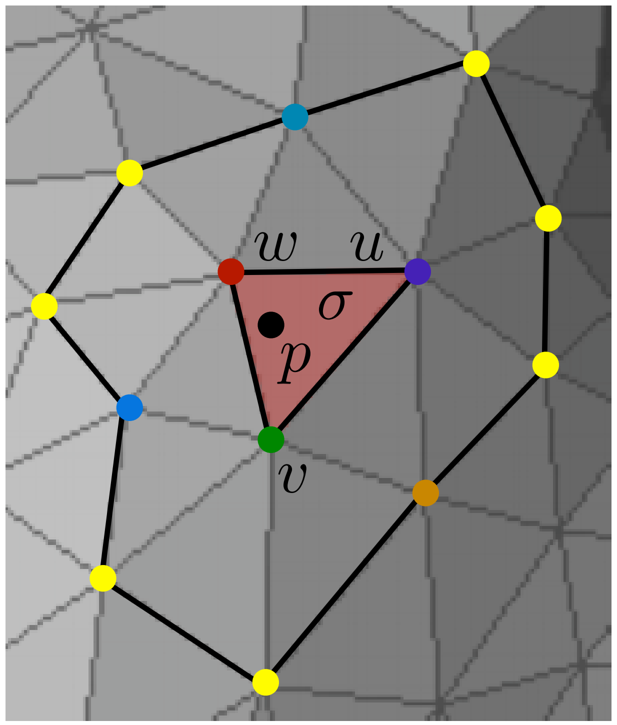


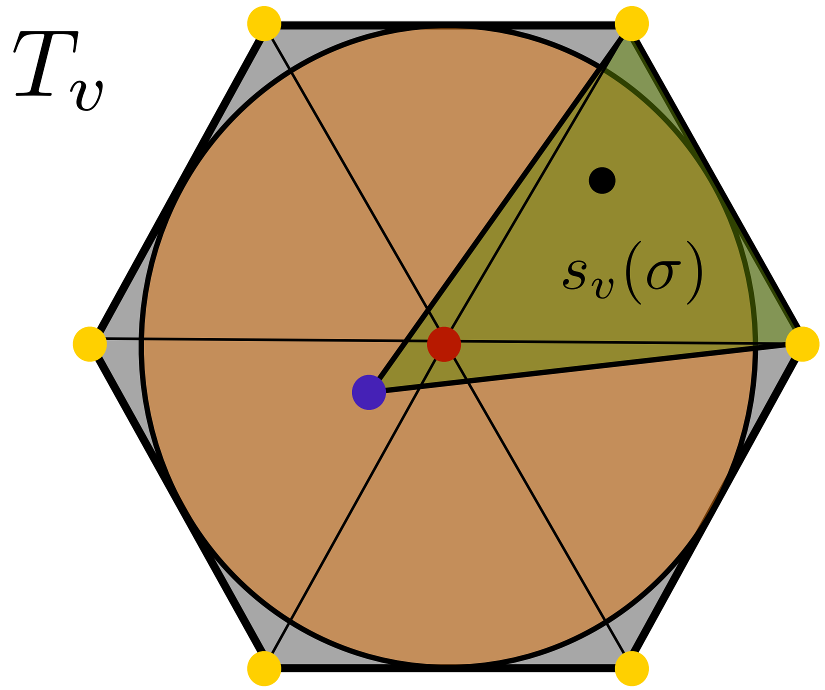
Building Parametrizations

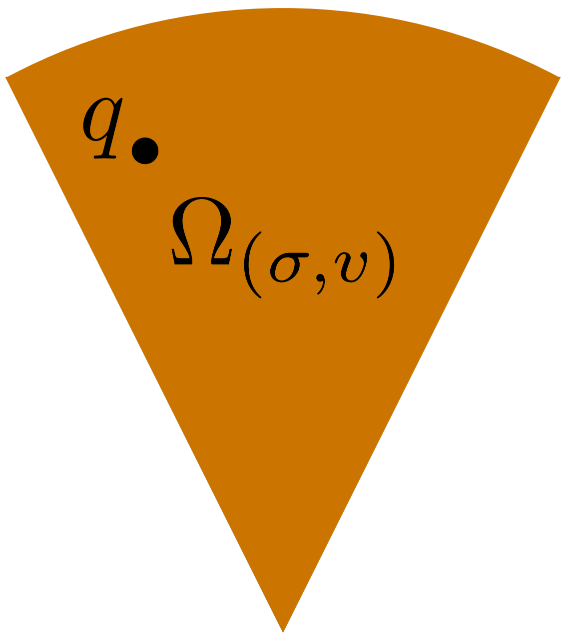




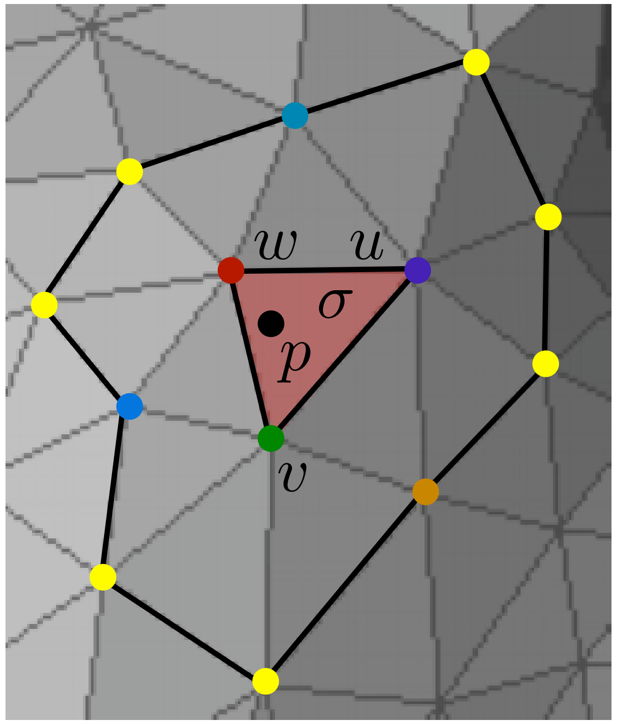
Building Parametrizations

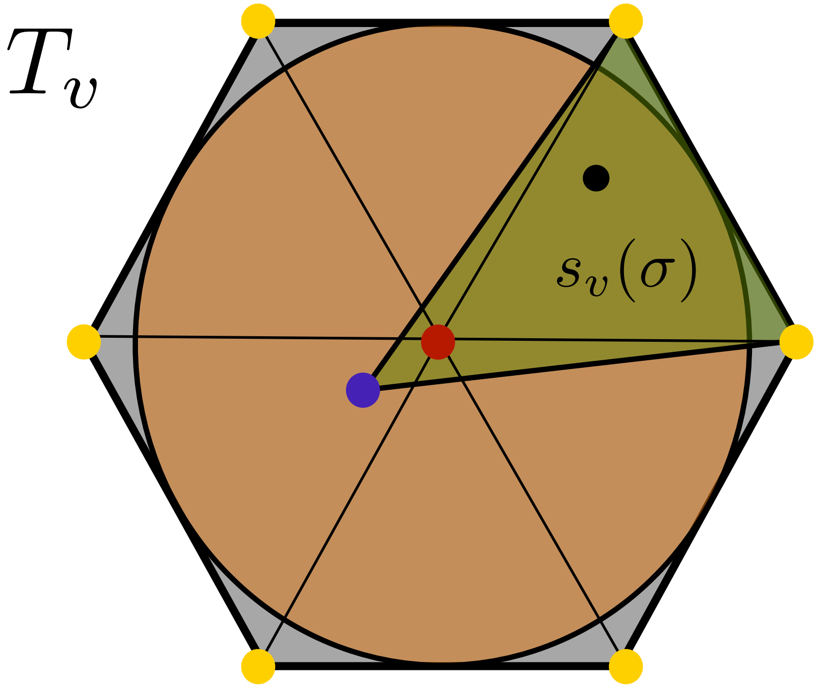


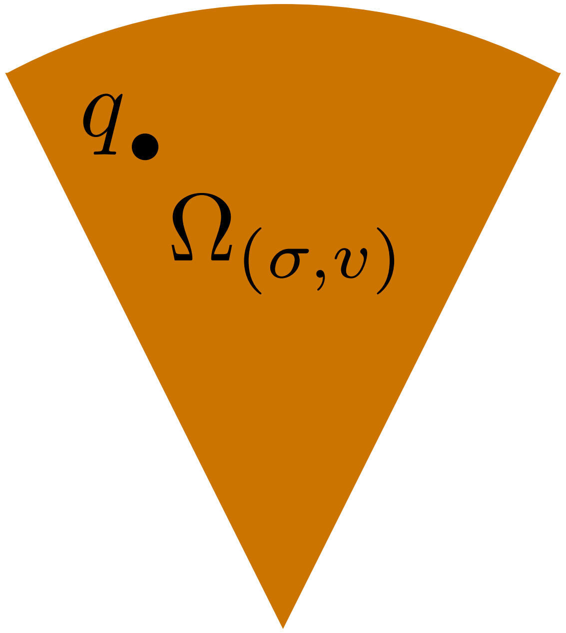


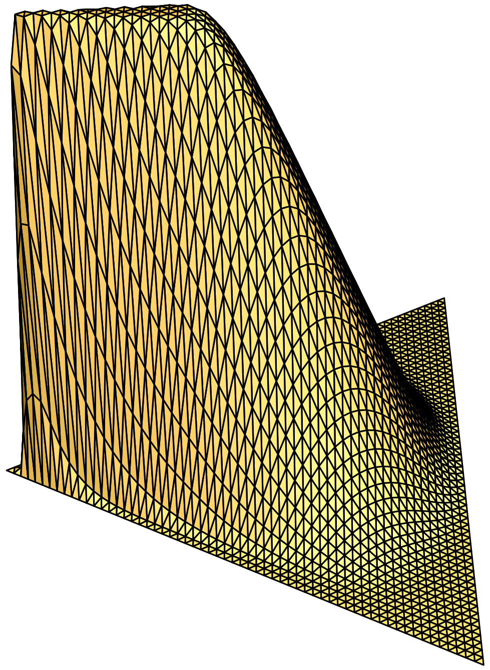


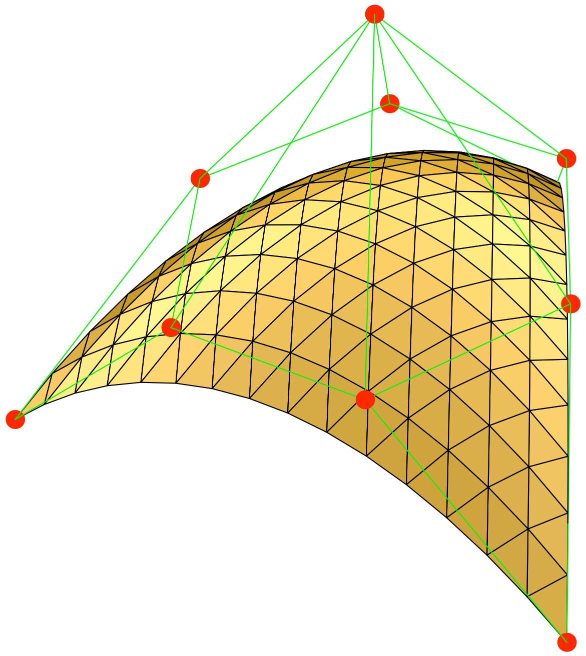
Building Parametrizations



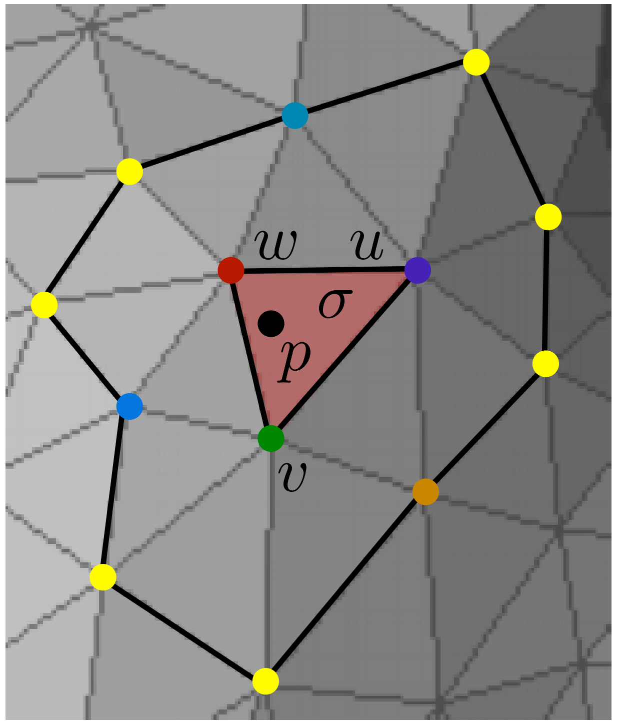


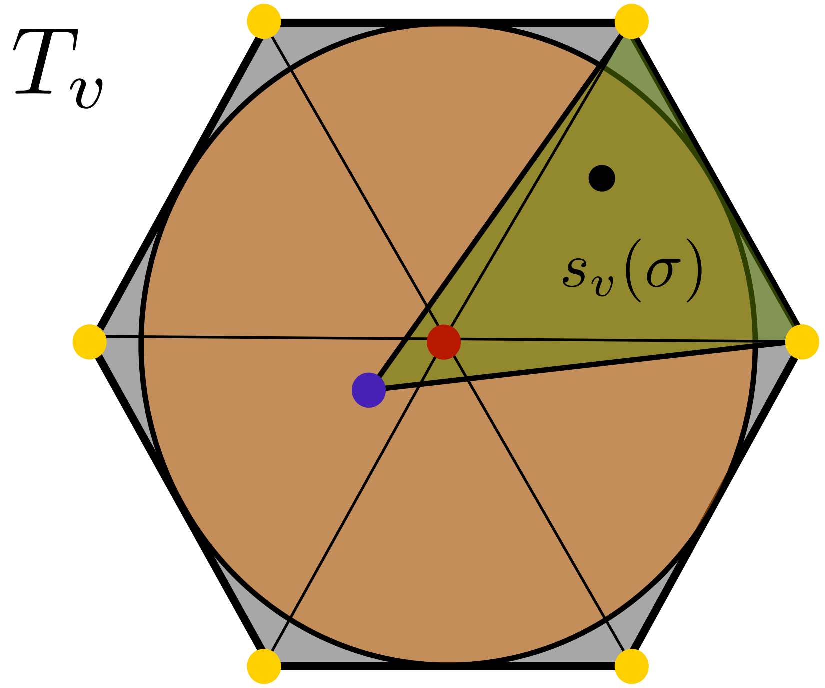


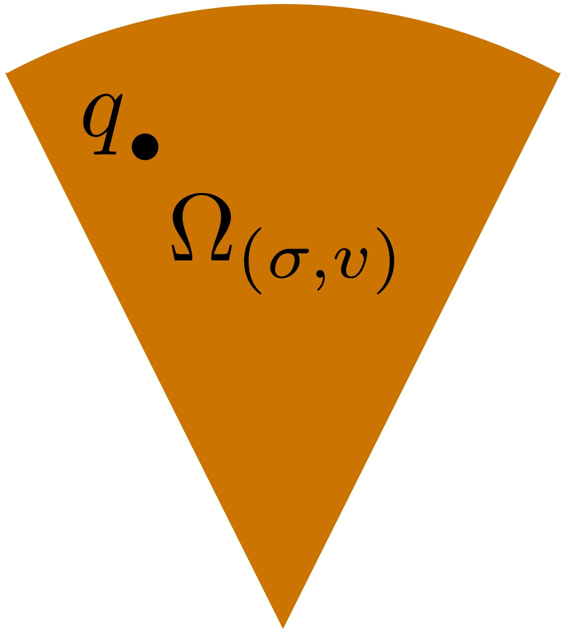


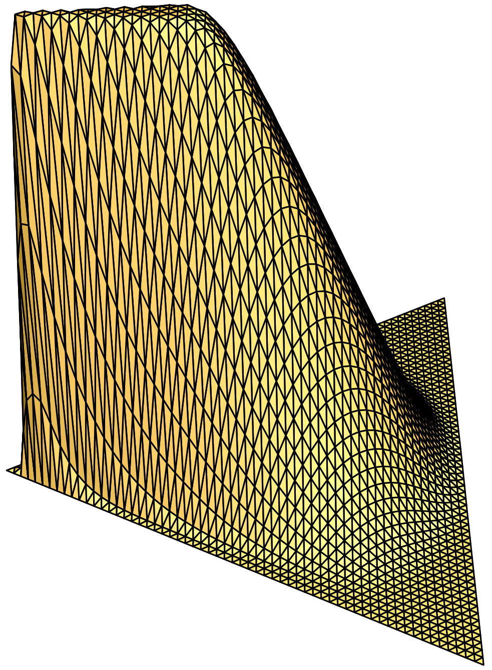


Building Parametrizations



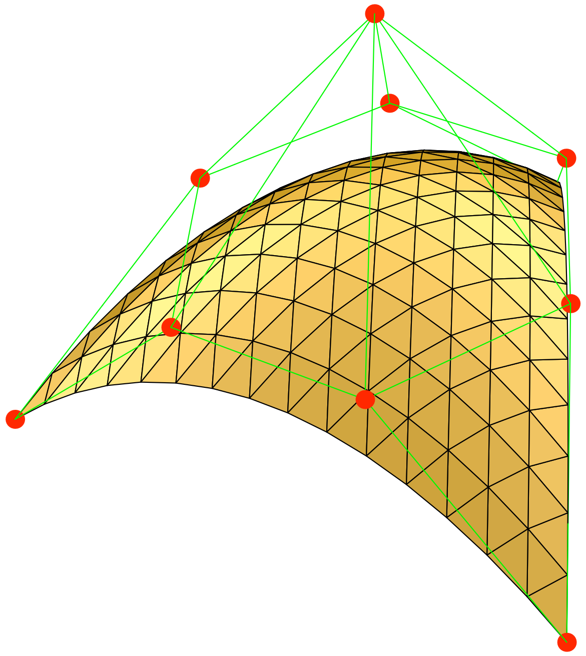






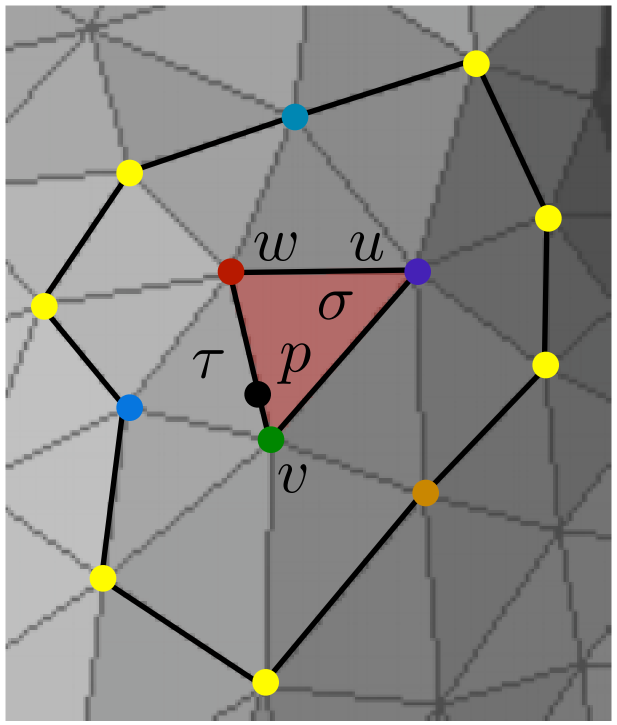
Contribution of $\Omega_{(\sigma,v)}$:

$$\gamma_{(\sigma,v)}(q) \cdot \psi_{(\sigma,v)}(q)$$

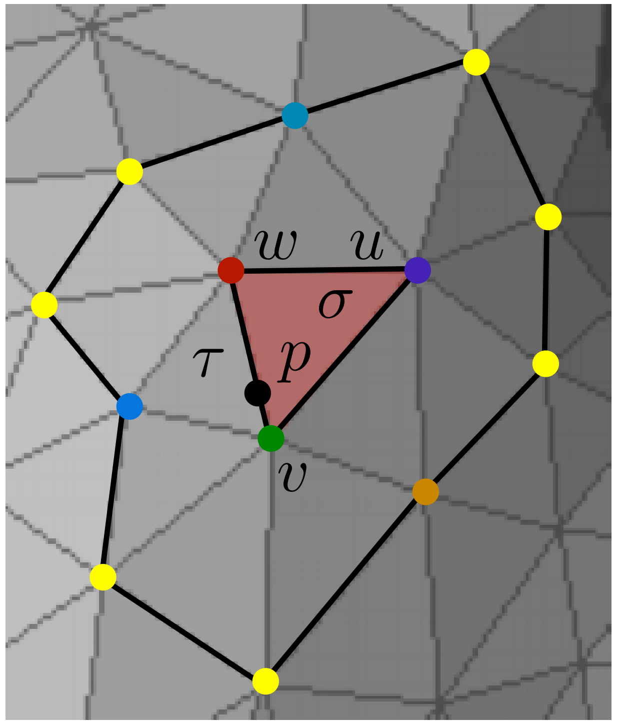


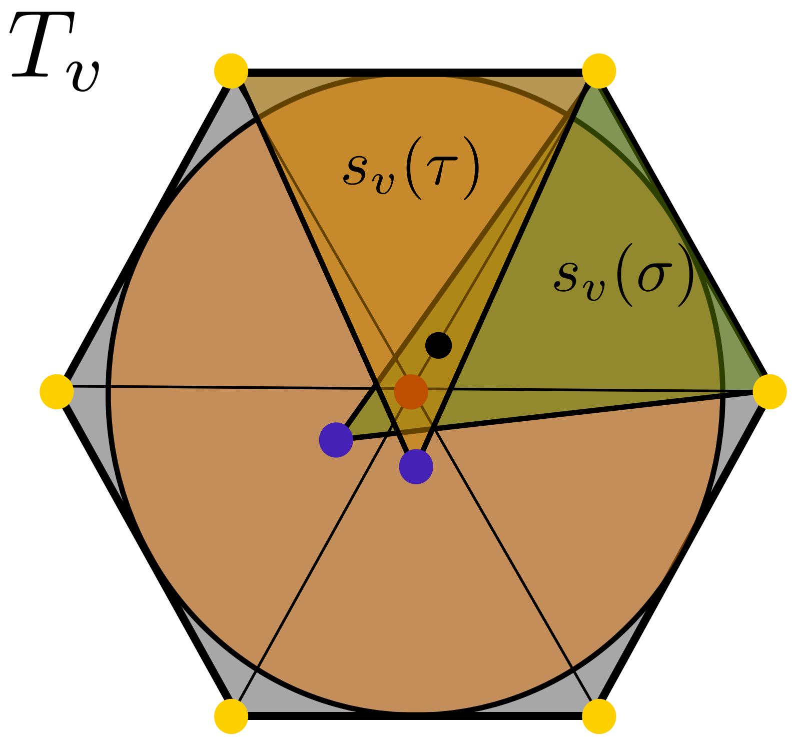
Building Parametrizations

Building Parametrizations

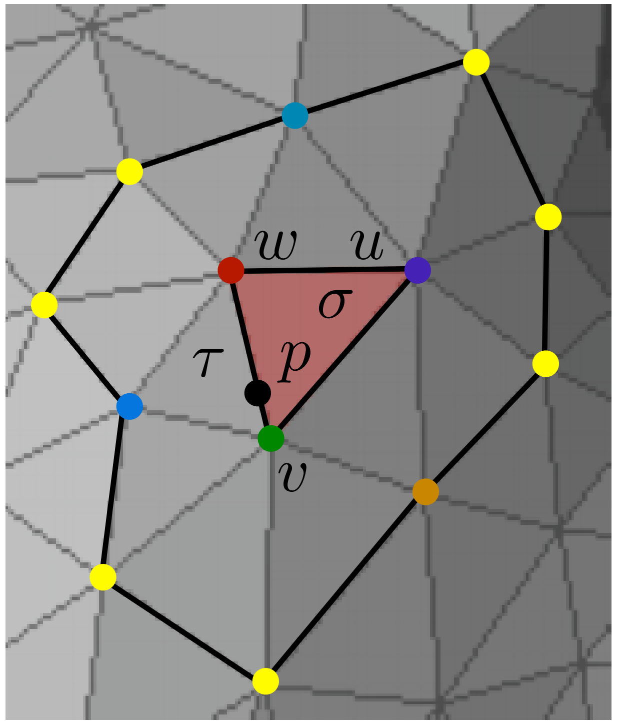


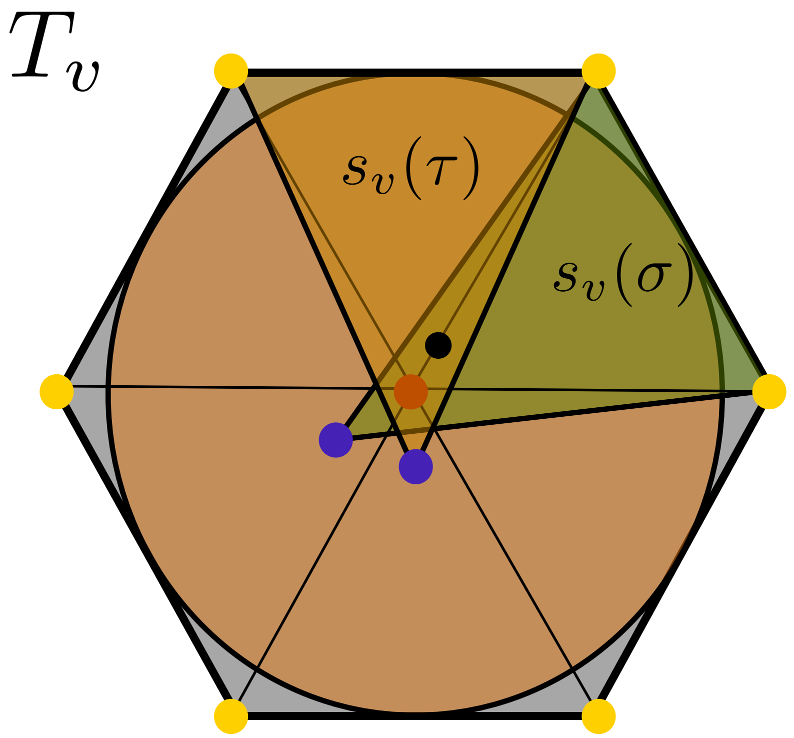
Building Parametrizations

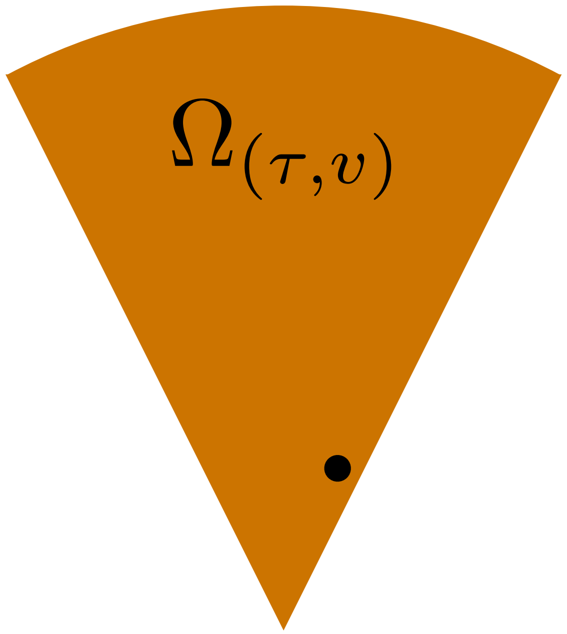




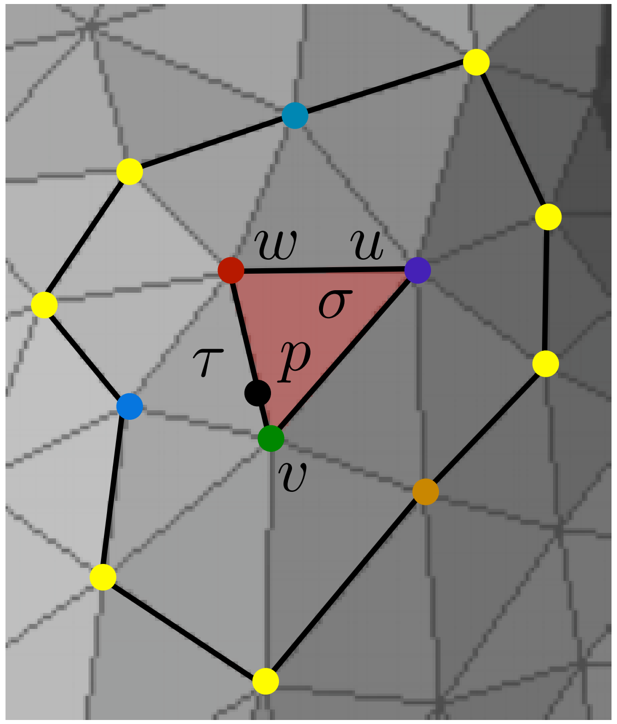
Building Parametrizations

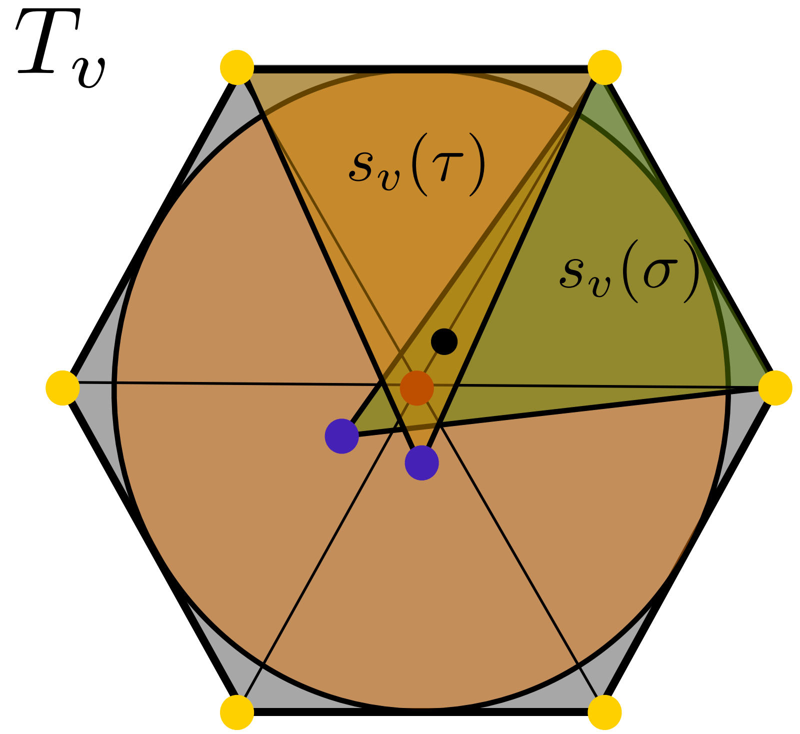


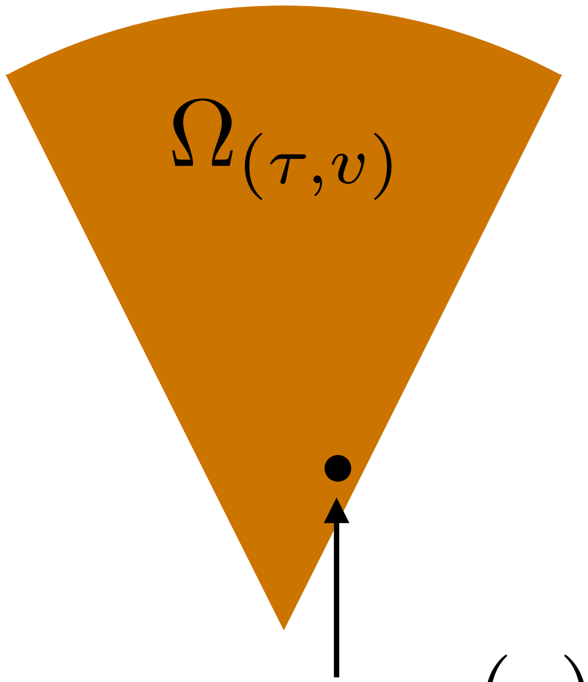




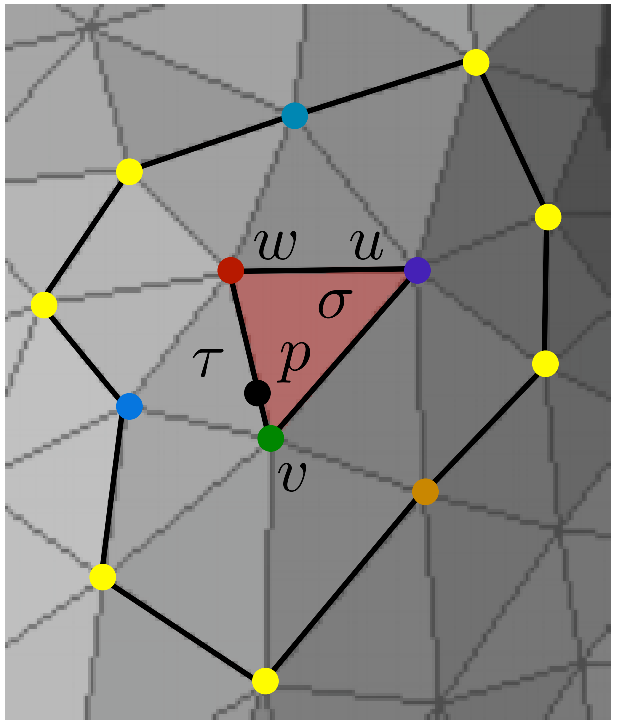
Building Parametrizations

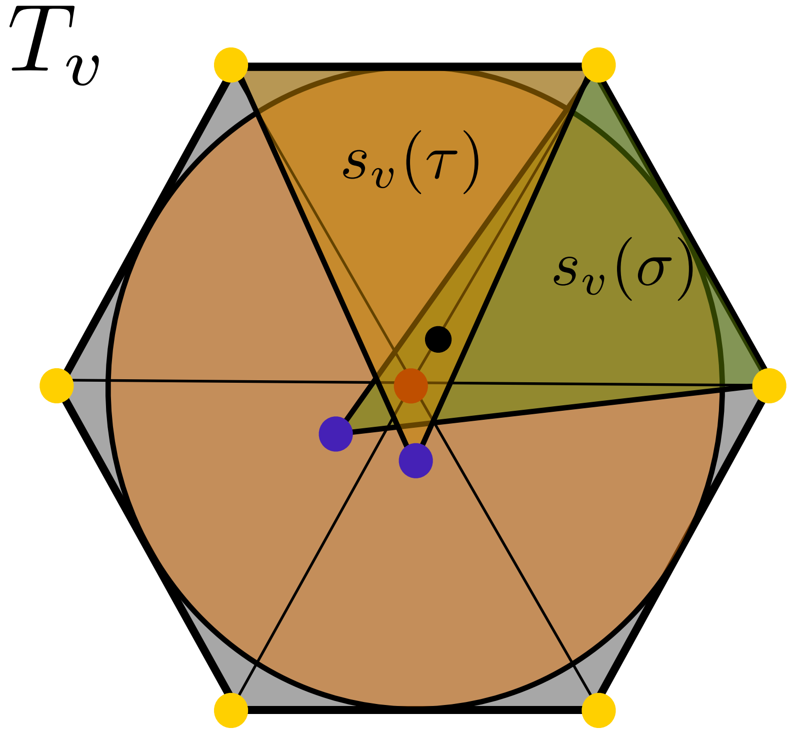


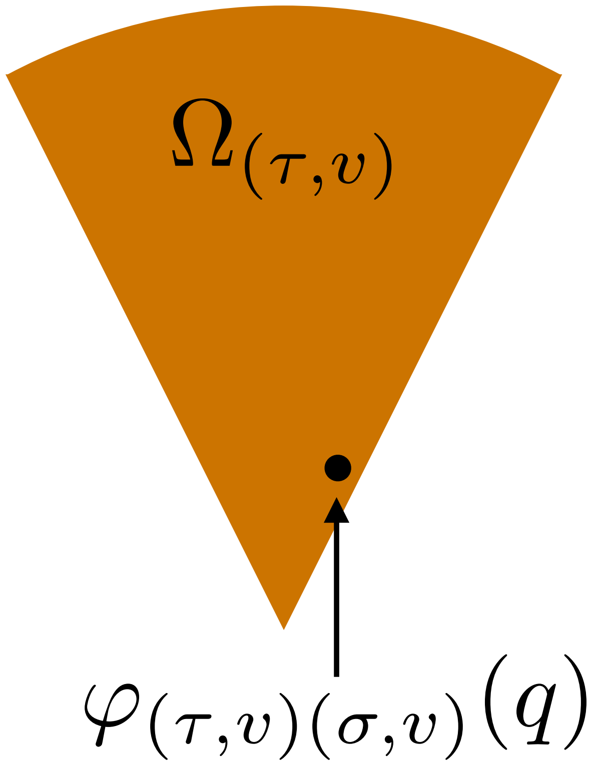


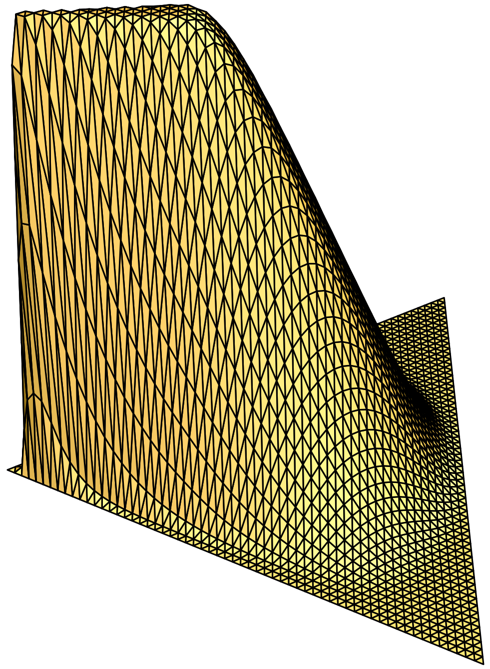


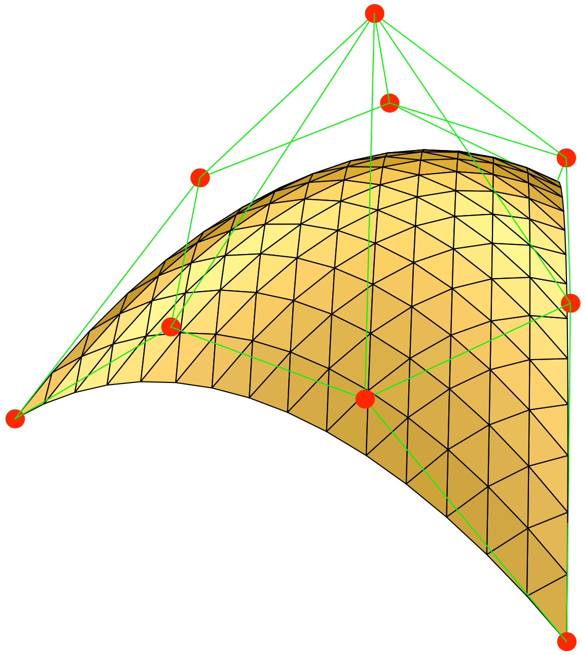
Building Parametrizations



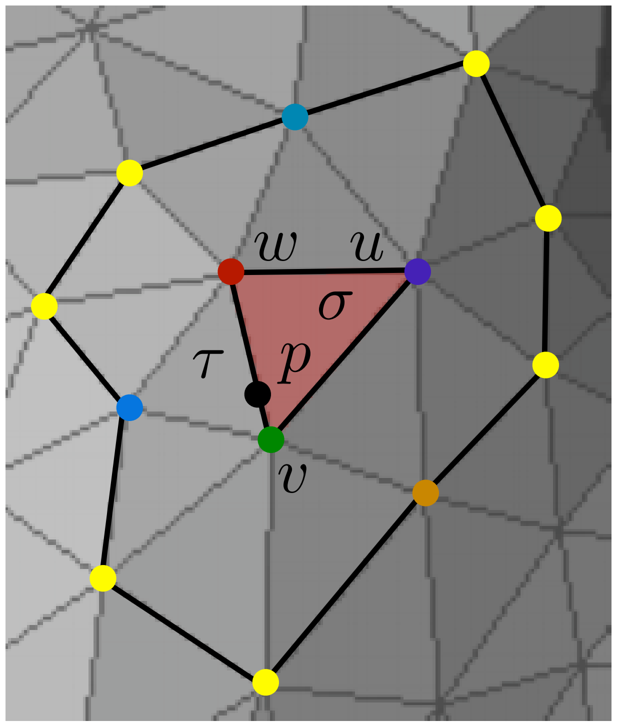


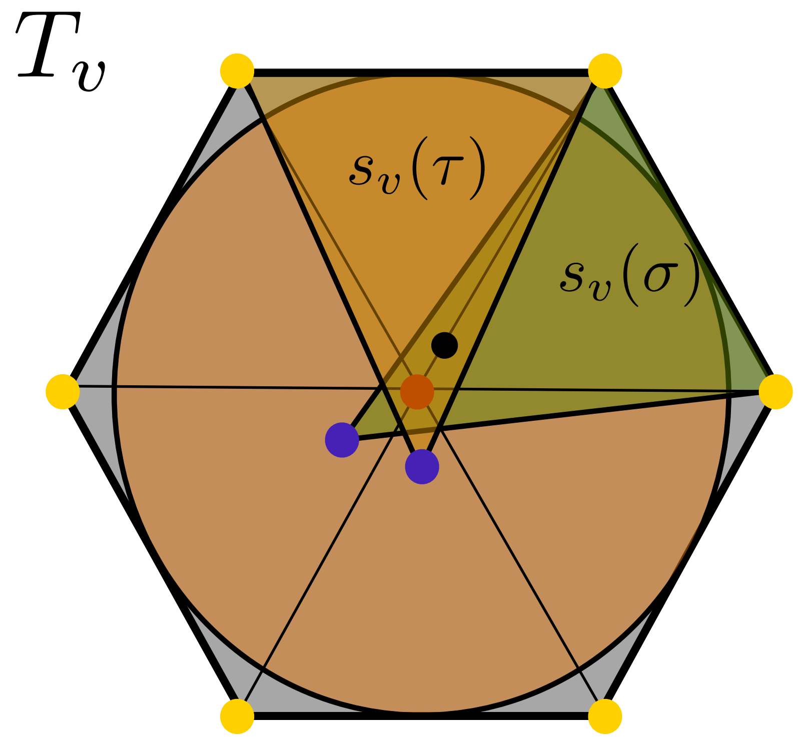


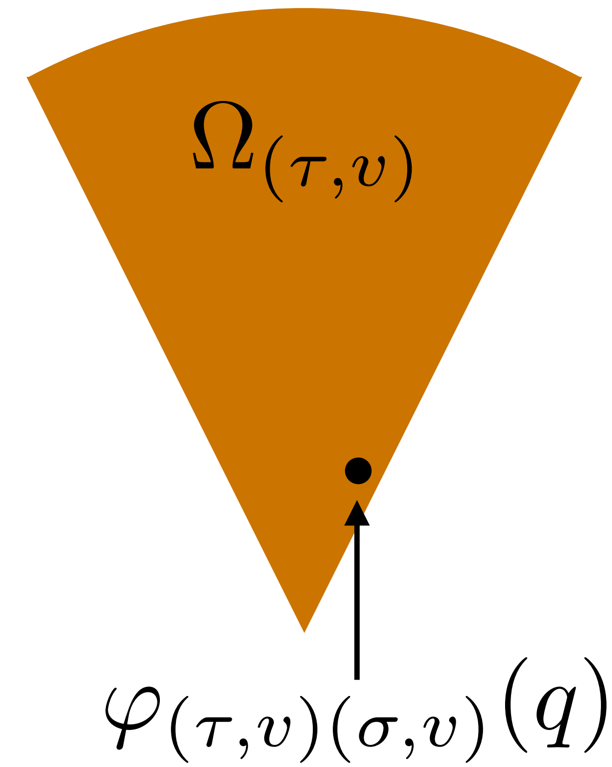


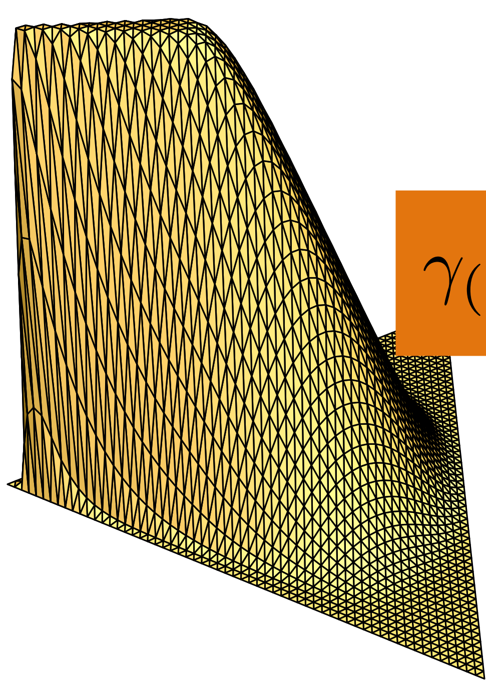


Building Parametrizations



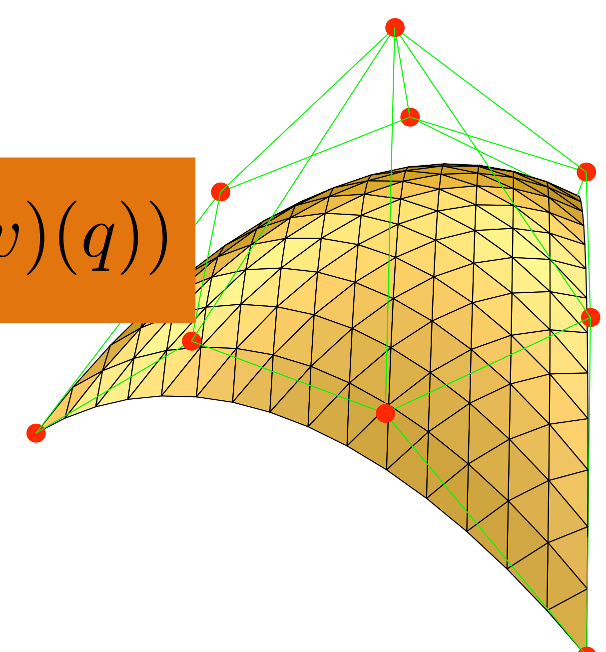






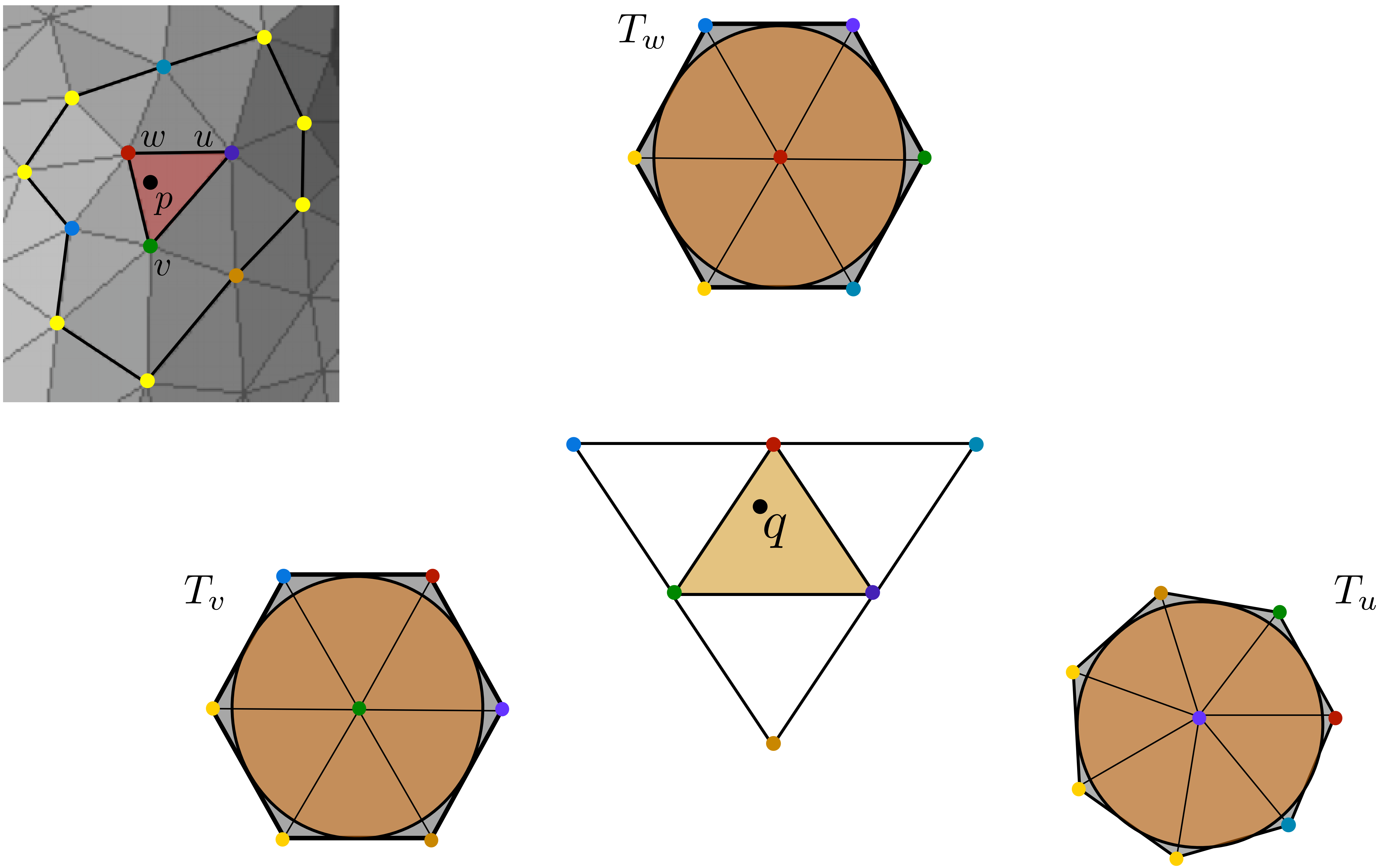
Contribution of $\Omega_{(\tau,v)}$:

$$\gamma_{(\tau,v)}(\varphi(\tau,v)(\sigma,v)(q)) \cdot \psi_{(\tau,v)}(\varphi(\tau,v)(\sigma,v)(q))$$

