Charles University in Prague
Faculty of Mathematics and Physics

BACHELOR THESIS

Eva Vidličková

Discrete differential geometry and its applications

Mathematical Institute of Charles University

Supervisor of the bachelor thesis:	Doc. RNDr. Zbyněk Šír, Ph.D.
Study programme:	Mathematics
Specialization:	General Mathematics

Praha 2014

I would like to express my thanks to my supervisor Doc. RNDr. Zbyněk Šír, Ph.D. for his patient guiding through my Bachelor thesis and valuable advice. I would also like to thank Keenan Crane, Ph.D. for providing me with some very useful documents. Last but not least, my thanks go to my family for their great support.

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In .... on ....   Author signature

Název práce: Diskrétní diferenciální geometrie a její aplikace

Autor: Eva Vidličková

Katedra: Matematický ústav UK

Vedoucí bakalářské práce: Doc. RNDr. Zbyněk Šír, Ph.D., Matematický ústav UK

Abstrakt: V této bakalářské práci představujeme úvod do Diskrétní diferenciální geometrie. Budeme pracovat s diskrétními křivkami i diskrétními plochami. Nejprve zpomeneme pár základních definic a vět z klasické Diferenciální geometrie a pak zavedeme jejich diskrétní verzi tak, aby některé globální vztahy nadále platily. Na konci implementujeme tok střední křivosti, definován na diskrétních plochách a spustíme jej na dvou příkladech, které ukazují jeho schopnost zmenšovat rozlohu plochy. Tato vlastnost se využívá pro zhlazování diskrétních ploch.

Klíčová slova: diskrétní křivka, diskrétní plocha, diskrétní křivost, tok střední křivosti

Title: Discrete differential geometry and its applications

Author: Eva Vidličková

Department: Mathematical Institute of Charles University

Supervisor of the bachelor thesis: Doc. RNDr. Zbyněk Šír, Ph.D., Mathematical Institute of Charles University

Abstract: In this thesis we present an elementary introduction to the Discrete differential geometry. We will work with both discrete curves and discrete surfaces. Firstly some basic definitions and theorems from classic Differential geometry will be mentioned and then we will translate these concepts to the discrete setting, so that some important global structures are still preserved. At the end we implement mean curvature flow defined on discrete surfaces and run it on two meshes, that show its area-minimizing feature. This can be used for denoising the discrete surfaces.

Keywords: discrete curve, discrete surface, discrete curvature, mean curvature flow

Contents

Introduction

Differential geometry, as a mathematical discipline that combines differential calculus, integral calculus, linear algebra and geometry, is one of the most beautiful fields of mathematics. To make this discipline accessible for computers and practical use in general, the discrete differential geometry was introduced.

This thesis is an elementary introduction to this field, in which some concepts and expressions are translated from the smooth into the discrete setting. Firstly we deal with the theory of curves (Chapter 1), then the theory of surfaces (Chapter 2). Both chapters are divided into two sections: Smooth setting and Discrete setting. In the Smooth setting we present some definitions, theorems and structures, all set in the classic Differential geometry. In the Discrete setting we look at some methods of defining discrete analogues of these constructs. While discretizing we try to satisfy the convergence condition, which says that in the limit of a refinement sequence, discrete measures of new expressions agree with continuous measures. We also want the discrete expressions to preserve some basic differential-geometric structures (in particular The Turning number theorem for curves and the Gauss-Bonnet theorem for surfaces).

Since whole discrete differential geometry was brought up to life for its practical usage, we present one application in the last chapter. Mean curvature flow is a well-known concept in the differential geometry. One of its properties is the area-minimizing feature. In this thesis we present the discrete version of mean curvature flow and show its area-minimization feature, which is used for the denoising ("smoothing") of some discrete surfaces. We implement the discrete mean curvature flow in MATLAB and run it on two examples: a "noisy" torus and a cube. As a result we have a smoothen torus and a discrete surface similar to a sphere.

My contribution consists above all in the combination of different sources into one coherent text. As a source for the smooth-setting definitions and theorems we take Souček [2012], for the concepts presented in the discrete setting we take Kazhdan [2009]. And the basis for the chapter Mean curvature flow was found in Crane [2013]. Since the Discrete differential geometry as a mathematical discipline is still quite new, many scientists take different approaches and the definitions of some expressions are often not the same. Therefore the preservation of consistancy was the most difficult part. I sharpened the proofs of Proposition 1.3 and Theorem 1.5. I implemented the discrete mean curvature flow in MATLAB and ran it on two discrete surfaces.

Chapter 1

Curves

This chapter is devoted to the theory of curves. We will start by formulating some elementary (but essential) definitions and theorems from the standard Differential Geometry. Then we will try to define their analogue versions in the discrete setting while preserving some important features, in particular some global theorems.

1.1 Smooth Setting

In this section we will describe the standard Differential Geometry (smooth setting). Let Pressley [2010] and Souček [2012] be our source of information. We name basic definitions and theorems at first and then present some ideas and relations as a basis for the discrete setting relations.

Definition 1.1. *Let $I = (a, b)$ be an open interval of the real line $\mathbb{R}$.*
A regular parametrized curve *is a smooth map*

$$\boldsymbol{\alpha} : I \to \mathbb{R}^3,$$
$$\text{where } \forall t \in I : \quad \boldsymbol{\alpha}'(t) \neq 0.$$

The tangent vector *of $\boldsymbol{\alpha}$ at its point $\boldsymbol{\alpha}(t)$ is defined as:*

$$\boldsymbol{\alpha}'(t) \equiv \frac{\mathrm{d}\boldsymbol{\alpha}}{\mathrm{d}t}(t) \subset \mathbb{R}^3$$

and the unit tangent vector *of $\boldsymbol{\alpha}$ is the vector:*

$$\boldsymbol{u}(t) := \frac{\boldsymbol{\alpha}'(t)}{|\boldsymbol{\alpha}'(t)|}.$$

The tangent line *of $\boldsymbol{\alpha}$ at its point $\boldsymbol{\alpha}(t)$ is the line that passes through $\boldsymbol{\alpha}(t)$ and is parallel to the tangent vector at $\boldsymbol{\alpha}(t)$.*

Definition 1.2. *Let