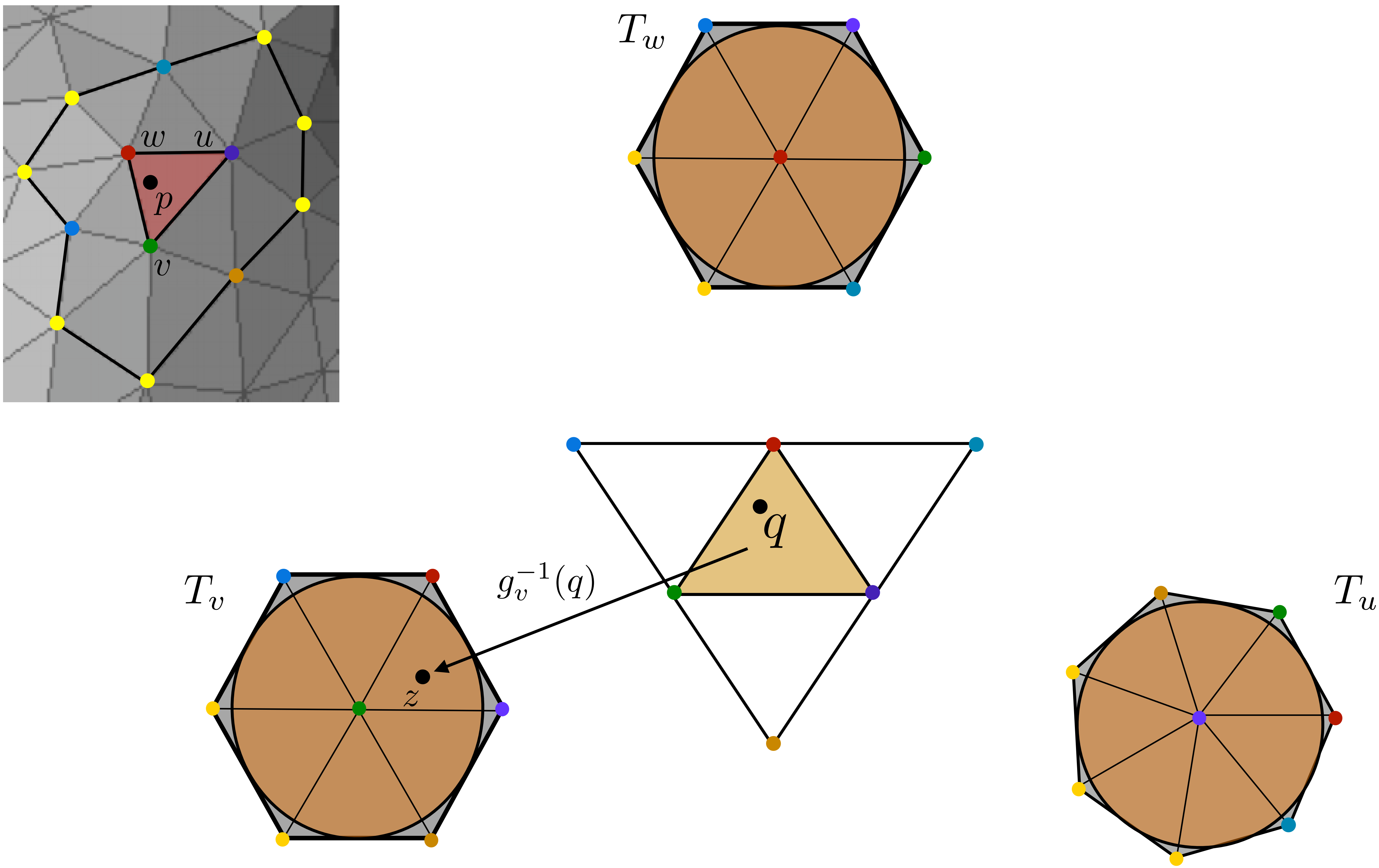


Building Parametrizations

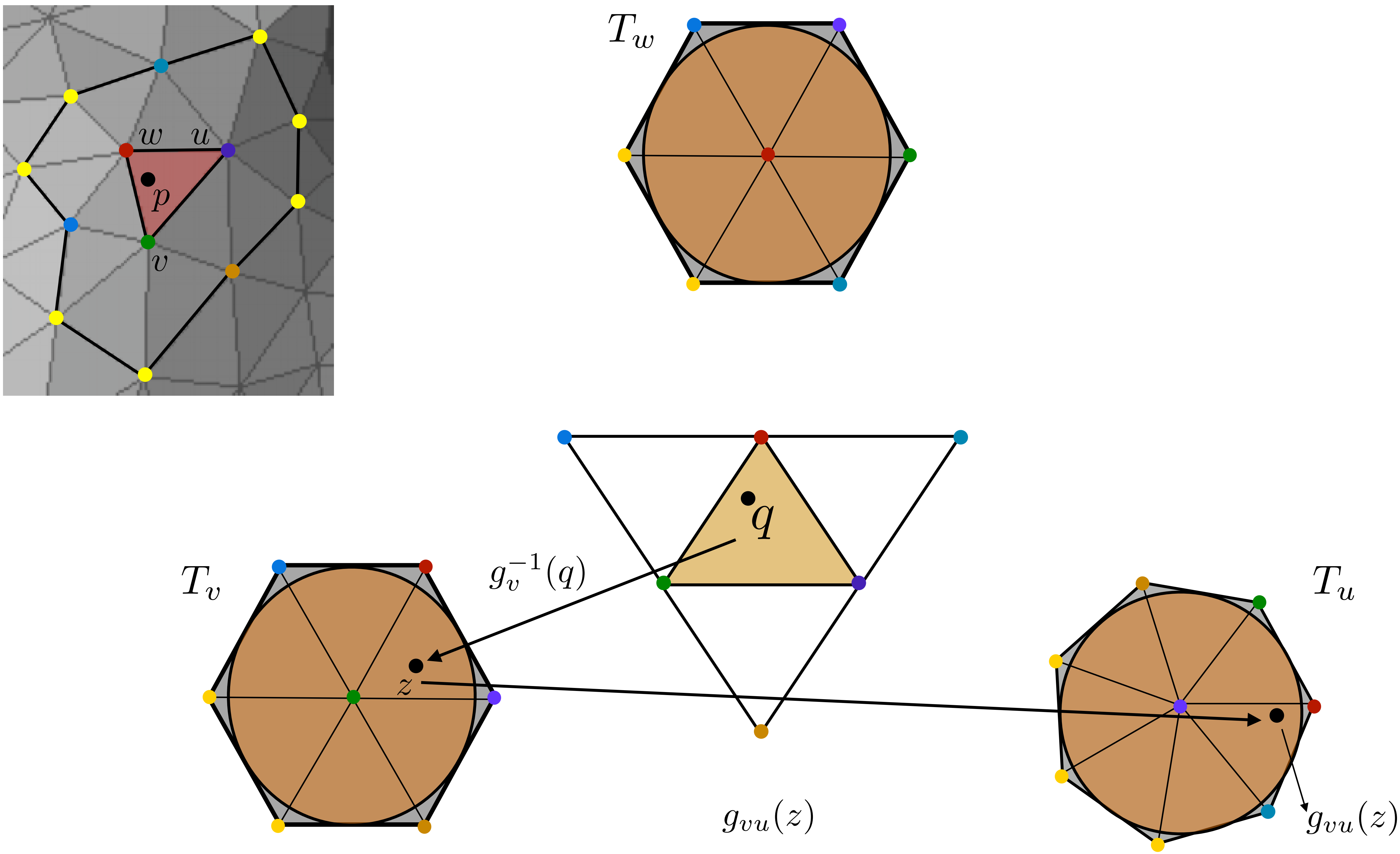
Building Parametrizations



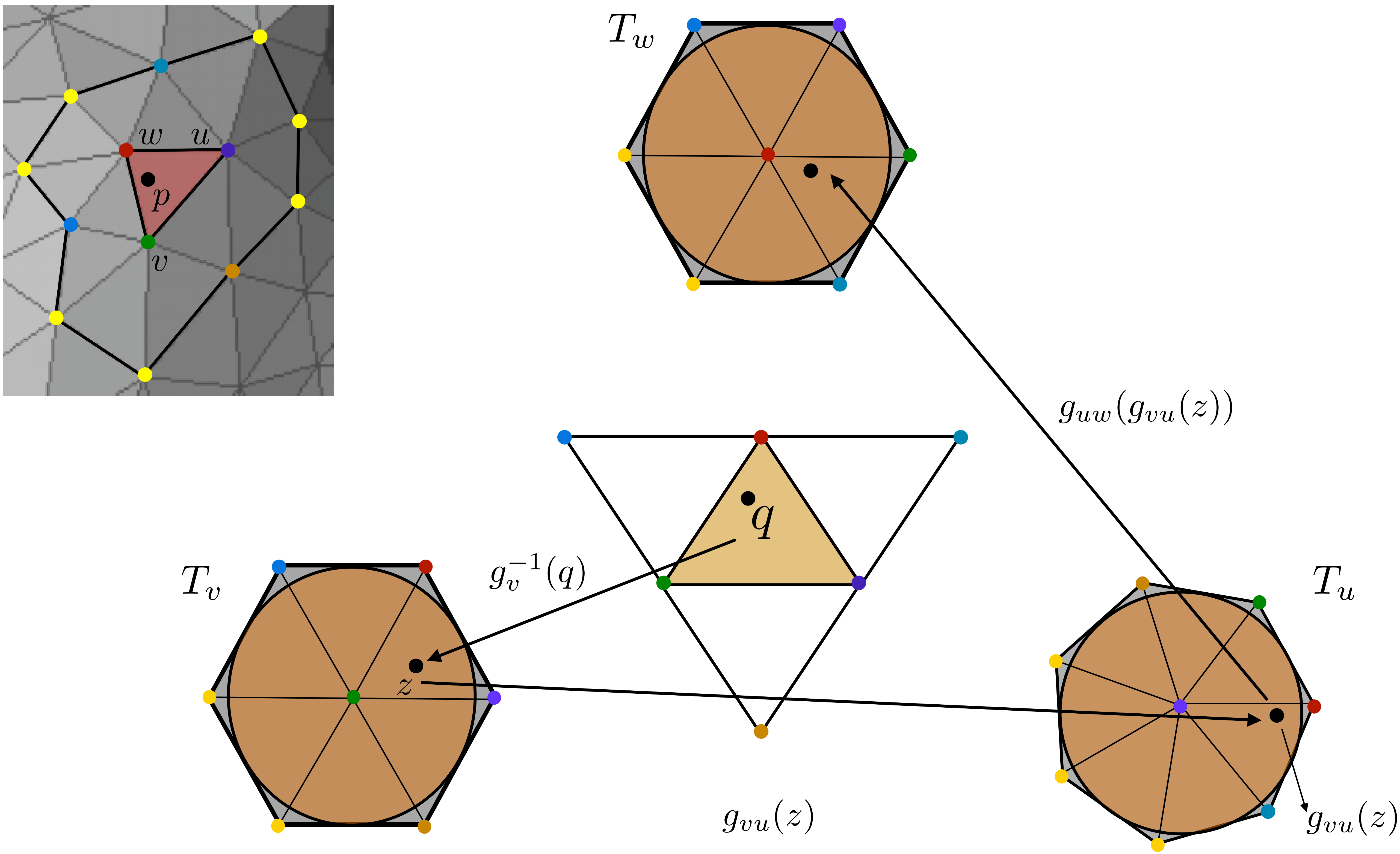
Building Parametrizations



Building Parametrizations



Building Parametrizations



Building Parametrizations

Building Parametrizations

For each $(\sigma, v) \in I$, we define a **parametrization**,

$$\theta_{(\sigma,v)} : \Omega_{(\sigma,v)} \to \mathbb{R}^3 \,,$$

such that for every $p \in \Omega_{(\sigma,v)}$,

$$\theta_{(\sigma,v)}(p) = \sum_{(\tau,u)\in J(p)} \nu_{(\tau,u)}(p) \cdot \psi_{(\tau,u)}(\varphi_{(\tau,u)(\sigma,v)}(p)) \,,$$

where

Building Parametrizations

Building Parametrizations

$$\nu_{(\tau,u)}(p) = \frac{\gamma_{(\tau,u)}(\varphi_{(\tau,u)(\sigma,v)}(p))}{\displaystyle\sum_{(\eta,w)\in J(p)} \gamma_{(\eta,w)}(\varphi_{(\eta,w)(\sigma,v)}(p))}$$

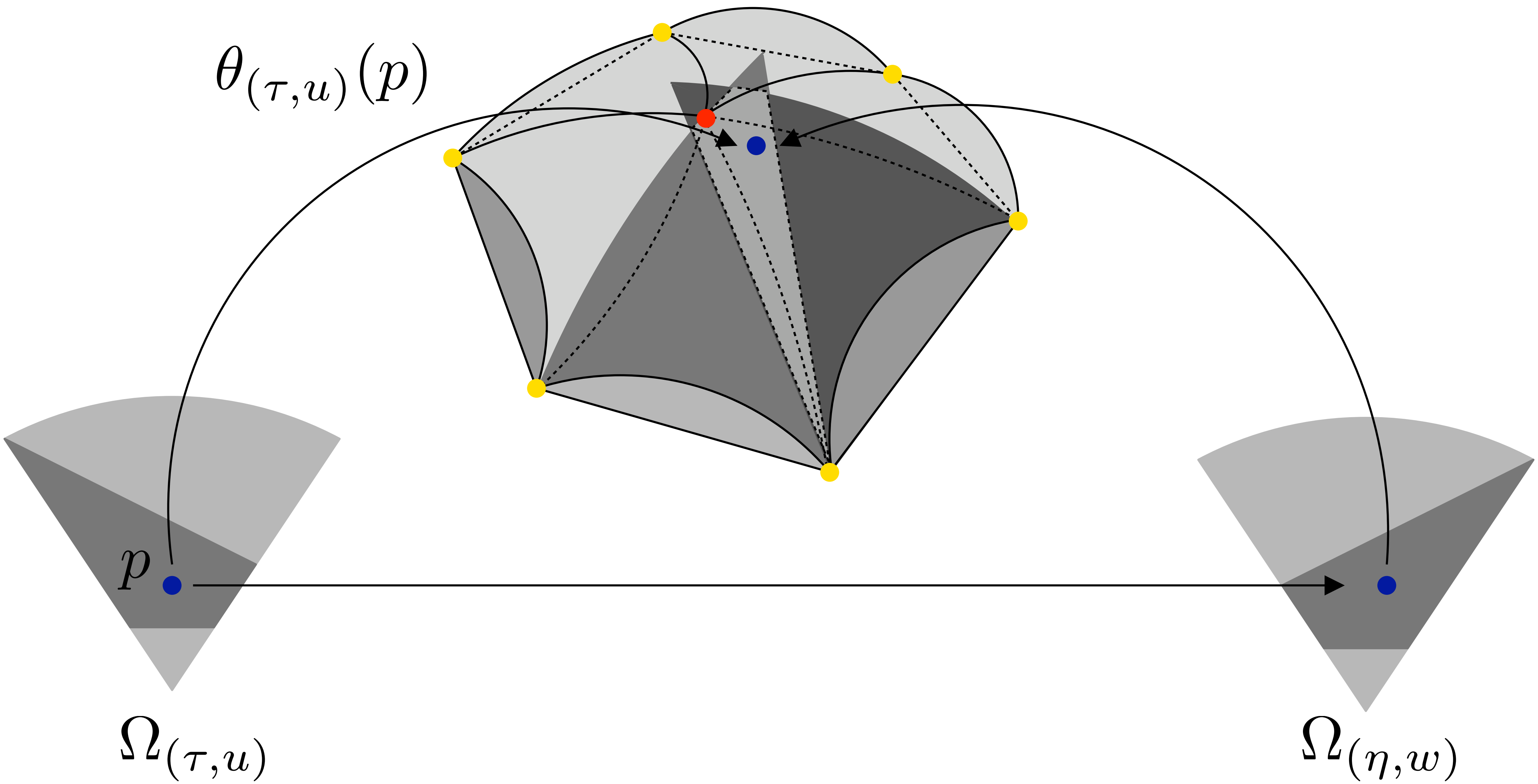
and

$$J(p) = \left\{(\eta, w) \in I \mid p \in \Omega_{(\sigma,v)(\eta,w)}\right\}.$$

Building Parametrizations

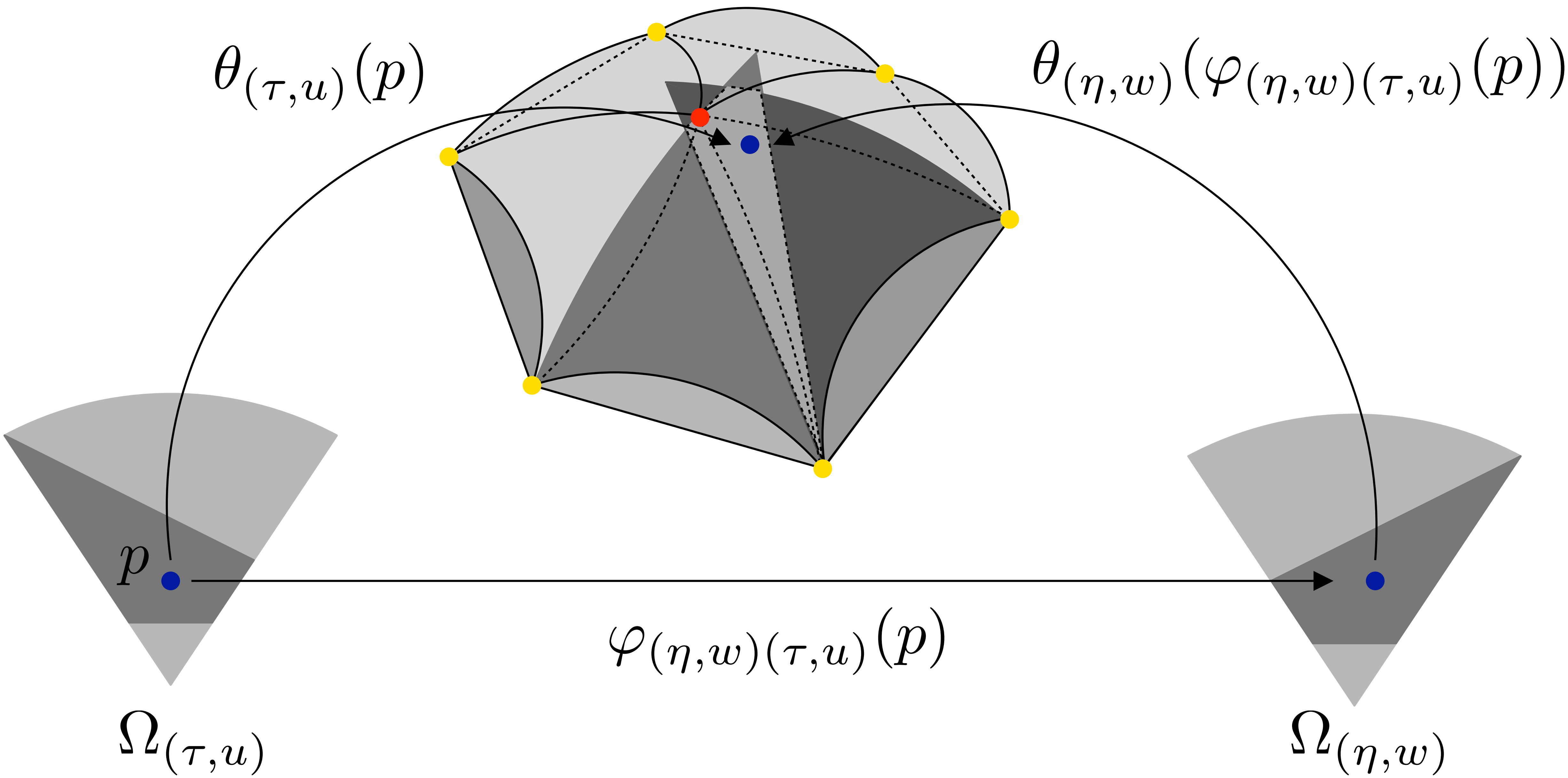
Building Parametrizations

Parametrizations are consistent!



Building Parametrizations

Parametrizations are consistent!

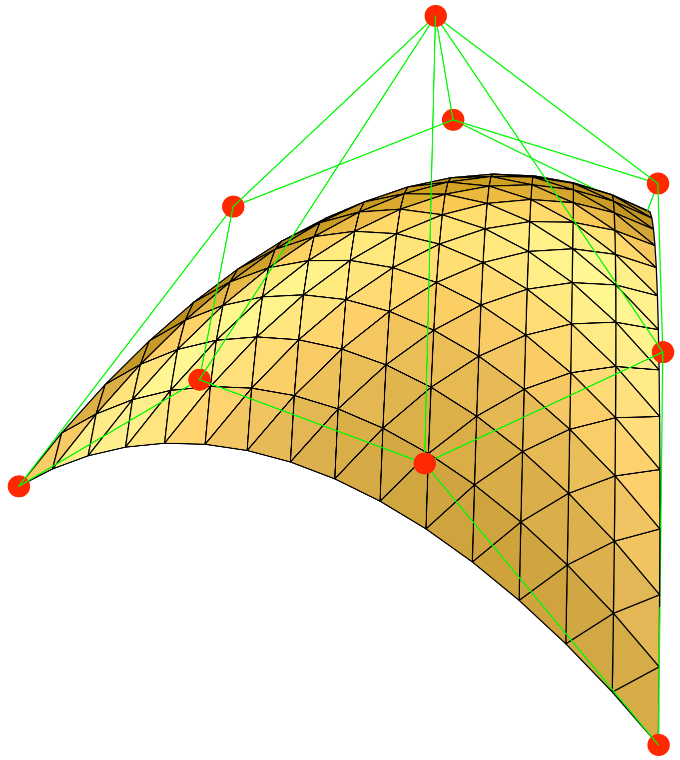


Building Parametrizations

$$\psi_{(\tau,u)}(\Omega_{(\tau,u)})$$

Building Parametrizations

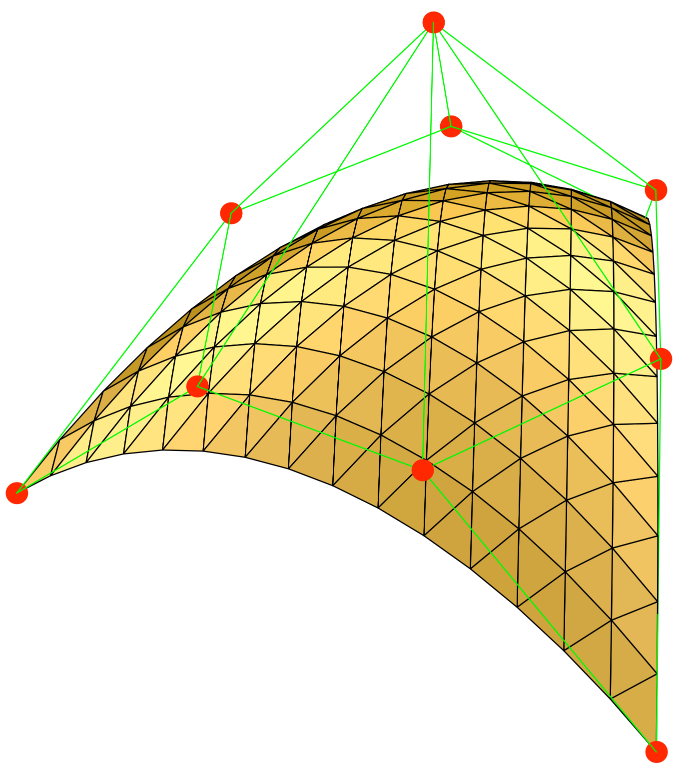
The control points of $\psi_{(\tau,u)}$ are the solutions of a least squares problem.

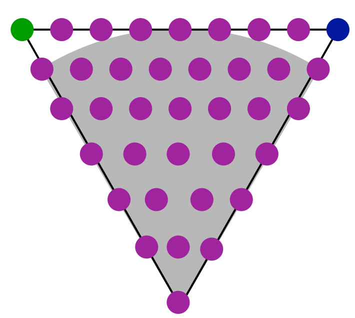


$\psi_{(\tau,u)}(\Omega_{(\tau,u)})$

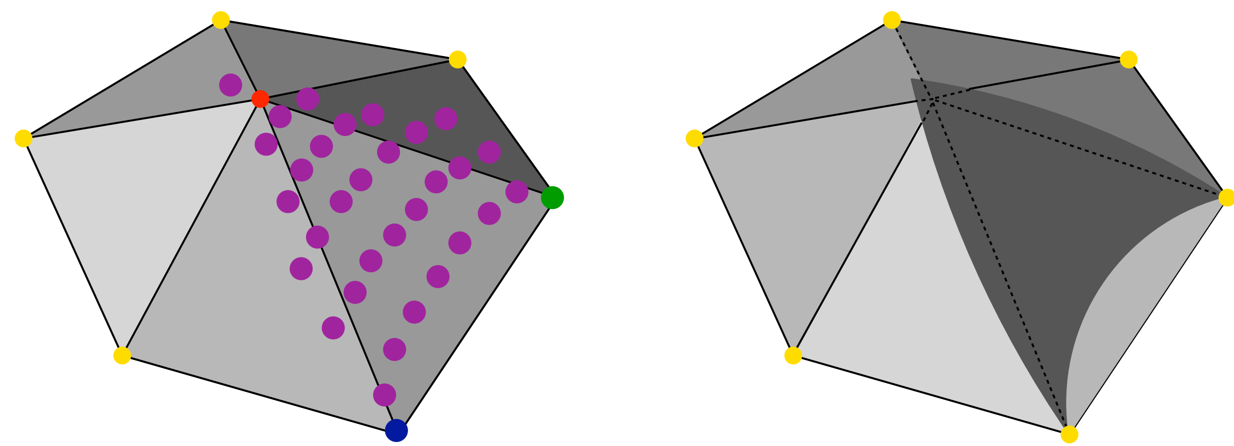
Building Parametrizations

The control points of $\psi_{(\tau,u)}$ are the solutions of a least squares problem.





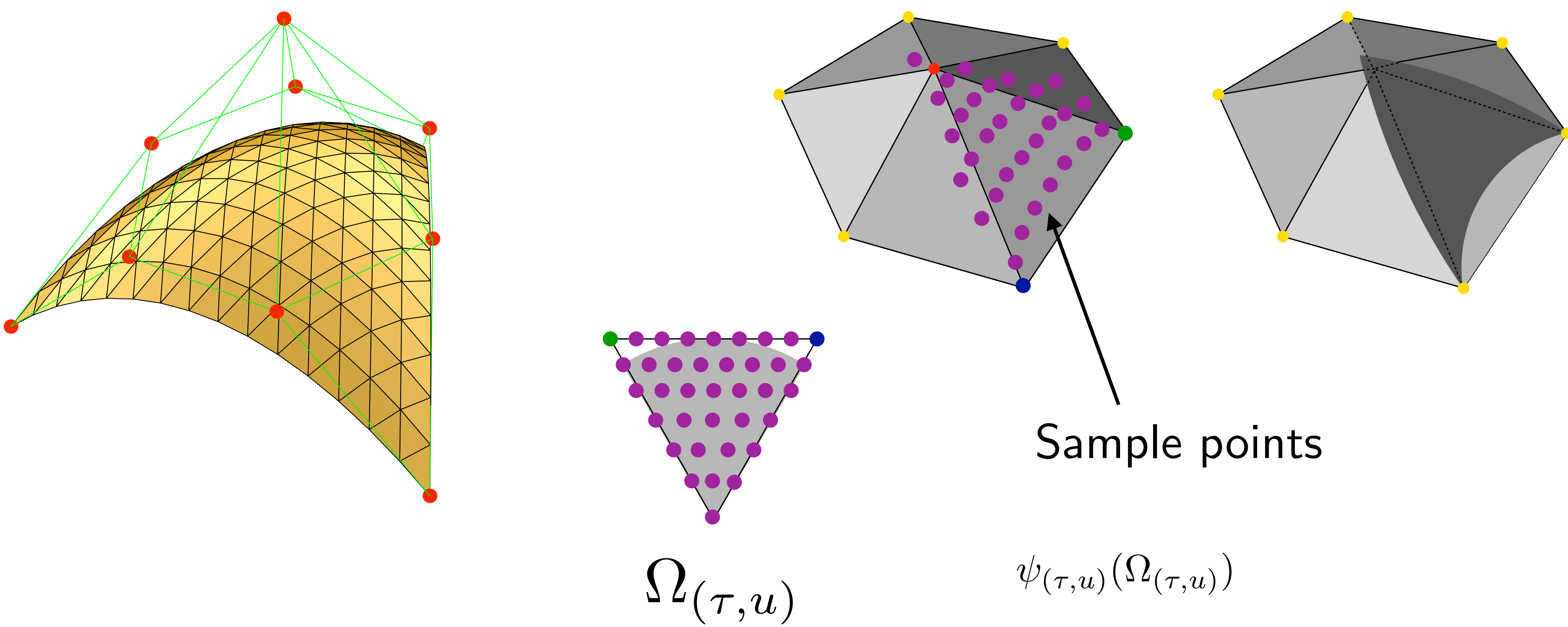
$\Omega_{(\tau,u)}$



$\psi_{(\tau,u)}(\Omega_{(\tau,u)})$

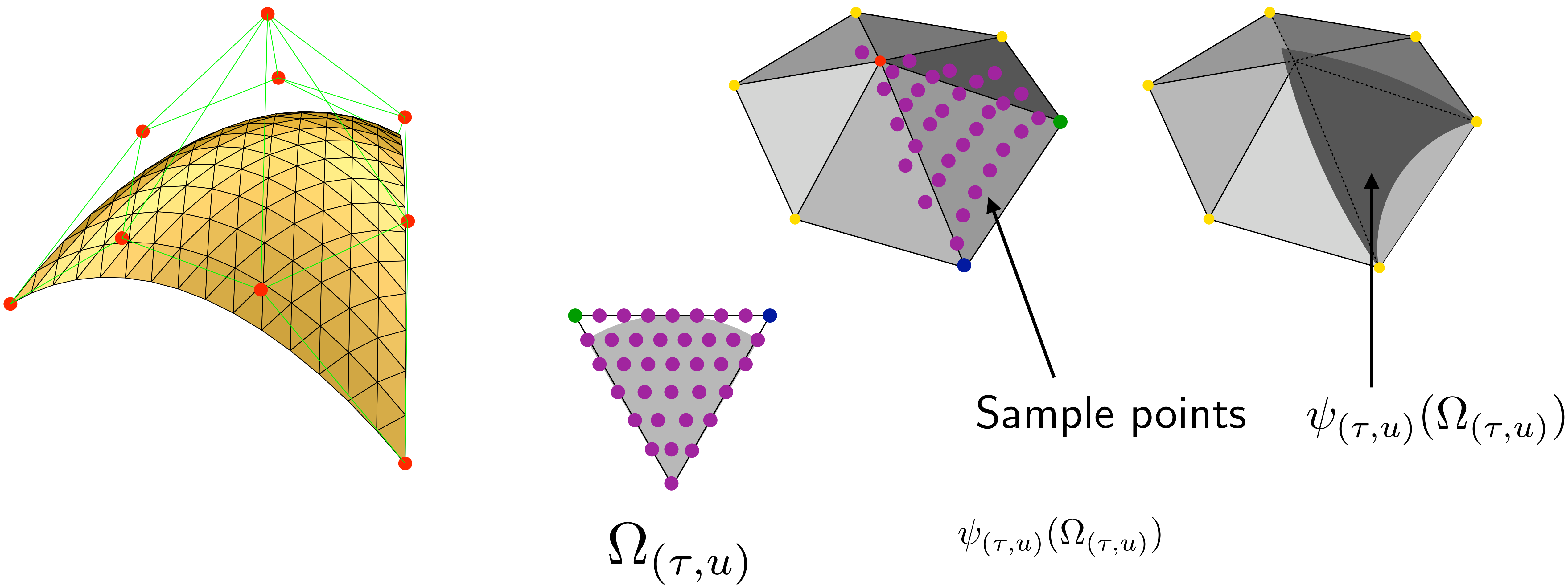
Building Parametrizations

The control points of $\psi_{(\tau,u)}$ are the solutions of a least squares problem.



Building Parametrizations

The control points of $\psi_{(\tau,u)}$ are the solutions of a least squares problem.



Building Parametrizations

Building Parametrizations

How can we find the sample points to start with?

Building Parametrizations

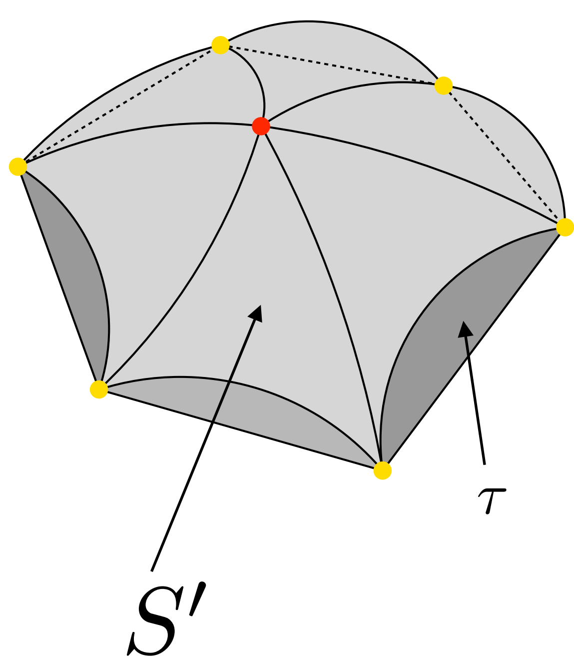
How can we find the sample points to start with?

Fit a “curved” surface, S', to S_T and then sample it!

Building Parametrizations

How can we find the sample points to start with?

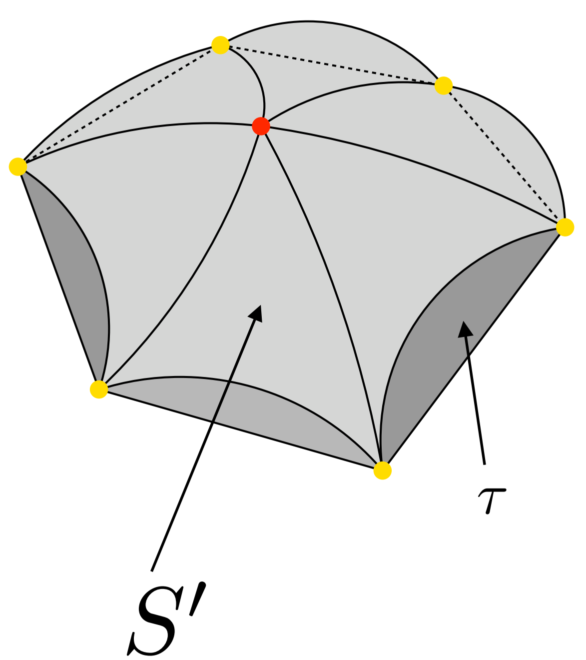
Fit a “curved” surface, S', to S_T and then sample it!



Building Parametrizations

How can we find the sample points to start with?

Fit a "curved" surface, S', to S_T and then sample it!

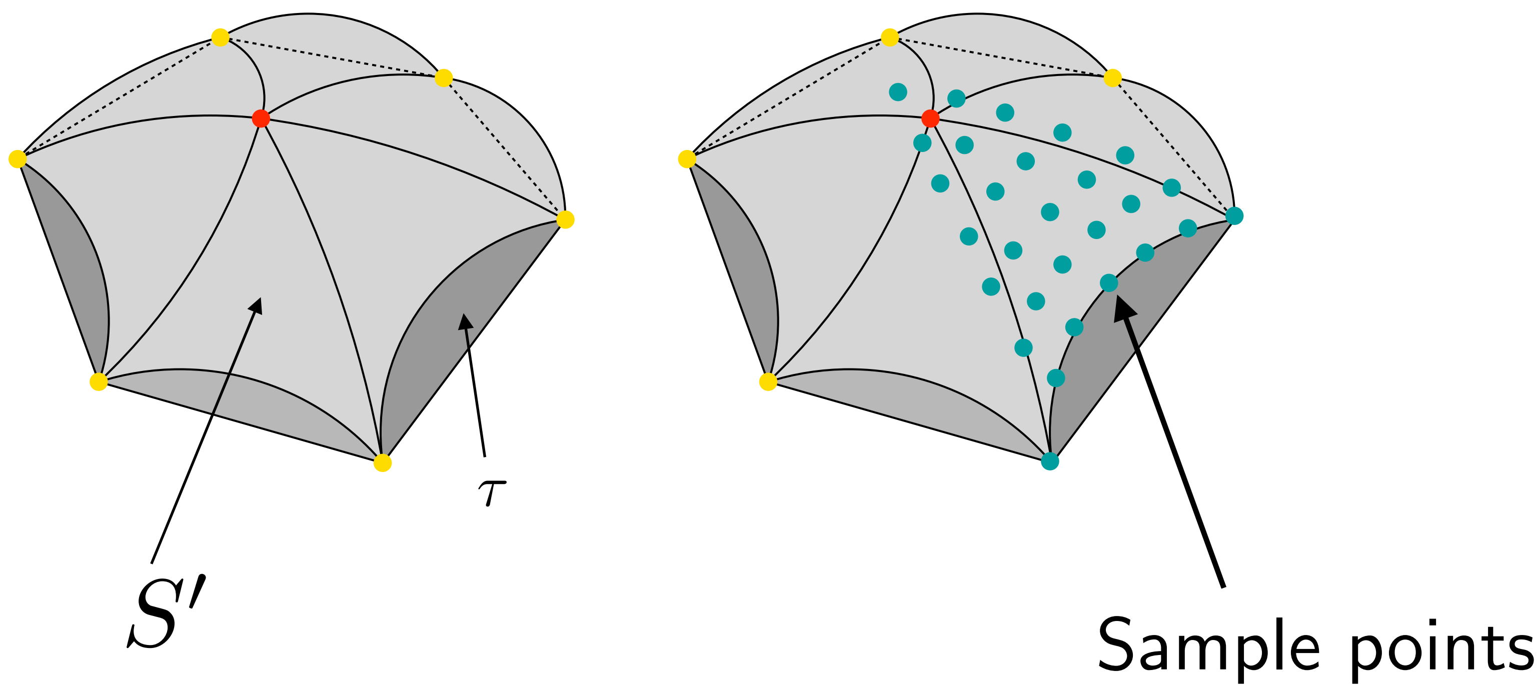


Good choices:

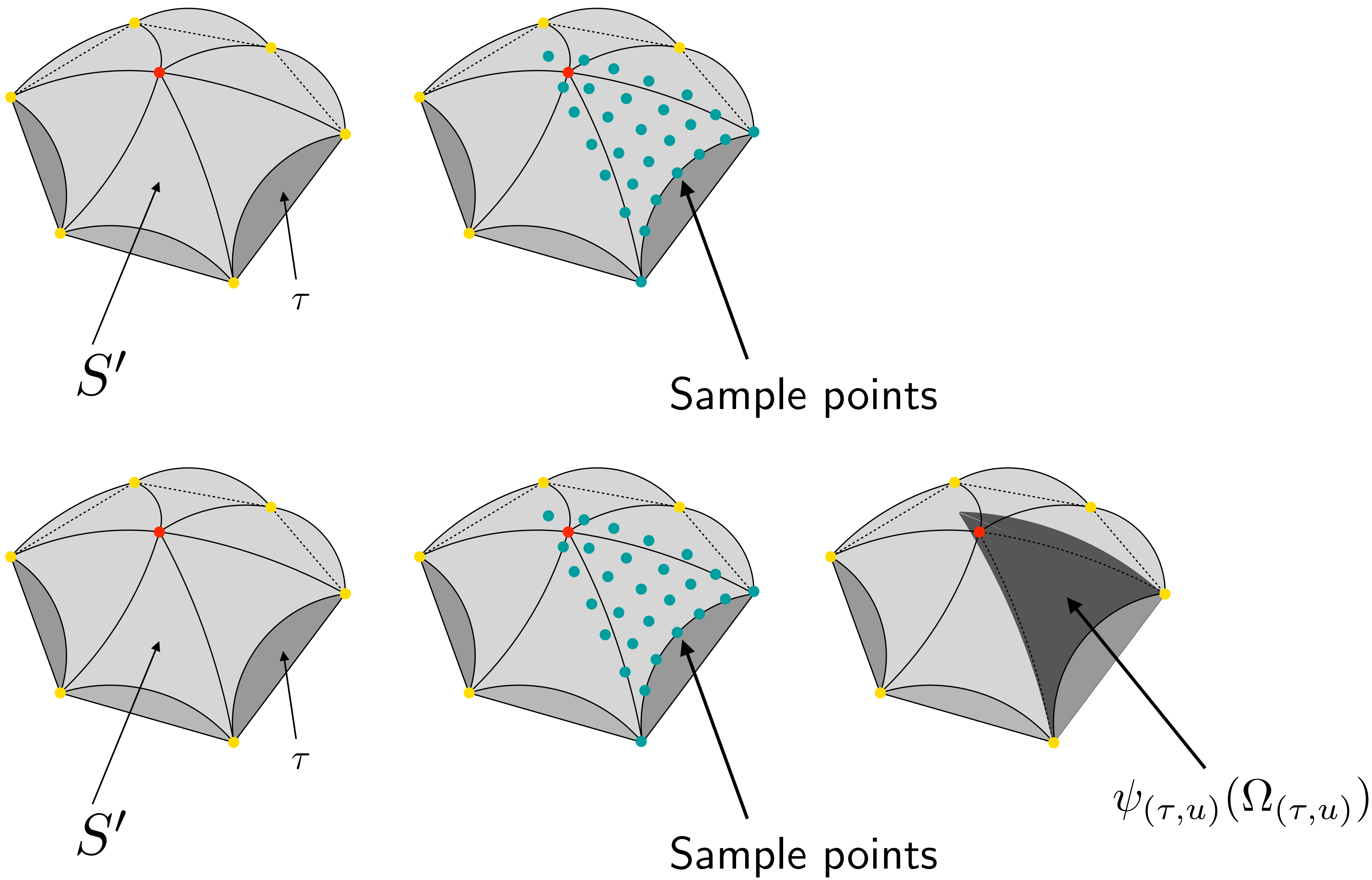
- PN triangle surfaces
- Subdivision surfaces

Building Parametrizations

Building Parametrizations

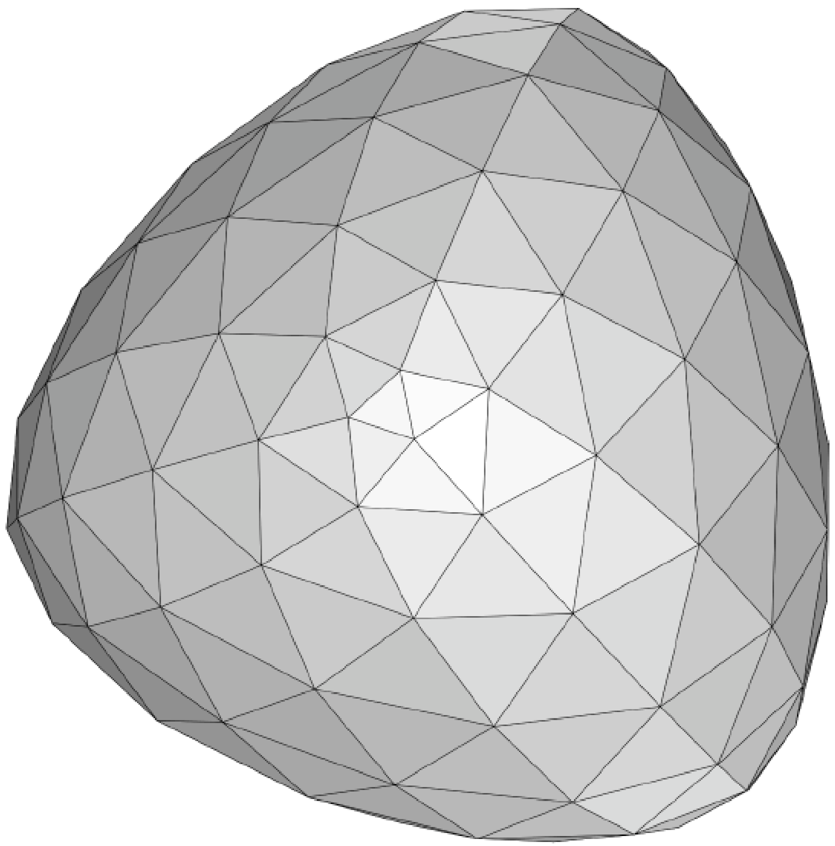


Building Parametrizations



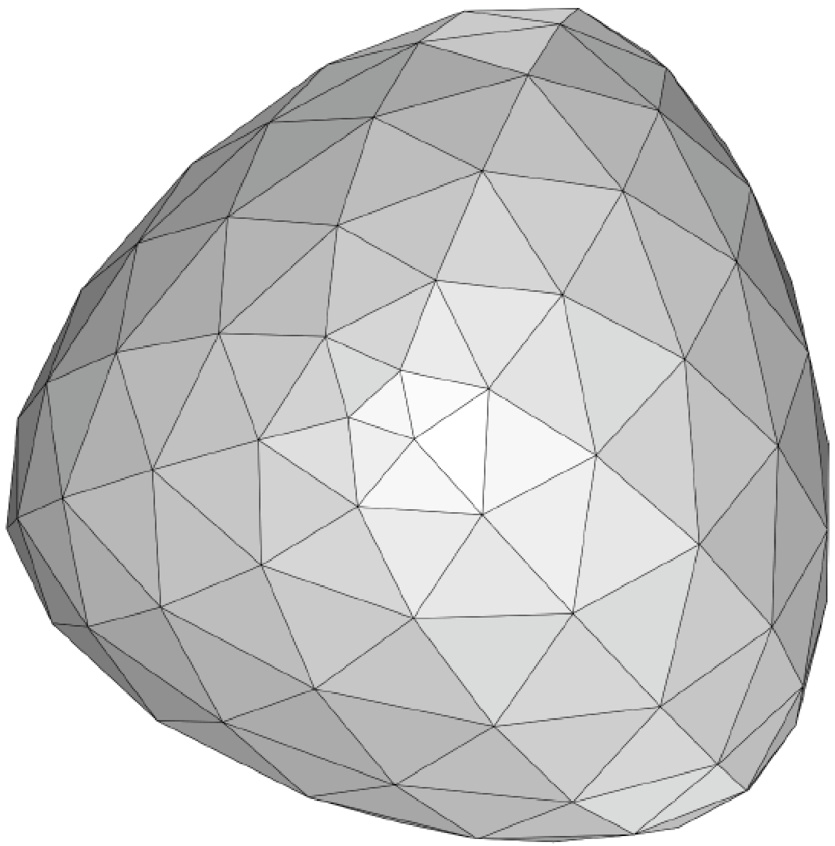
Results

Results

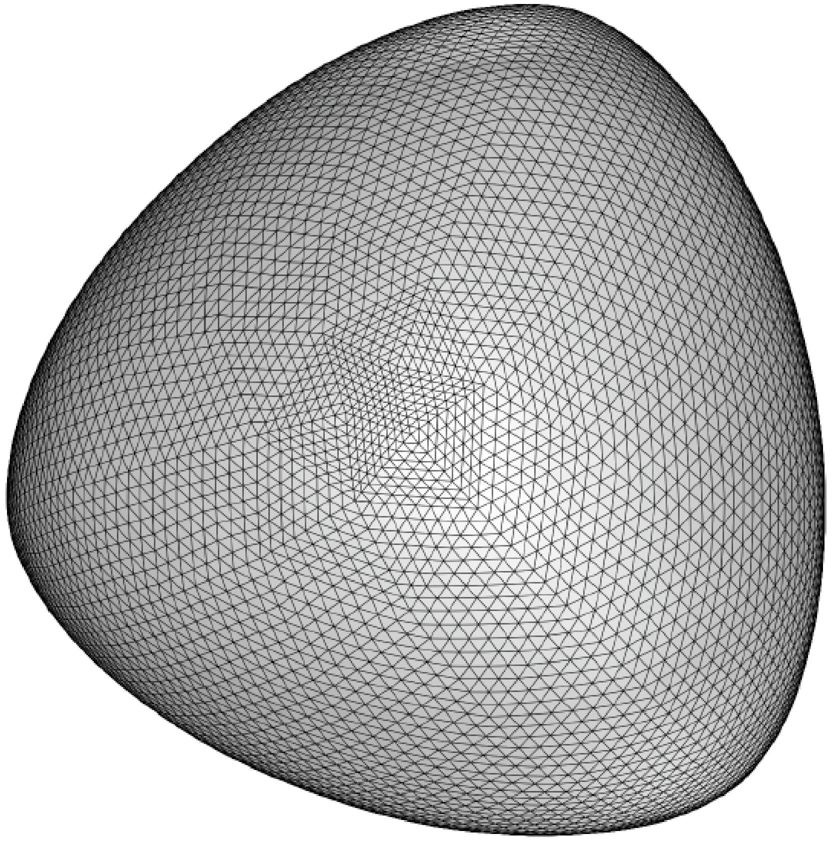


Mesh

Results

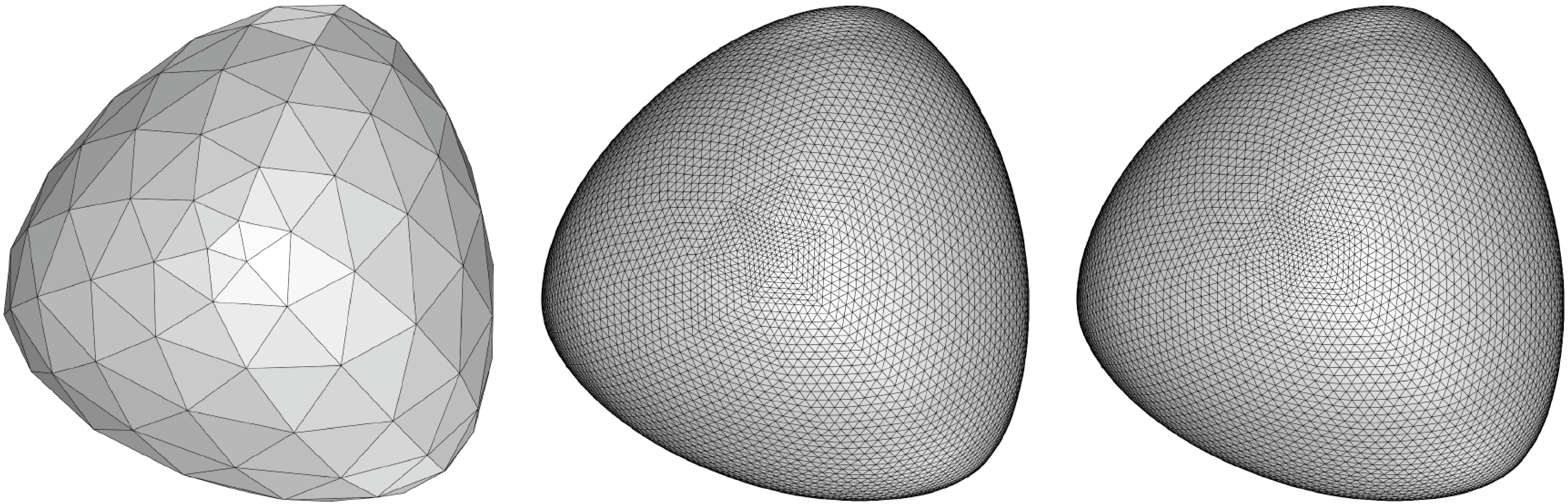


Mesh



PN triangle

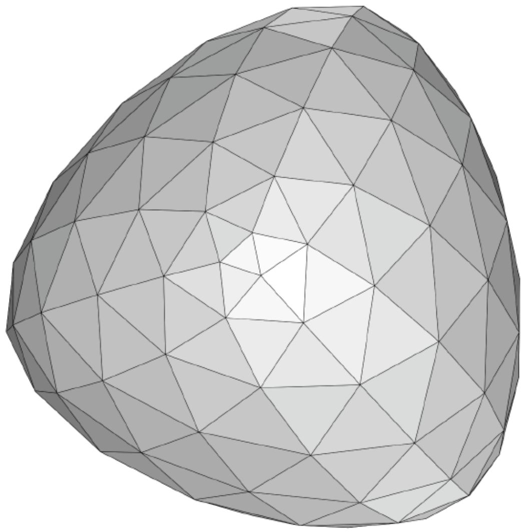
Results



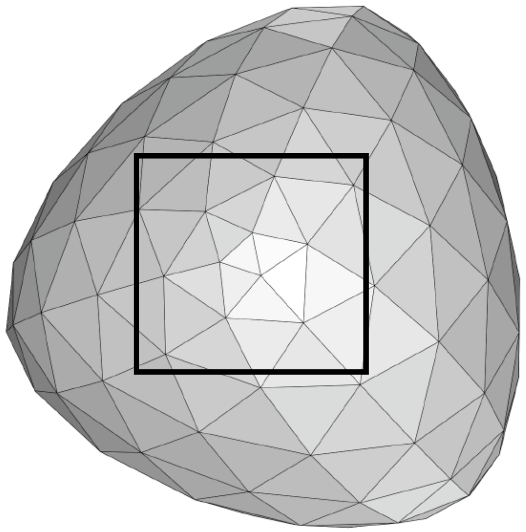
Mesh PN triangle PPS

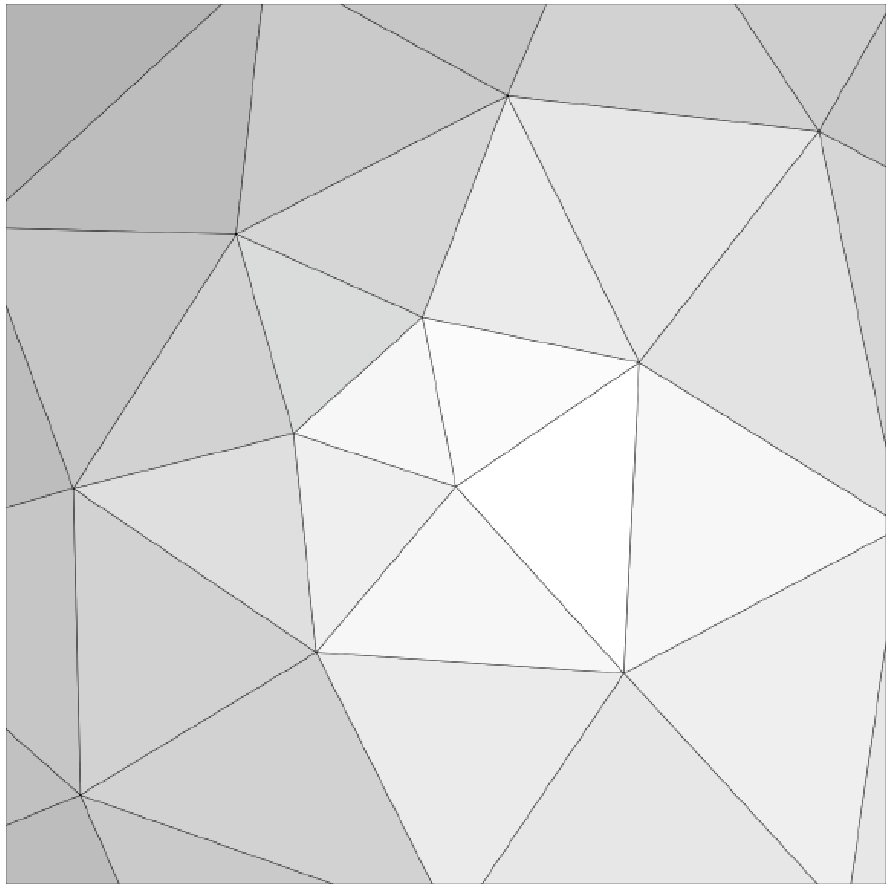
Results

Results

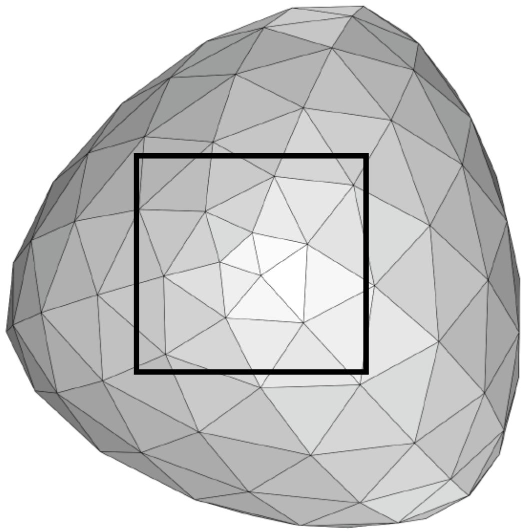


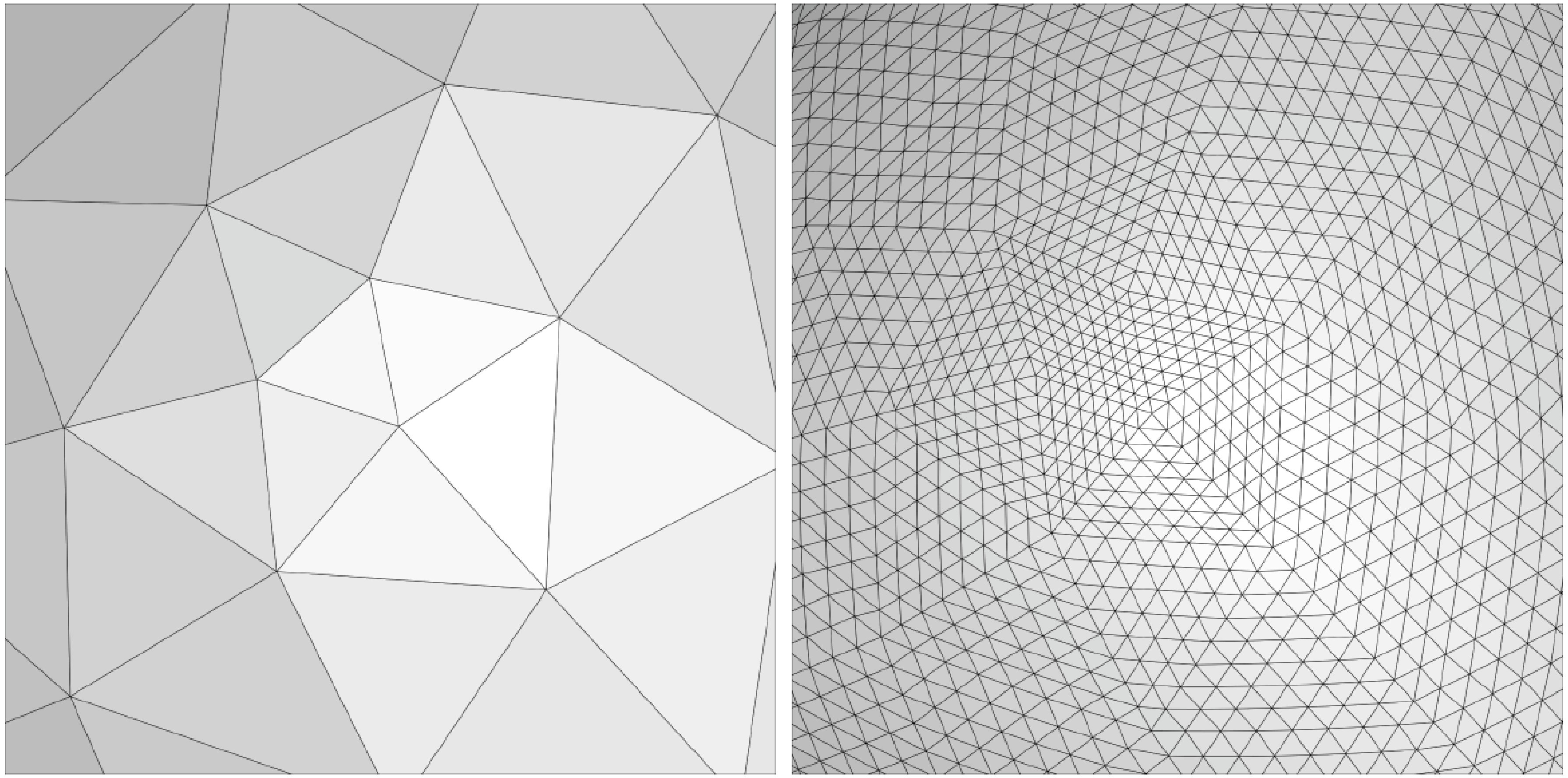
Results



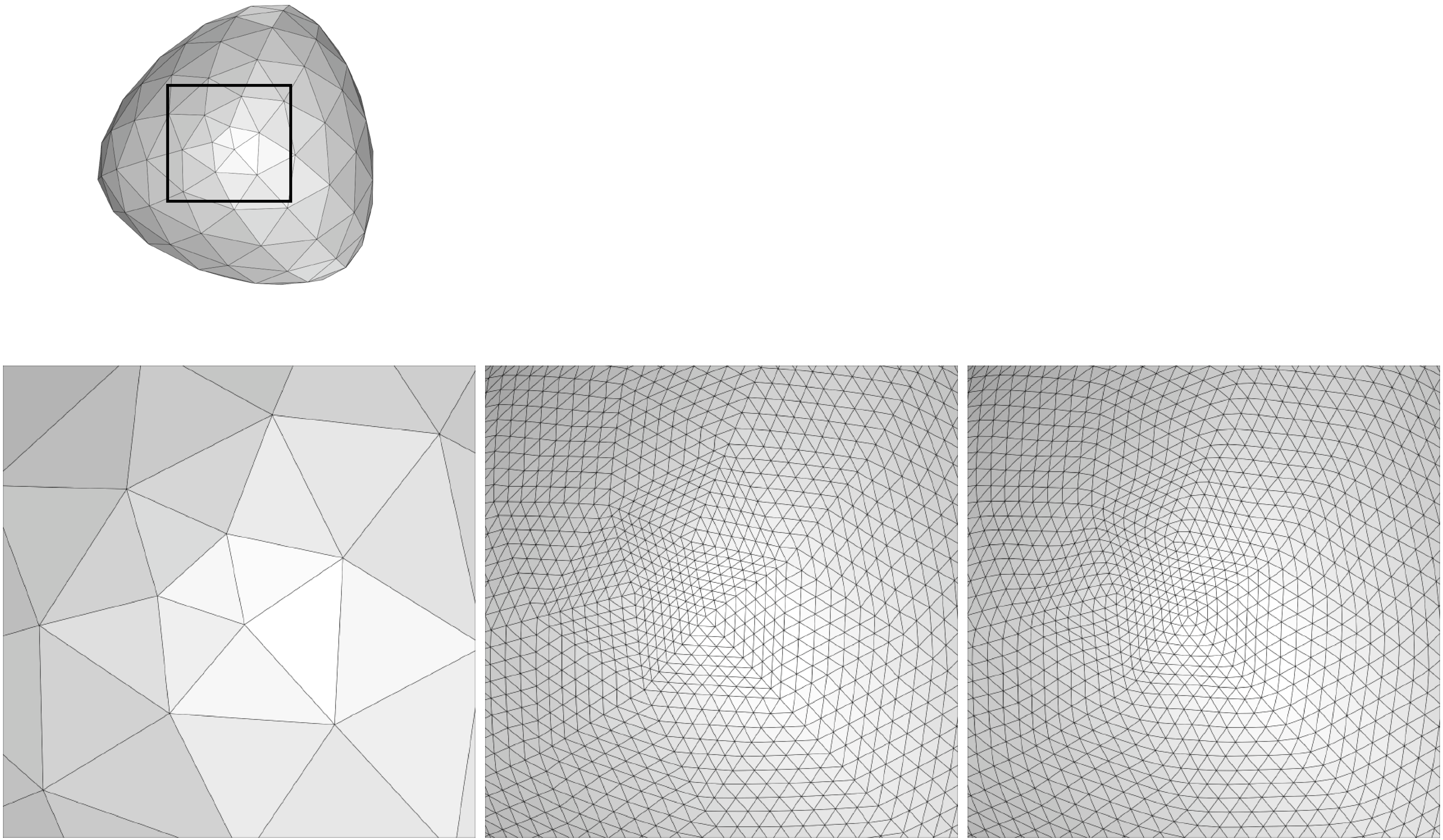


Results



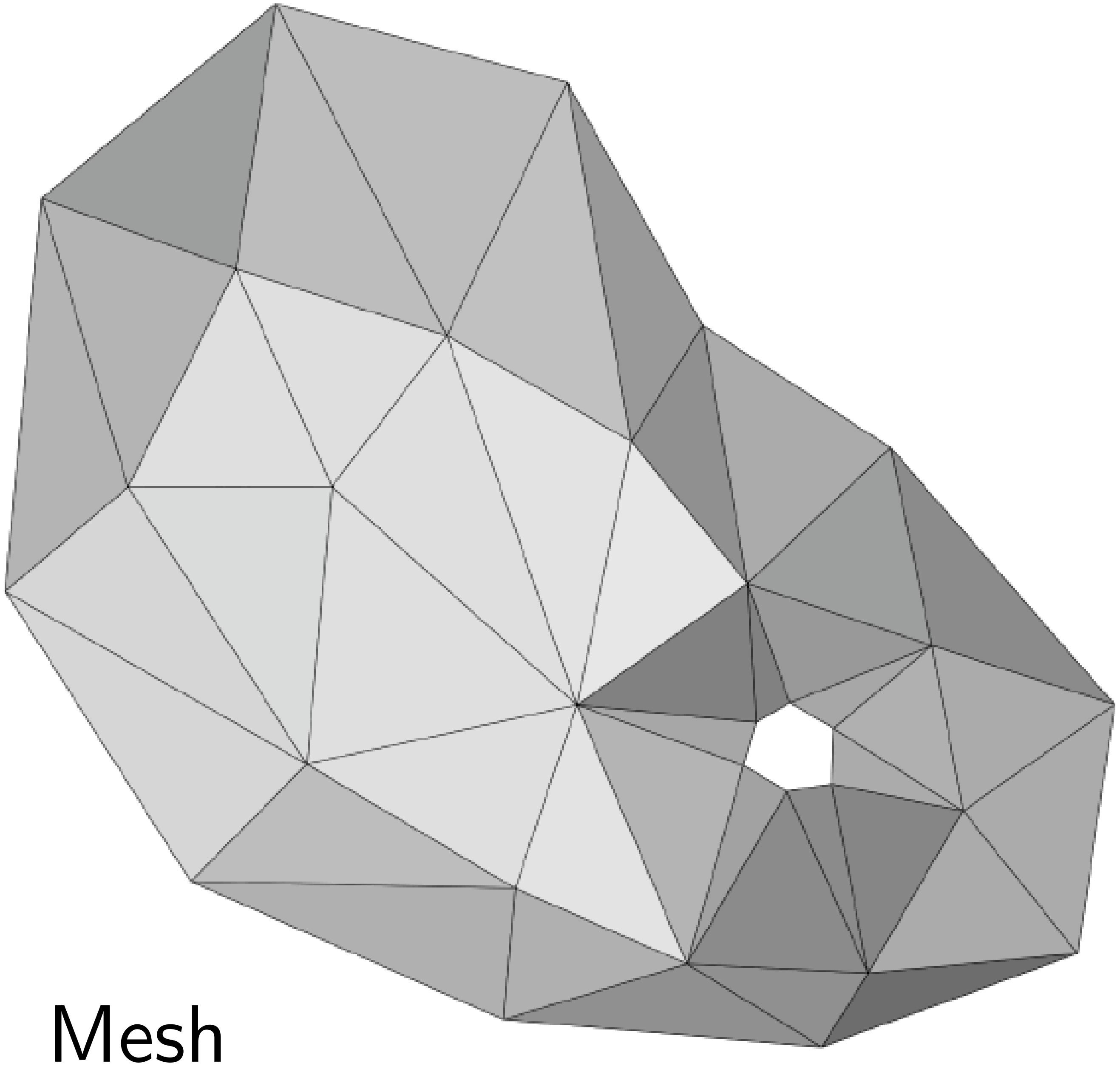


Results



Results

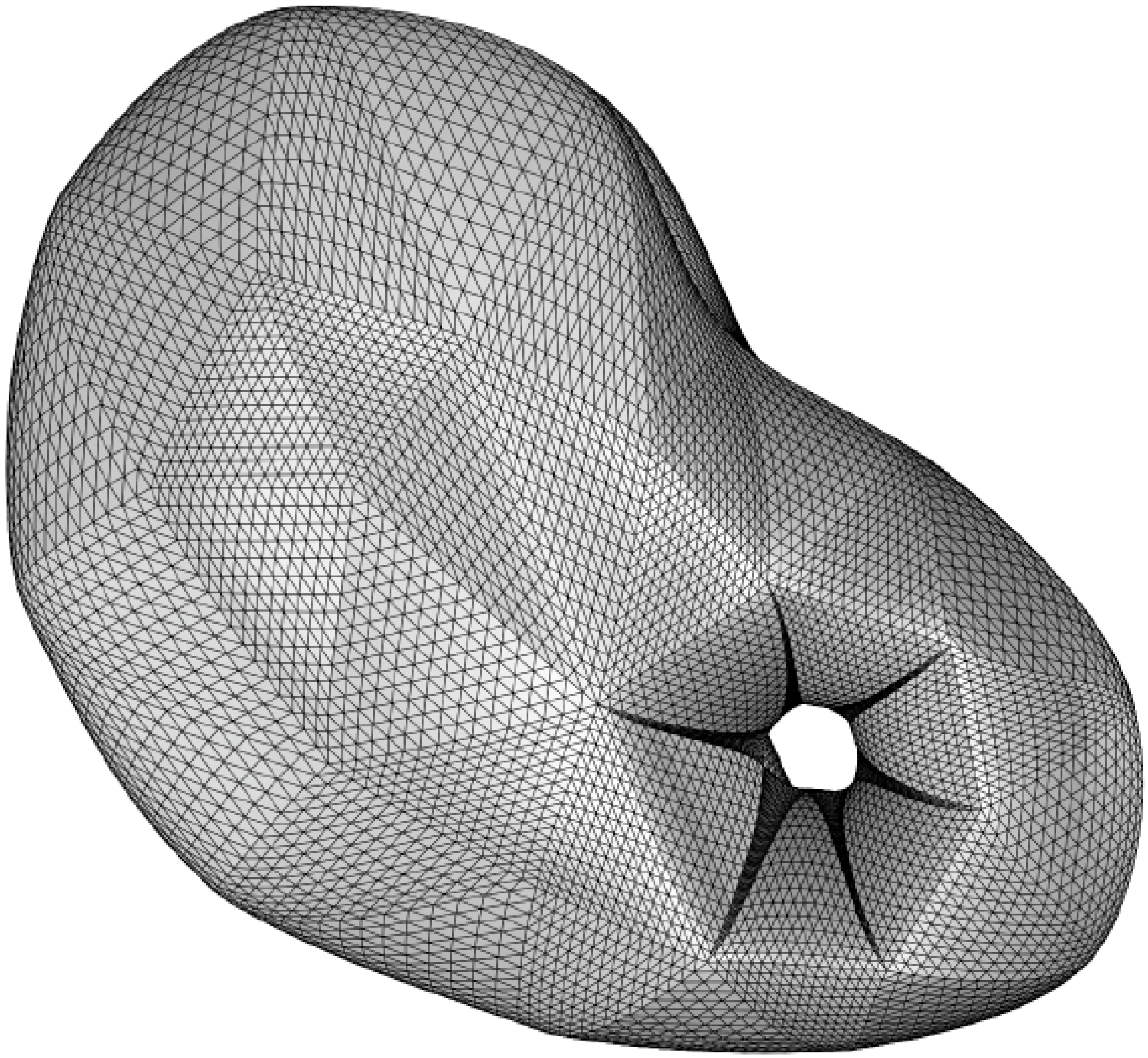
Results



Mesh

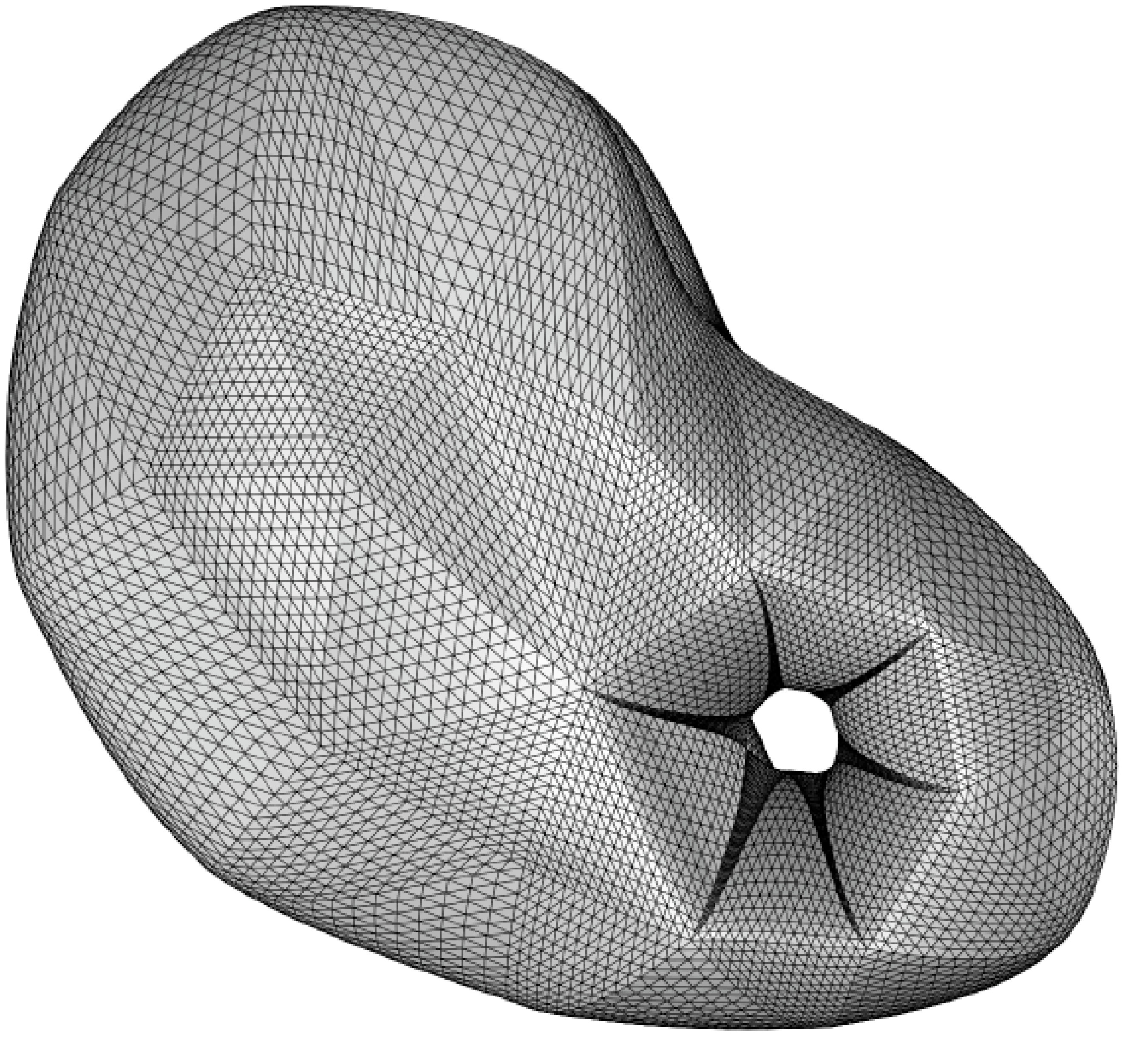
Results

PN triangle

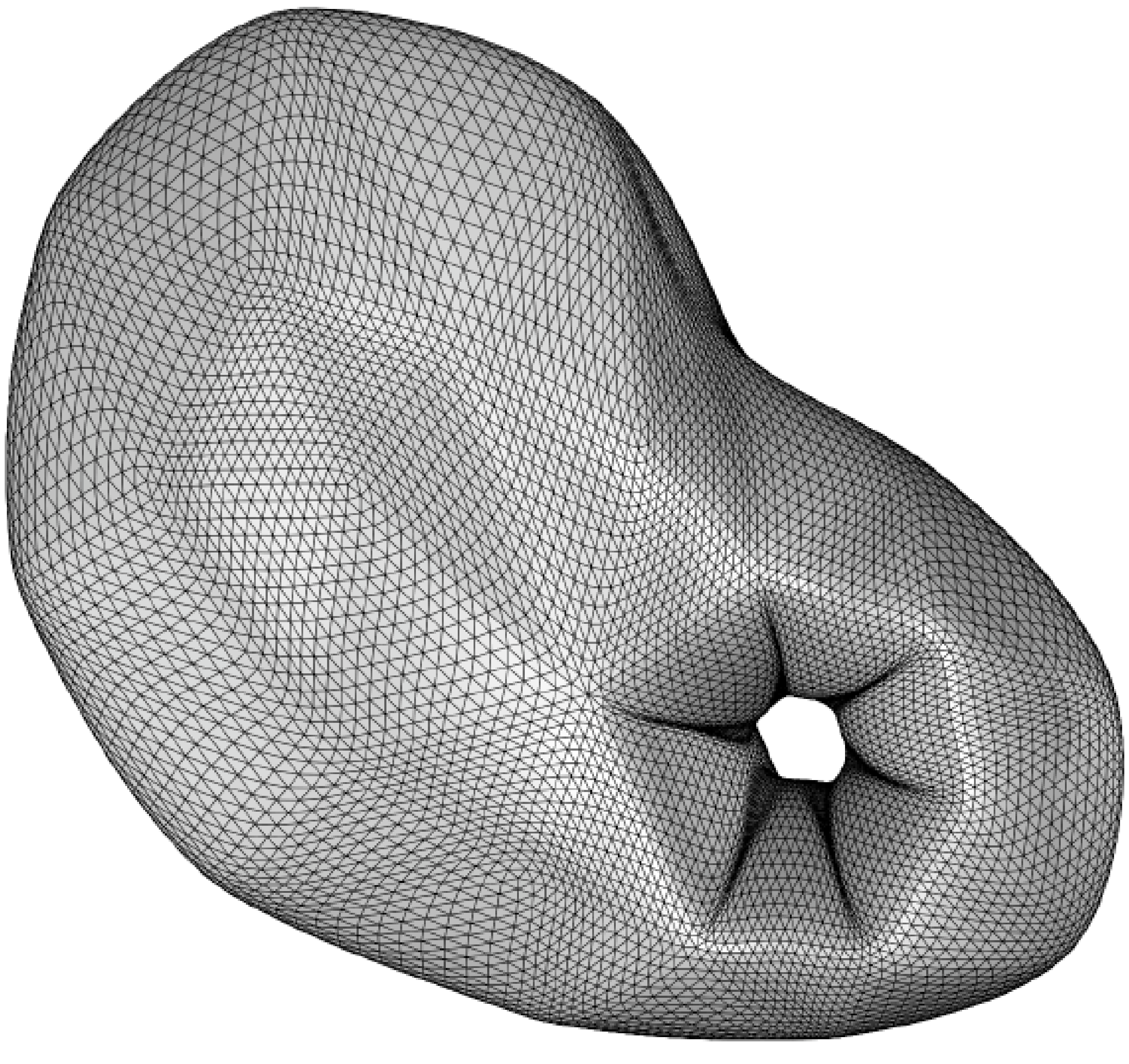


Results

PN triangle



PPS



Results

Results

Mesh

Results

Mesh

PN triangle

Results

Mesh

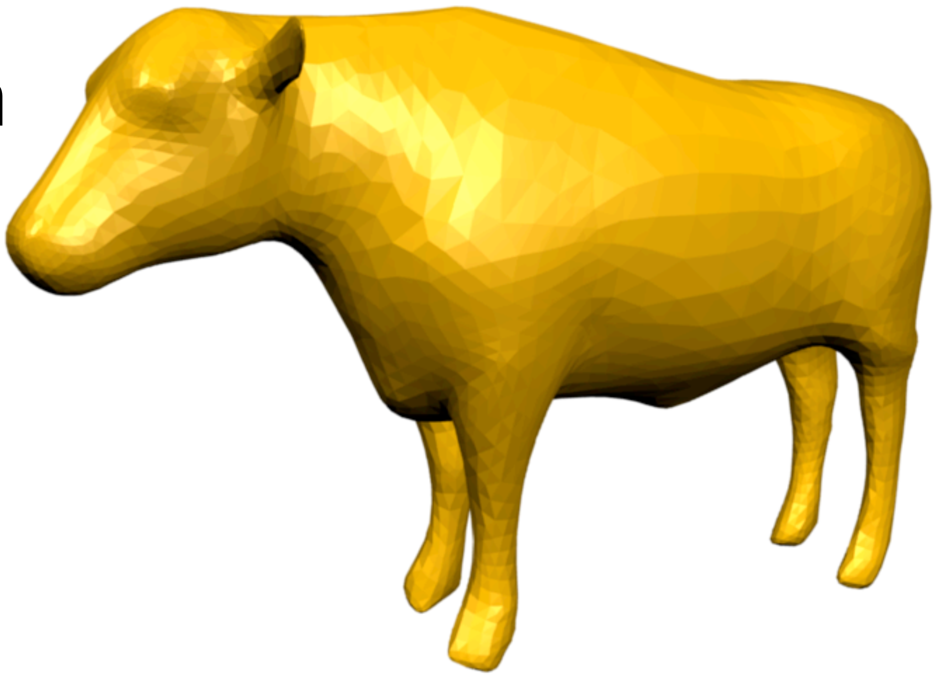
PN triangle

PPS

Results

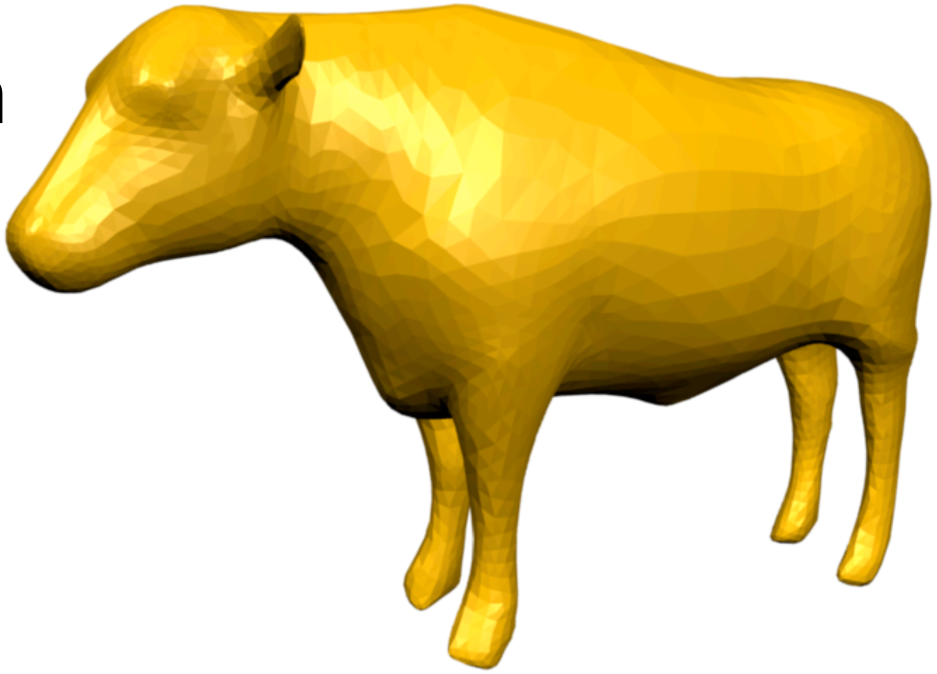
Results

Mesh



Results

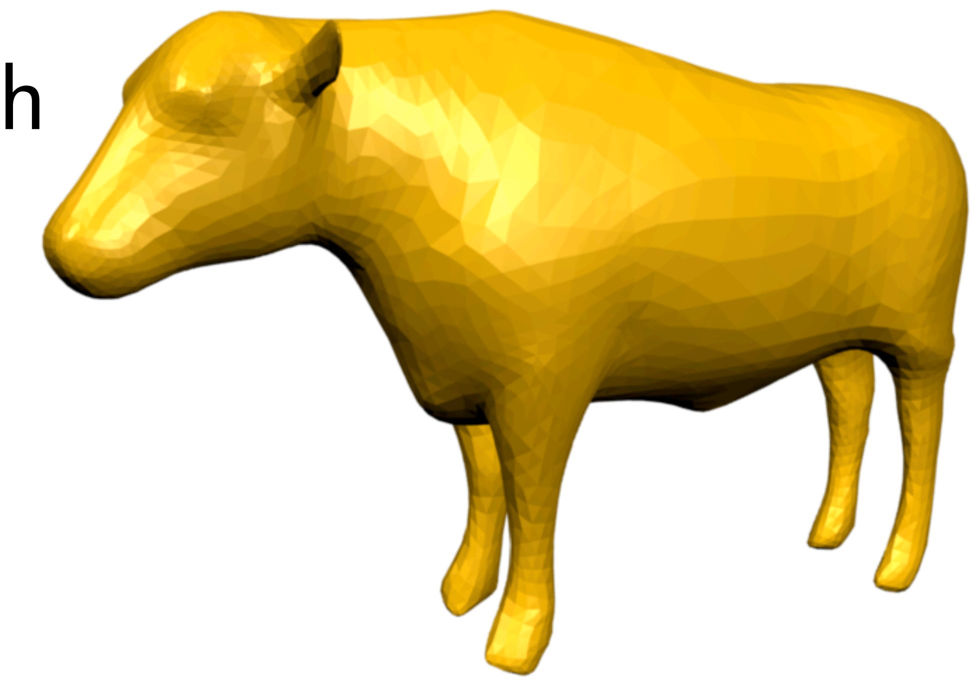
Mesh



PN triangle

Results

Mesh



PN triangle

PPS

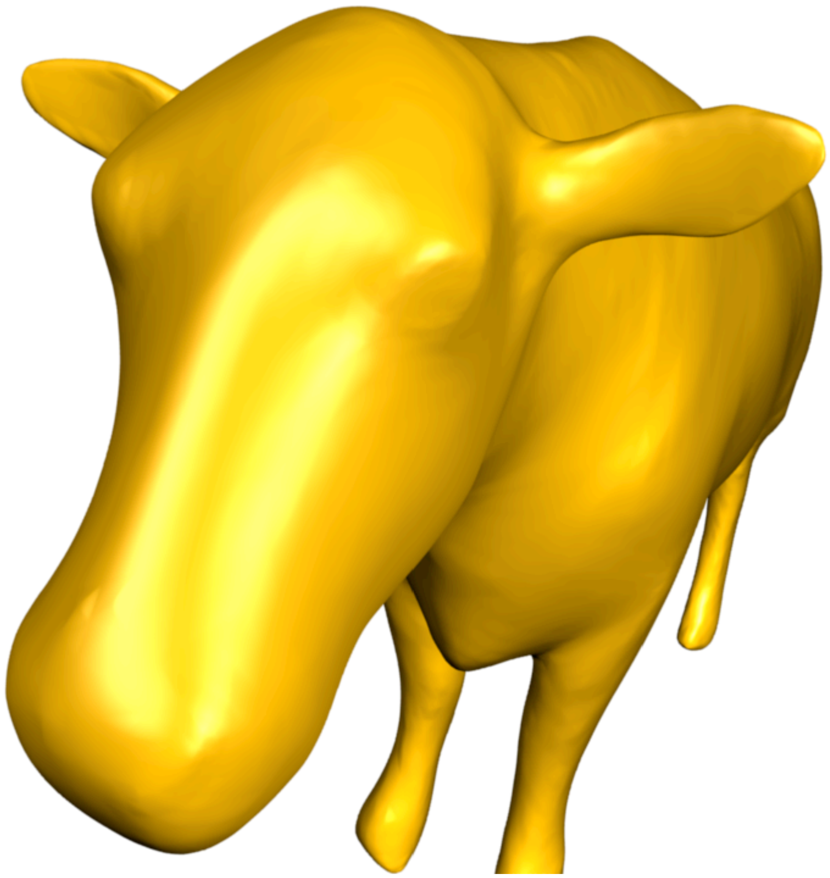
Results

Results

Mesh

Results

Mesh

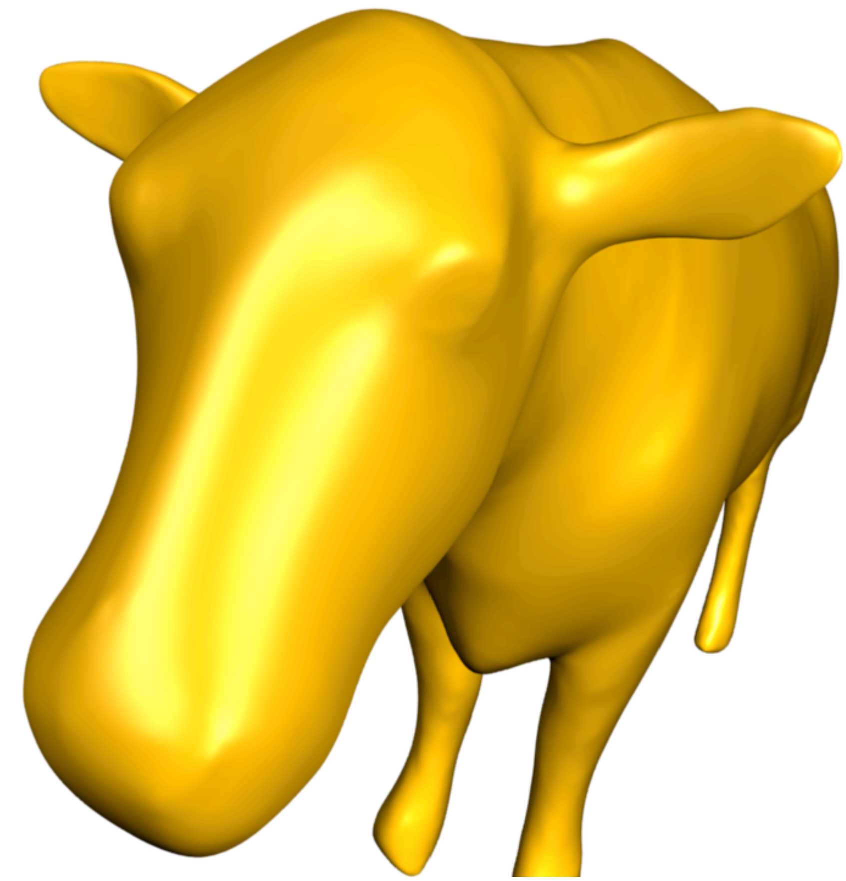


PN triangle

Results

Mesh

PN triangle



PPS

Results

Results

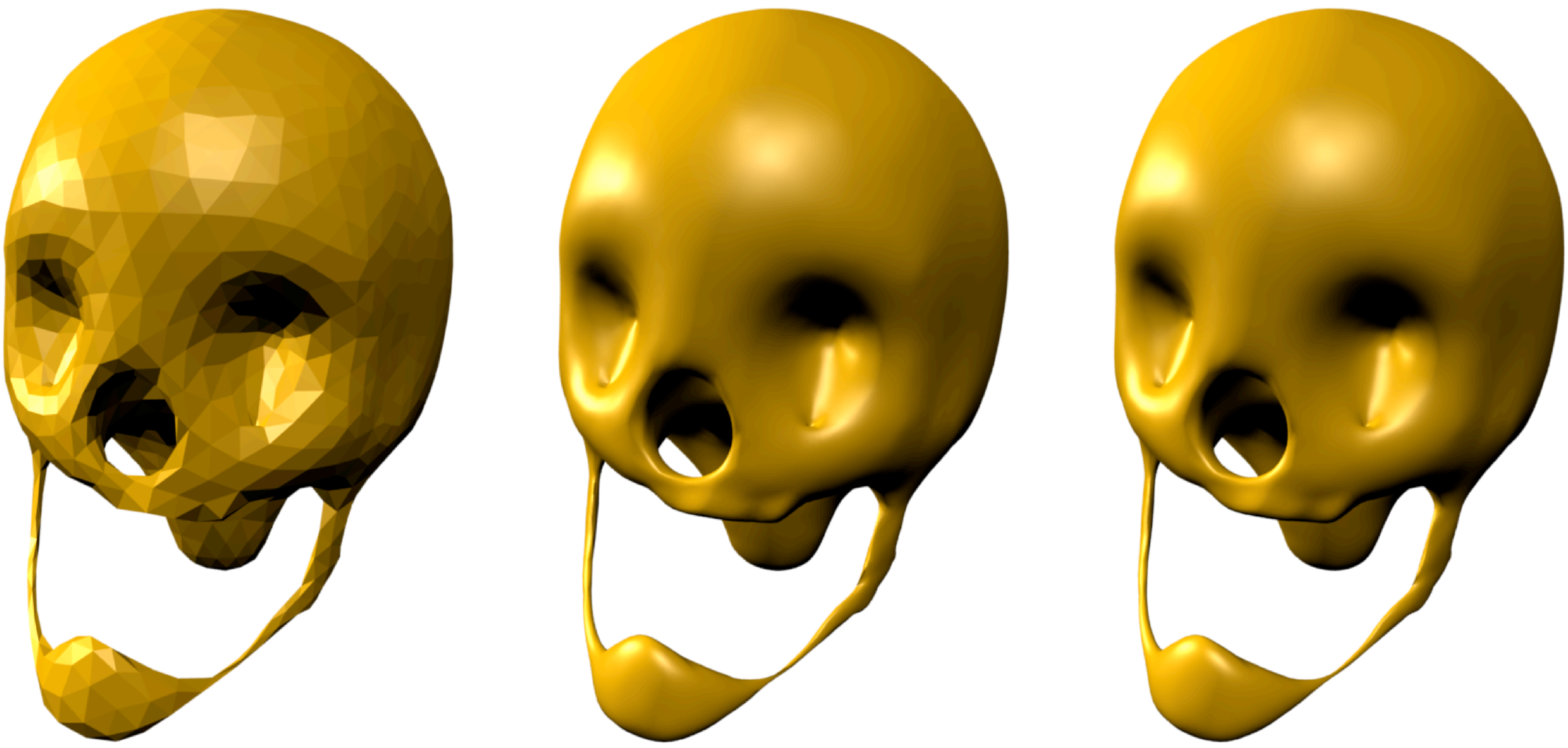
Mesh

Results

Mesh

PN triangle

Results



Mesh PN triangle PPS

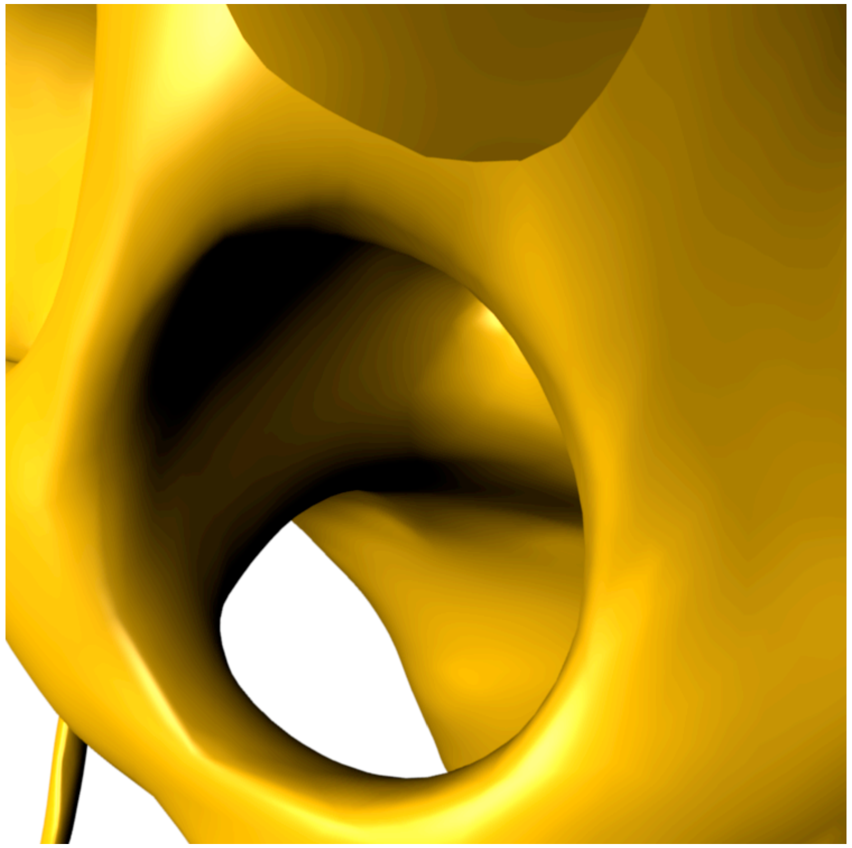
Results

Results

Mesh

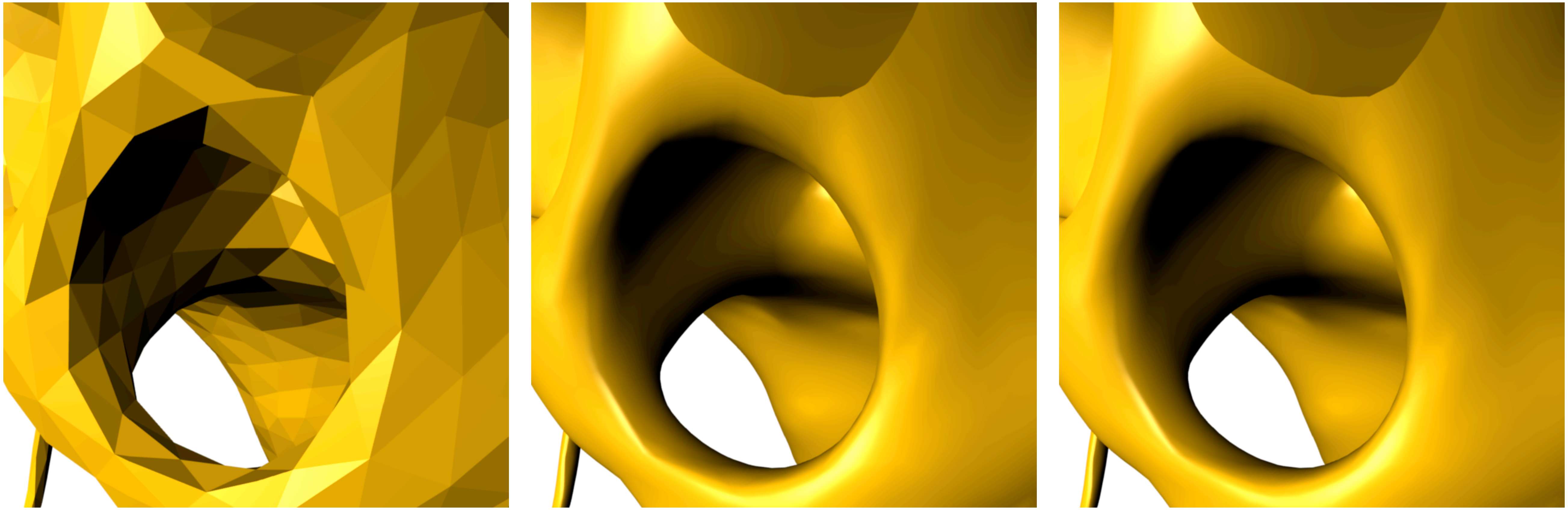
Results

Mesh



PN triangle

Results



Mesh PN triangle PPS

Conclusions

Conclusions

The image of our C^k parametric pseudo-surface is given by

$$M = \bigcup_{(\sigma,v)} \theta_{(\sigma,v)}(\Omega_{(\sigma,v)})\,.$$

Conclusions

The image of our C^k parametric pseudo-surface is given by

$$M = \bigcup_{(\sigma,v)} \theta_{(\sigma,v)}(\Omega_{(\sigma,v)})\,.$$

The map $\theta_{(\sigma,v)}$ is actually C^∞.

Conclusions

The image of our C^k parametric pseudo-surface is given by

$$M = \bigcup_{(\sigma,v)} \theta_{(\sigma,v)}(\Omega_{(\sigma,v)})\,.$$

The map $\theta_{(\sigma,v)}$ is actually C^∞.

There are $3 \times n_t$ p-domains and Bézier patches in our construction, where n_t is the number of triangles of the input mesh, S_T.

Conclusions

Conclusions

Unfortunately, the map $\theta_{(\sigma,v)}$ is NOT polynomial.

Conclusions

Unfortunately, the map $\theta_{(\sigma,v)}$ is NOT polynomial.

OPEN PROBLEM: Can we make it polynomial?

Conclusions

Conclusions

Recall that

$$\theta_{(\sigma,v)}(p) = \sum_{(\tau,u)\in J(p)} \nu_{(\tau,u)}(p) \cdot \psi_{(\tau,u)}(\varphi_{(\sigma,v)(\tau,u)}(p))\,,$$

where

$$\nu_{(\tau,u)}(p) = \frac{\gamma_{(\tau,u)}(\varphi_{(\tau,u)(\sigma,v)}(p))}{\sum_{(\eta,w)\in J(p)} \gamma_{(\eta,w)}(\varphi_{(\eta,w)(\sigma,v)}(p))}$$

and

$$J(p) = \{(\eta, w) \in I \mid p \in \Omega_{(\sigma,v)(\eta,w)}\}\,.$$

Conclusions

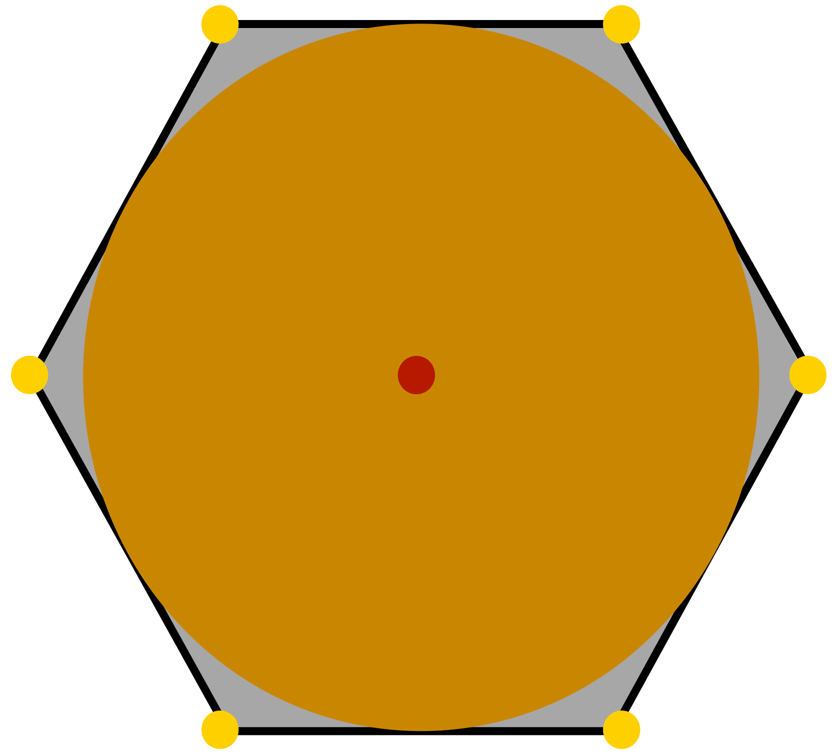
We can easily make $\gamma_{(\tau,u)}$ a C^k rational polynomial, for any finite k.

However, the difficult lies in making $\varphi_{(\tau,u)(\sigma,v)}$ (rational) polynomial!.

Conclusions

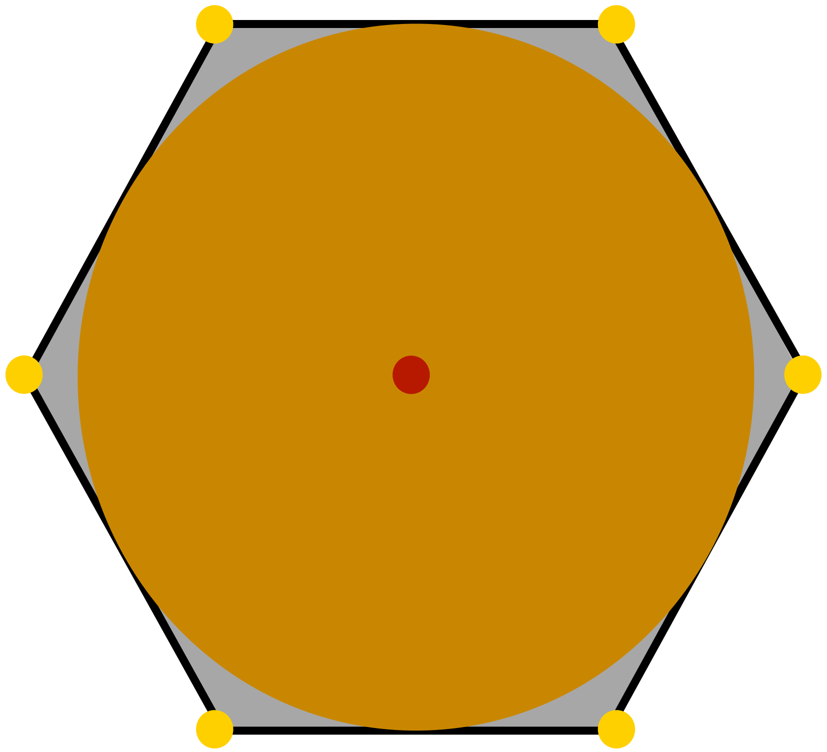
Conclusions

We can create a much simpler construction by letting the p-domains be the inscribed circles of the P-polygons, as shown below:



Conclusions

We can create a much simpler construction by letting the p-domains be the inscribed circles of the P-polygons, as shown below:

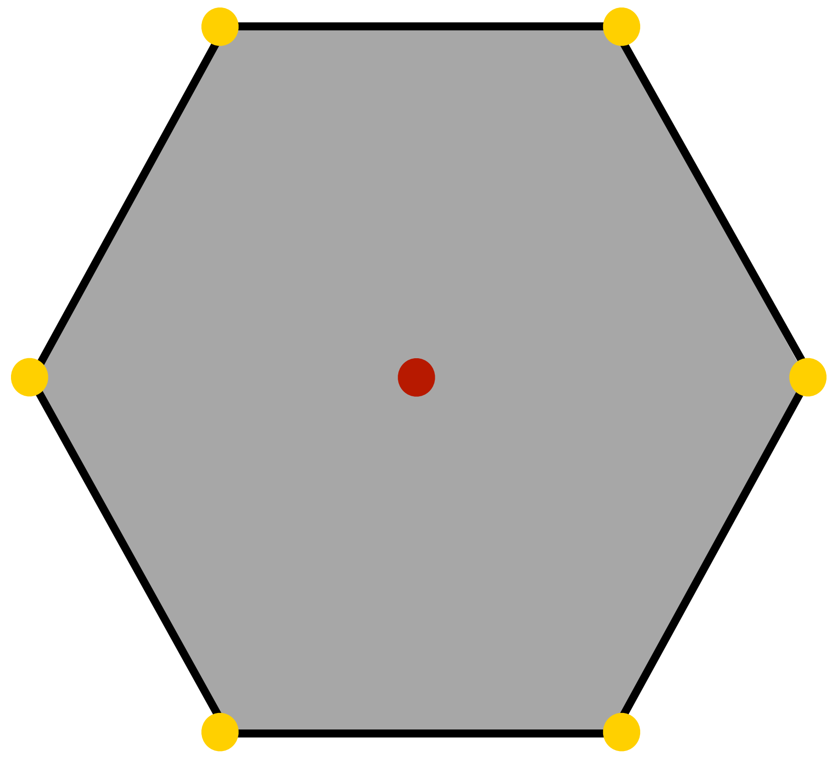


The transition maps do not change, but the shape functions do!

Conclusions

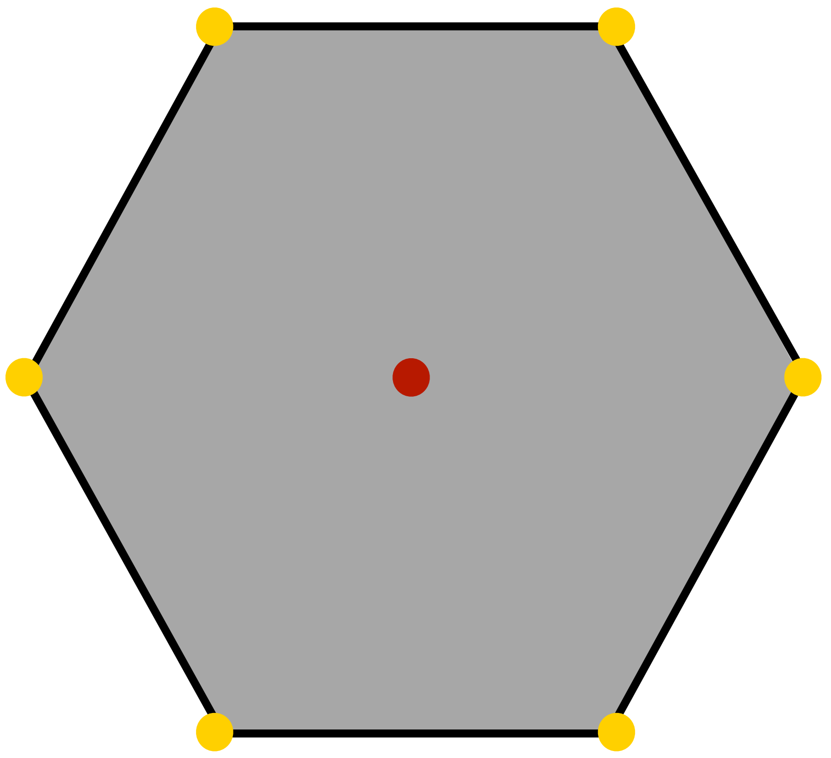
Conclusions

Why didn't we let the interior of the P-polygons be the p-domains?



Conclusions

Why didn't we let the interior of the P-polygons be the p-domains?

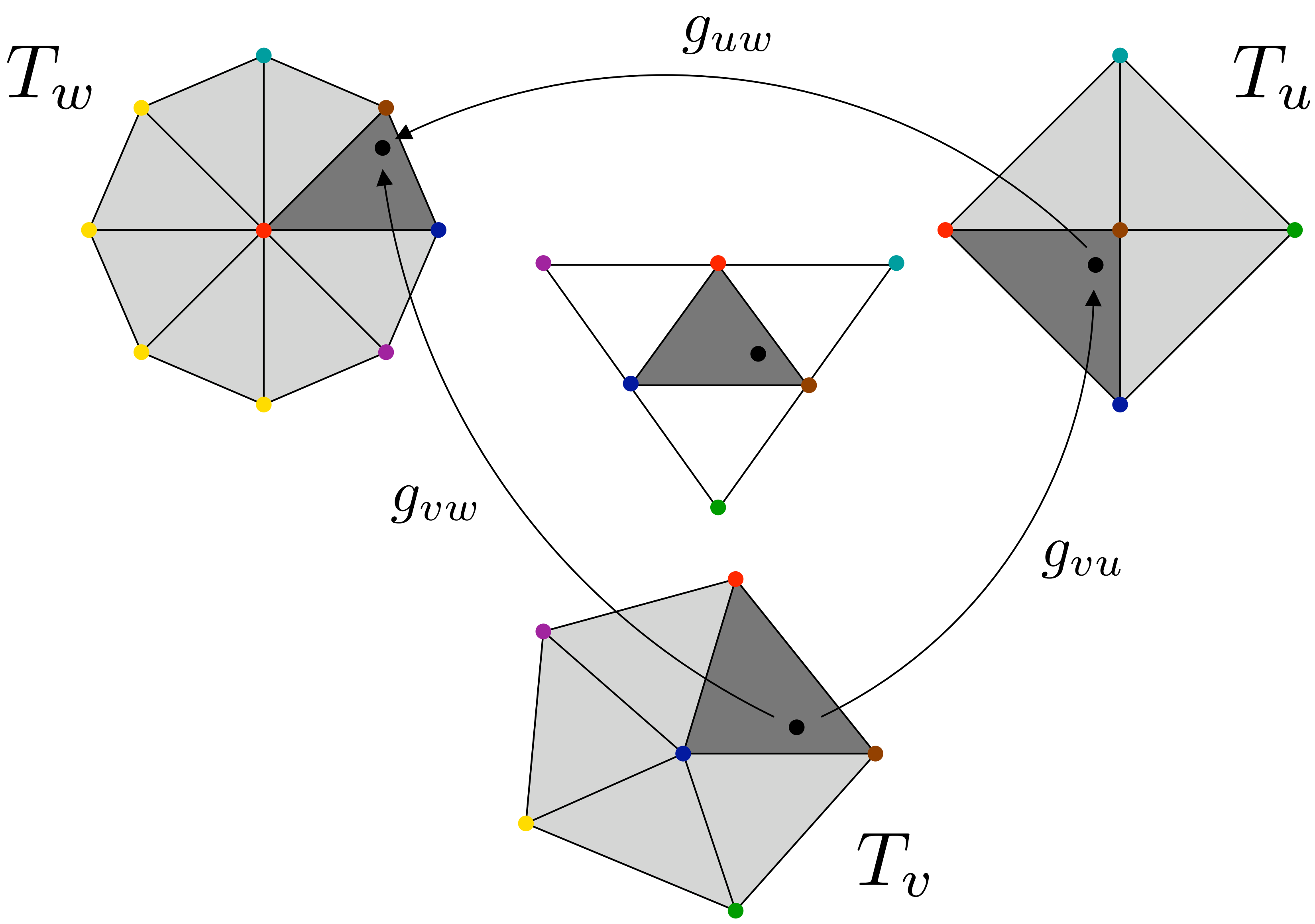


Simple answer: we failed to figure out the transition maps!

Conclusions

Conclusions

OPEN PROBLEM: Can you find a **simple** C^∞ bijective map g satisfying $g_{vw} = g_{uw} \circ g_{vu}$ (this has to do with the cocycle condition)?



Conclusions

Conclusions

For a good survey on the existing constructions, see

- Cindy M. Grimm and Denis Zorin. Surface Modeling and Parametrization with Manifolds. In ACM SIGGRAPH 2006 Courses (SIGGRAPH'06), pages 1-81, New York, NY, USA, 2006. ACM Press.

Adaptive Manifold Fitting (Part I)

Luiz Velho
IMPA

Outline

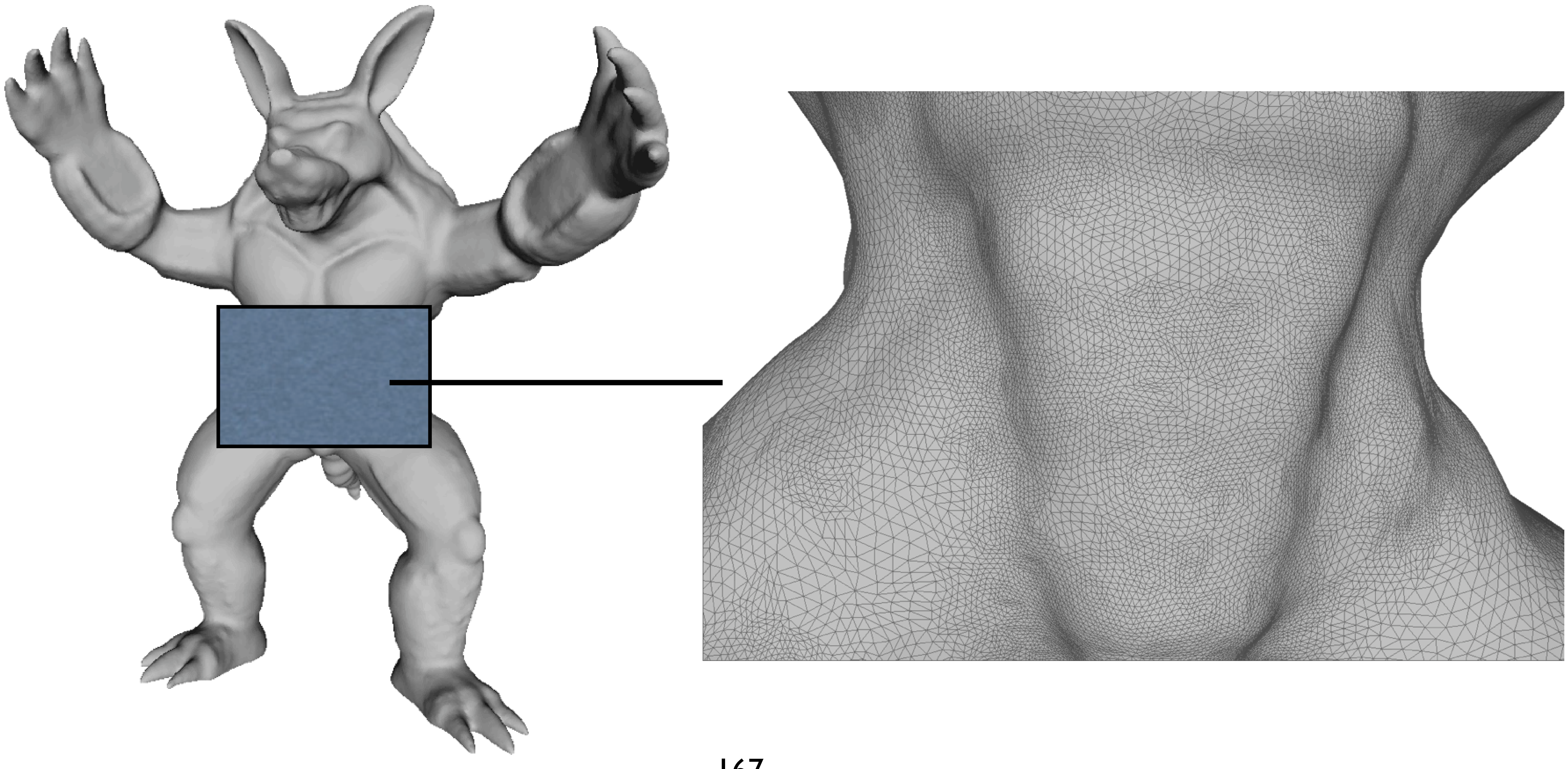
- Fitting Surfaces to Very Large Meshes
- Multiresolution Operators
- Building Base Meshes by Simplification
- Adaptive Mesh Refinement
- Conclusions

Surface Fitting

- Very Large Meshes (10^6 vertices)
  - Challenging Problem!

Surface Fitting

- Very Large Meshes (10^6 vertices)
  - Challenging Problem!



Manifolds and Fitting

- Basic Setting
  - Gluing Data proportional to Mesh Size

- Problem: *Very Large Meshes*
  - Computationally Inefficient
  - Do not Exploit Approximation Power
- Solution:
  - Adaptation

Adaptive Fitting

- Optimization Formulation:
  - Given an Approximation Error ϵ
  - Find $\mathcal{M}$ with Smallest Number of Charts

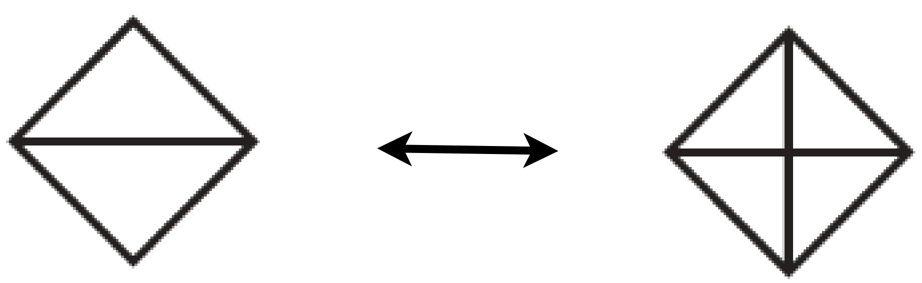
- Strategy:
  - Combine
    - Multiresolution Structure
    - Manifold Surface Approximation

Multiresolution Framework

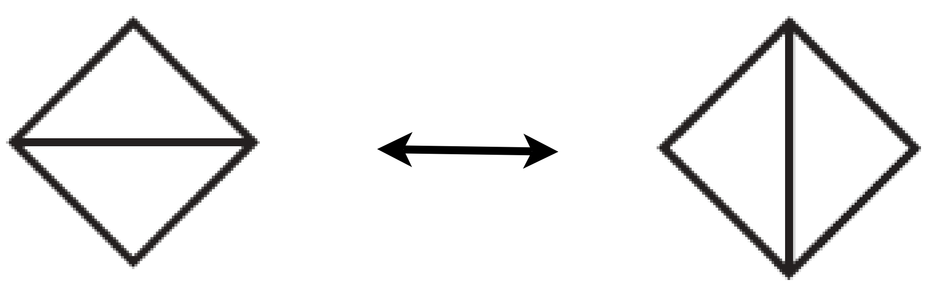
- Simplicial Multi-triangulation
  - Stellar Theory
- Building Base Meshes
  - Surface Simplification
- Adaptive Fitting
  - *4-8* Refinement

Stellar Theory

- Topological Operators
- Edge Split and Weld
  - Change Mesh Resolution



- Edge Flip
  - Change Mesh Connectivity

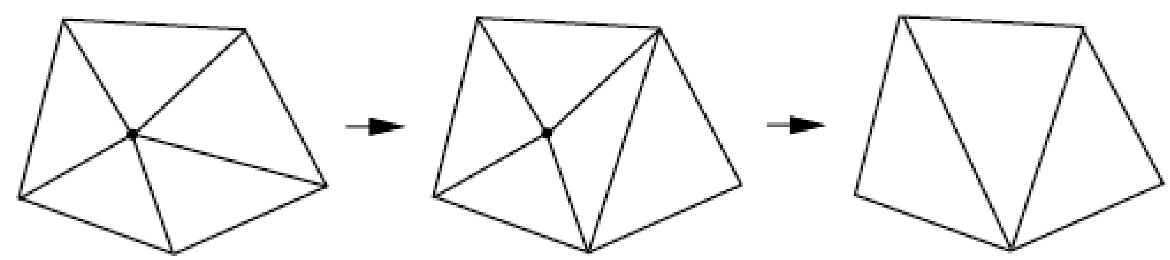


Stellar Simplification

- Basic Elements:
  I. Operator Factorization

     - Edge Collapse

     - Flip + Weld



  II. Quadric Error Metric

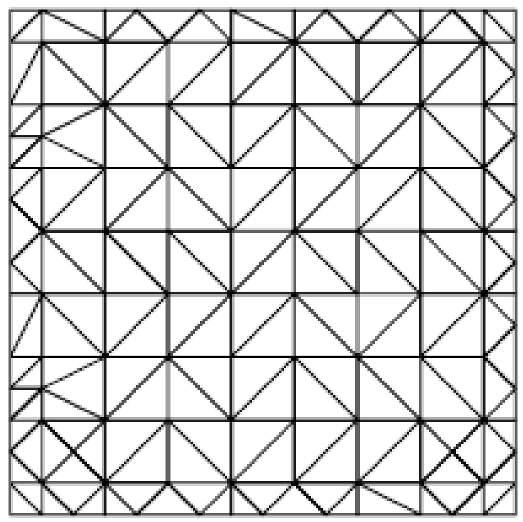
Basic Algorithm

- Repeat for *N* Resolution Levels
  1. Rank Vertices Based on Quadric Error
  2. Select Independent Set of Clusters
  3. Simplify Mesh using Stellar Operators

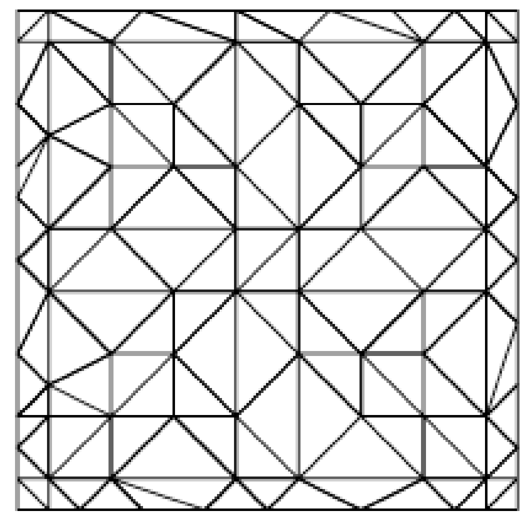
✳ *Properties*

- *Logarithmic Height*
- *Good Aspect Ratios*

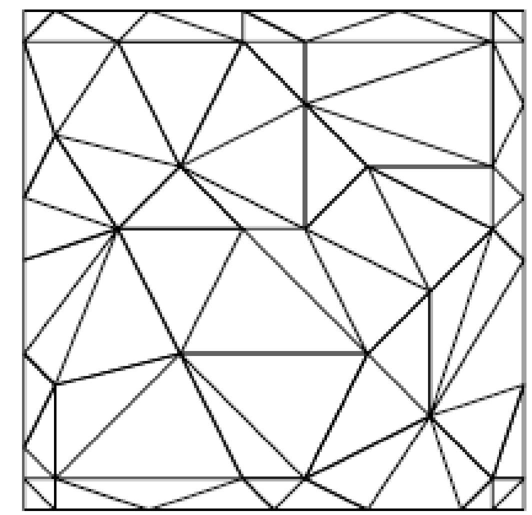
Example 1: Plane



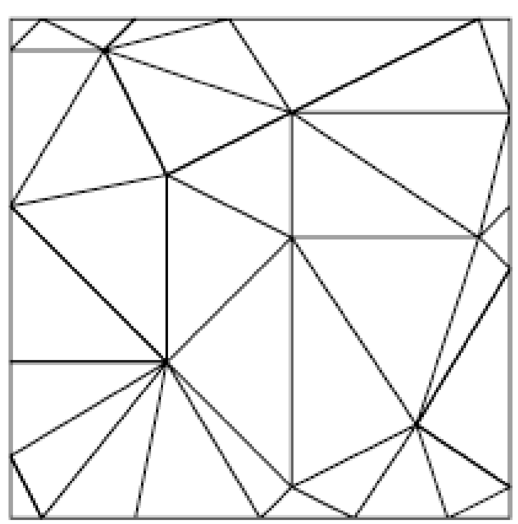
(a) original mesh



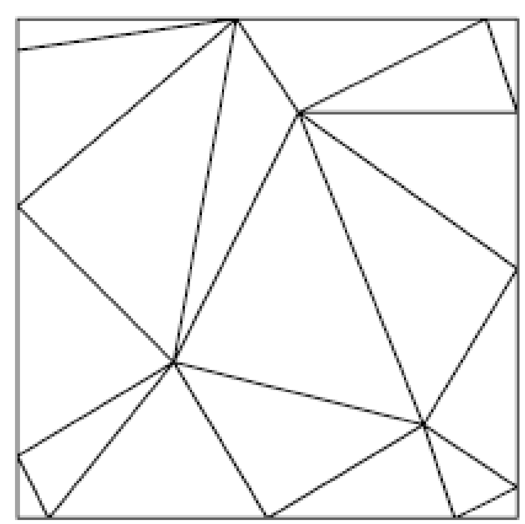
(b) level 1



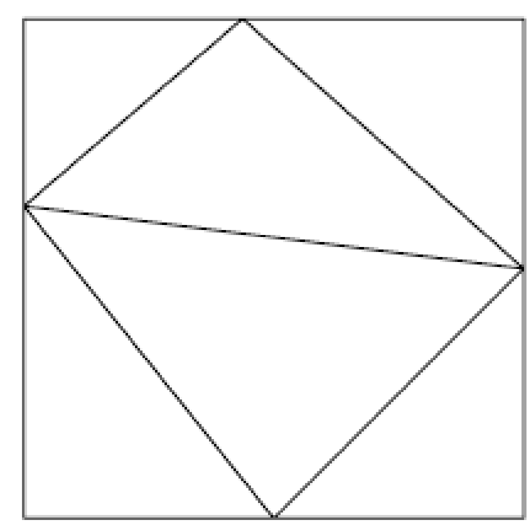
(c) level 3



(d) level 5

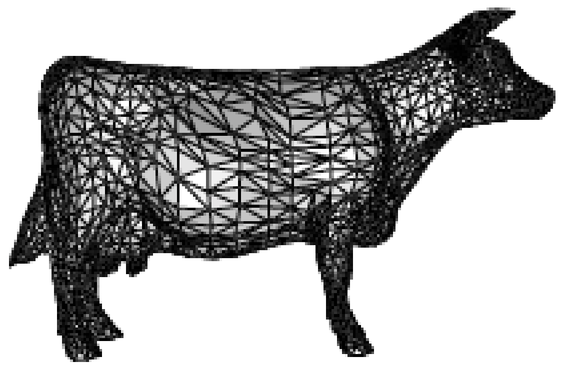


(e) level 7

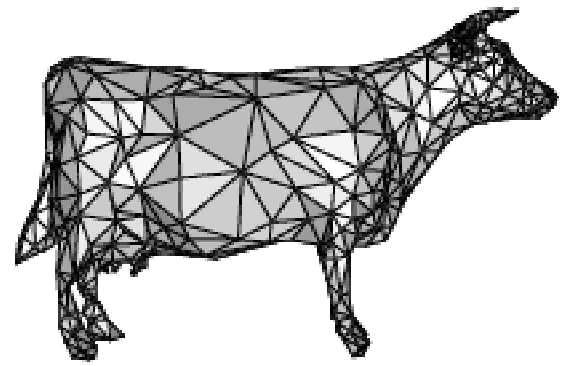


(f) level 9

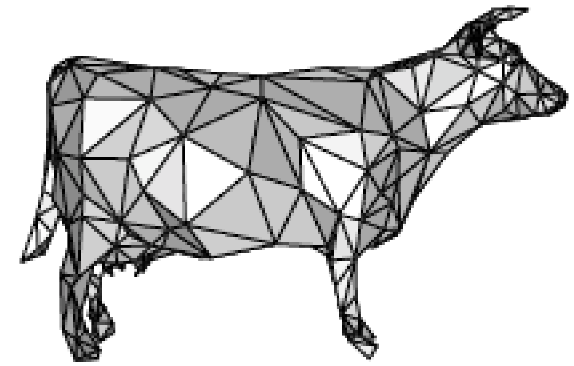
Example 2: Cow



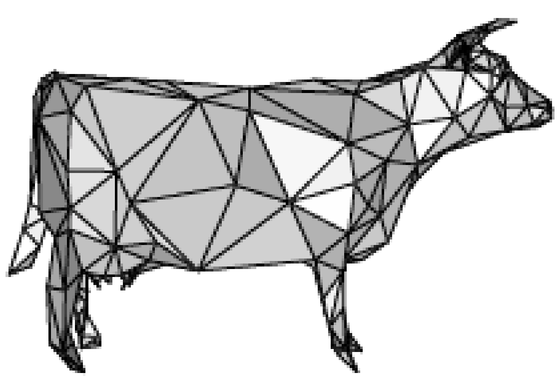
(a) original mesh



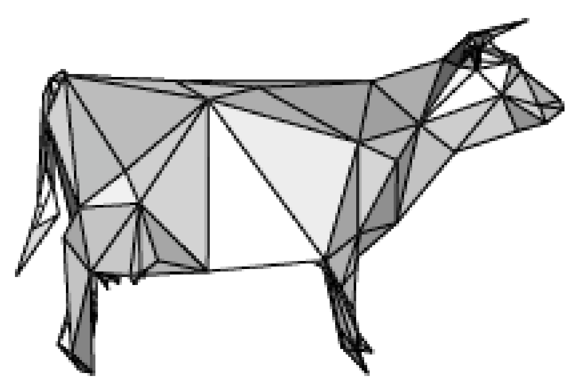
(b) level 1



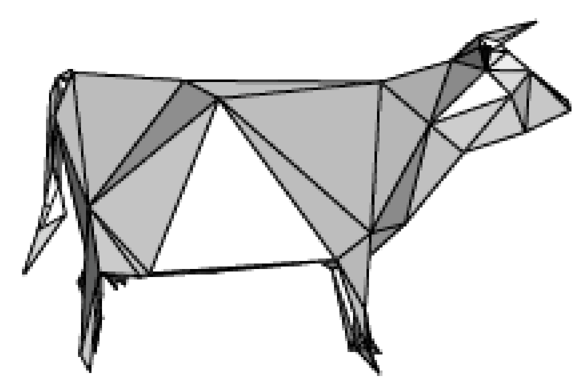
(c) level 3



(d) level 5



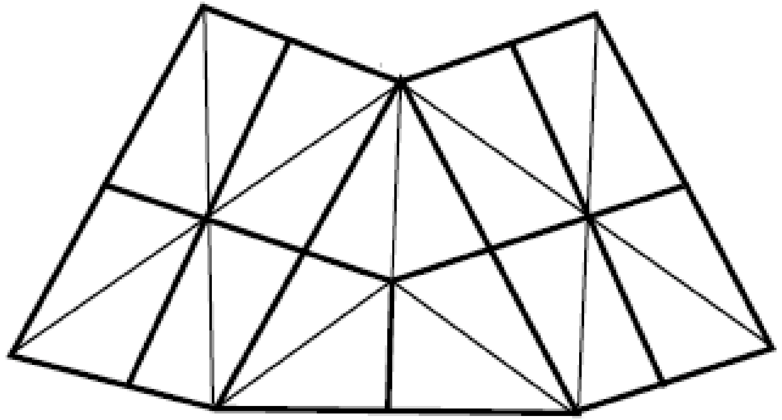
(e) level 7



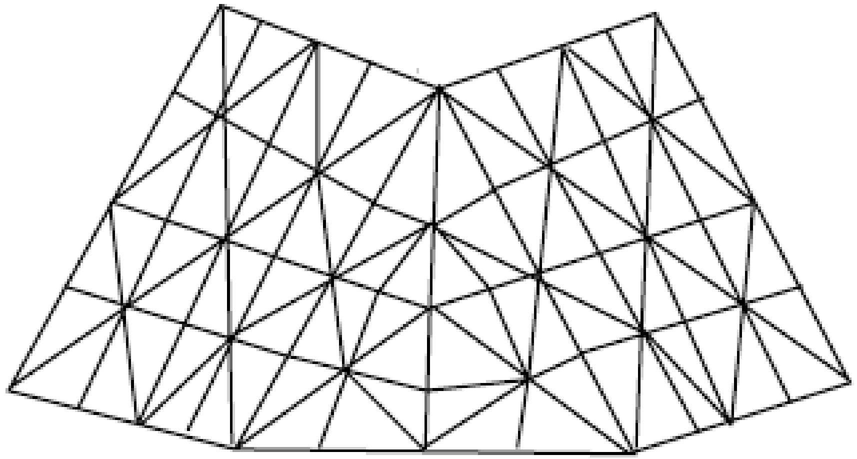
(f) level 9

Variable Resolution Mesh

- Underlying Semi-Regular Structure
  - Tri-quad Base Mesh

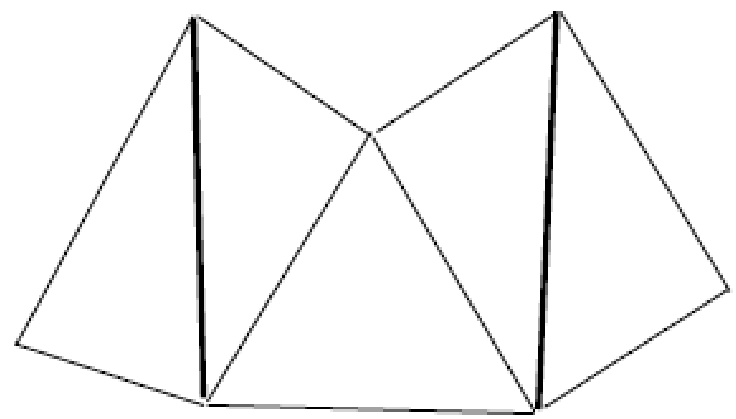


  - *4-8* Subdivision

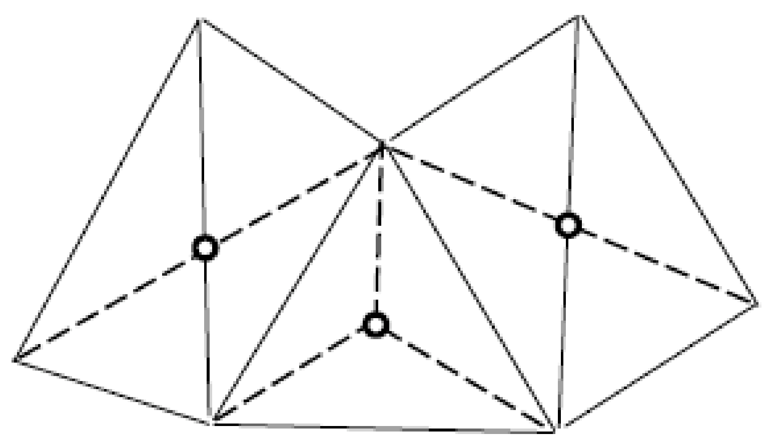


Building the Base Mesh

1. Two-Face Clusters + Single Triangles

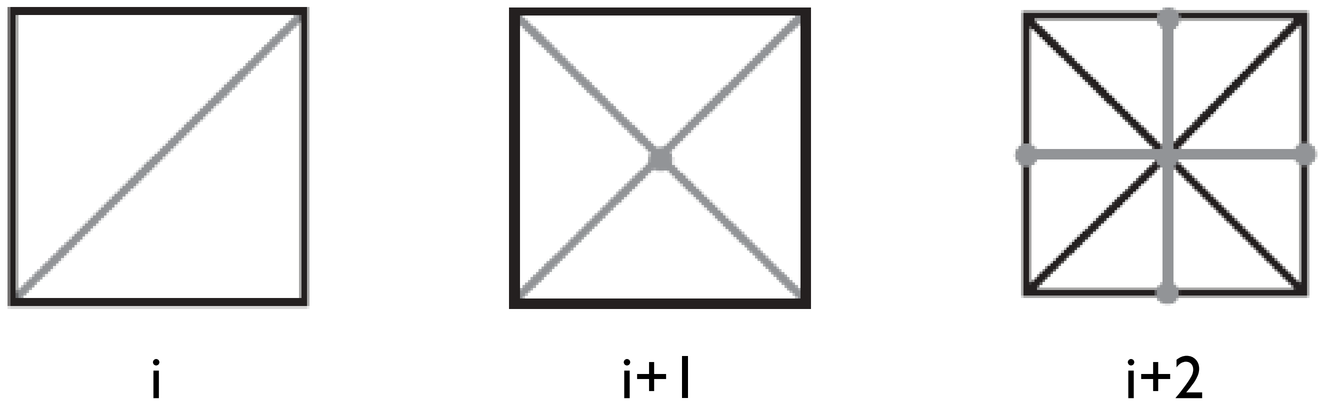


2. Barycenter Subdivision

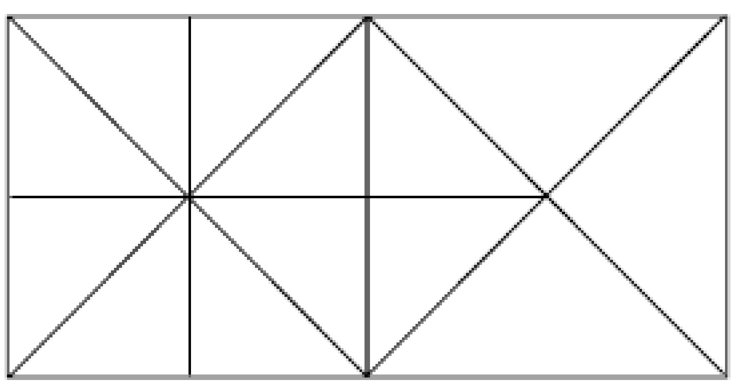


4-8 Subdivision

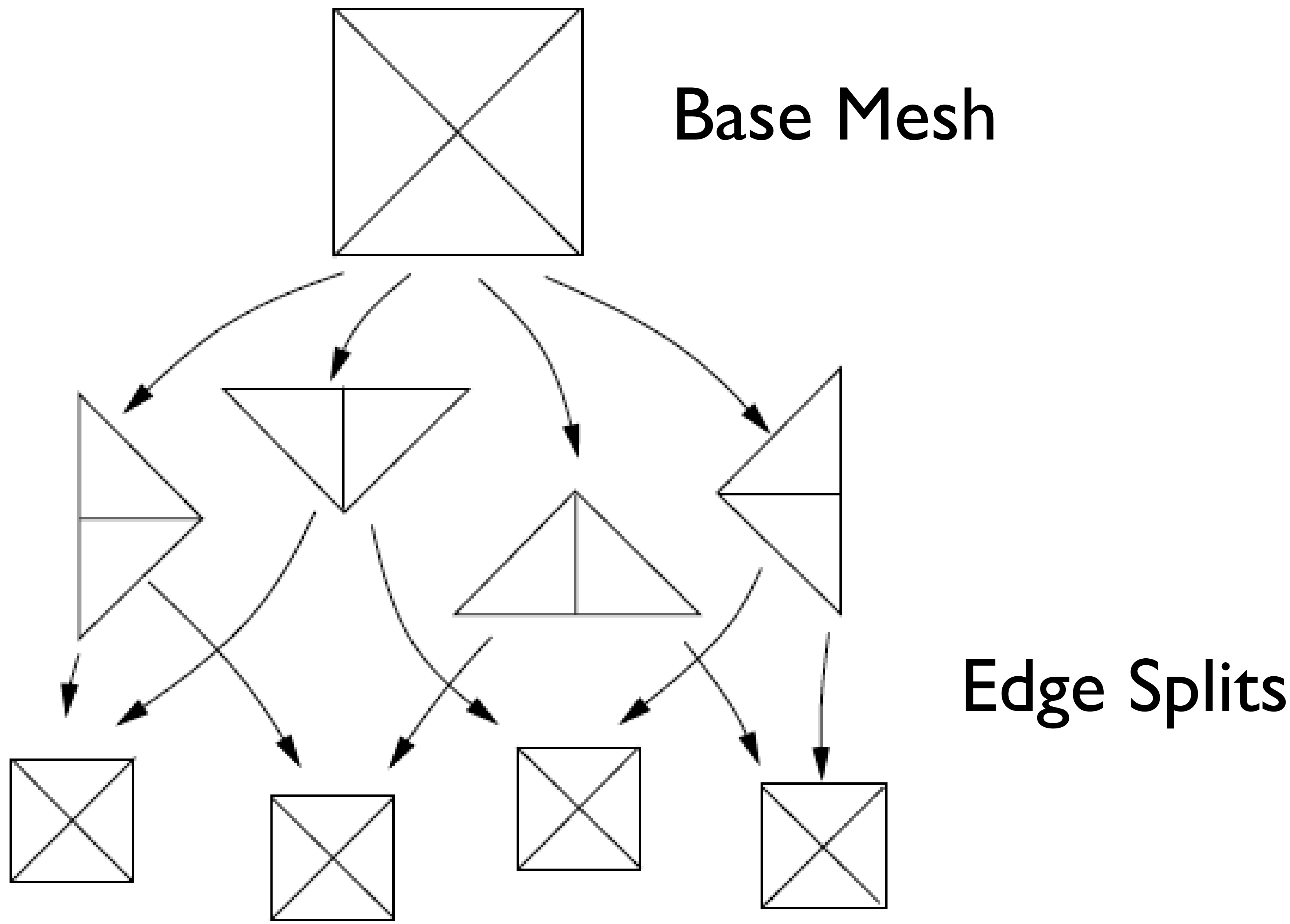
- Interleaved Binary Subdivision



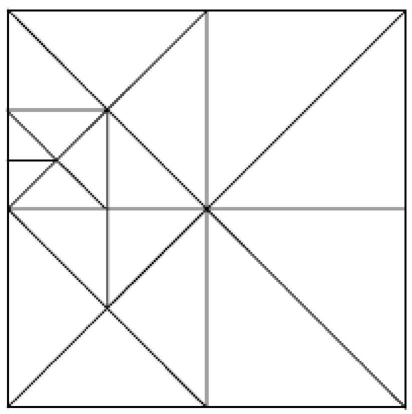
- Non-Uniform Refinement

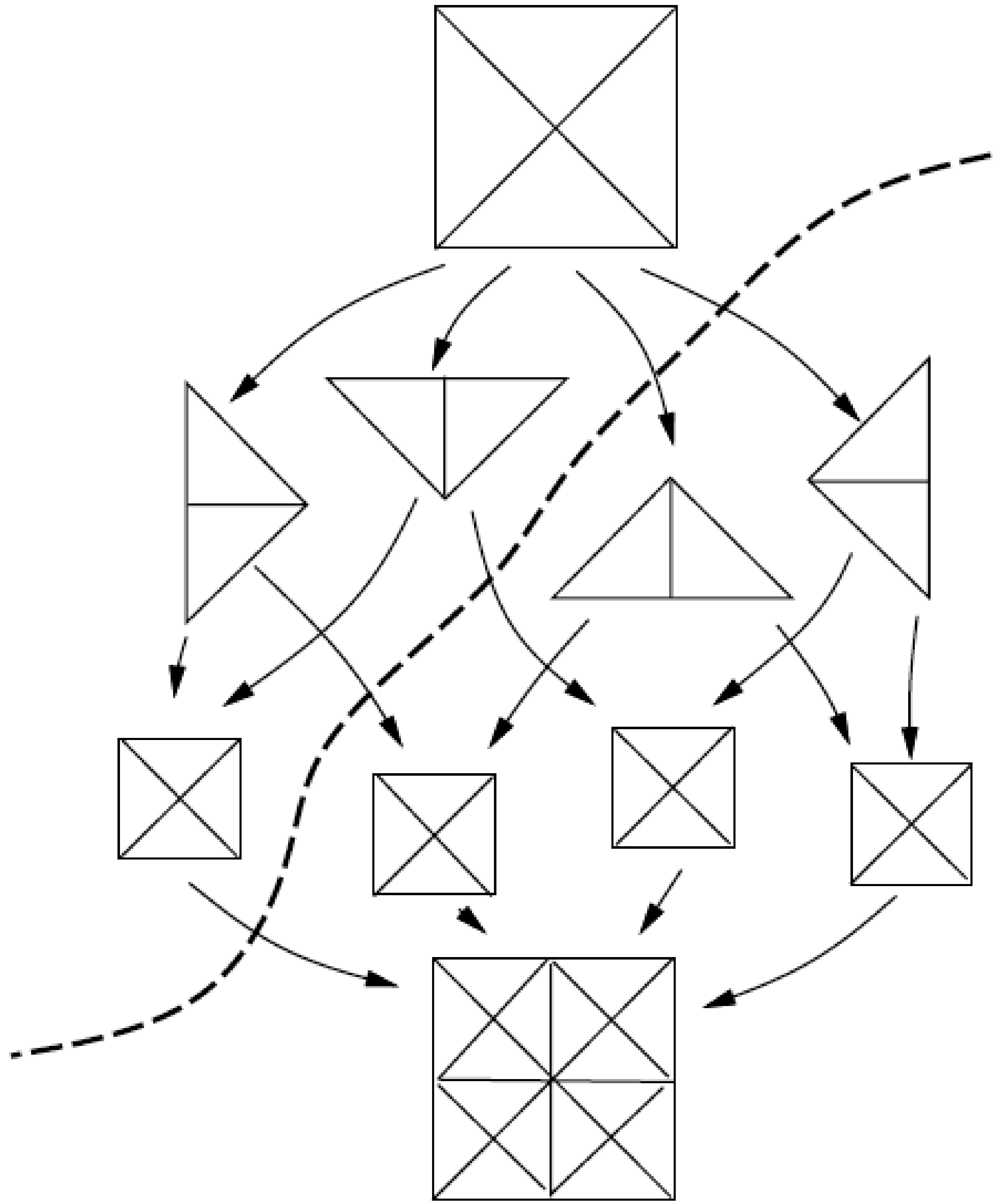


Binary Multi-Triangulation

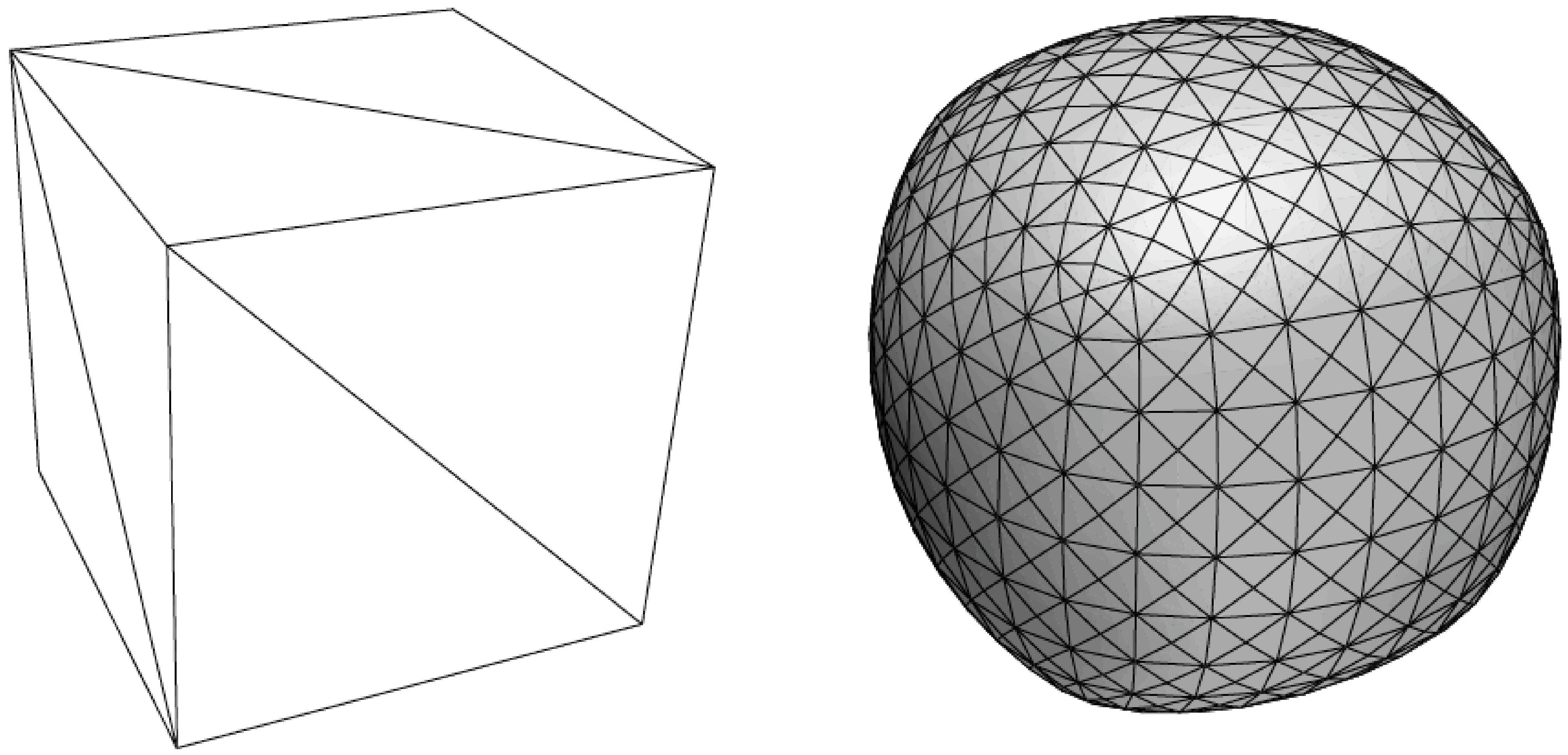


Adaptive Refinement



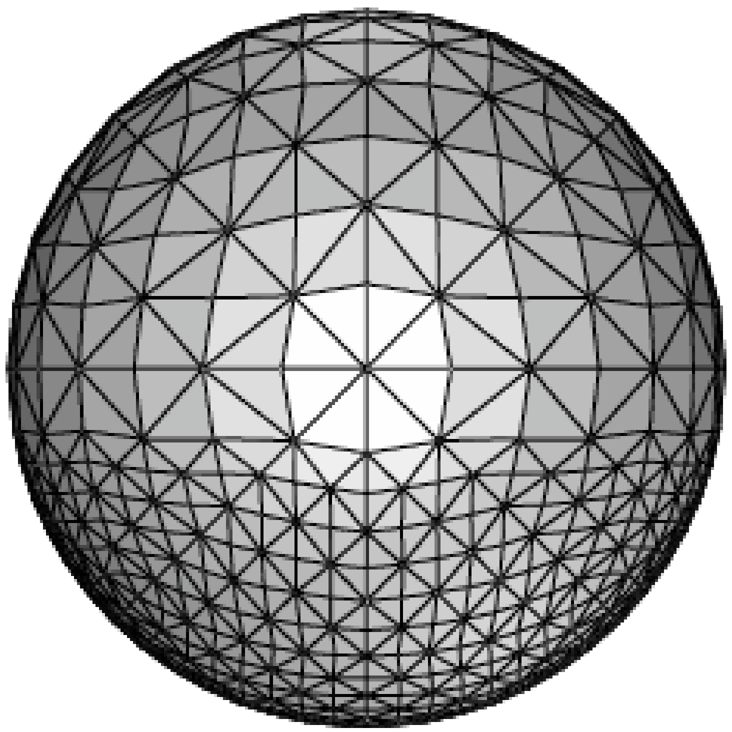


Example I: Uniform

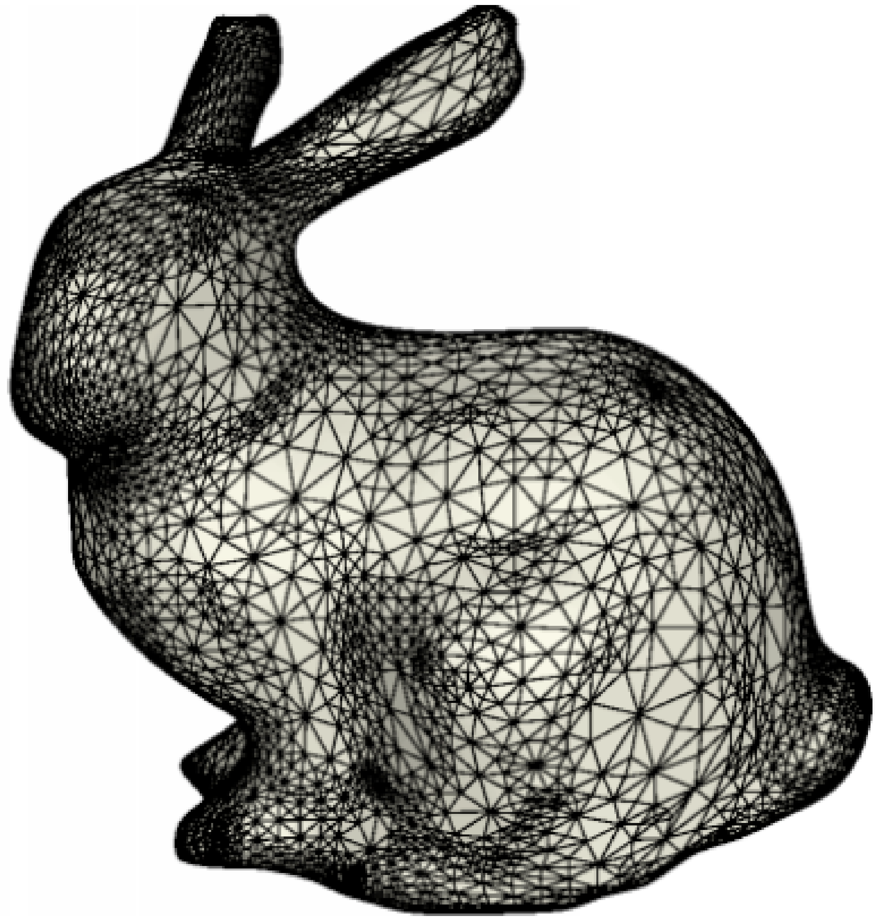


Example 2: Adaptive

- Application-Dependent Criteria



Spatial Selection



<u>Curvature</u>

Conclusions

- Simplicial Multiresolution
  - Powerful Mechanism for Adaptation
- First Part of the Solution for Surface Fitting
  - Simplification
  - Adaptive Refinement
- Second Part (Next)
  - Geodesic Parametrization
  - Bezier Approximation

Adaptive Manifold Fitting (Part II)

Dimas Martínez Morera
UFAL

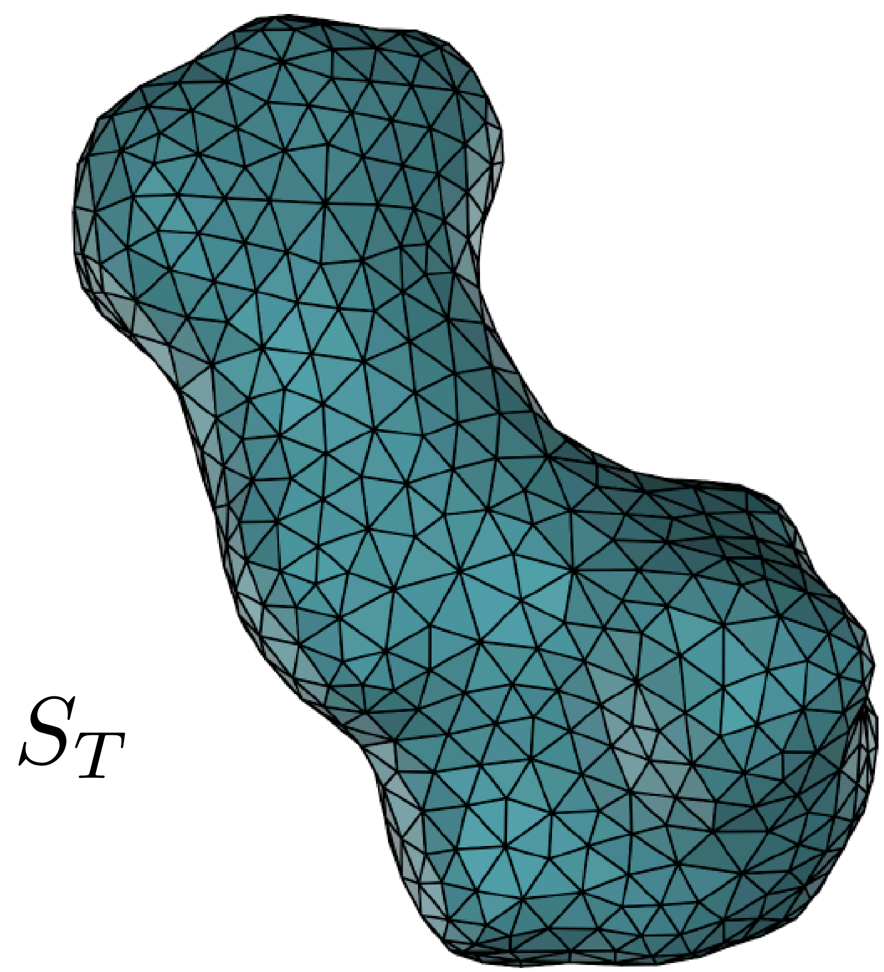
Outline

- The Surface Fitting Problem
- Adaptive Fitting
- Discrete Geodesics
- Conclusions

The Surface Fitting Problem

The Surface Fitting Problem

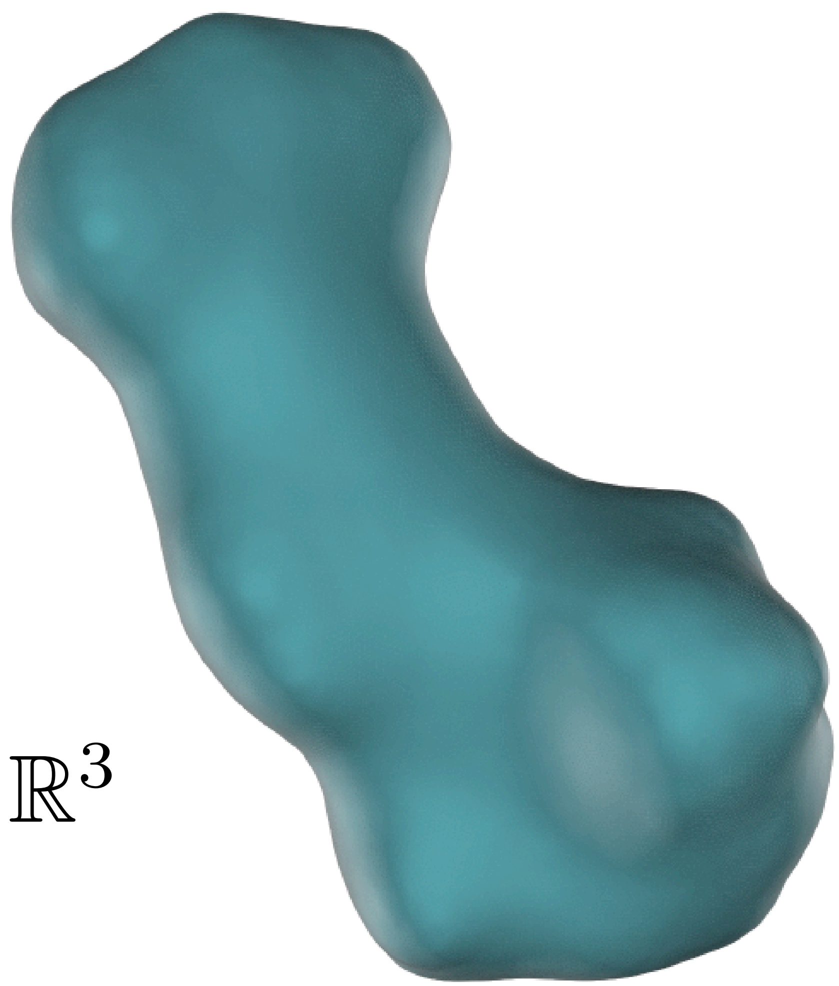
We are a given a piecewise-linear surface, S_T, in $\mathbb{R}^3$, with an empty boundary, a positive integer k, and a positive number ϵ, ...



The Surface Fitting Problem

The Surface Fitting Problem

We want to find a C^k surface $S \subset \mathbb{R}^3$ ...



$S \subset \mathbb{R}^3$

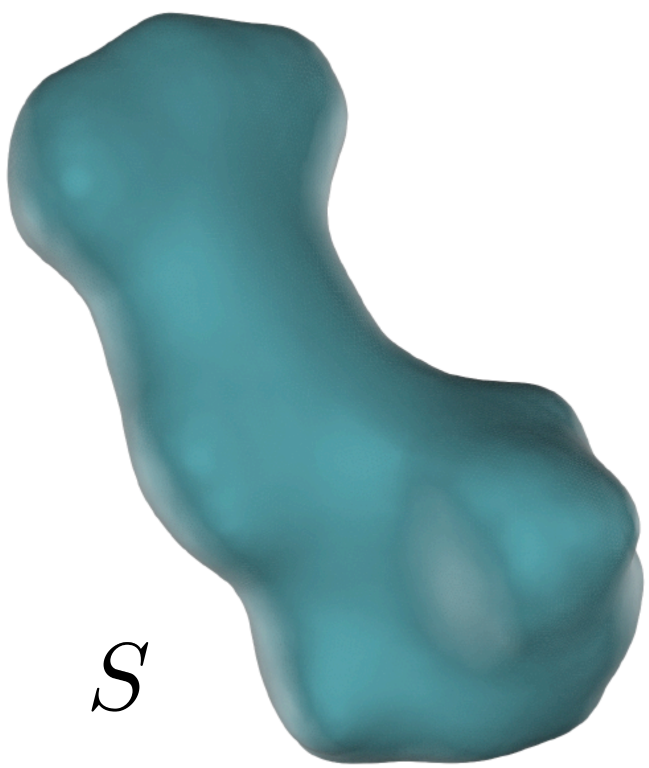
The Surface Fitting Problem

The Surface Fitting Problem

such that there exists a homeomorphism, $h : S \rightarrow |S_T|$, satisfying

$$\|h(v) - v\| \leq \epsilon \,,$$

for every vertex v of S_T.

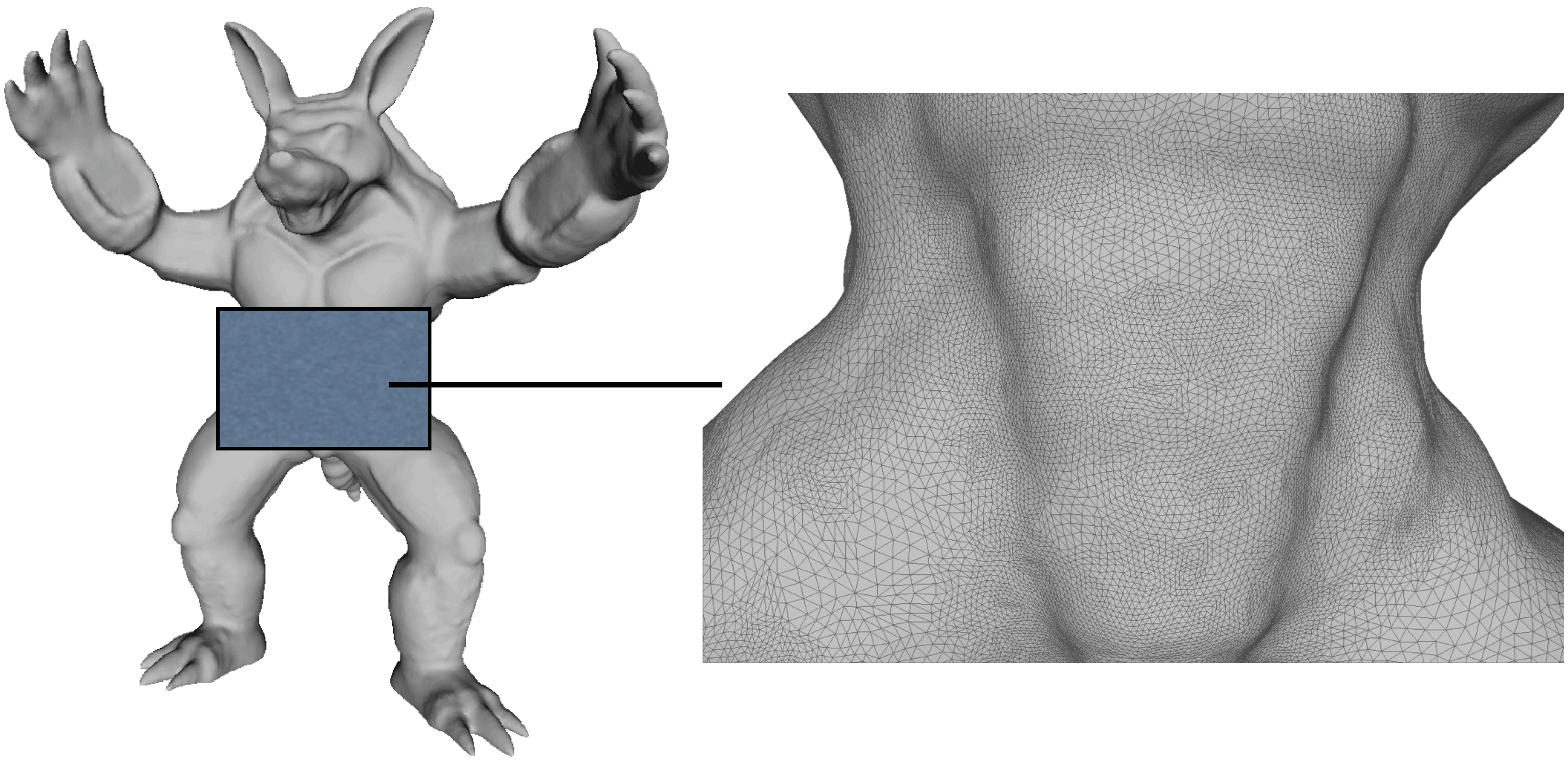


The Surface Fitting Problem

The Surface Fitting Problem

REMARK:

S_T is expected to be "very large" ($\sim 10^6$ vertices).



Adaptive Fitting

Adaptive Fitting

PIPELINE

Adaptive Fitting

S_T

PIPELINE

Adaptive Fitting

$$S_T$$

$$\downarrow$$

$$\boxed{\tilde{S}_T = \text{Simplify } S_T}$$

PIPELINE

Adaptive Fitting

S_T

$\downarrow$

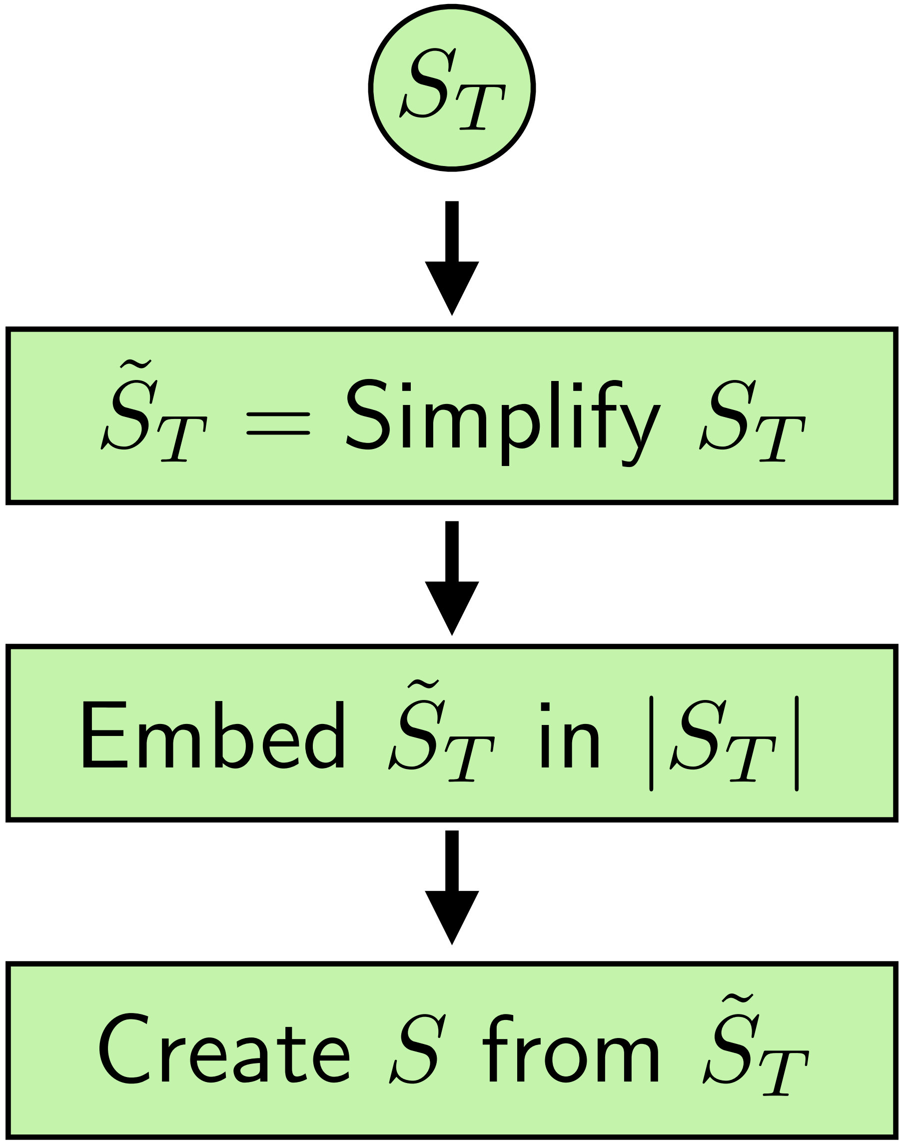
$\tilde{S}_T = \text{Simplify } S_T$

$\downarrow$

Embed $\tilde{S}_T$ in $|S_T|$

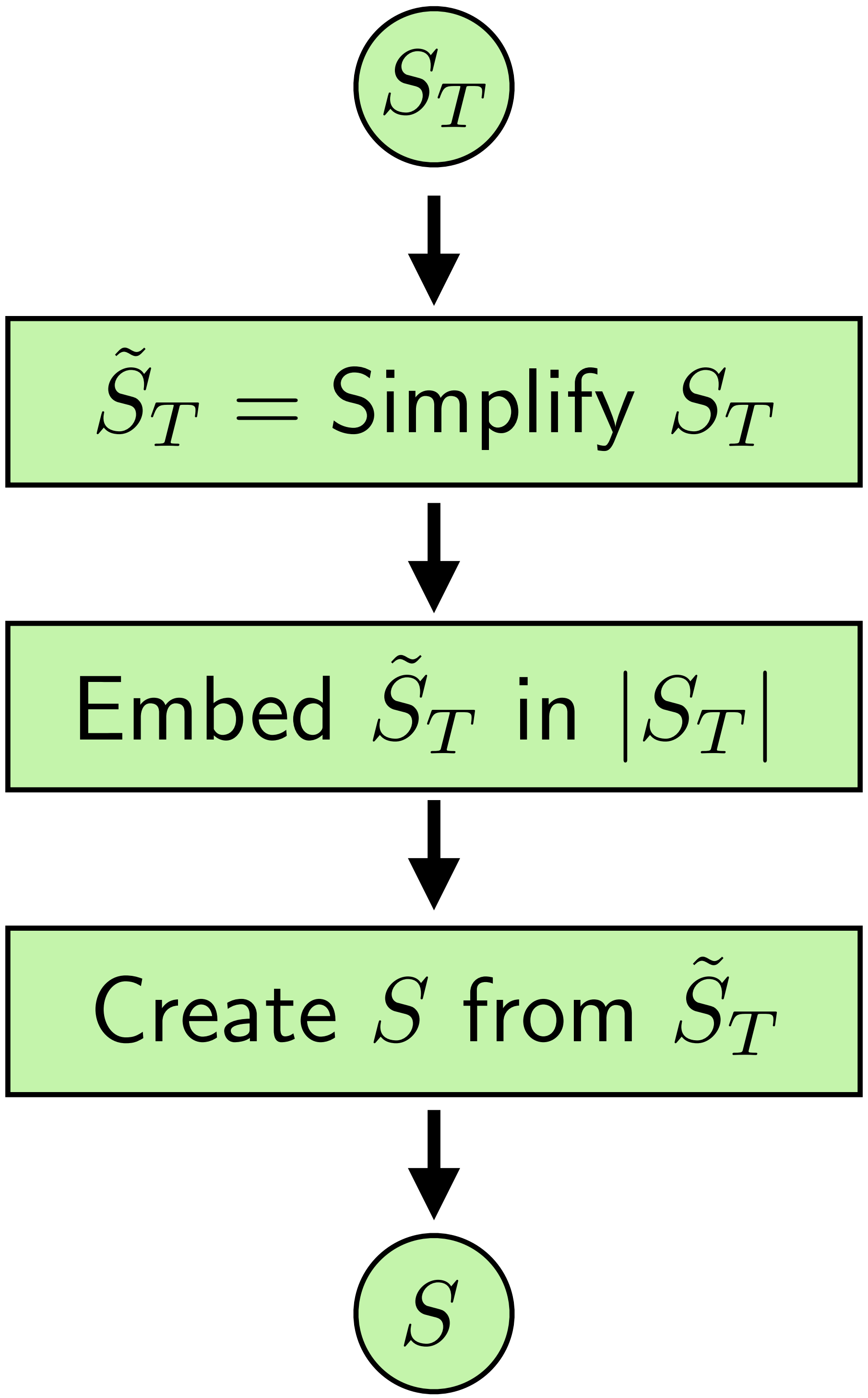
PIPELINE

Adaptive Fitting



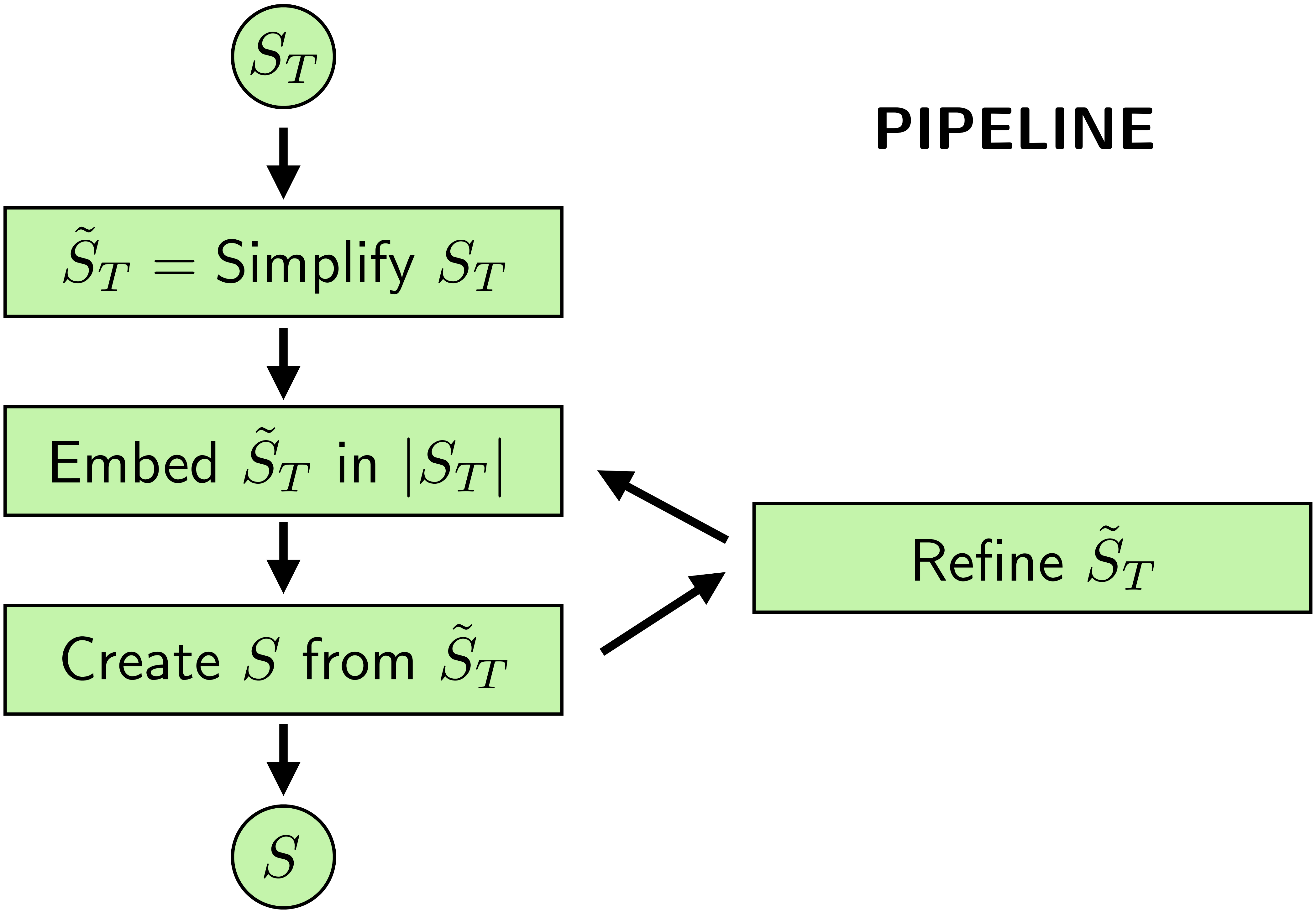
PIPELINE

Adaptive Fitting



PIPELINE

Adaptive Fitting



PIPELINE

Adaptive Fitting

$S_T \longrightarrow \boxed{\tilde{S}_T = \text{Simplify } S_T}$

Adaptive Fitting

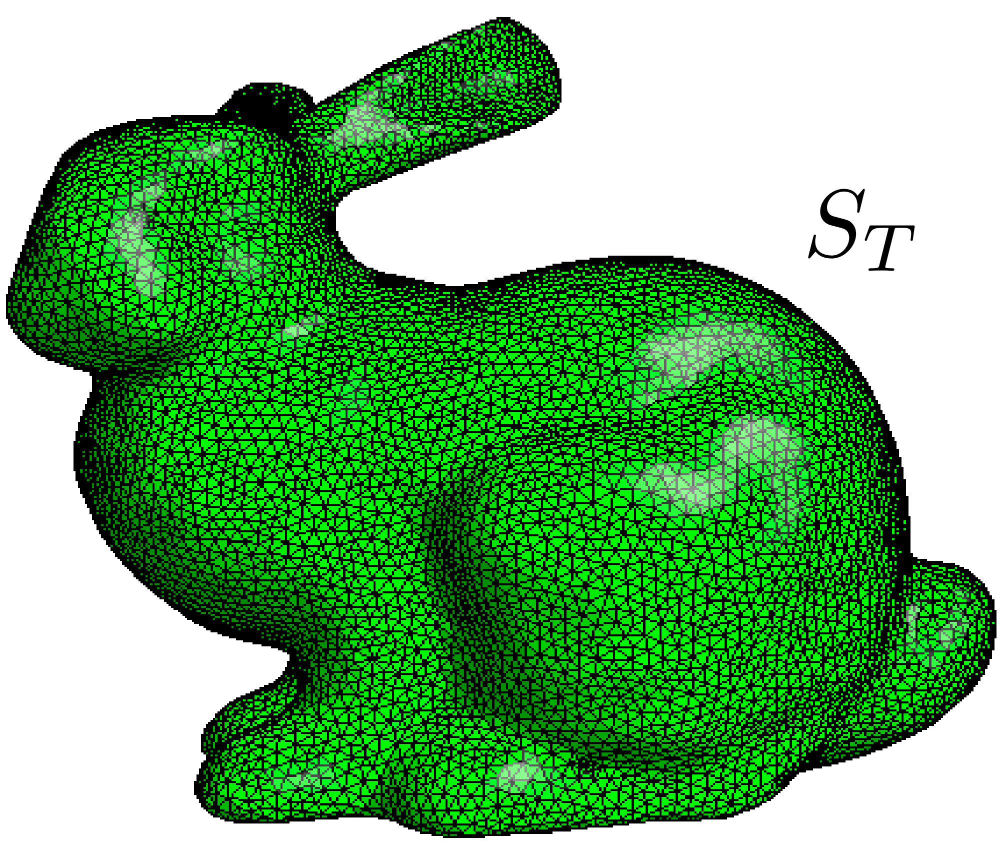
$S_T \longrightarrow \boxed{\tilde{S}_T = \text{Simplify } S_T}$

- Four-Face Clusters Algorithm

Adaptive Fitting

$S_T \longrightarrow \boxed{\tilde{S}_T = \text{Simplify } S_T}$

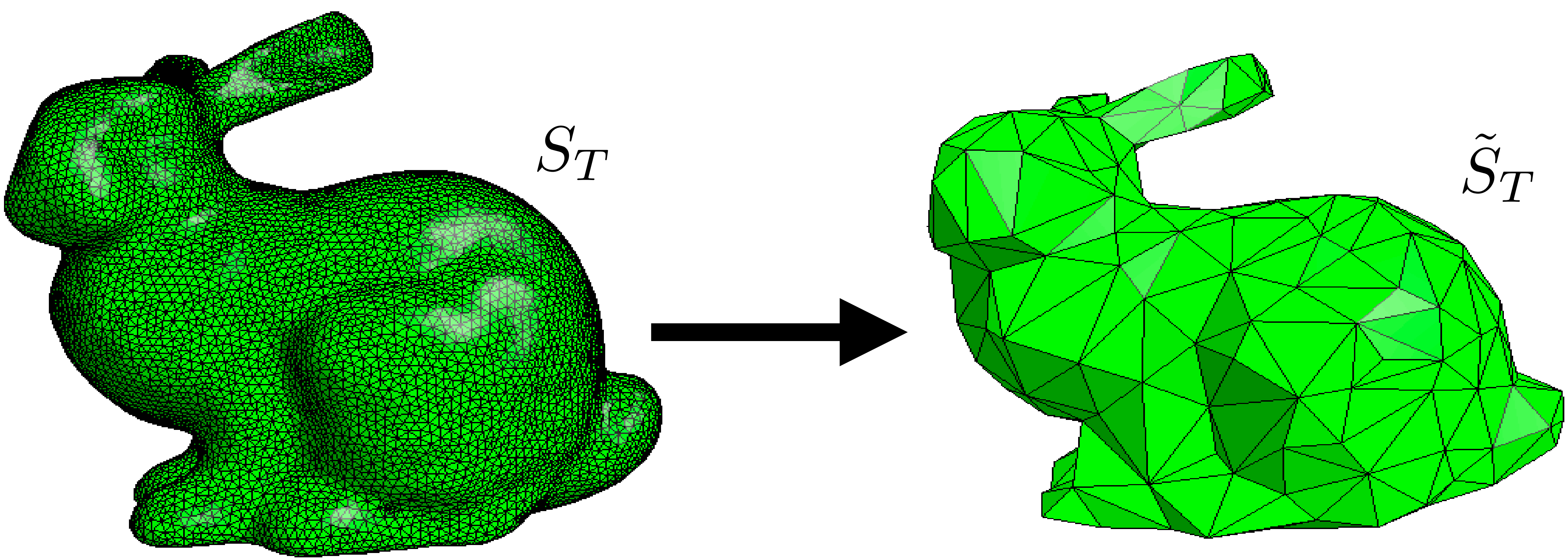
- Four-Face Clusters Algorithm



Adaptive Fitting

$$S_T \longrightarrow \boxed{\tilde{S}_T = \text{Simplify } S_T}$$

- Four-Face Clusters Algorithm



Adaptive Fitting

Adaptive Fitting

Embed $\tilde{S}_T$ in $|S_T|$

Adaptive Fitting

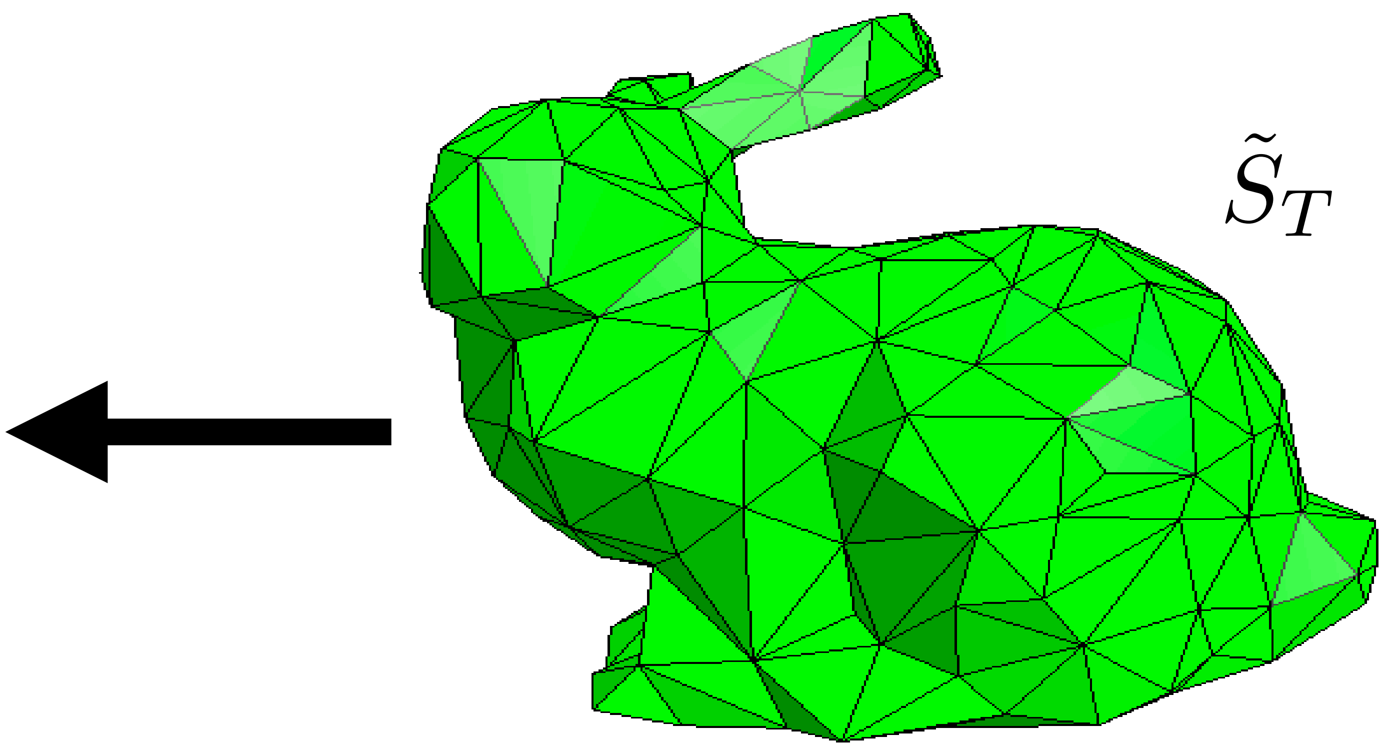
Embed $\tilde{S}_T$ in $|S_T|$

- Each edge of $\tilde{S}_T$ is embedded in $|S_T|$ as a “geodesic”.

Adaptive Fitting

Embed $\tilde{S}_T$ in $|S_T|$

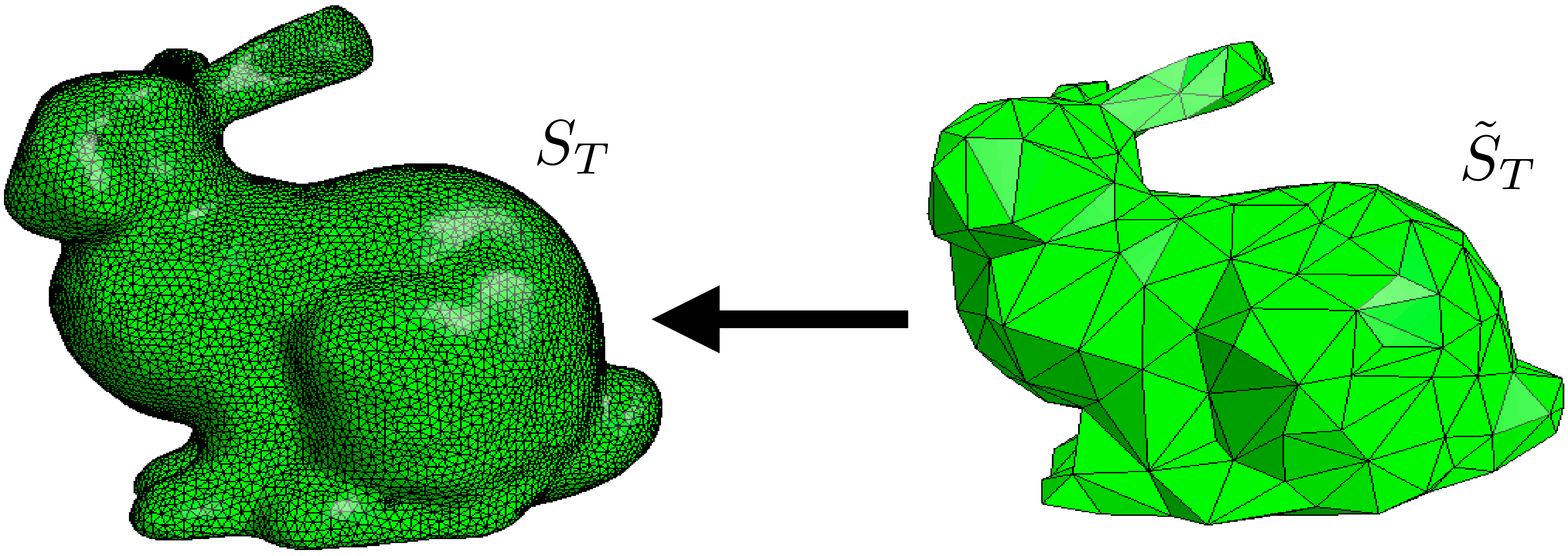
- Each edge of $\tilde{S}_T$ is embedded in $|S_T|$ as a "geodesic".



Adaptive Fitting

$\boxed{\text{Embed } \tilde{S}_T \text{ in } |S_T|}$

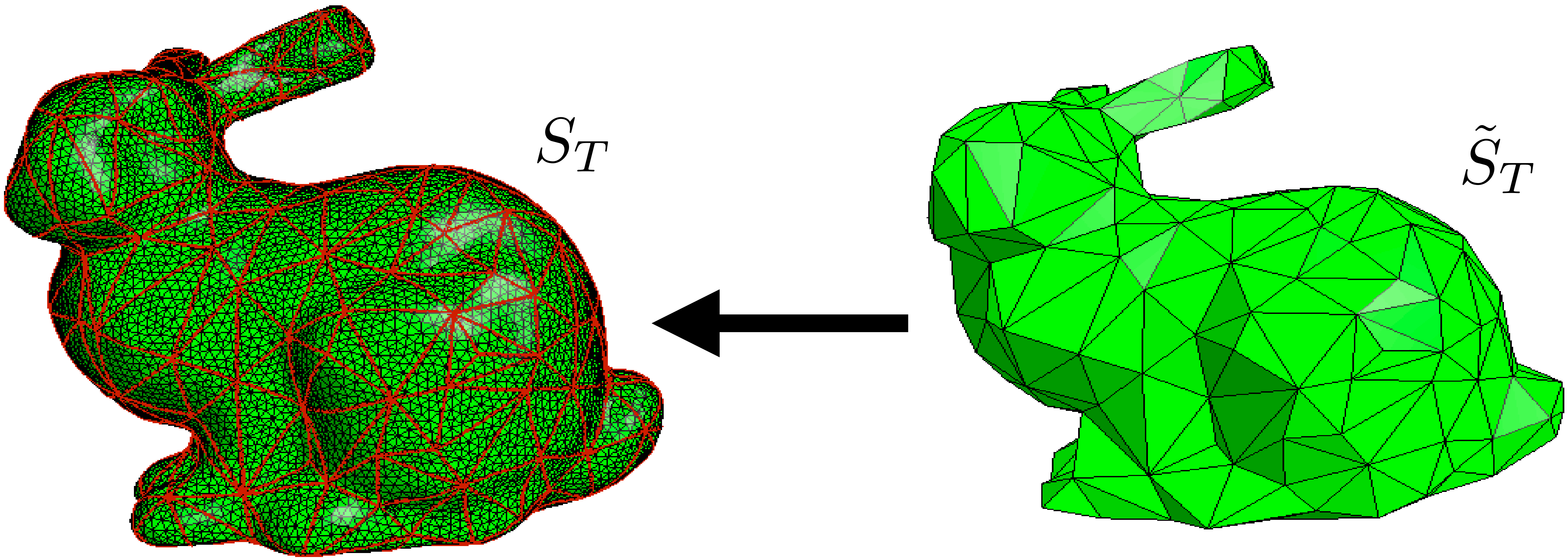
- Each edge of $\tilde{S}_T$ is embedded in $|S_T|$ as a "geodesic".



Adaptive Fitting

Embed $\tilde{S}_T$ in $|S_T|$

- Each edge of $\tilde{S}_T$ is embedded in $|S_T|$ as a “geodesic”.

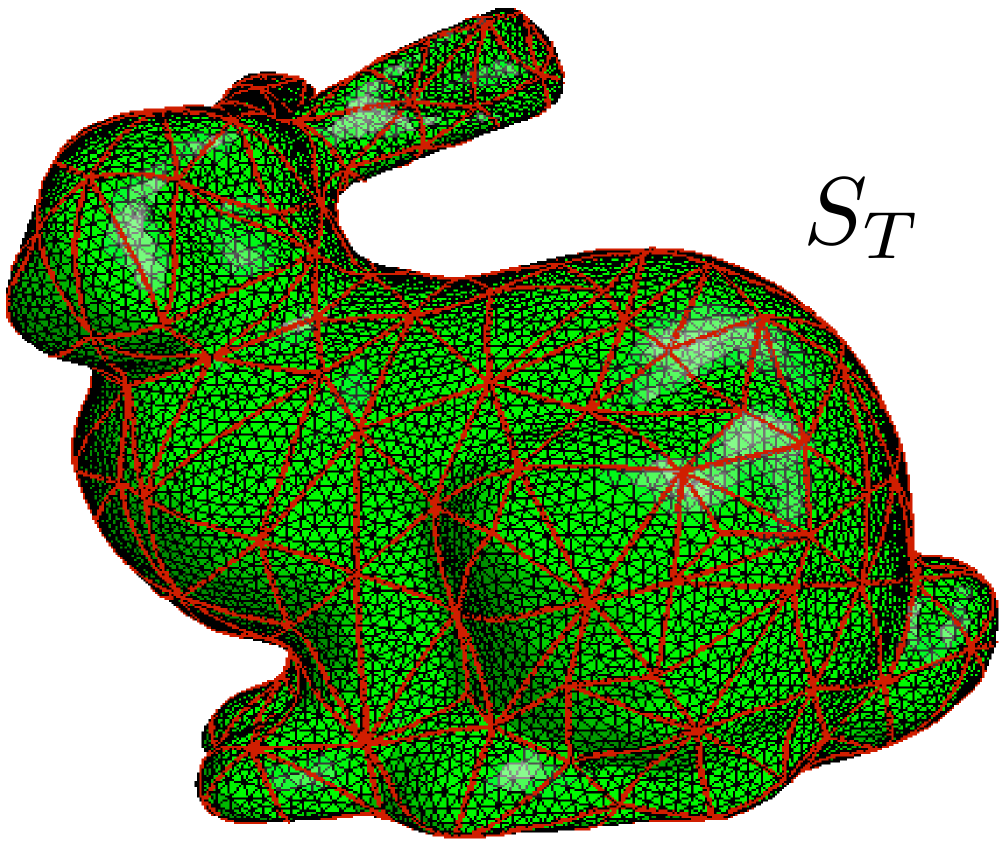


Adaptive Fitting

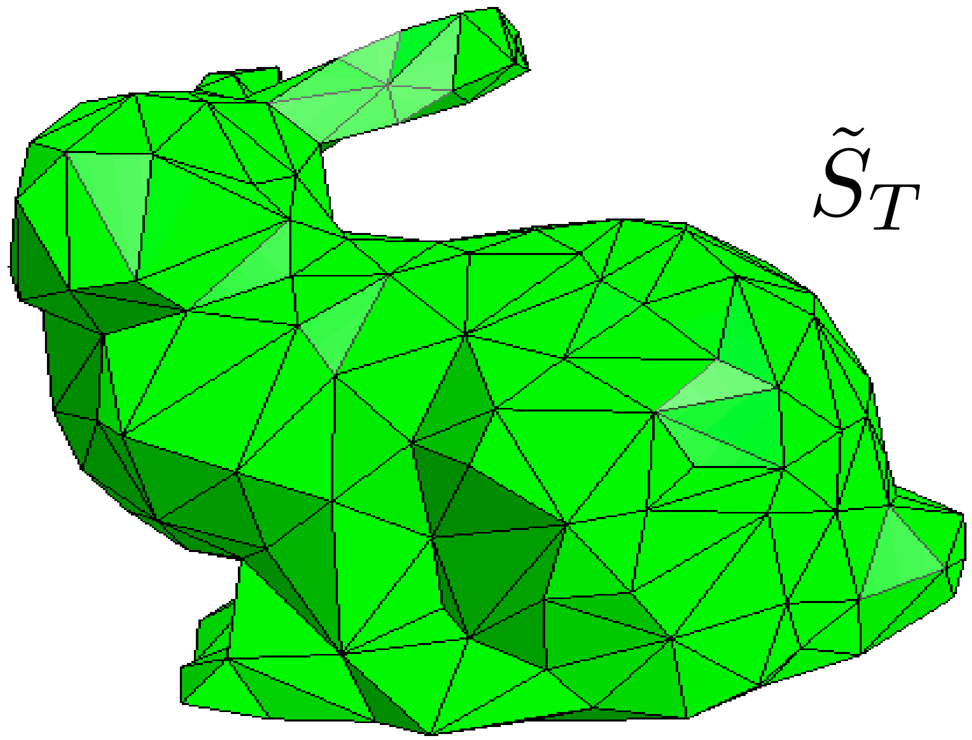
Adaptive Fitting

REMARK:

The vertices of $\tilde{S}_T$ ARE vertices of S_T.



S_T

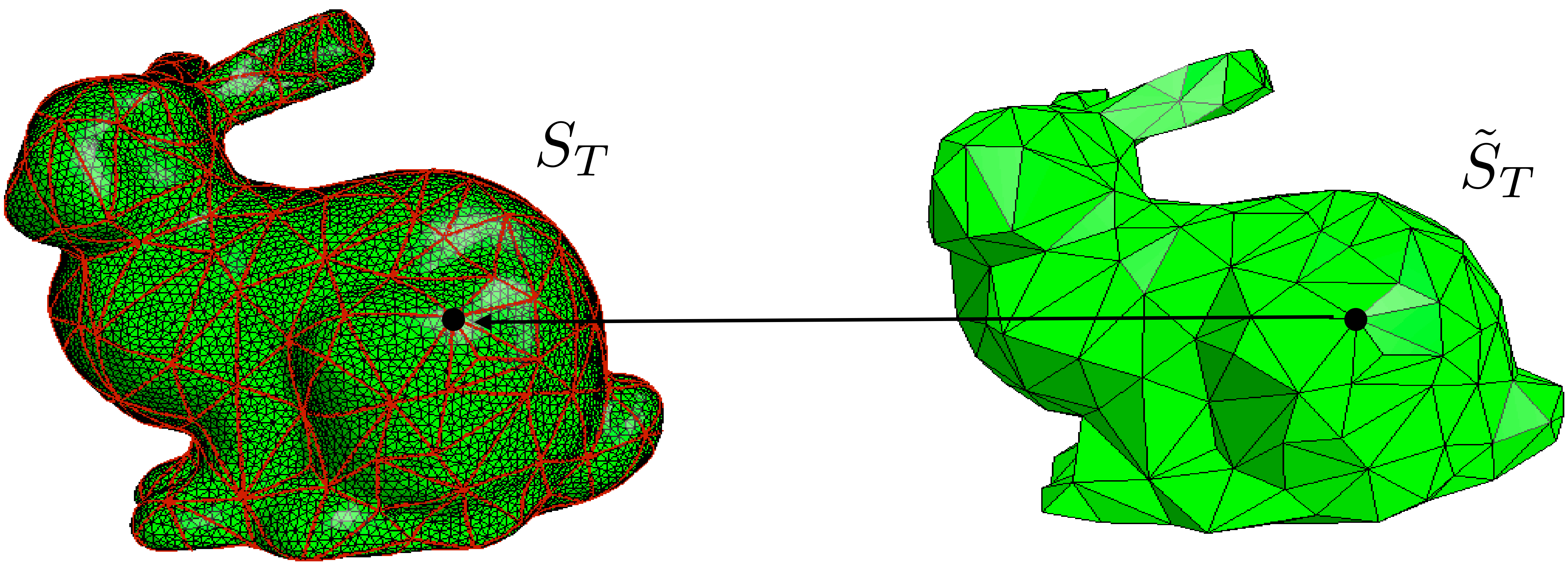


$\tilde{S}_T$

Adaptive Fitting

REMARK:

The vertices of $\tilde{S}_T$ ARE vertices of S_T.



Adaptive Fitting

Adaptive Fitting

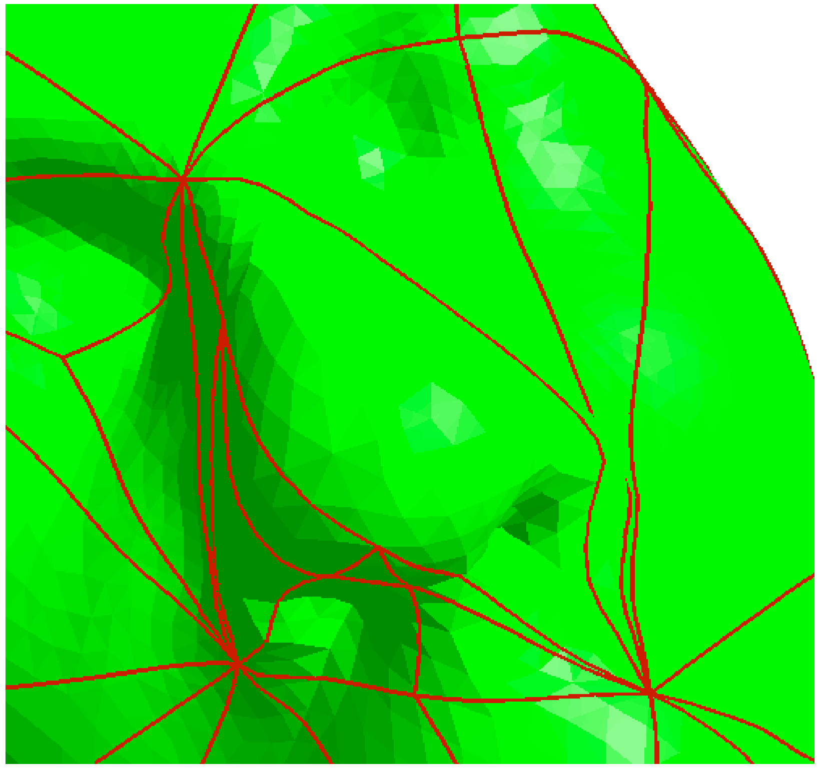
PROBLEM:

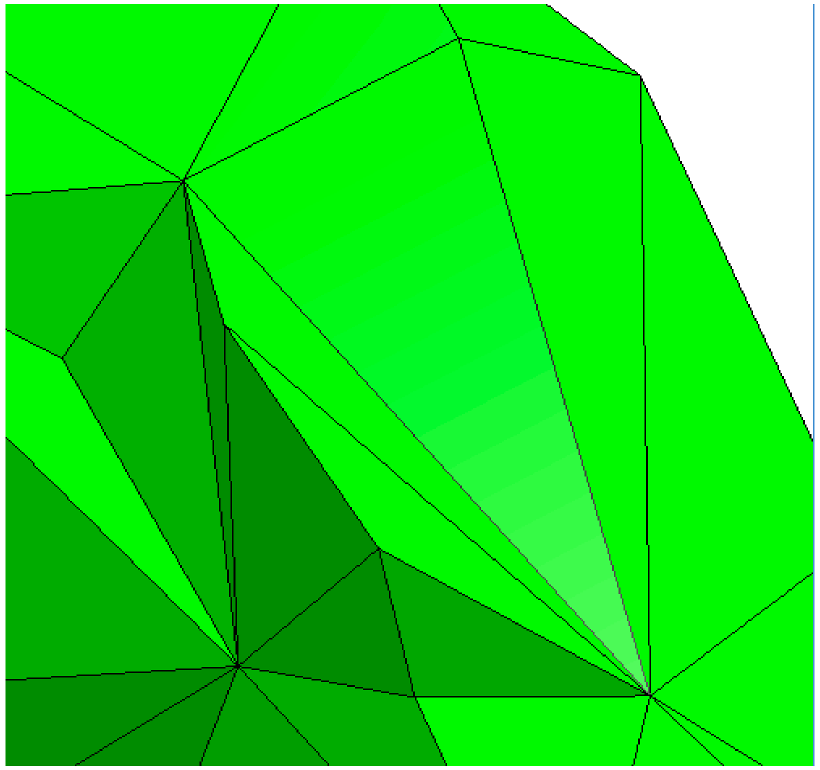
When defining geodesic triangles, we can violate the manifold property of the geodesic mesh, as illustrated by the figure below:

Adaptive Fitting

PROBLEM:

When defining geodesic triangles, we can violate the manifold property of the geodesic mesh, as illustrated by the figure below:





Adaptive Fitting

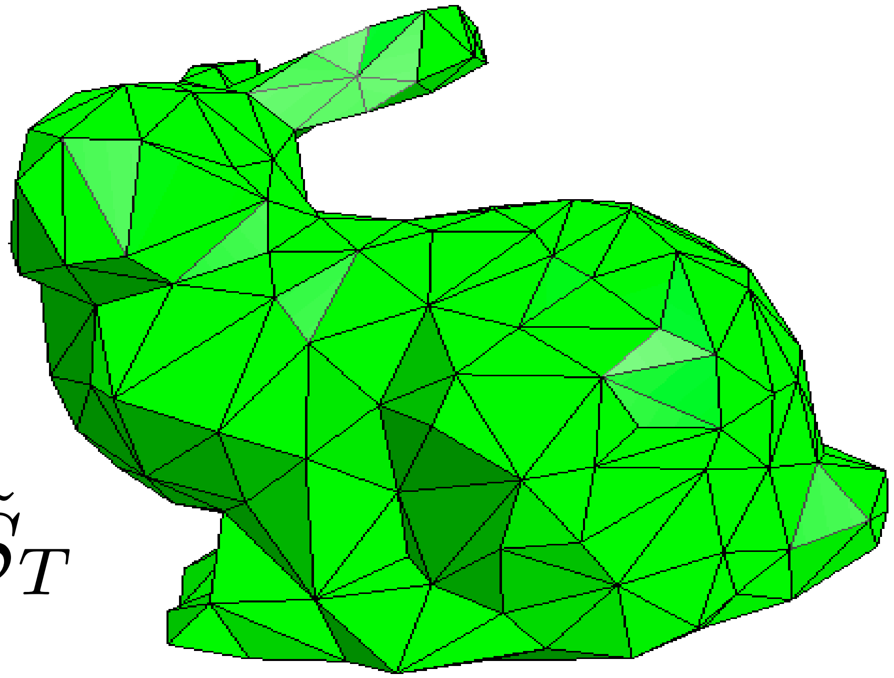
Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

- For each vertex v of $\tilde{S}_T$, we consider the P-polygon, P_v, of v in $\mathbb{R}^2$, and the standard triangulation, T_v, of the P-polygon P_v.

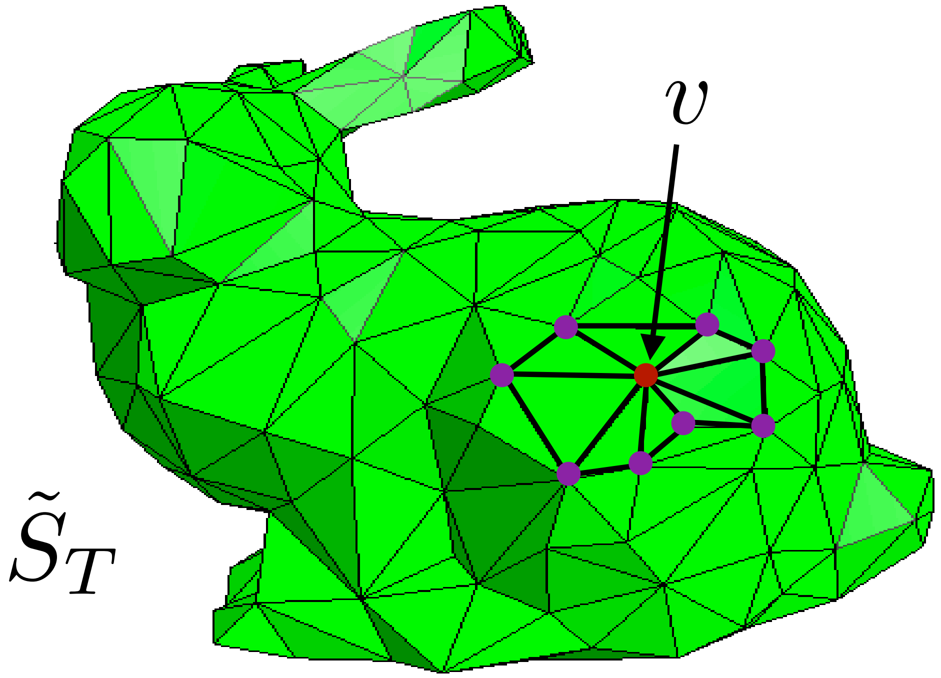


$\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

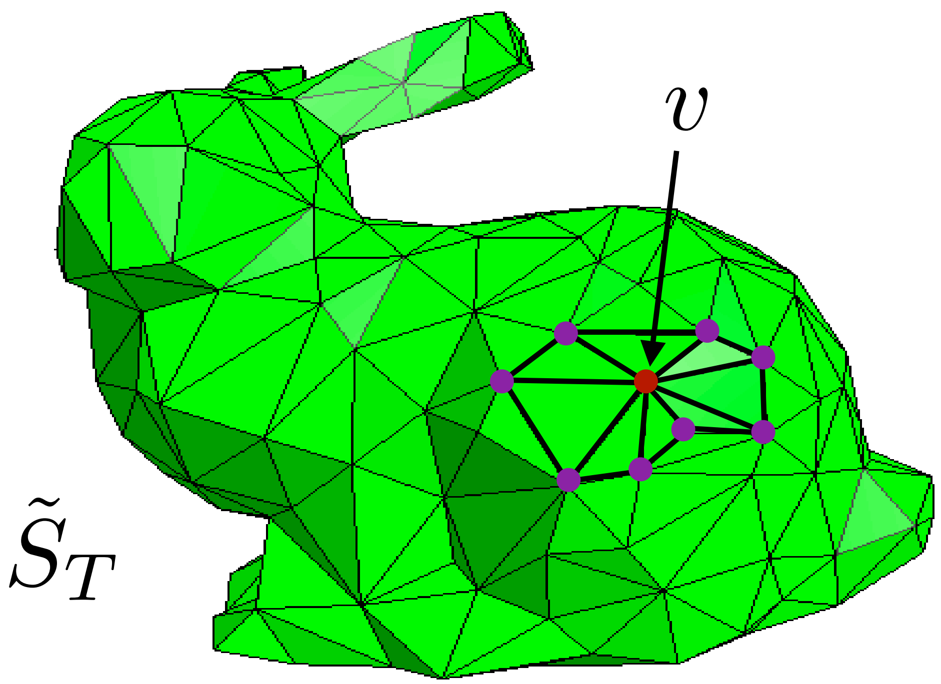
- For each vertex v of $\tilde{S}_T$, we consider the P-polygon, P_v, of v in $\mathbb{R}^2$, and the standard triangulation, T_v, of the P-polygon P_v.

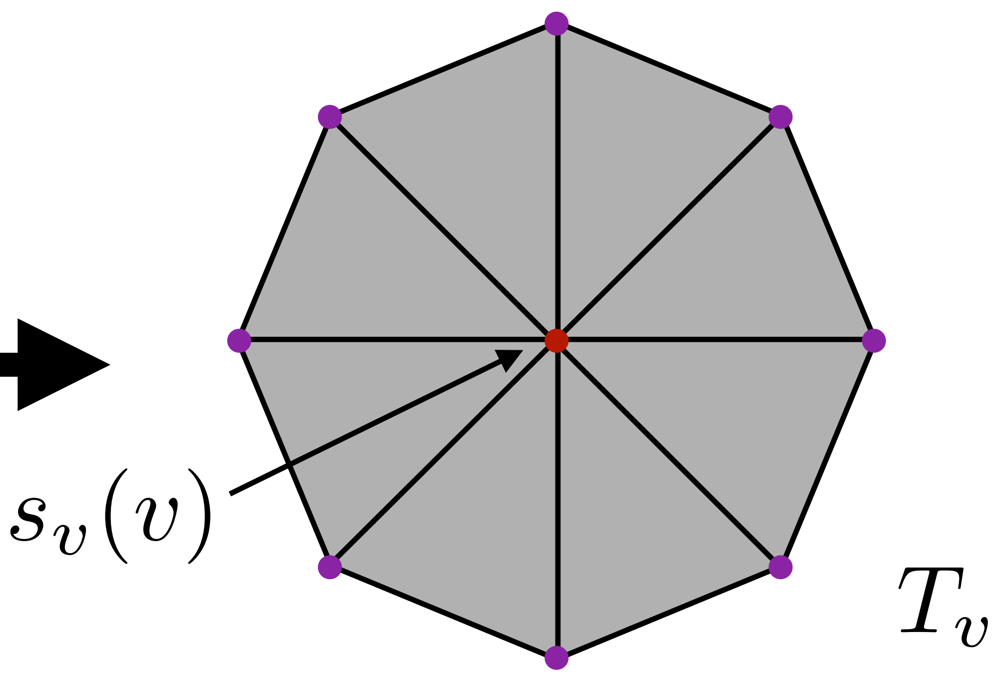


Adaptive Fitting

Create S from $\tilde{S}_T$

- For each vertex v of $\tilde{S}_T$, we consider the P-polygon, P_v, of v in $\mathbb{R}^2$, and the standard triangulation, T_v, of the P-polygon P_v.





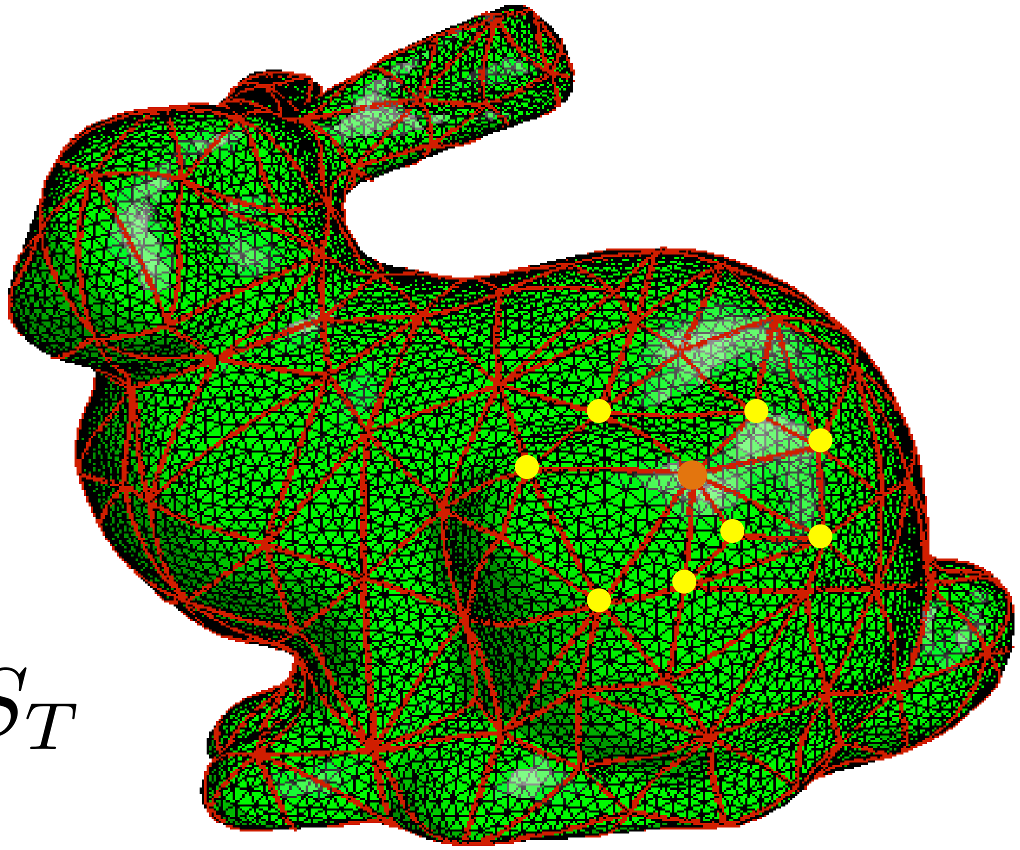
Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

- Consider the embedding of the star, $st(v, \tilde{S}_T)$, of v in S_T.



S_T

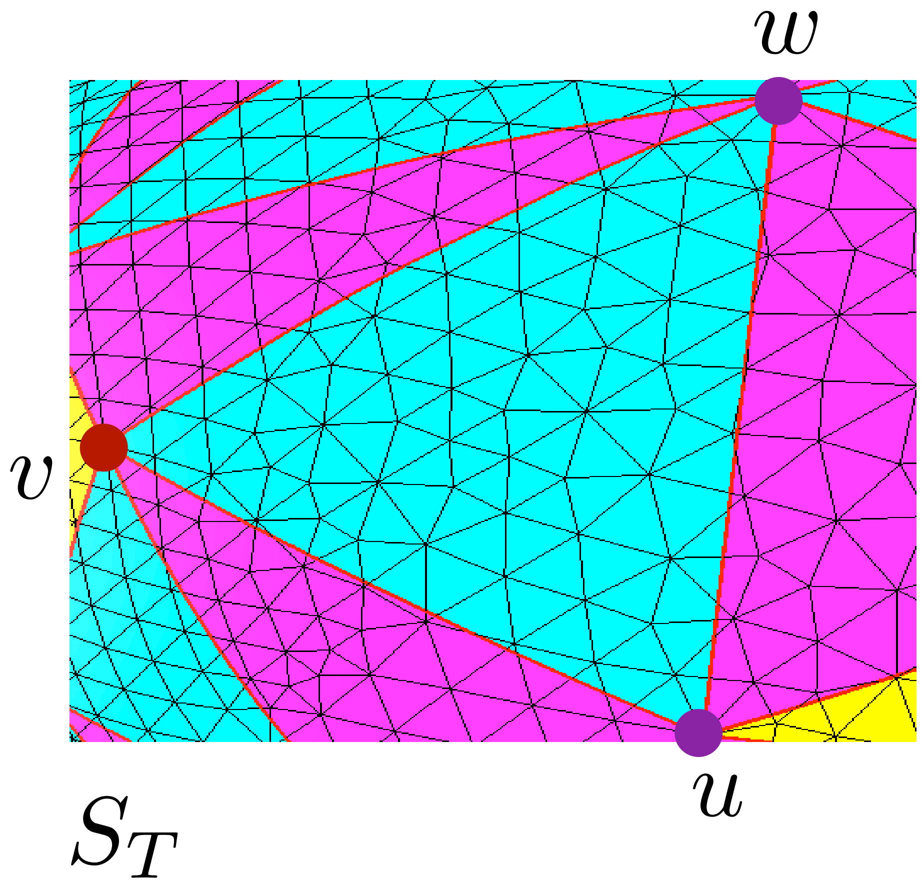
Adaptive Fitting

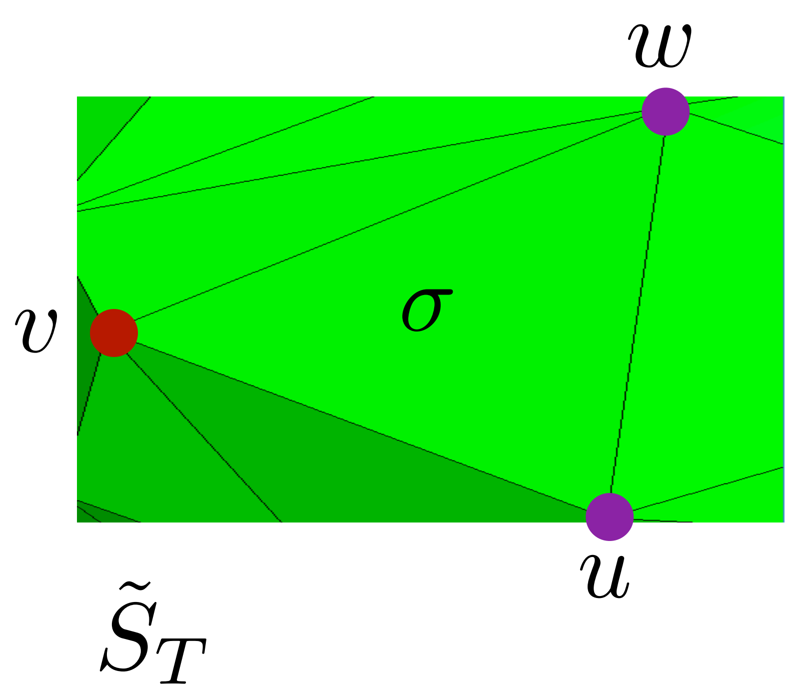
Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

- Map the vertices of S_T bounded by the embedding of $st(v, \tilde{S}_T)$ to T_v.

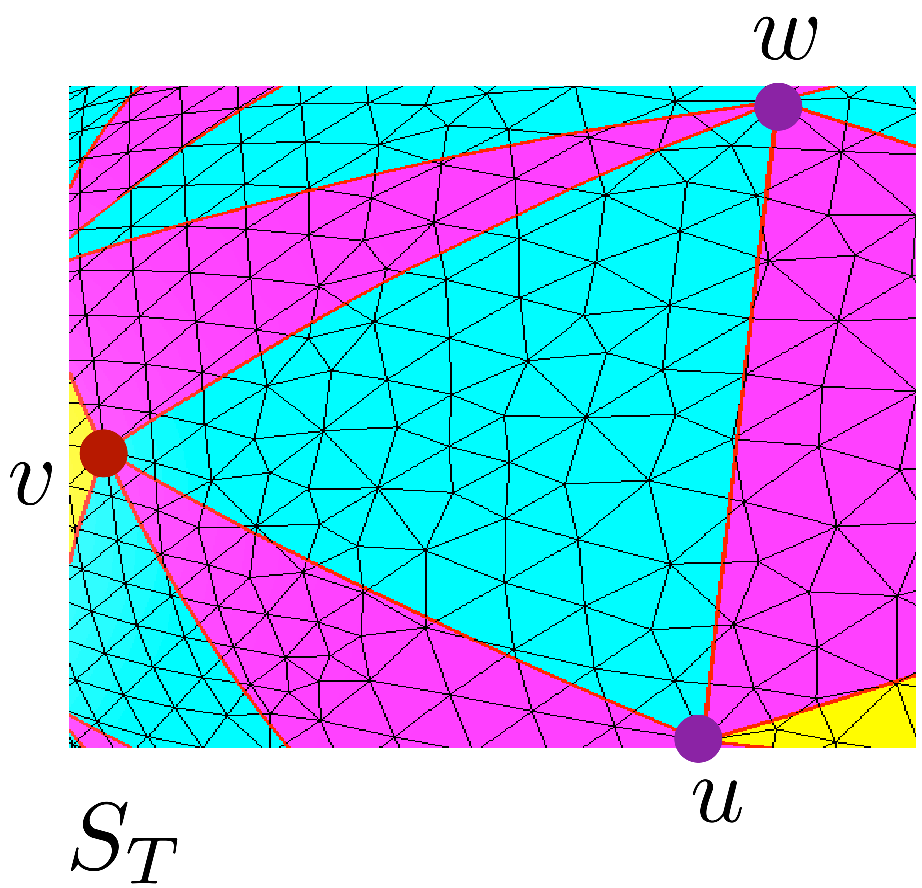




Adaptive Fitting

Create S from $\tilde{S}_T$

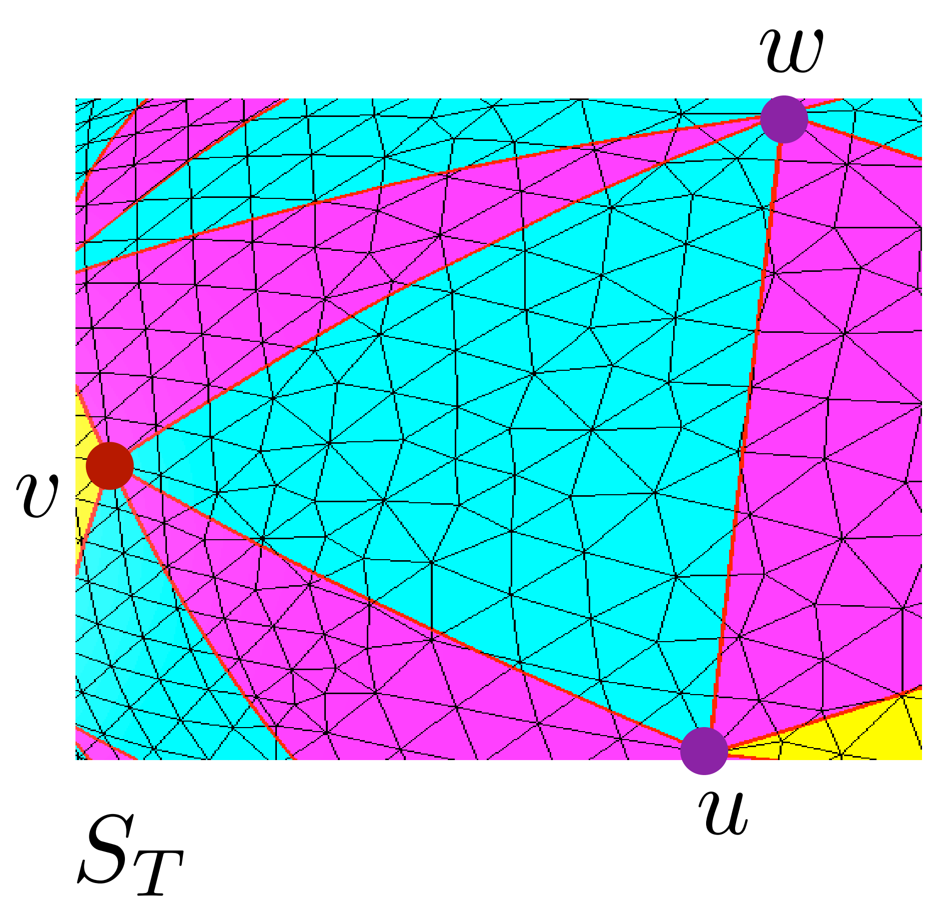
- Map the vertices of S_T bounded by the embedding of $st(v, \tilde{S}_T)$ to T_v.

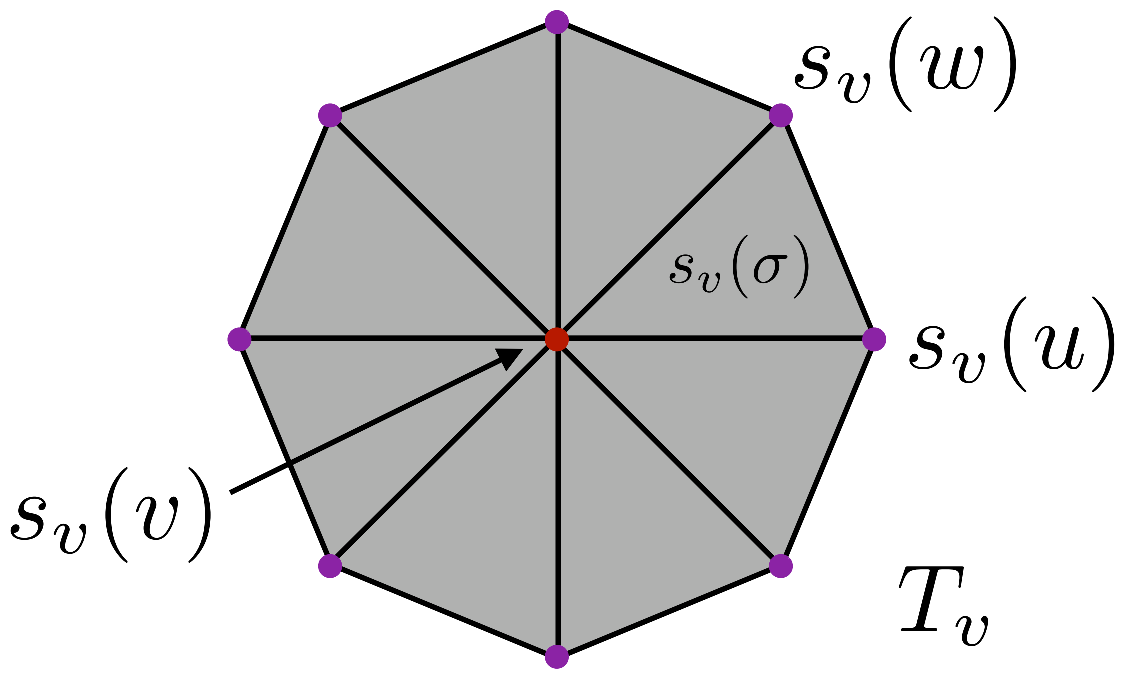


Adaptive Fitting

Create S from $\tilde{S}_T$

- Map the vertices of S_T bounded by the embedding of $st(v, \tilde{S}_T)$ to T_v.

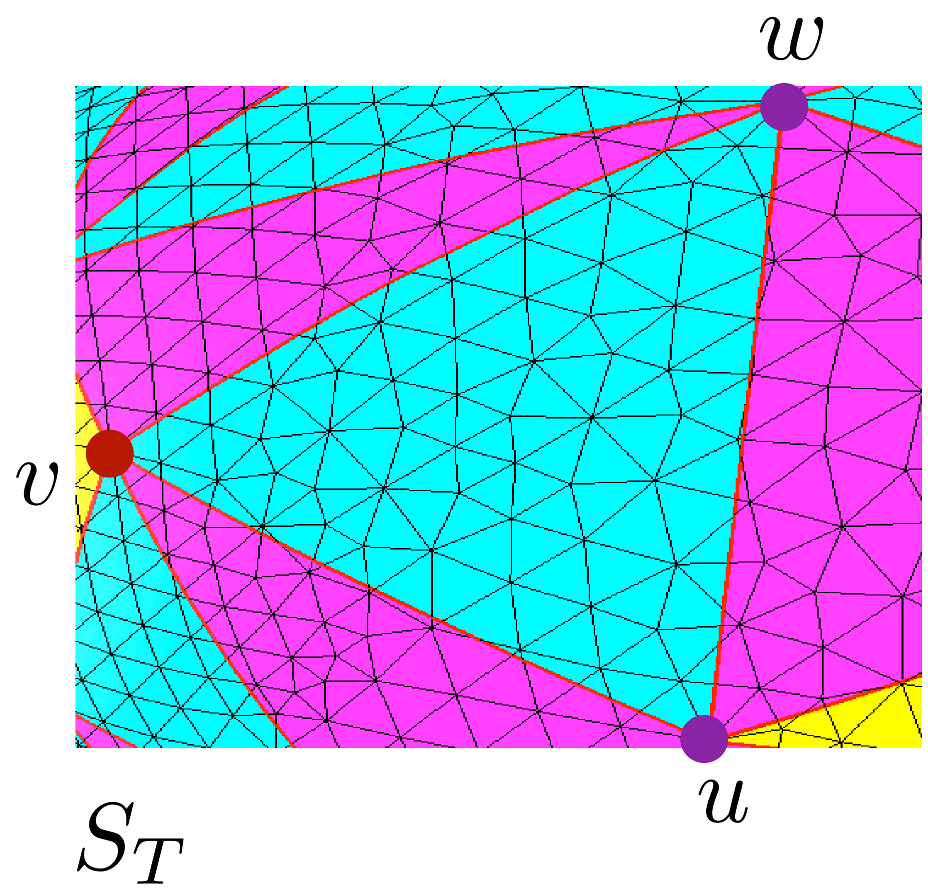


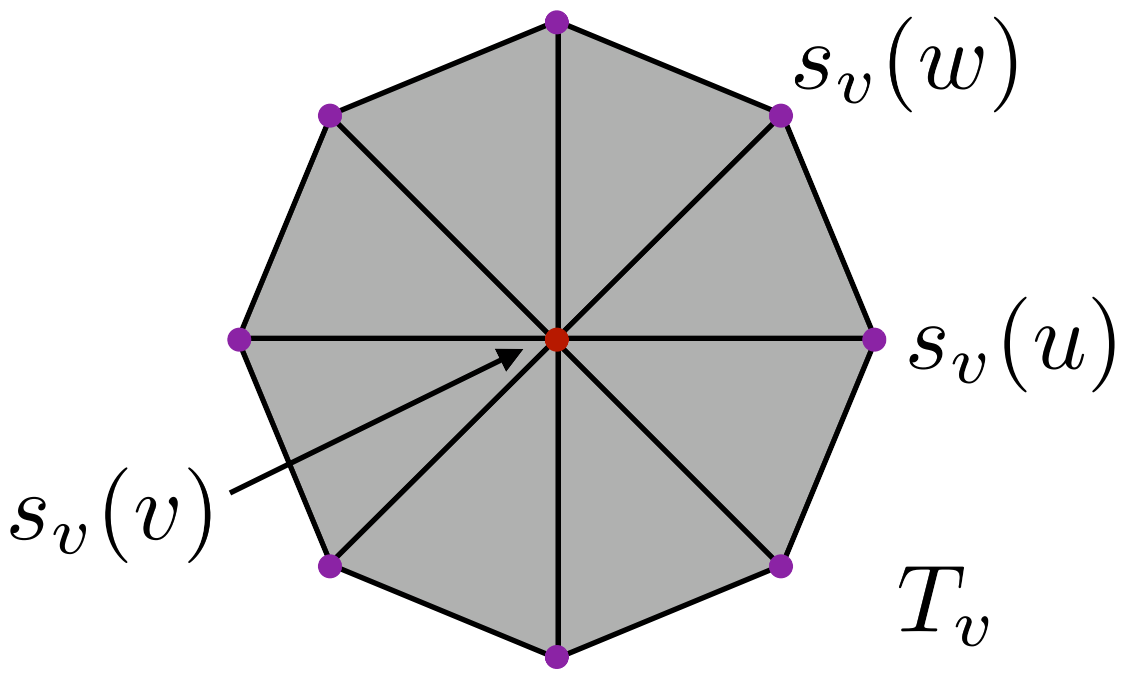


Adaptive Fitting

Create S from $\tilde{S}_T$

- Map the vertices of S_T bounded by the embedding of $st(v, \tilde{S}_T)$ to T_v.

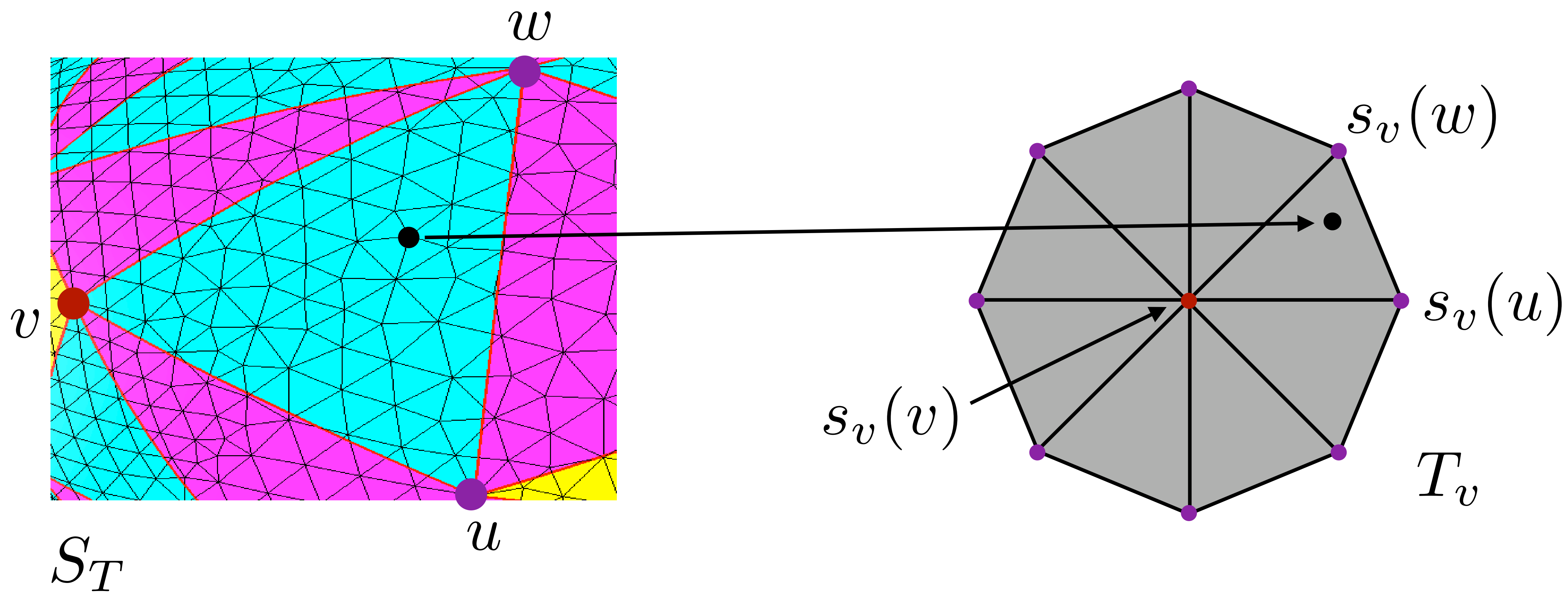




Adaptive Fitting

Create S from $\tilde{S}_T$

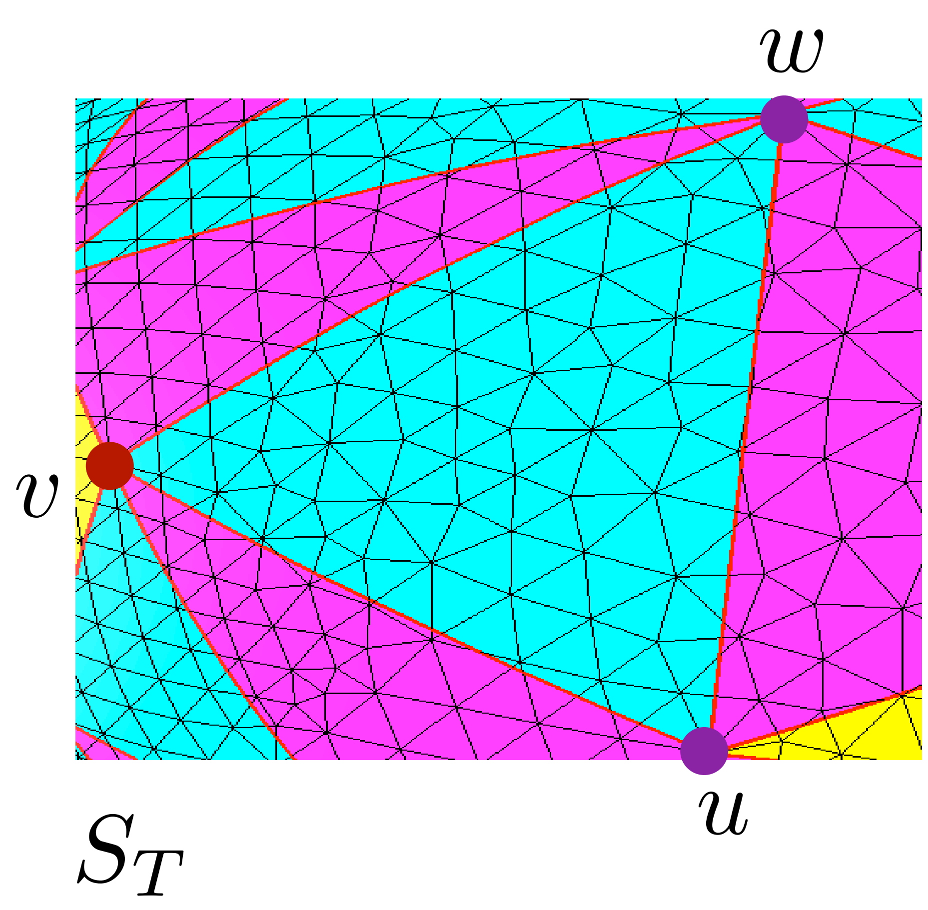
- Map the vertices of S_T bounded by the embedding of $st(v, \tilde{S}_T)$ to T_v.

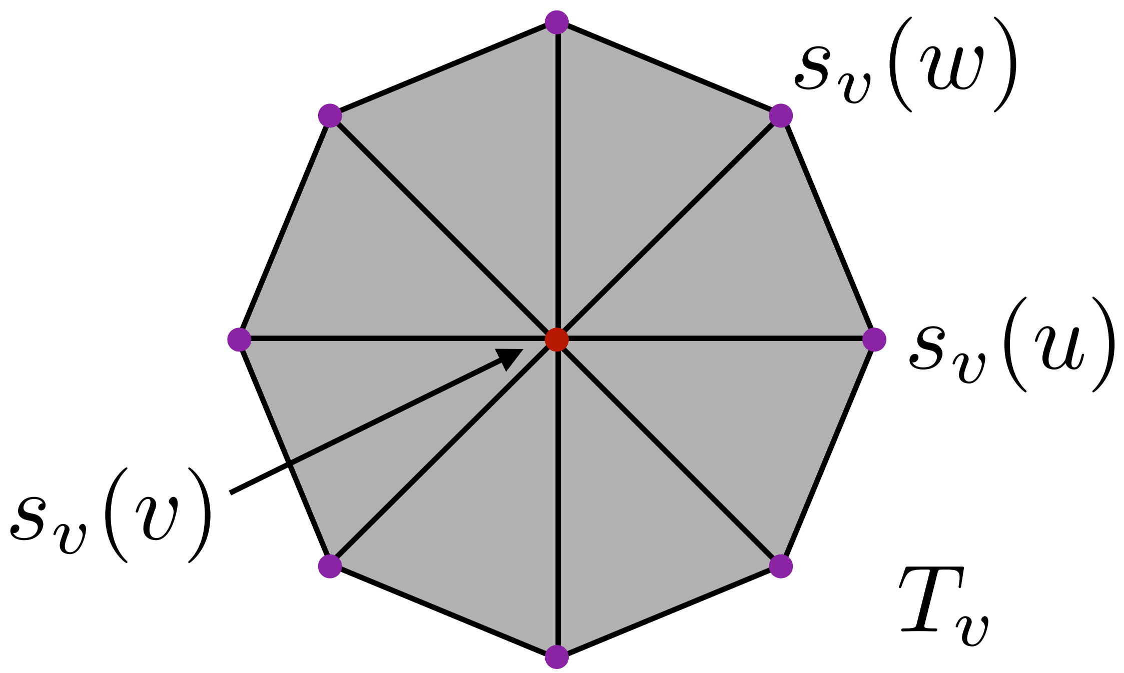


Adaptive Fitting

Create S from $\tilde{S}_T$

- Points where geodesics intersect edges of S_T are also mapped to T_v.

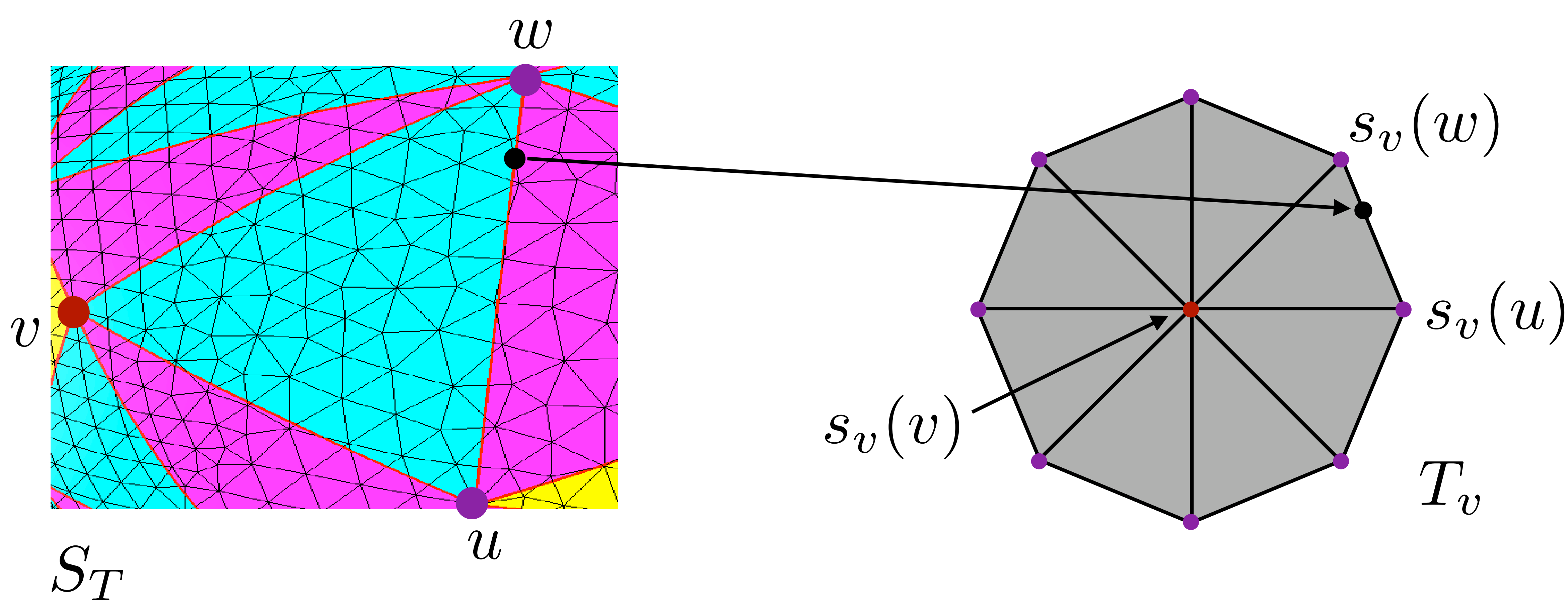




Adaptive Fitting

Create S from $\tilde{S}_T$

- Points where geodesics intersect edges of S_T are also mapped to T_v.



Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

- How is this mapping done?

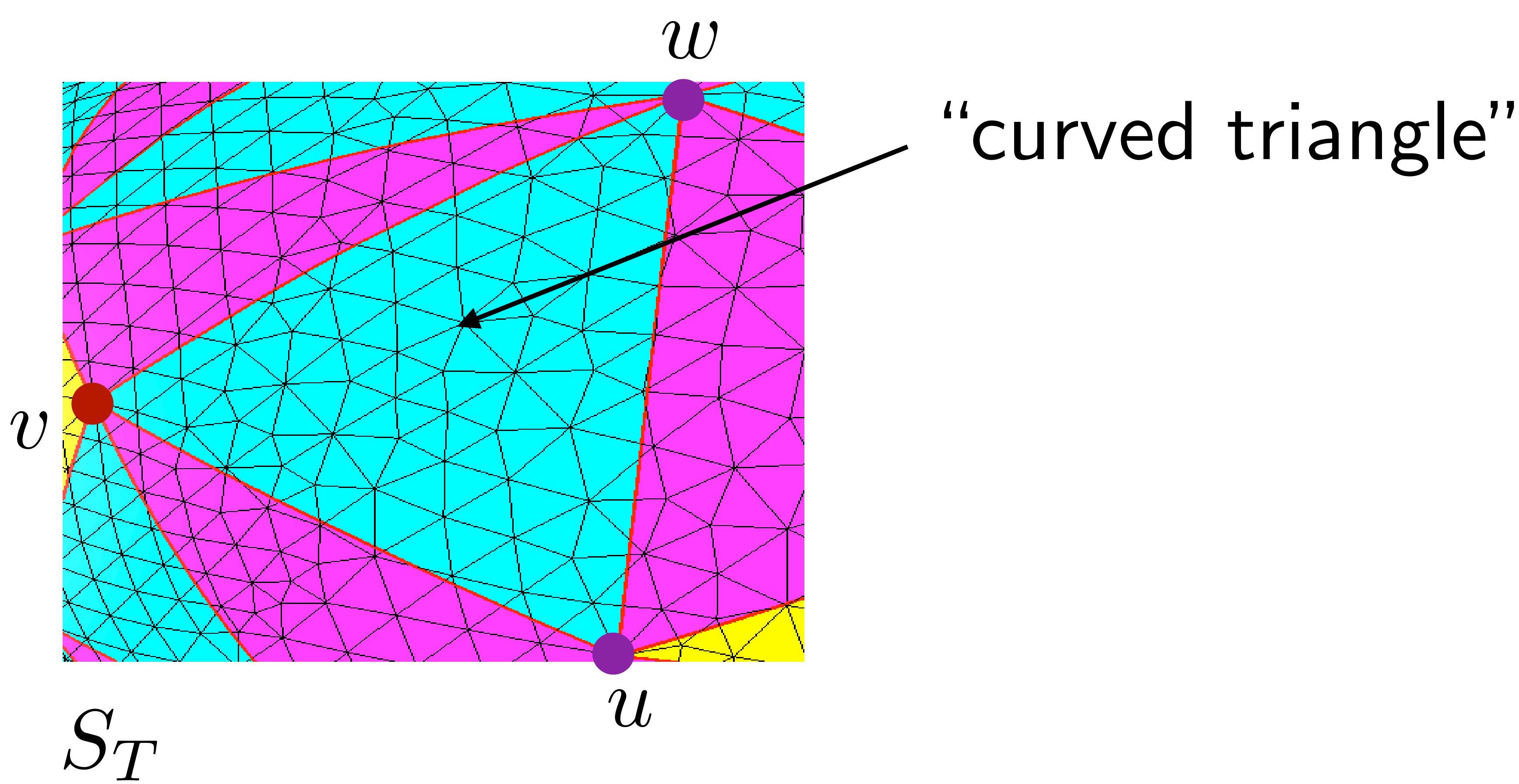
Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

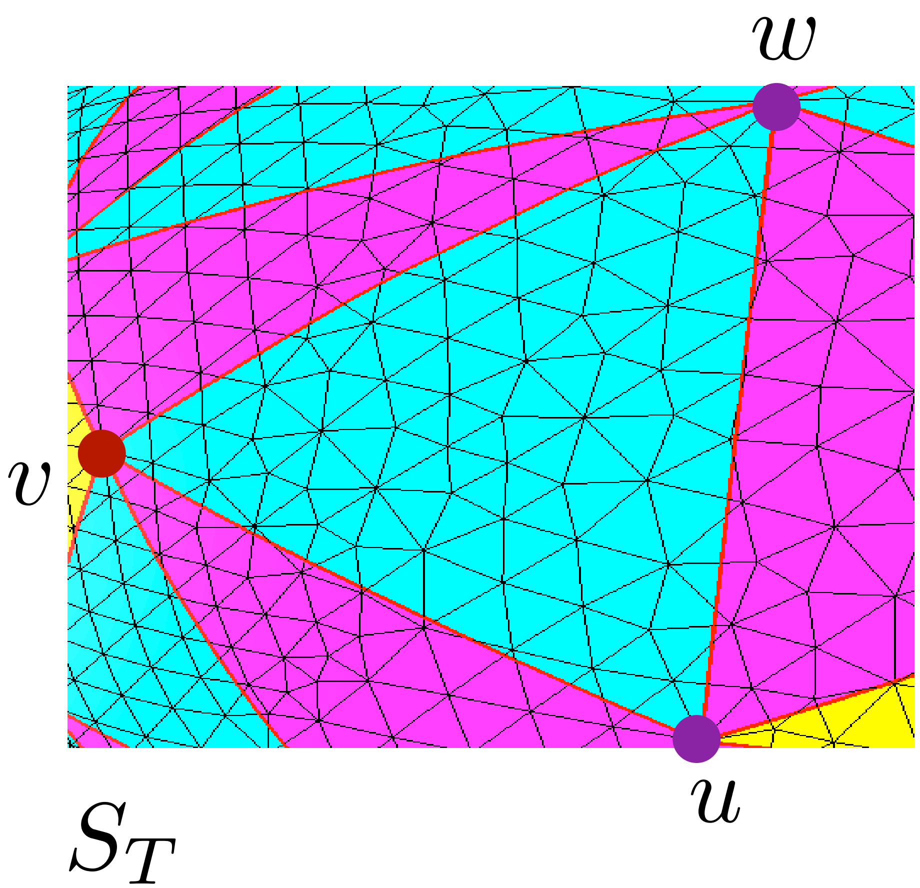
- We map the vertices in each "curved" triangle separately.



Adaptive Fitting

Create S from $\tilde{S}_T$

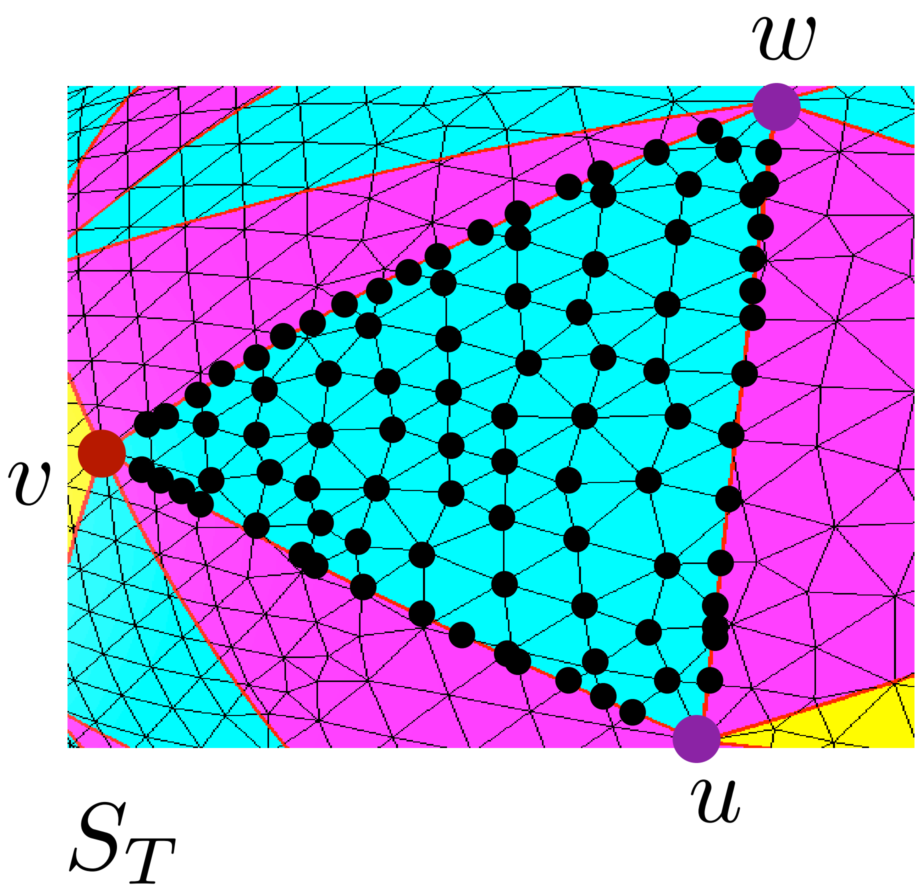
- We use Floater's parametrization to build the map for each "curved" triangle.



Adaptive Fitting

Create S from $\tilde{S}_T$

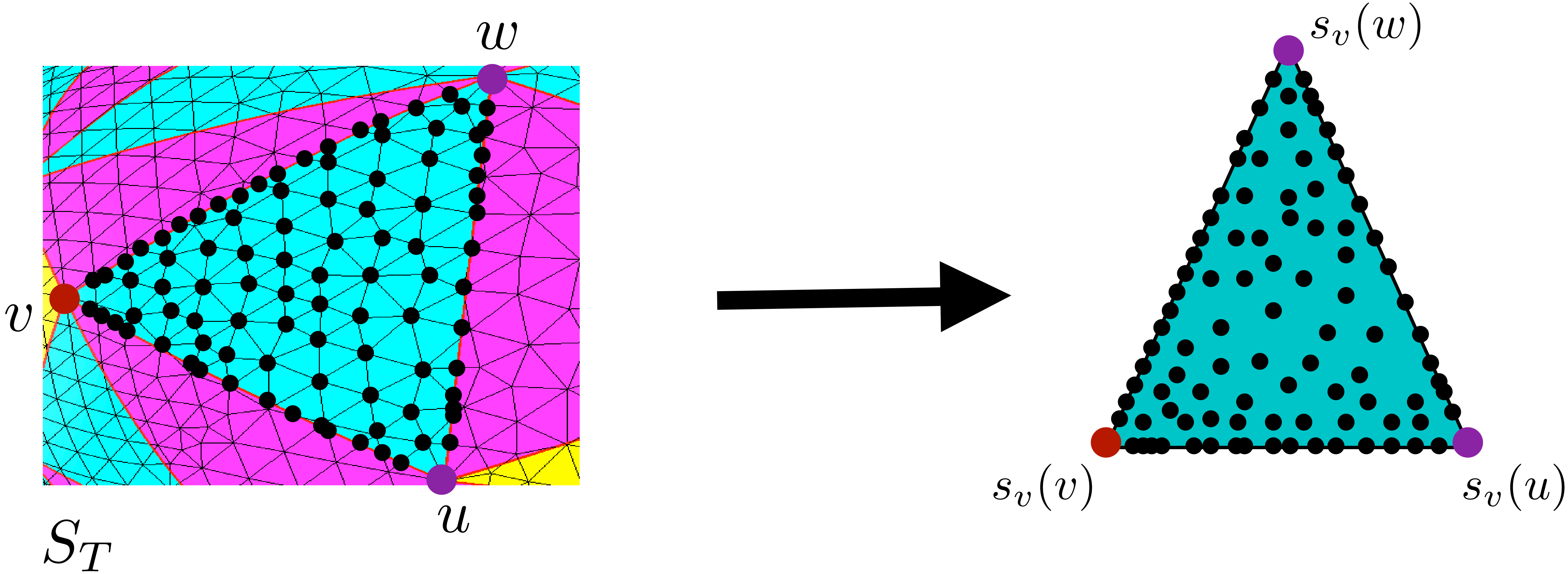
- We use Floater's parametrization to build the map for each "curved" triangle.



Adaptive Fitting

Create S from $\tilde{S}_T$

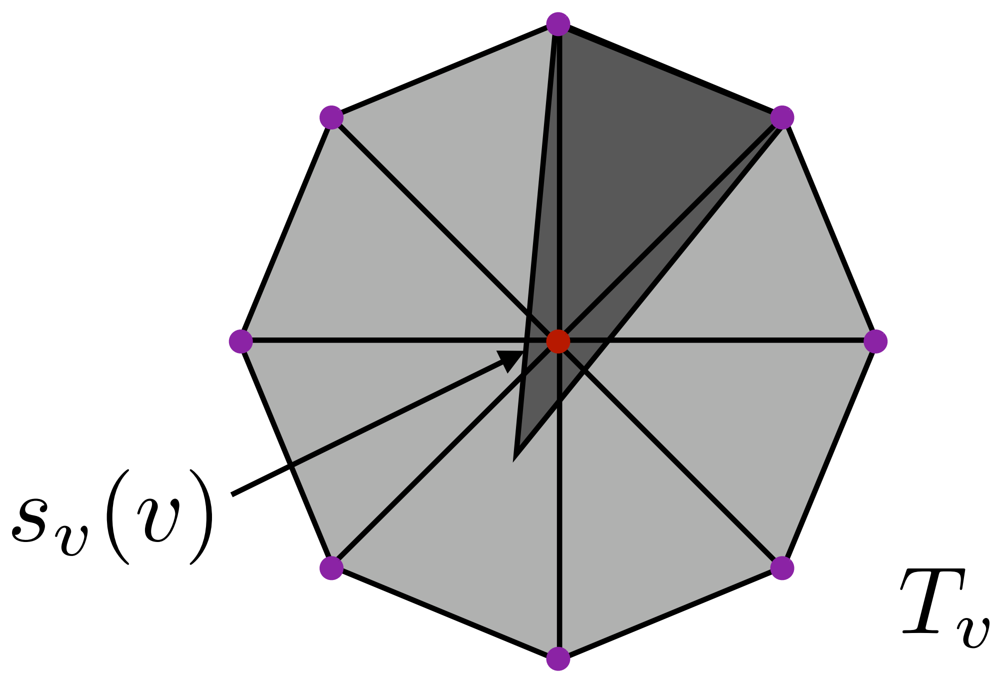
- We use Floater's parametrization to build the map for each "curved" triangle.



Adaptive Fitting

Create S from $\tilde{S}_T$

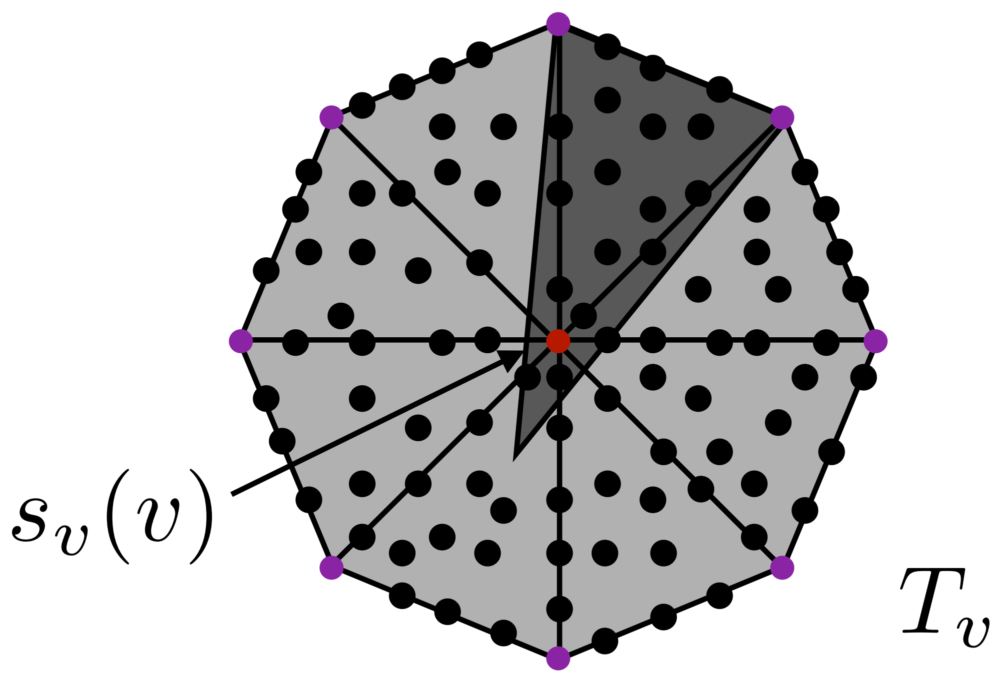
- For each triangle in $st(v, \tilde{S}_T)$, compute the shape function $\psi_{(\sigma,v)}$.



Adaptive Fitting

Create S from $\tilde{S}_T$

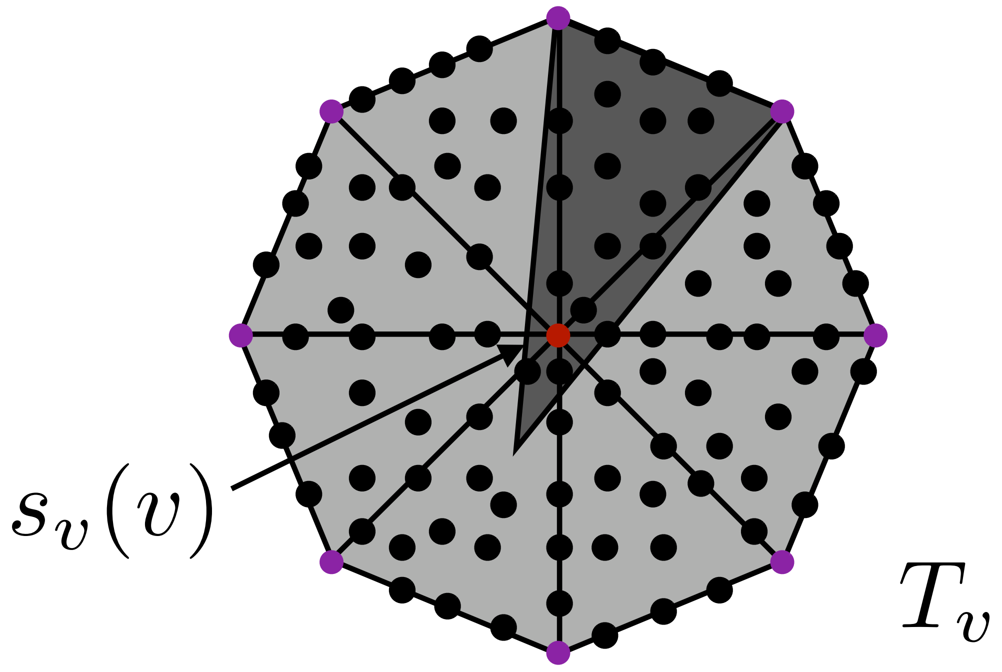
- For each triangle in $st(v, \tilde{S}_T)$, compute the shape function $\psi_{(\sigma,v)}$.

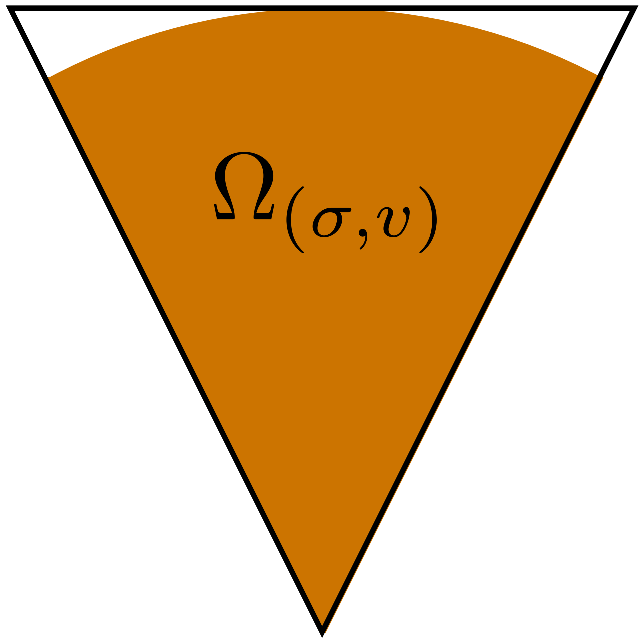


Adaptive Fitting

Create S from $\tilde{S}_T$

- For each triangle in $st(v, \tilde{S}_T)$, compute the shape function $\psi_{(\sigma,v)}$.

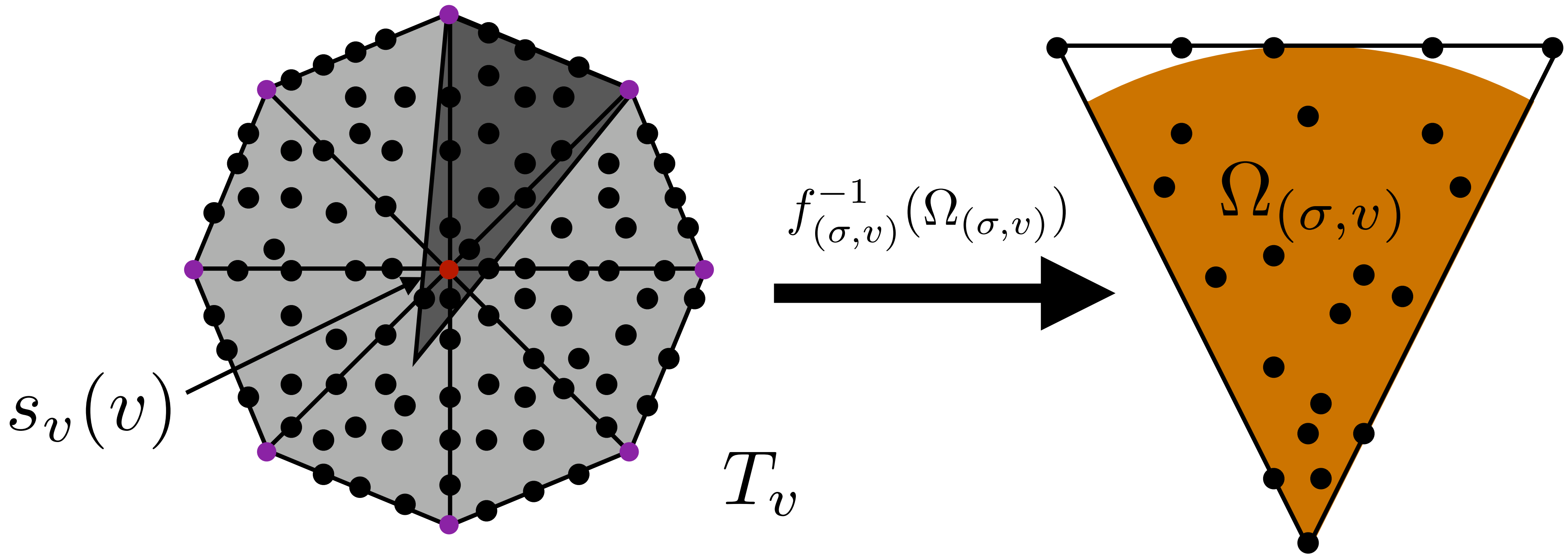




Adaptive Fitting

$\boxed{\text{Create } S \text{ from } \tilde{S}_T}$

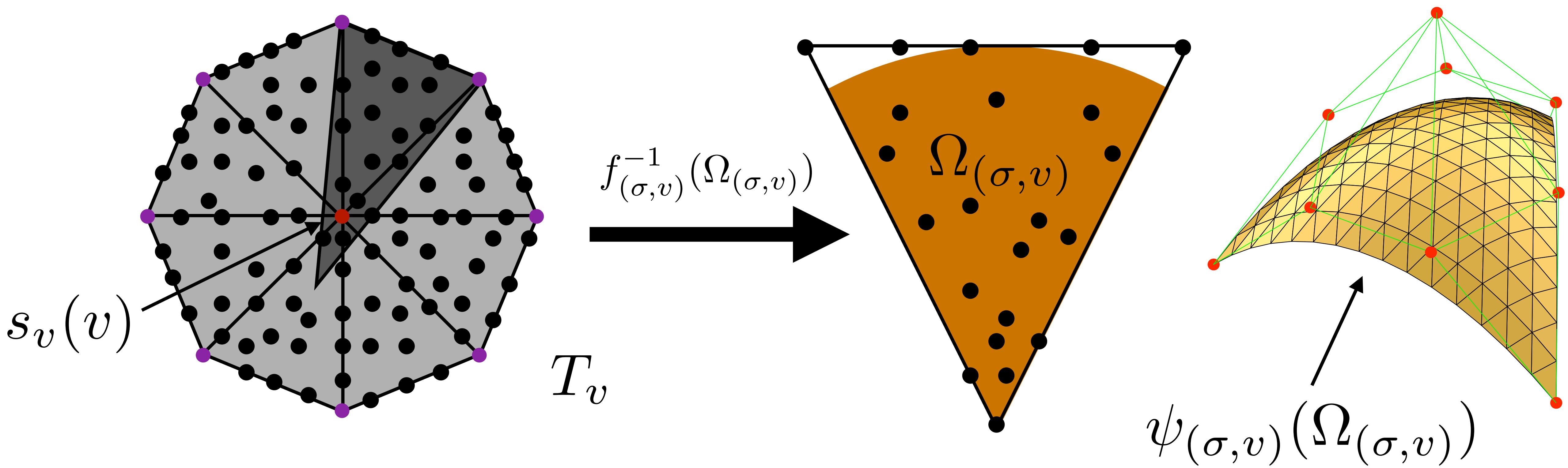
- For each triangle in $st(v, \tilde{S}_T)$, compute the shape function $\psi_{(\sigma,v)}$.



Adaptive Fitting

Create S from $\tilde{S}_T$

- For each triangle in $st(v, \tilde{S}_T)$, compute the shape function $\psi_{(\sigma,v)}$.



Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

- Control points of $\psi_{(\sigma,v)}$ are computed by a least squares procedure.

Adaptive Fitting

Create S from $\tilde{S}_T$

- Control points of $\psi_{(\sigma,v)}$ are computed by a least squares procedure.

- But, this time, the sample points are the vertices of S_T that correspond to the points in T_v through Floater's parametrization!

Adaptive Fitting

Create S from $\tilde{S}_T$

Adaptive Fitting

Create S from $\tilde{S}_T$

- For each point p in T_v, we compute the **approximation error**,

$$\|q - \psi_{(\sigma,v)(p)}\|,$$

where q is the vertex of S_T corresponding to p through Floater's parametrization.

Adaptive Fitting

Create S from $\tilde{S}_T$

- For each point p in T_v, we compute the **approximation error**,

$$\left\| q - \psi_{(\sigma,v)(p)} \right\| ,$$

  where q is the vertex of S_T corresponding to p through Floater's parametrization.

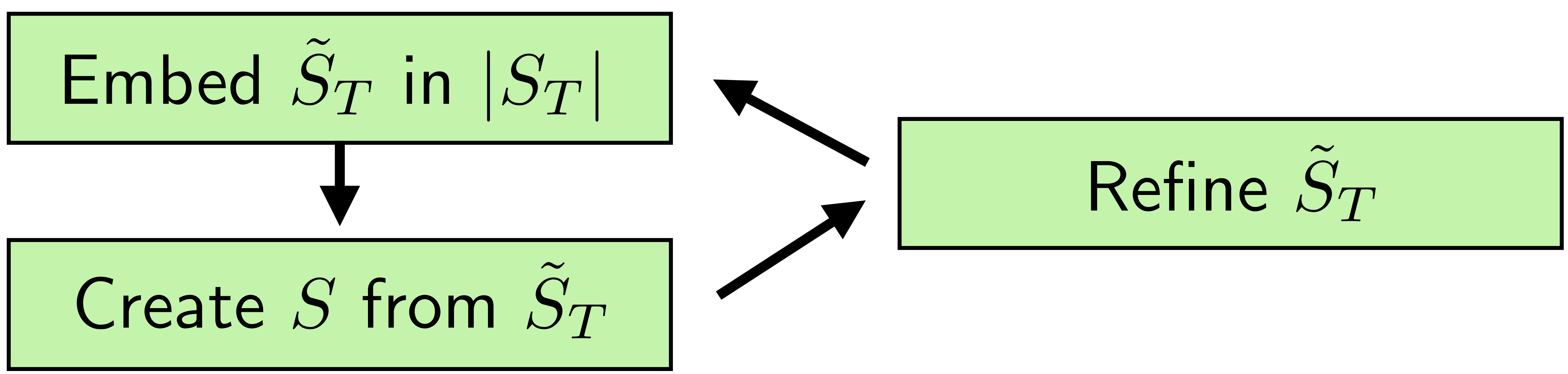
- If the above error is smaller than the given number ϵ, we keep computing $\psi_{(\tau,u)}$, for all pairs $(\tau, u) \in I$. Otherwise, we stop this process and go to the refinement step.

Adaptive Fitting

Adaptive Fitting

Refine $\tilde{S}_T$

- We locally refine $\tilde{S}_T$ using the stellar operations and the 4-8 refinement, and then embed the resulting $\tilde{S}_T$ in $|S_T|$ again.



Discrete Geodesics

Discrete Geodesics

- **Locally Shortest Geodesic**:
  A curve joining two points, A and B, on a polyhedral surface. It is a local minimum of the length functional.

Discrete Geodesics

- **Locally Shortest Geodesic**:
  A curve joining two points, A and B, on a polyhedral surface. It is a local minimum of the length functional.

- **Straighest Geodesic**:
  A curve beginning at point A and moving in the direction of the tangent vector. It has zero *discrete geodesic curvature* everywhere.

Discrete Geodesics

Discrete Geodesics

Locally shortest geodesics:

Discrete Geodesics

Locally shortest geodesics:

Exact algorithms:

- Mitchell, Mount, and Papadimitriou (1987)
- Chen and Han (1996)
- Kapoor (1999)
- Surazhsky, Surazhsky, Kirsanov, Gortler, and Hoppe (2005)

Discrete Geodesics

Locally shortest geodesics:

Discrete Geodesics

Locally shortest geodesics:

Approximate algorithms:

- Kimmel and Sethian (1998)
- Martínez, Velho, and Carvalho (2004)
- Surazhsky, Surazhsky, Kirsanov, Gortler, and Hoppe (2005)

Discrete Geodesics

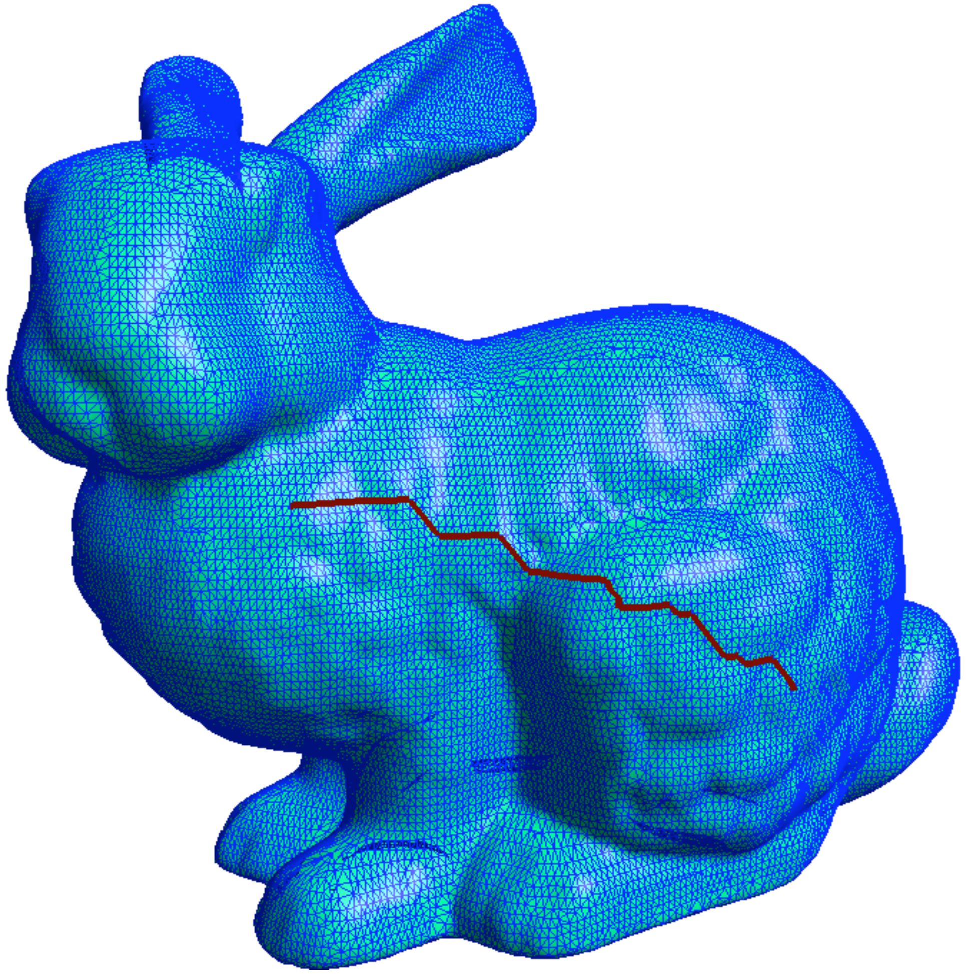
Discrete Geodesics

A Two-Step Algorithm:

Discrete Geodesics

A Two-Step Algorithm:

Step 1:
Find an initial curve joining A and B.



Discrete Geodesics

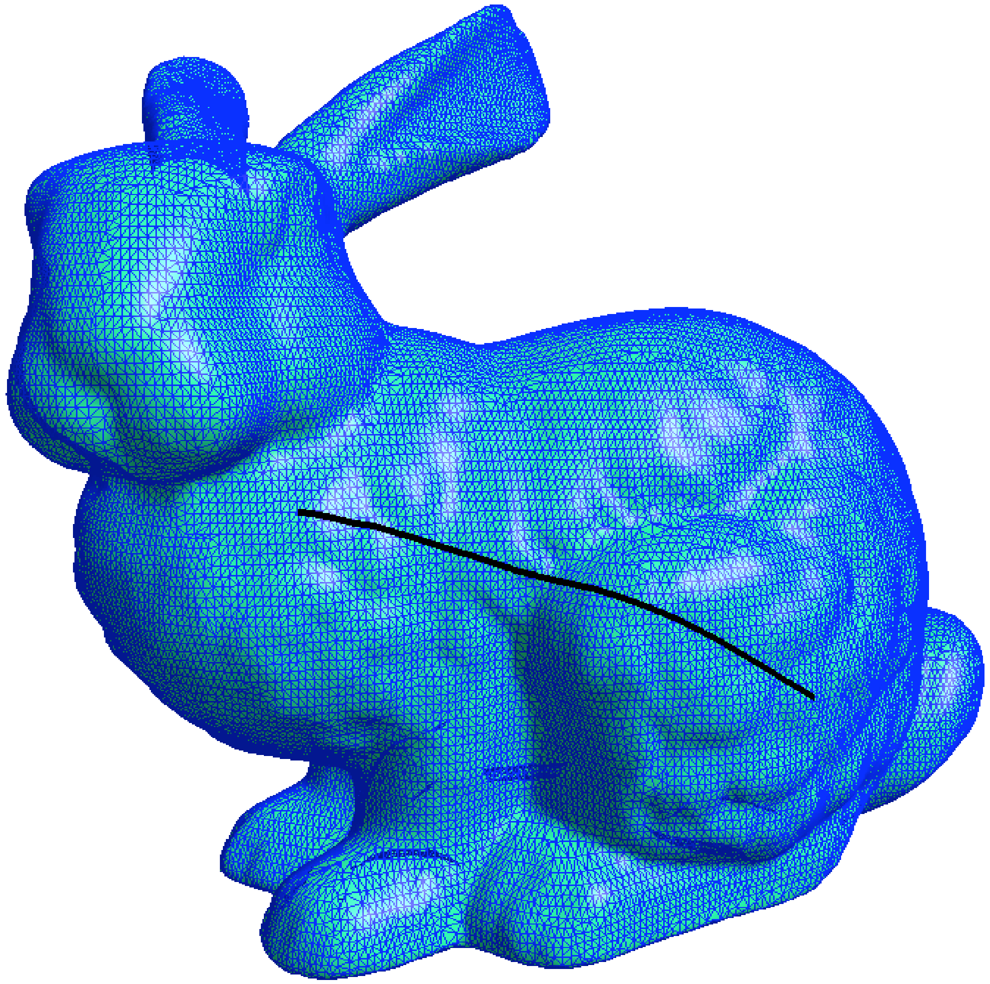
A Two-Step Algorithm:

Discrete Geodesics

A Two-Step Algorithm:

Step 2:

Iteratively modify the position of each curve vertex.



Discrete Geodesics

Discrete Geodesics

Step 1:
Find an initial curve joining A and B.

Discrete Geodesics

Step 1:
Find an initial curve joining A and B.

- Fast Marching Method

Discrete Geodesics

Step 1:
Find an initial curve joining A and B.

- Fast Marching Method

- Define a distance function at the vertices, $d(v) = \mathrm{dist}(A, V)$, using an approximation of the eikonal equation

$$|\nabla d| = 1 \, .$$

Discrete Geodesics

Step 1:

Find an initial curve joining A and B.

Discrete Geodesics

Step 1:
Find an initial curve joining A and B.

- Back-integrate the differential equation:

$$\begin{cases} \dfrac{d\Gamma_0}{ds}(s) & = & -\nabla d(\Gamma_0(s)) \\ \\ \Gamma_0(0) & = & B\,. \end{cases}$$

Discrete Geodesics

Discrete Geodesics

Step 2:

Iteratively modify the position of each curve vertex.

Discrete Geodesics

Step 2:

Iteratively modify the position of each curve vertex.

- Given a curve Γ_i, we want to get a shorter curve, Γ_{i+1}, with the same endpoints.

Discrete Geodesics

Step 2:
Iteratively modify the position of each curve vertex.

- Given a curve Γ_i, we want to get a shorter curve, Γ_{i+1}, with the same endpoints.
  - a geodesic should be a line segment in the interior of a face;

Discrete Geodesics

Step 2:

Iteratively modify the position of each curve vertex.

- Given a curve Γ_i, we want to get a shorter curve, Γ_{i+1}, with the same endpoints.
  - a geodesic should be a line segment in the interior of a face;
  - the curve will be a polygonal with nodes belonging to the edges of the mesh;

Discrete Geodesics

Step 2:
Iteratively modify the position of each curve vertex.

- Given a curve Γ_i, we want to get a shorter curve, Γ_{i+1}, with the same endpoints.
  - a geodesic should be a line segment in the interior of a face;
  - the curve will be a polygonal with nodes belonging to the edges of the mesh;
  - the algorithm will correct the position of the curve nodes;

Discrete Geodesics

Step 2:
Iteratively modify the position of each curve vertex.

- Given a curve Γ_i, we want to get a shorter curve, Γ_{i+1}, with the same endpoints.
  - a geodesic should be a line segment in the interior of a face;
  - the curve will be a polygonal with nodes belonging to the edges of the mesh;
  - the algorithm will correct the position of the curve nodes;
  - distinct behavior for "edge nodes" and "vertex nodes".

Discrete Geodesics

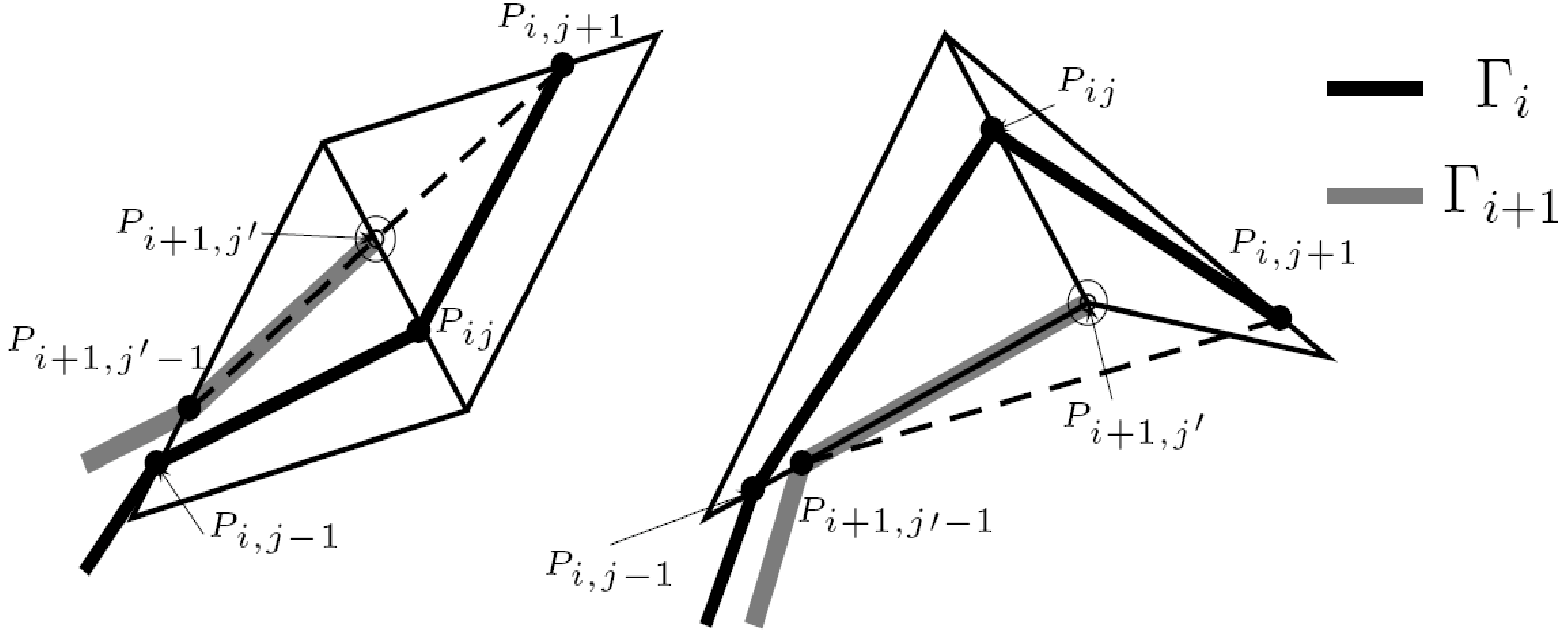
Step 2:

Iteratively modify the position of each curve vertex.

Discrete Geodesics

Step 2:
Iteratively modify the position of each curve vertex.

Edges nodes:



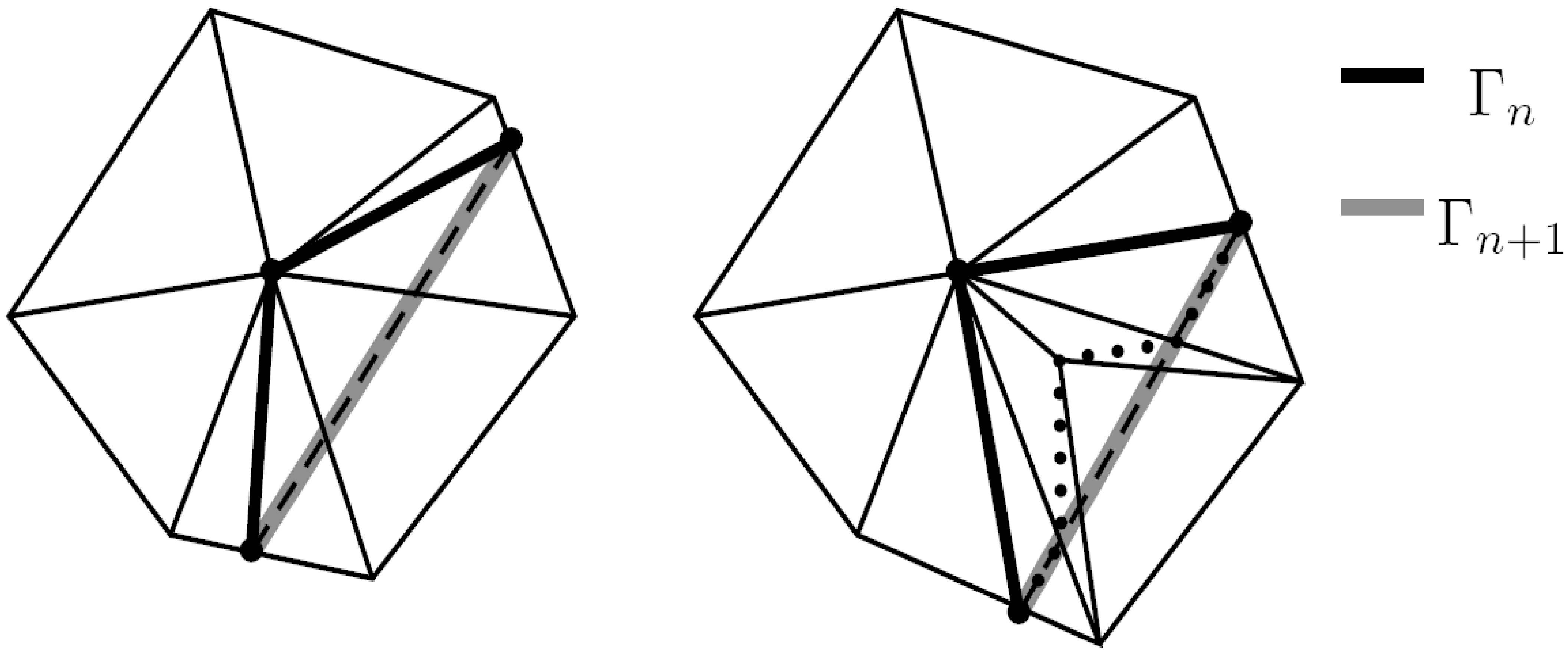
Discrete Geodesics

Step 2:
Iteratively modify the position of each curve vertex.

Discrete Geodesics

Step 2:
Iteratively modify the position of each curve vertex.

Vertex nodes:



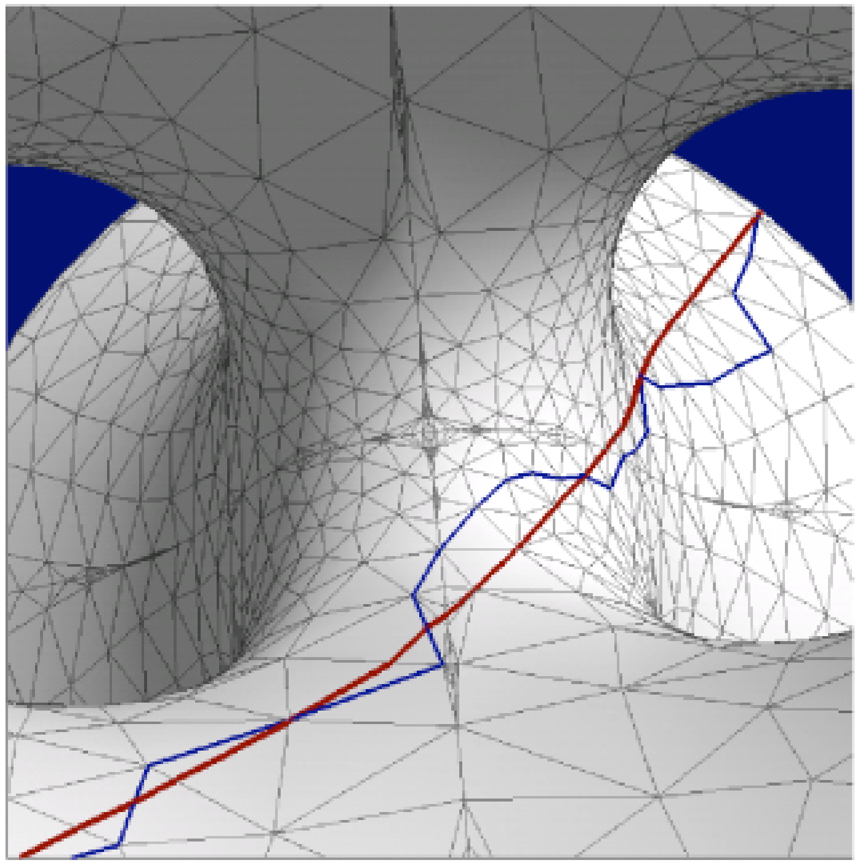
Discrete Geodesics

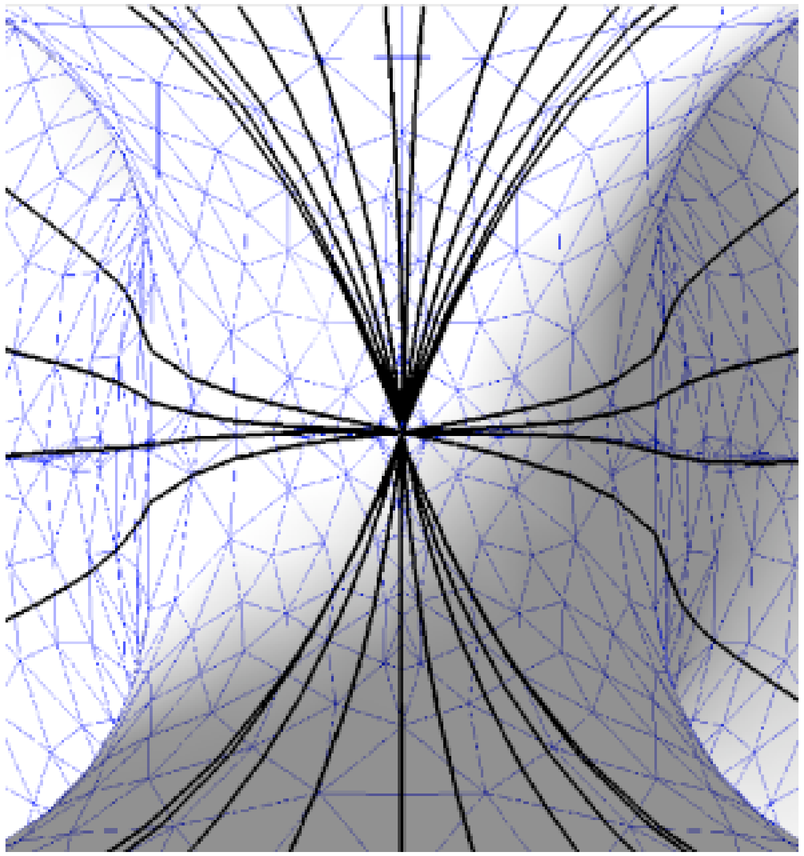
Discrete Geodesics

Examples:

Discrete Geodesics

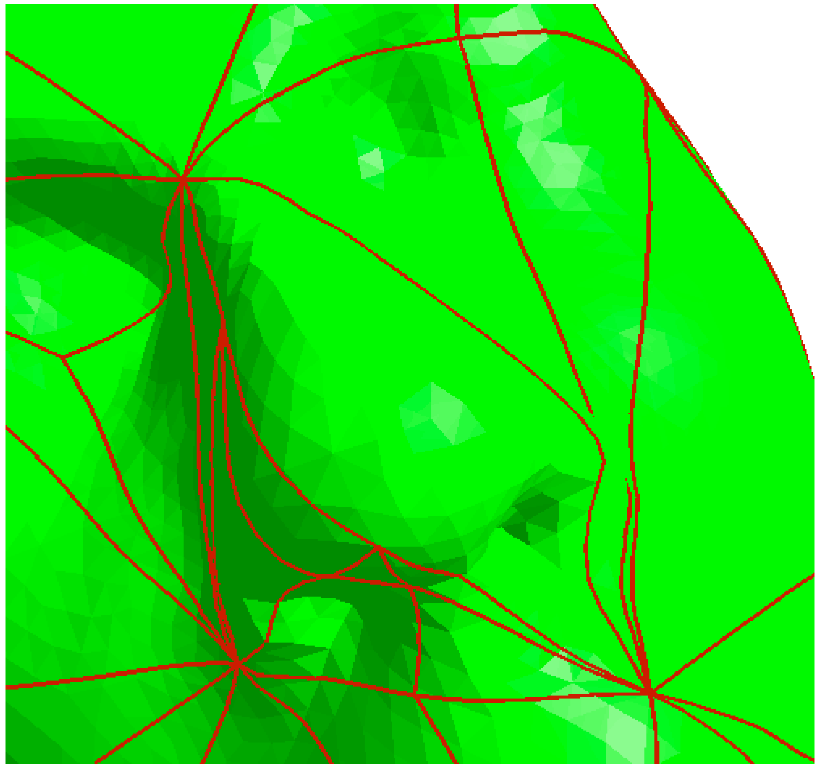
Examples:

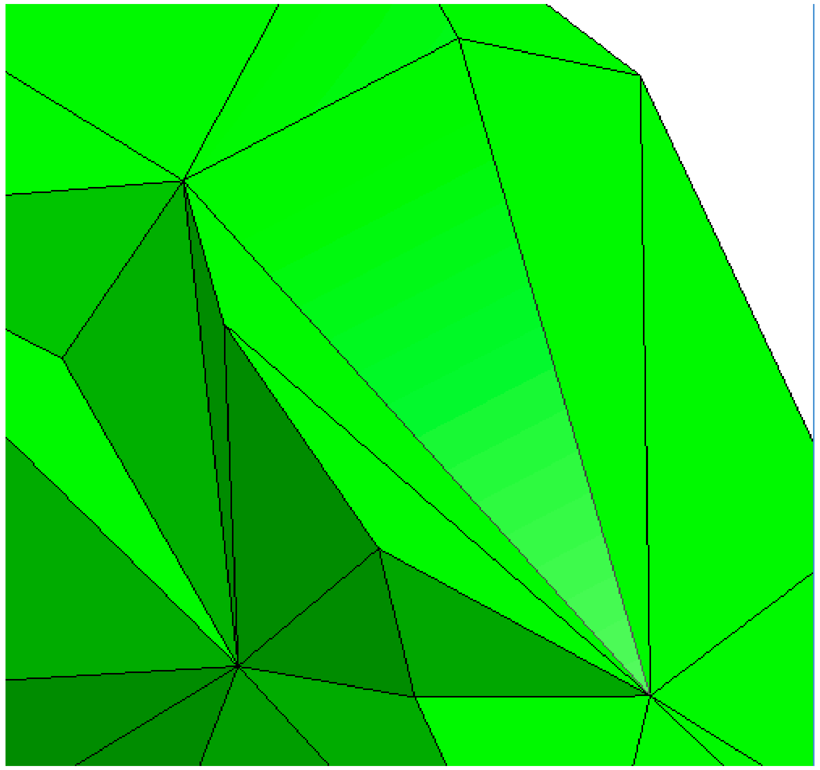




Discrete Geodesics

Discrete Geodesics

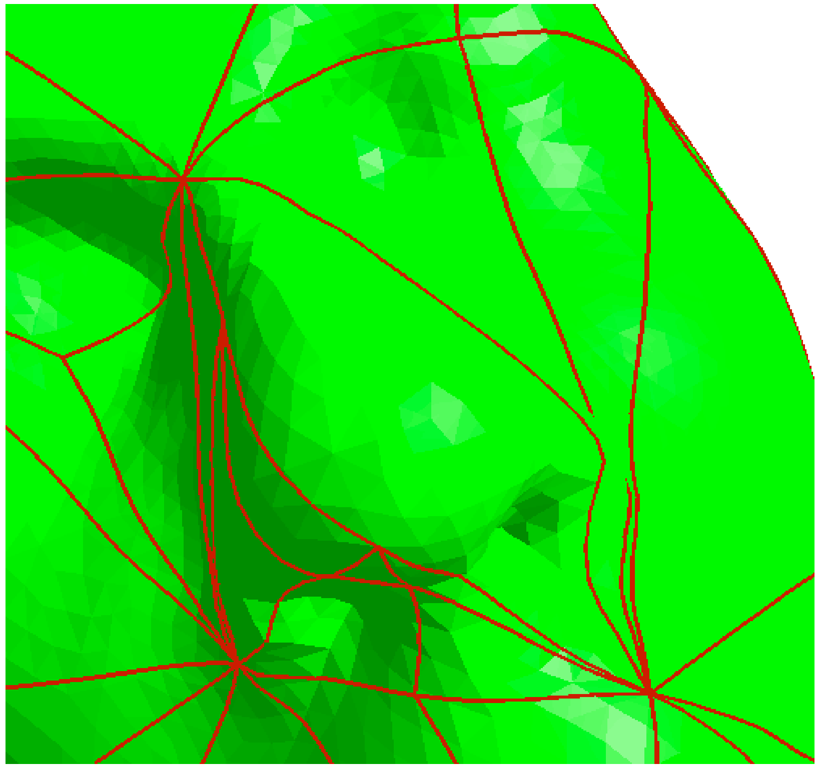


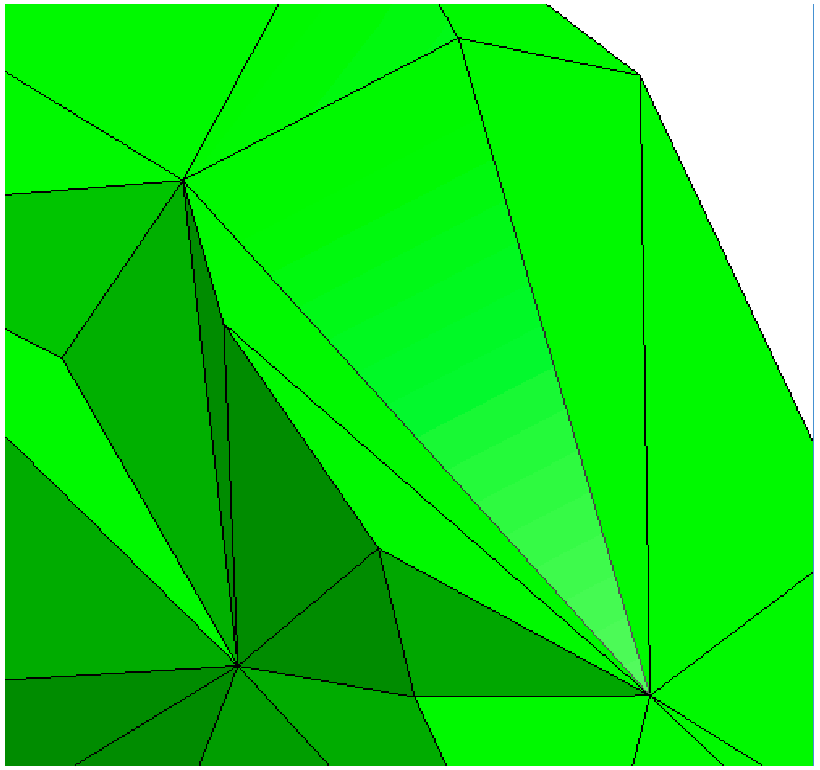


Discrete Geodesics

Adaptive Fitting:

When defining geodesic triangles, we can violate the manifold property of the geodesic mesh, as illustrated by the figure below:





Discrete Geodesics

Discrete Geodesics

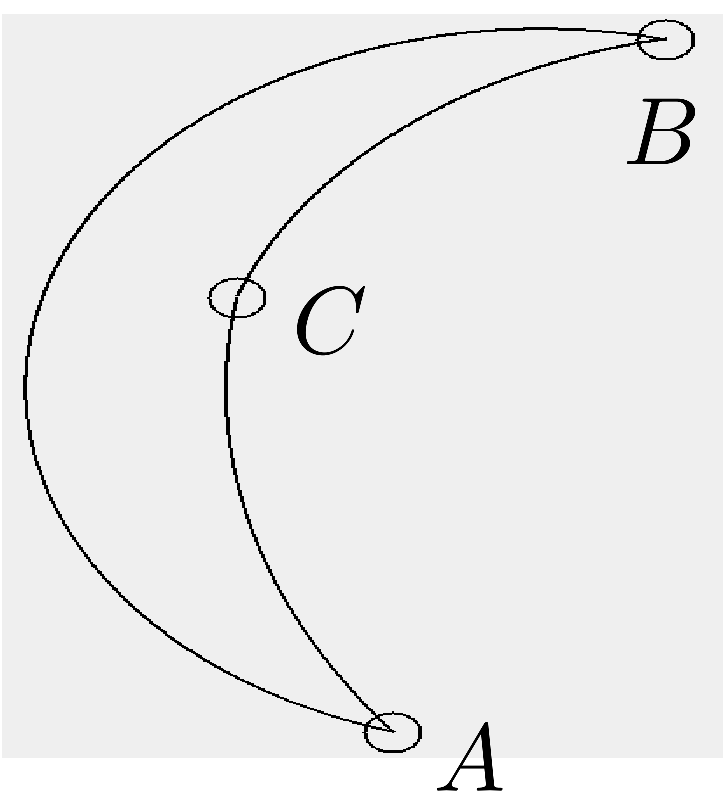
Solution:

Compute the geodesic curve AB as the geodesic resulting from using the concatenation of geodesics AC and CB as initial approximation:

Discrete Geodesics

Solution:

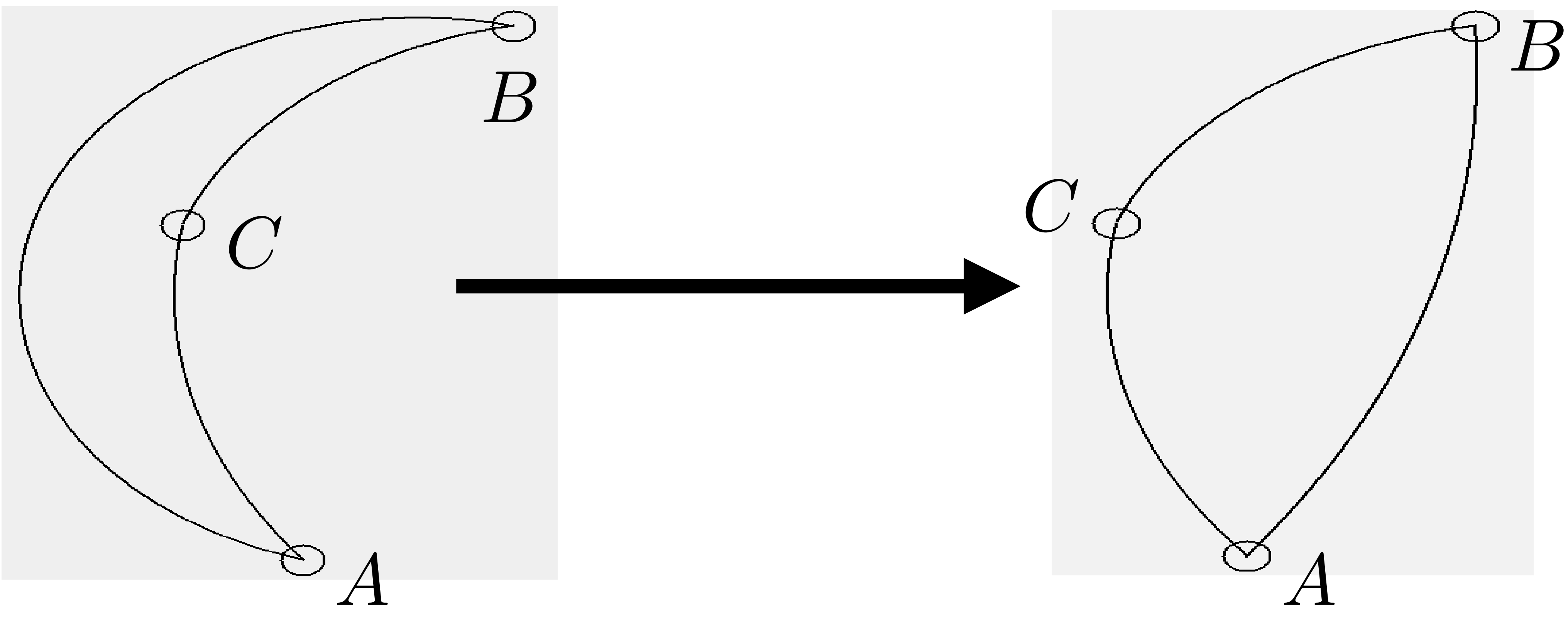
Compute the geodesic curve AB as the geodesic resulting from using the concatenation of geodesics AC and CB as initial approximation:



Discrete Geodesics

Solution:

Compute the geodesic curve AB as the geodesic resulting from using the concatenation of geodesics AC and CB as initial approximation:



Conclusions

Conclusions

- Adaptive fitting pipeline is not new, but its elements are.

Conclusions

- Adaptive fitting pipeline is not new, but its elements are.
- This is the "real deal" when it comes to comparisons between smooth surfaces and very dense polygonal meshes.

Conclusions

- Adaptive fitting pipeline is not new, but its elements are.
- This is the "real deal" when it comes to comparisons between smooth surfaces and very dense polygonal meshes.
- Implementation of the adaptive fitting is still under development.

Conclusions

- Adaptive fitting pipeline is not new, but its elements are.
- This is the "real deal" when it comes to comparisons between smooth surfaces and very dense polygonal meshes.
- Implementation of the adaptive fitting is still under development.
- More specifically, the refinement step has not been completed.

Applications of Manifolds and Research Challenges

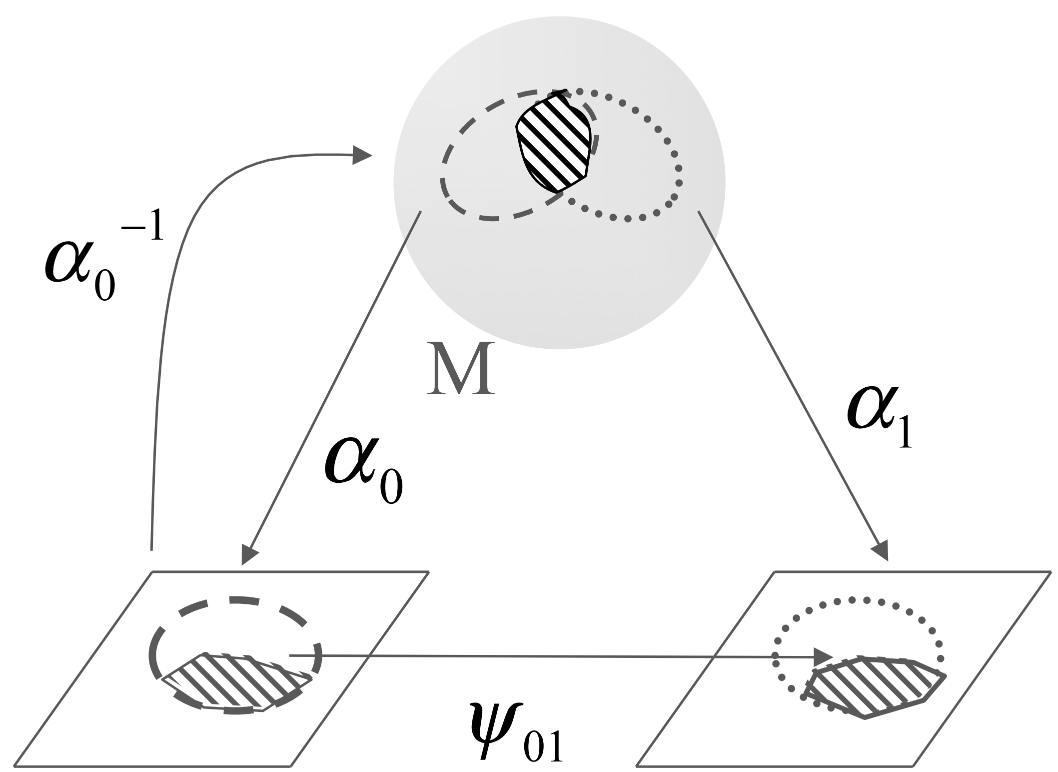
Luiz Velho
IMPA

Outline

- Concepts
- Illumination
- Appearance
- Simulation
- Faces
- Manifold Learning
- Wrap-up

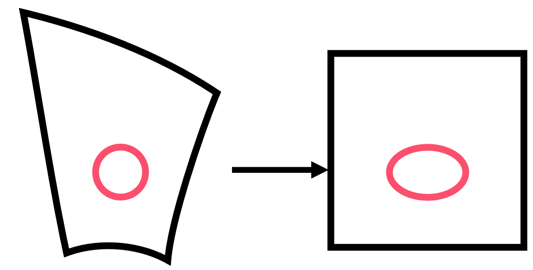
Manifolds & Parametrization

- Two Points of View
  - Functions on surfaces
  - Functions defining surfaces

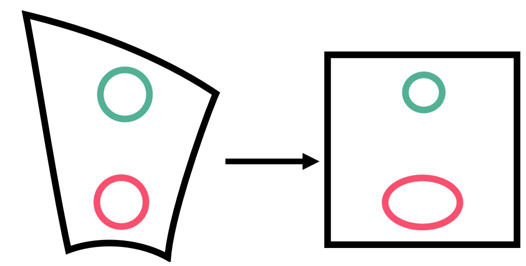


Desirable Properties

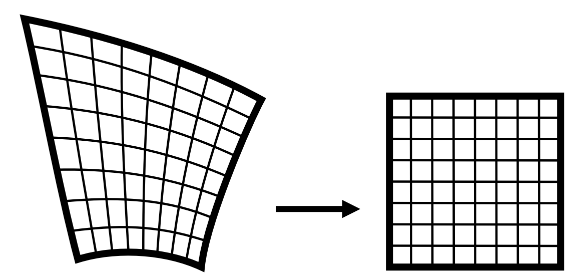
- Minimal Distortion
  - Angle



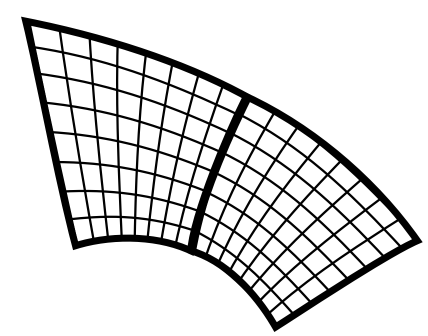
  - Area



- Smoothness
  - Differentiability



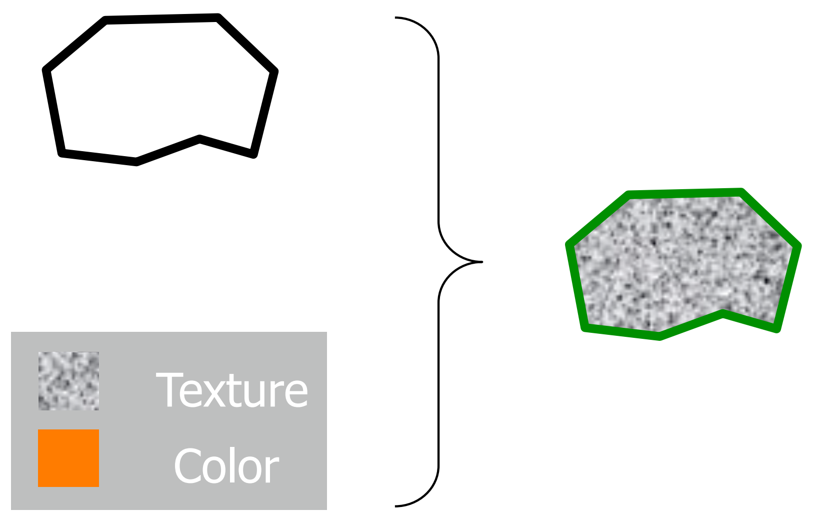
  - Continuity



Graphical Objects

- Shape U
  - Topology (*domain*)
    - Abstract Manifold
  - Geometry (*function*)
    - Embedding
- Attributes f
  - Functions (*co-domain*)

$$O = (U, f)$$



G.O. Manifold Setting

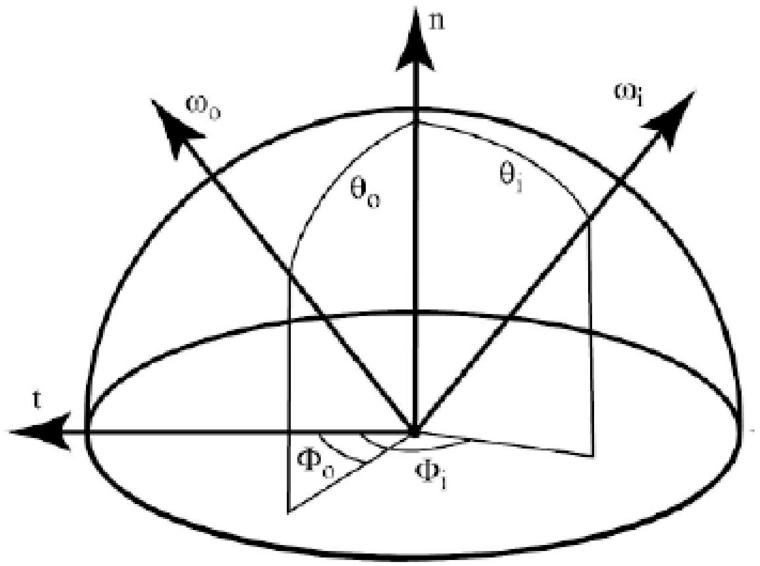
- Canonical Surfaces
  - Fixed Shape (defined *apriori*)
  - Variable Functions (complex)
    - *ex: Sphere*
- Arbitrary Surfaces
  - Complex Shape
  - Computation on Surfaces (attributes)
  - Building / Transforming (shape)
    - *ex: Triangle Meshes*

Applications

- Illumination
  - Canonical Manifold + Functions
- Appearance and Simulation
  - Pseudo-Manifold + Attributes
- Faces
  - Manifold + Geometric Deformation
- Surface Reconstruction
  - Pseudo-Manifold / Topology Estimation

Illumination

- Functions on the Sphere
  - Light Fields / BRDFs
- Applications
  - Capture / Synthesis
- Construction [Grimm 2002]



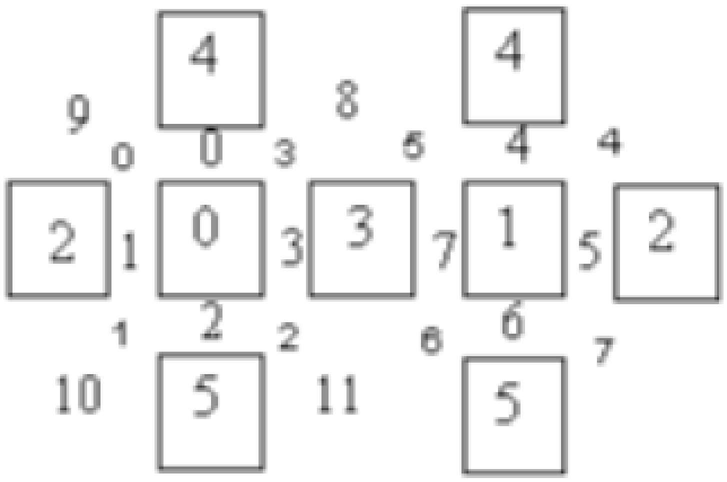
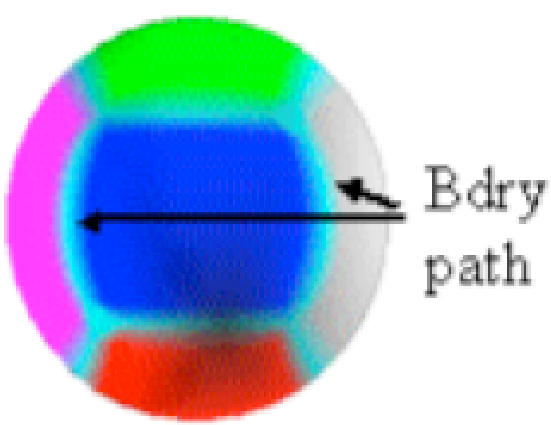
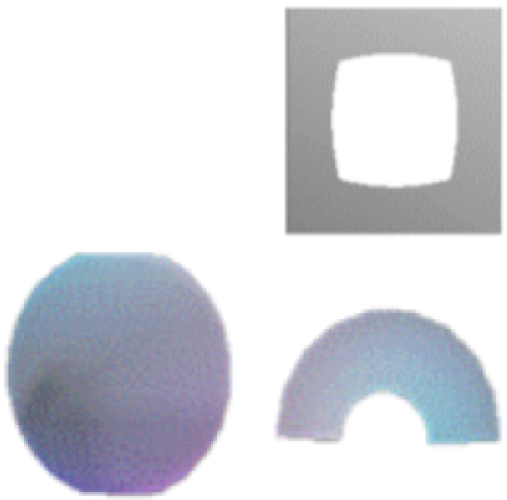


Chart (squares), edge, and



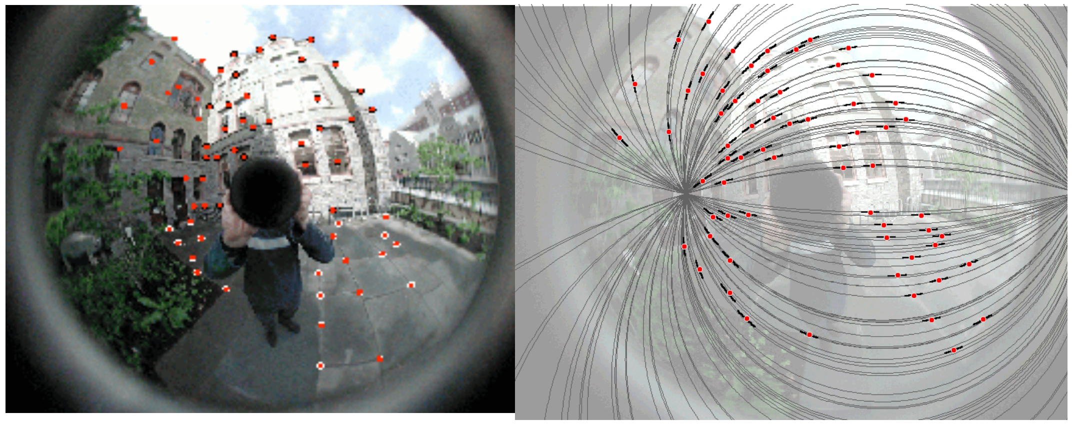
Bottom cap



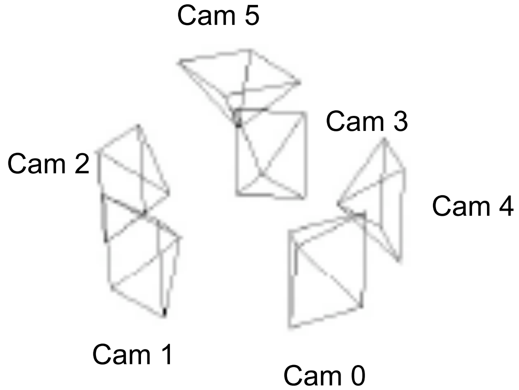
A single chart on the sphere

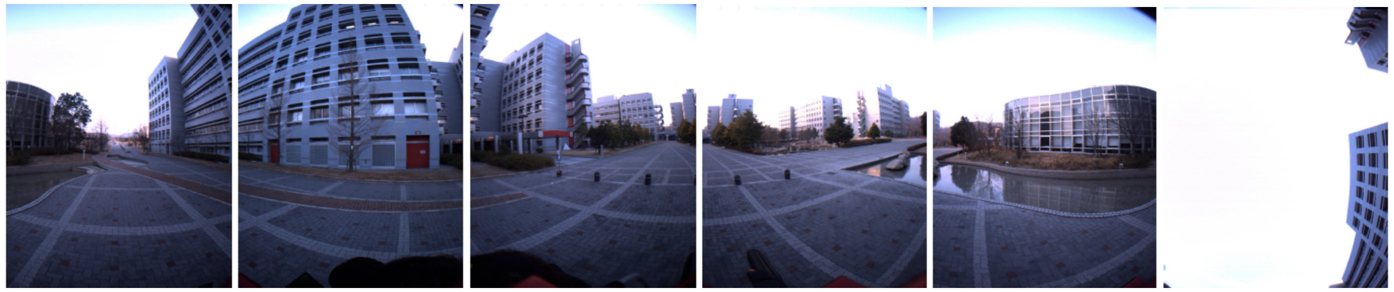
Omnidirectional Images

- Panoramic Cameras
  - Processing



- Multi-Camera Assembly
  - Stitching / Blending

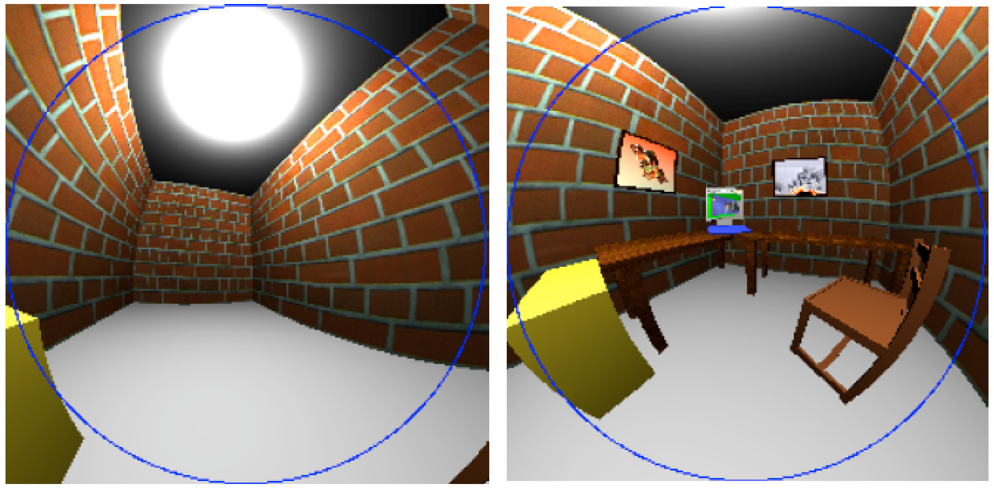


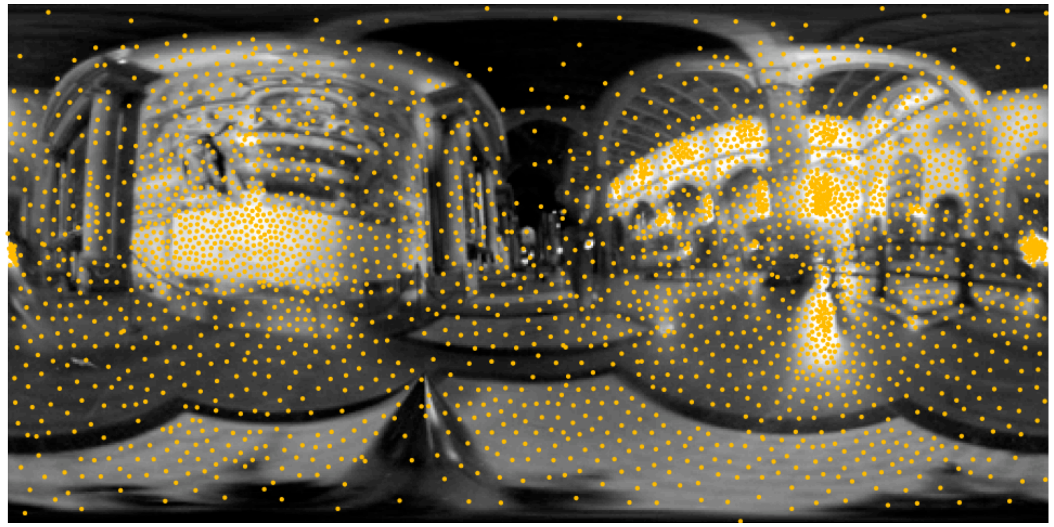


Cam 0 Cam 1 Cam 2 Cam 3 Cam 4 Cam 5

Illumination Maps

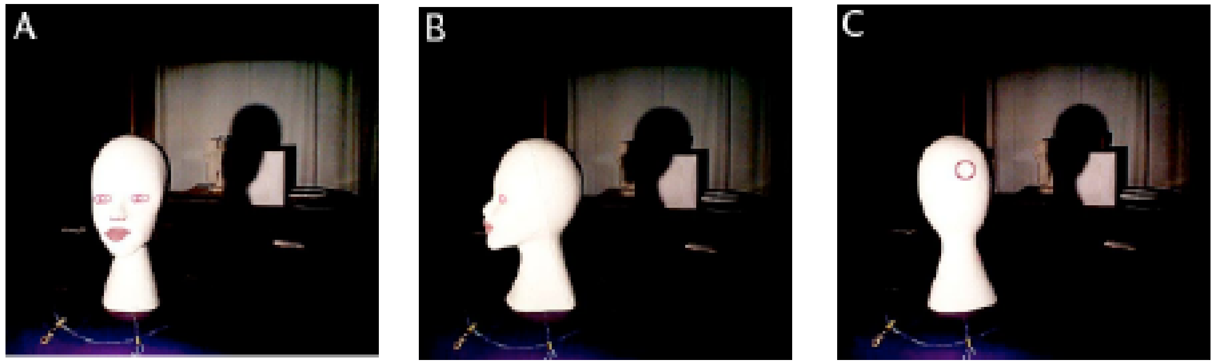
- Environment Maps
  - Area Sampling
- Light Maps
  - Stratification

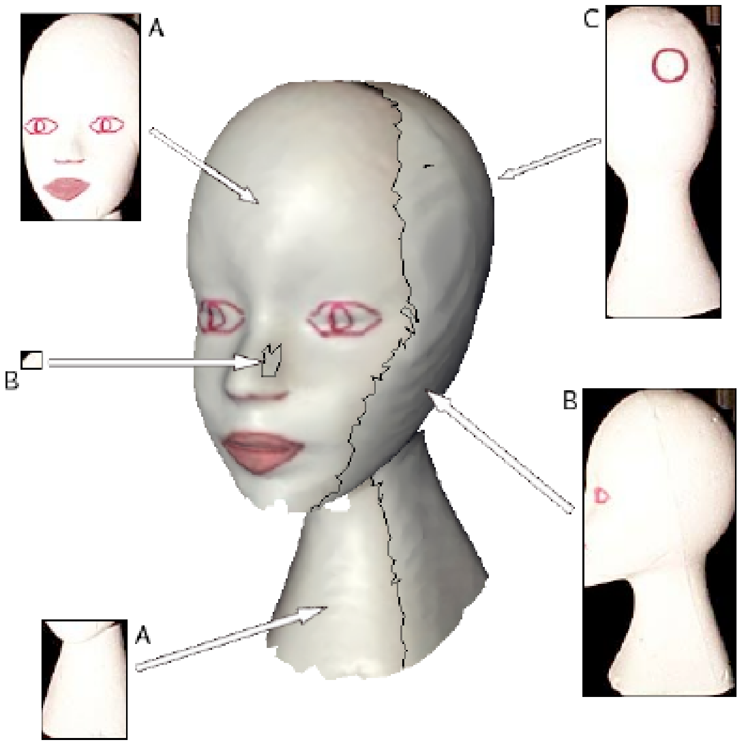




Surface Properties

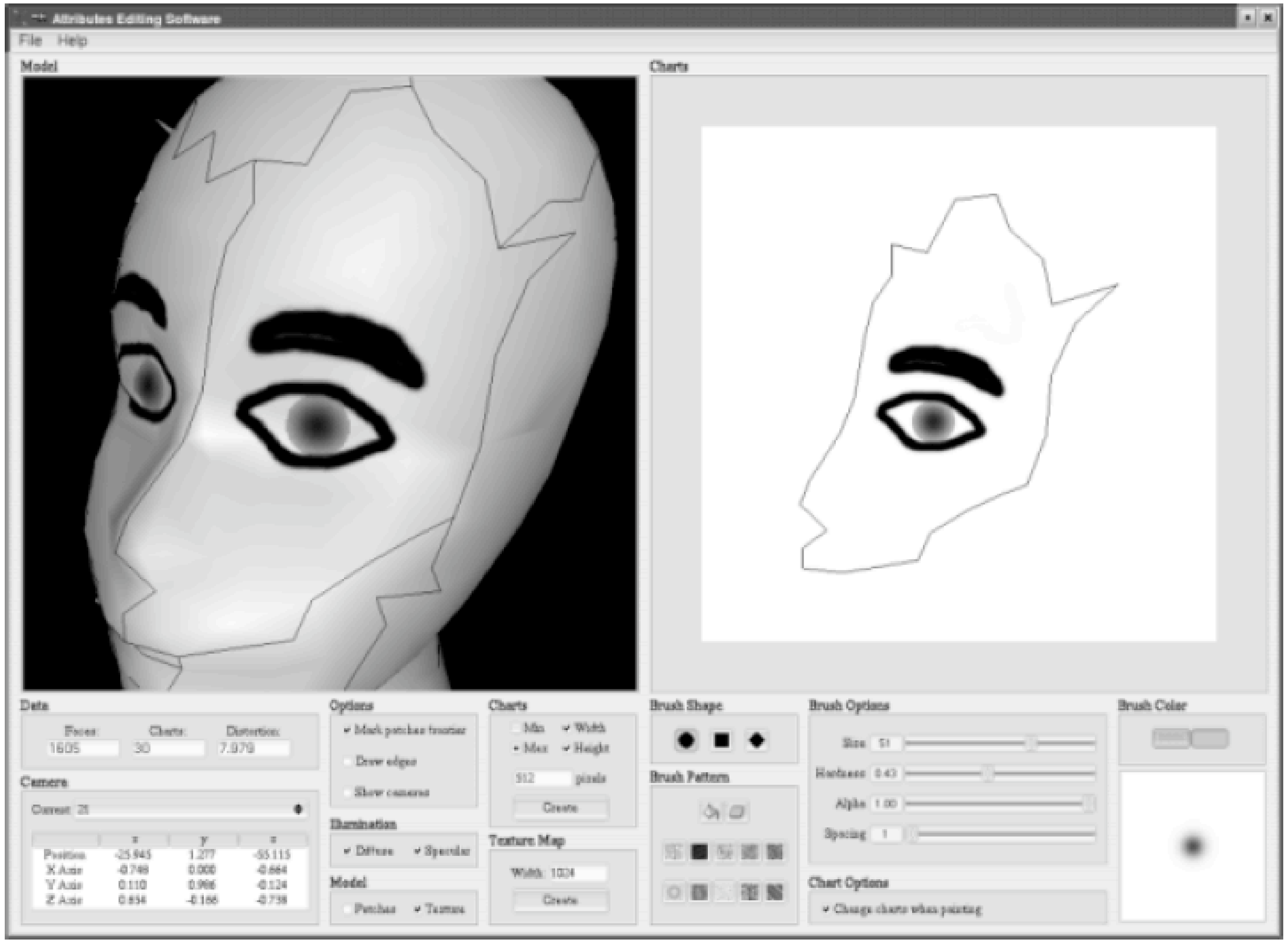
- Texture Atlas
  - Albedo
  - Normal Field
- Building from Images
  - Projective Map





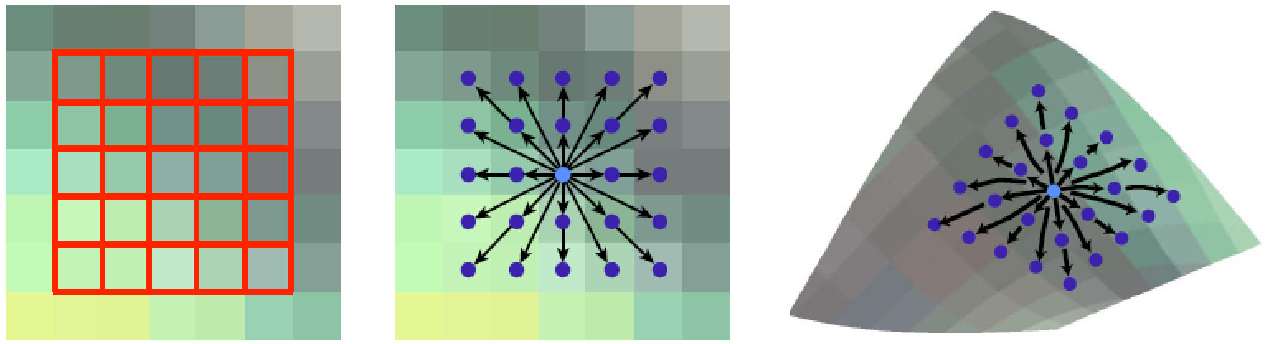
Painting

- Color
- Normals



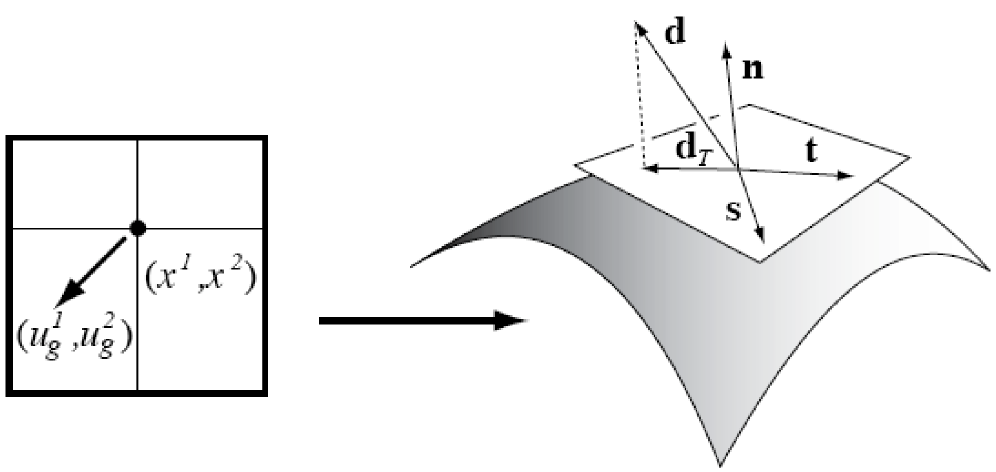
Texture Synthesis

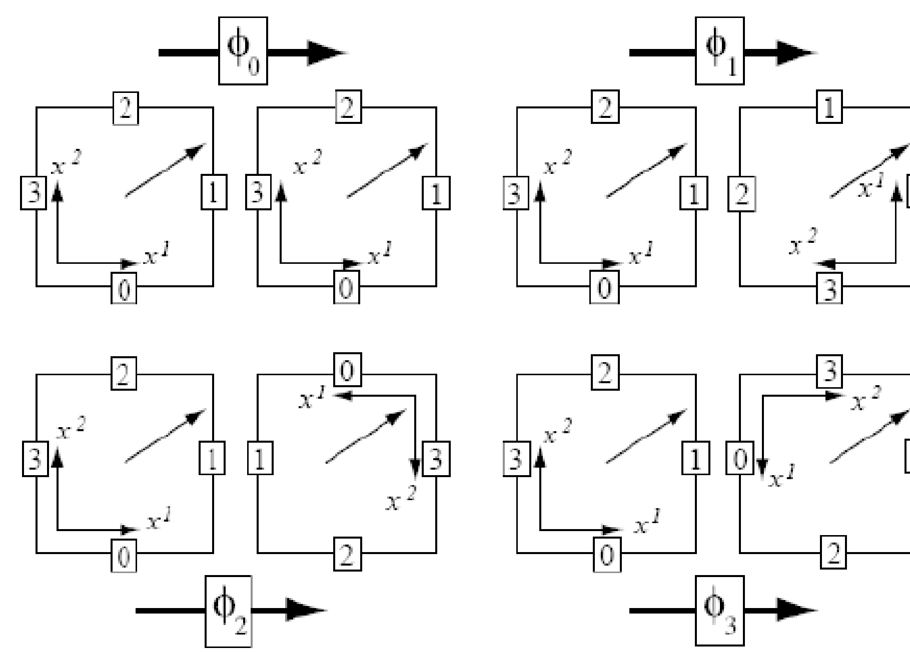
- Stationary / Quasi Stationary



Simulation

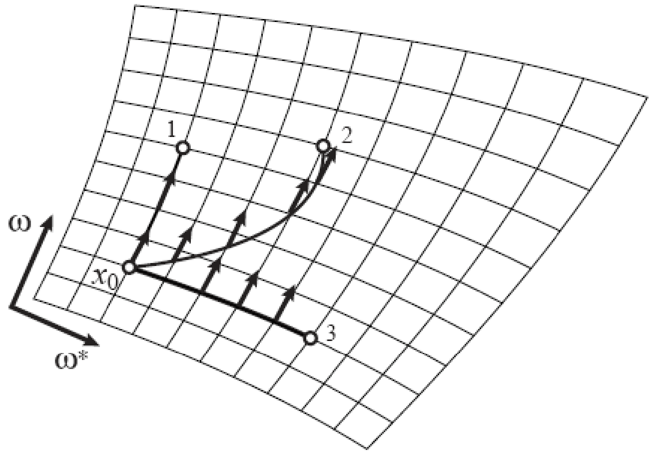
- Solving Equations on Manifolds
  - Surface Points
  - Local Neighborhoods

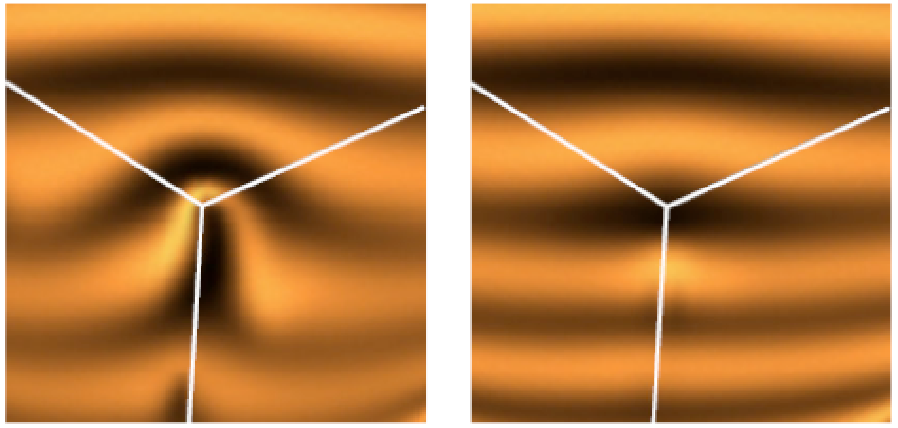


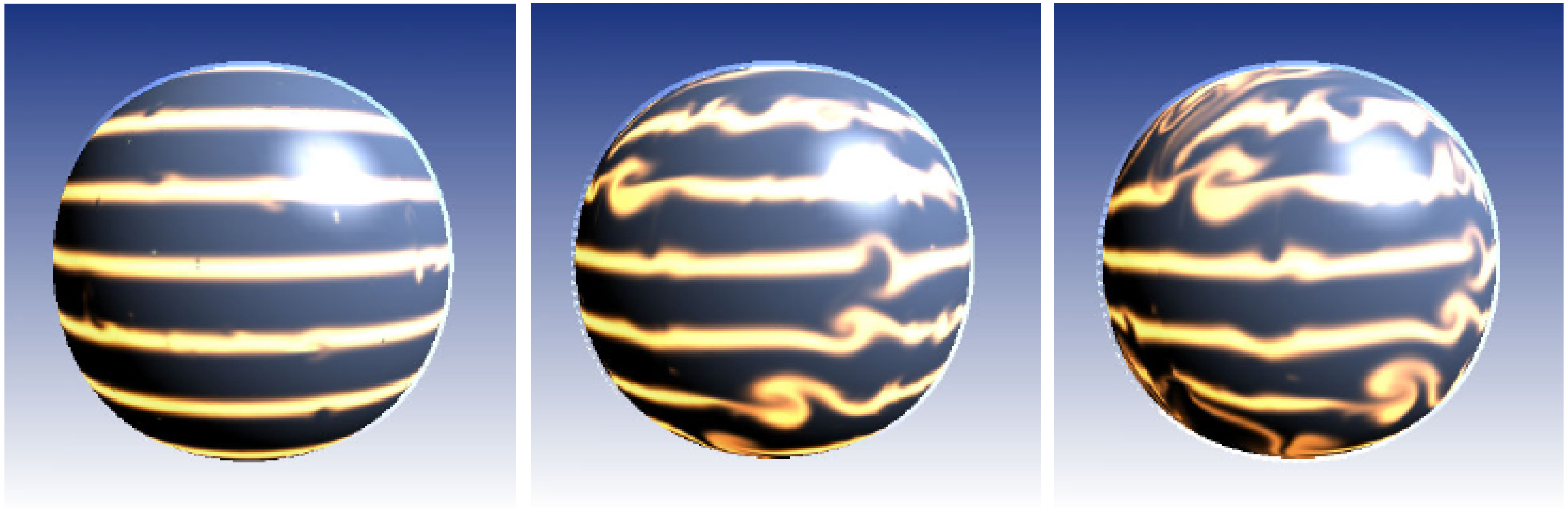


Fluids

- Vector Fields on Surfaces

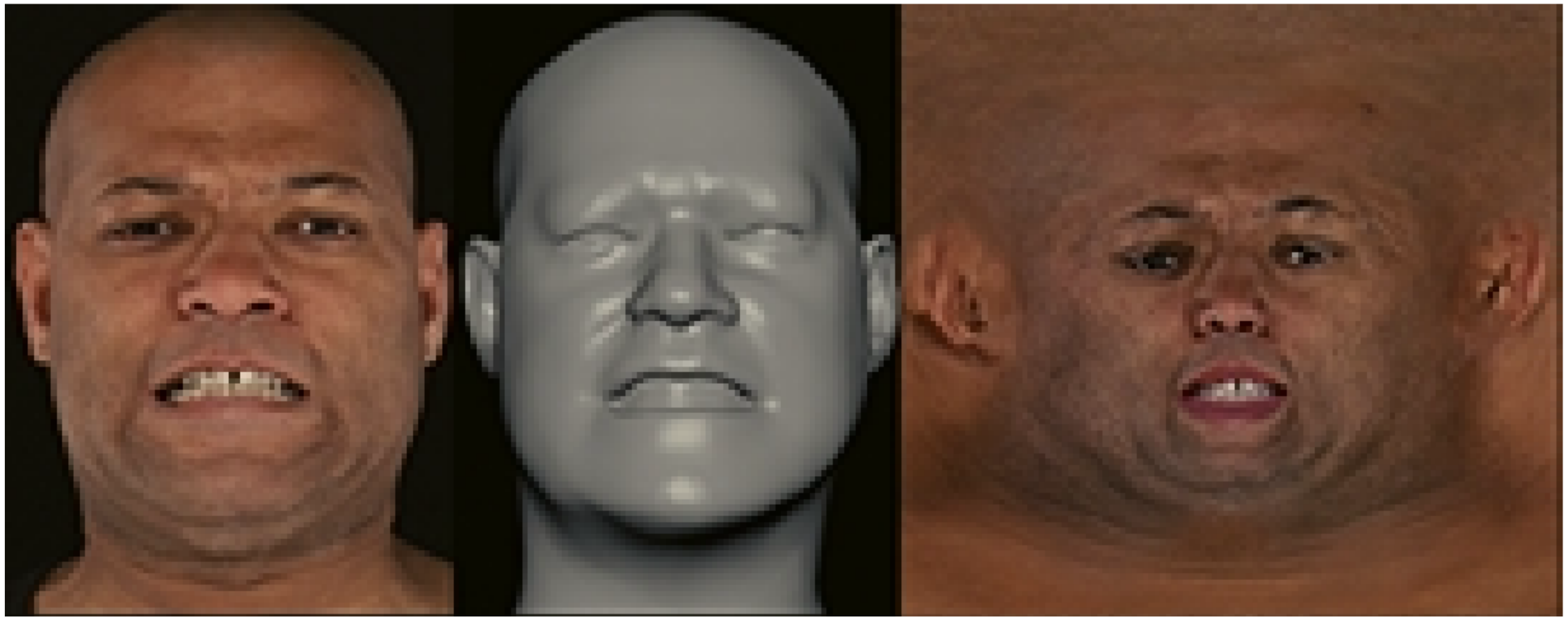






Faces

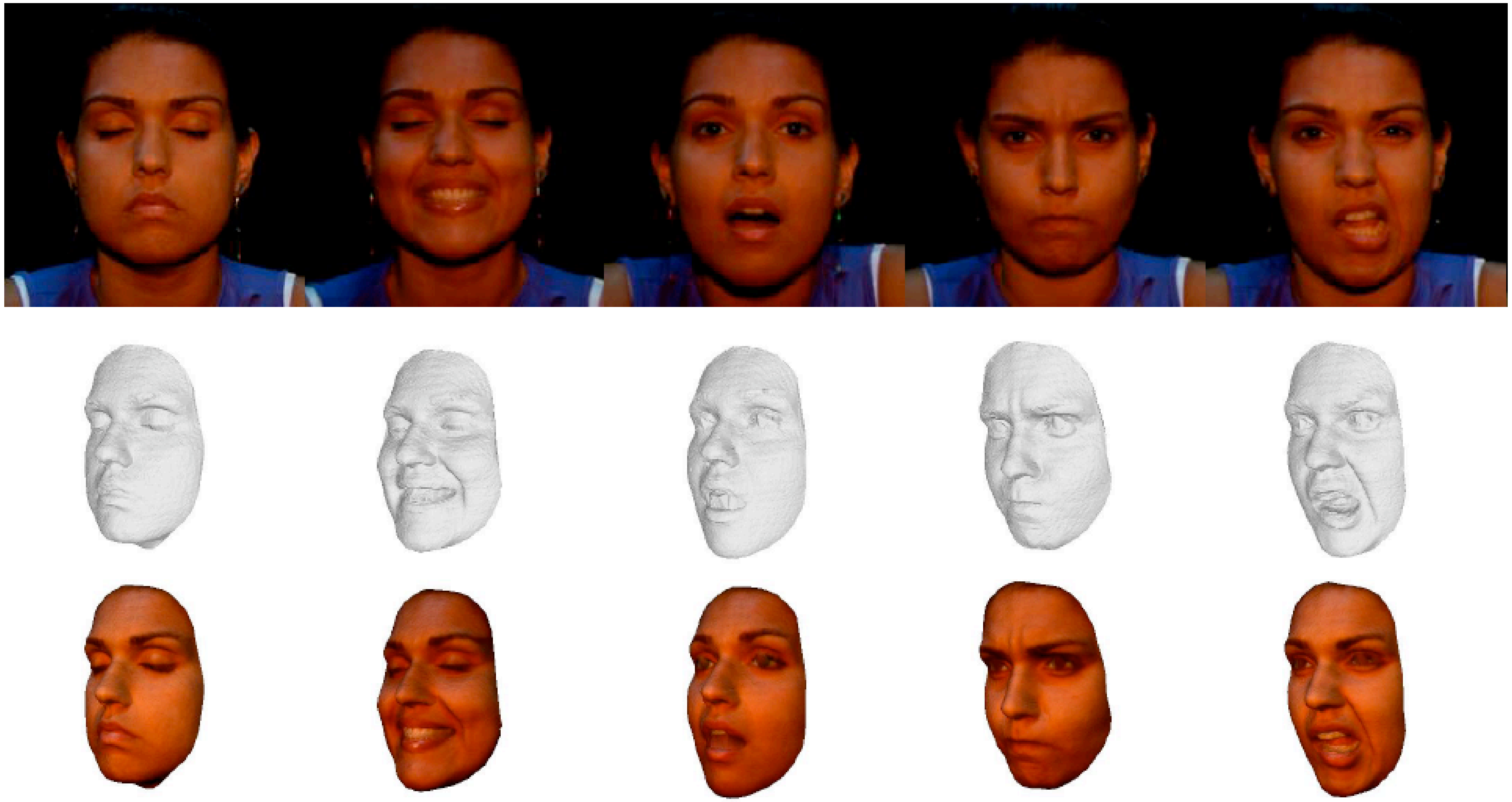
- Geometry + Appearance



[G. Borshukov et al SIGGRAPH 2003]

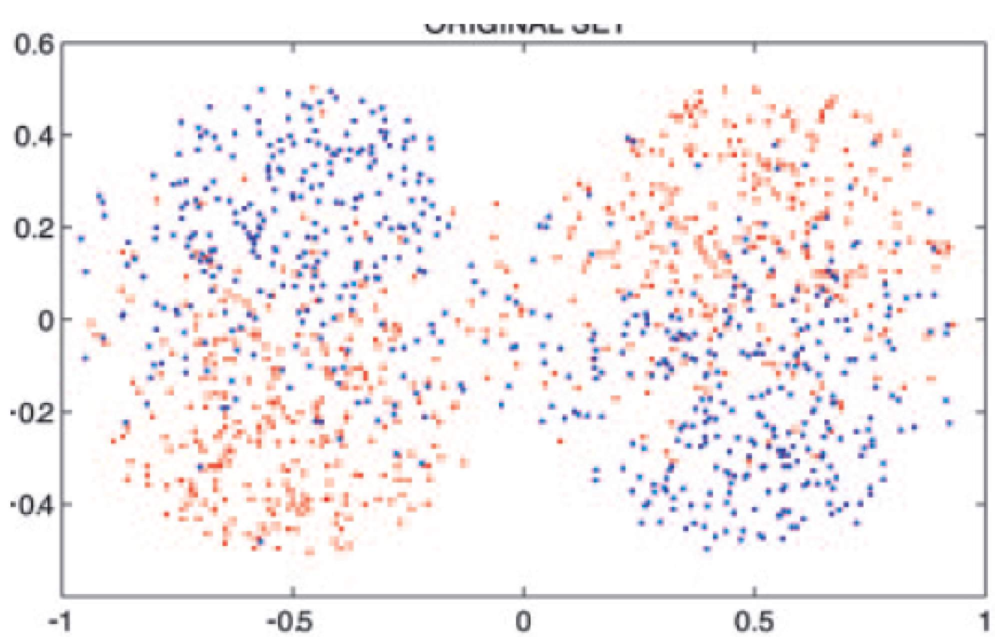
Facial Expressions

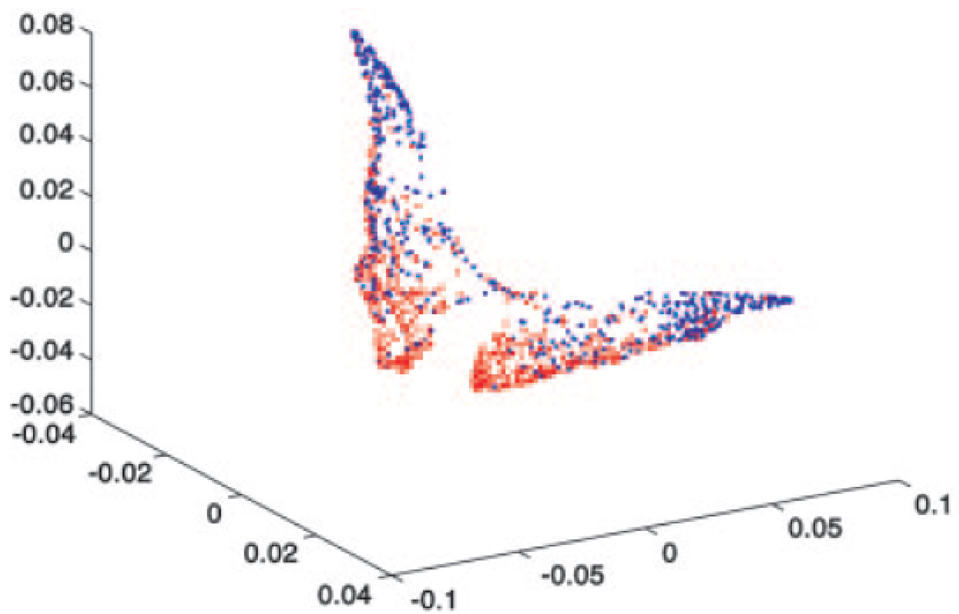
- Deformations



Manifold Learning

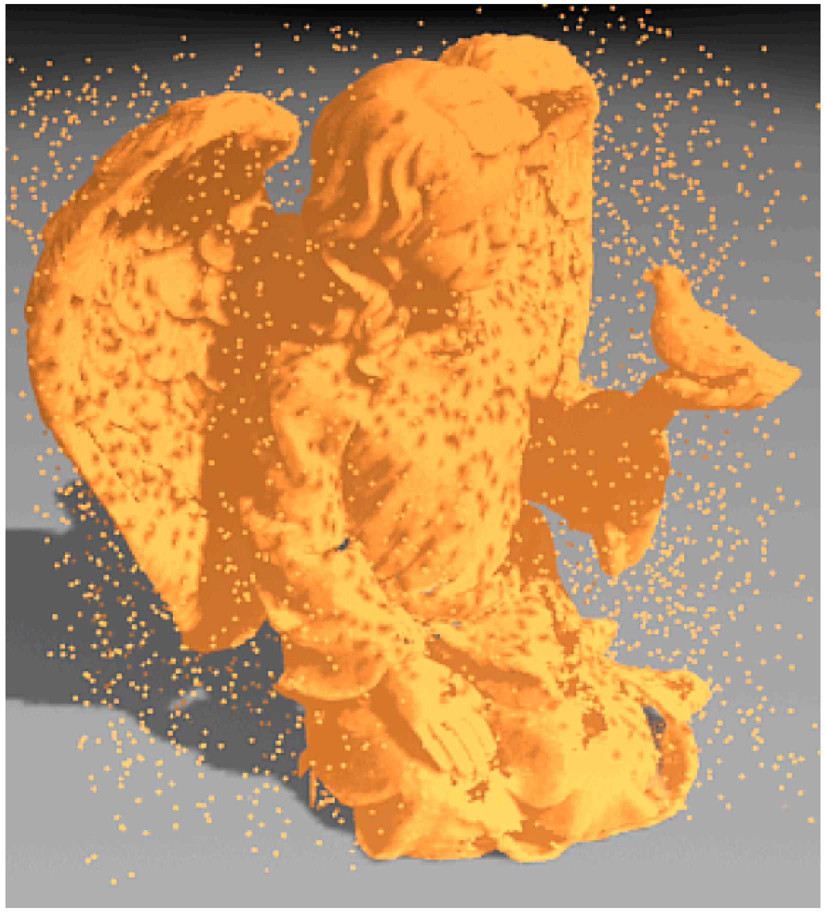
- Estimate from Data Samples
  - Topology
  - Geometry

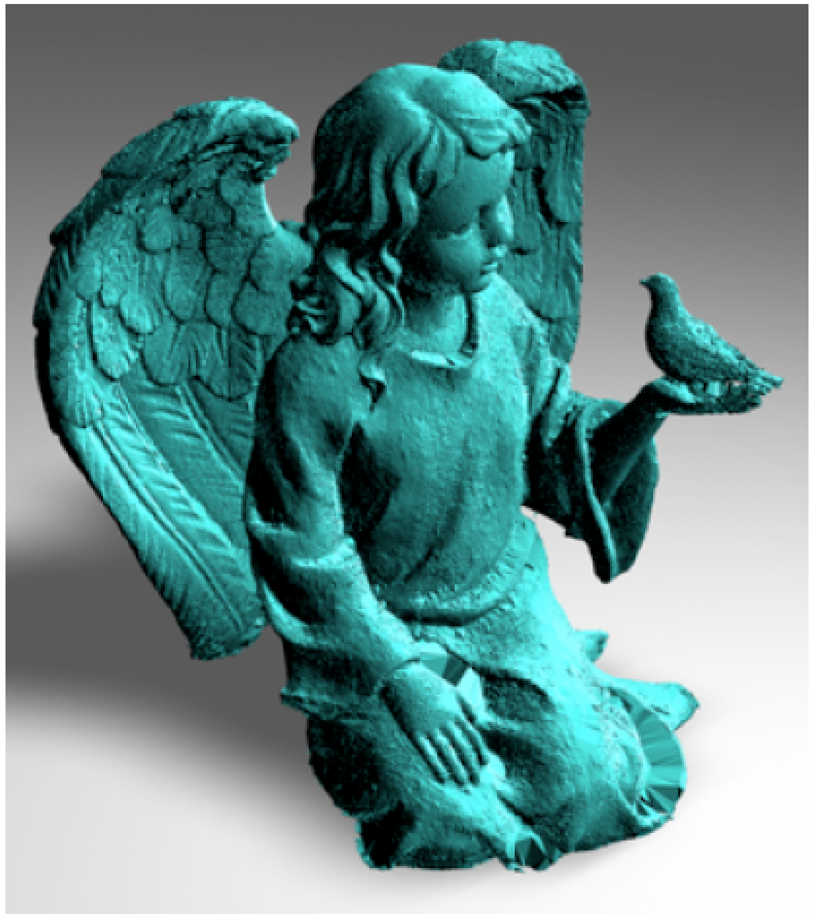




Surfaces

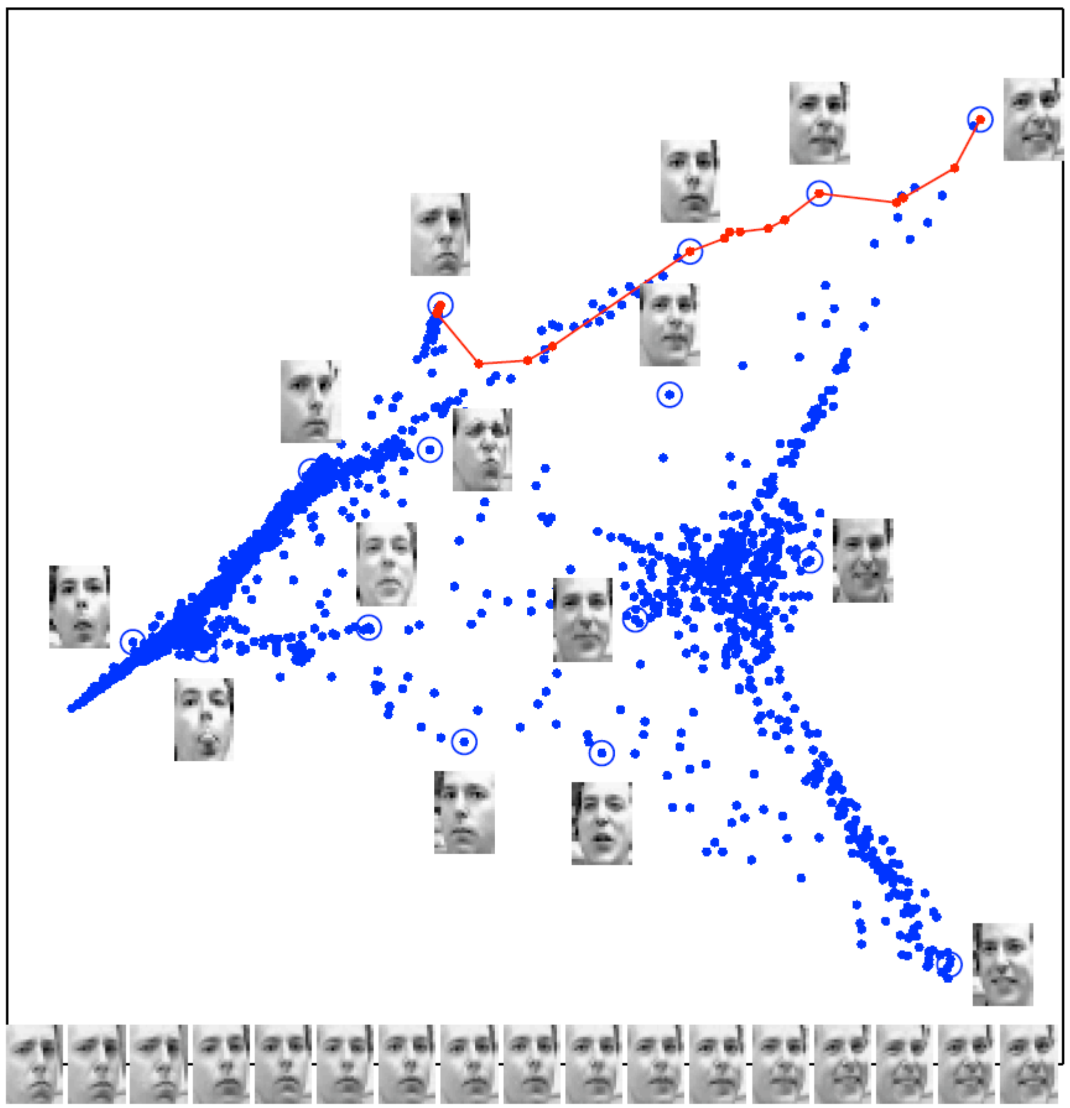
- Point Sets

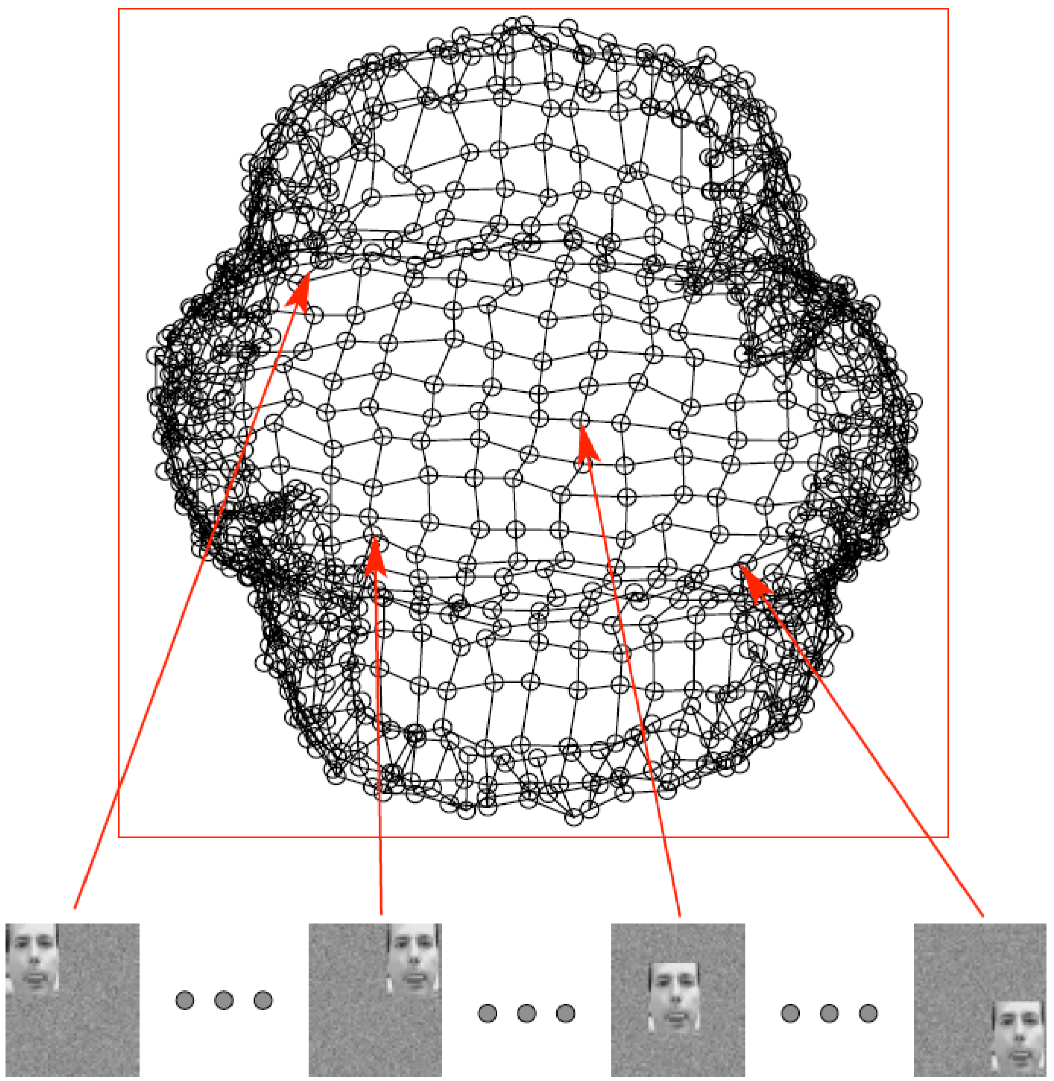




N-Dimensional Case

- ex: Facial Expressions





Challenges

- Multi-Resolution
  - Hierarchical Atlas
  - Dynamic Setting
- API
  - Intuitive
  - General

Questions ?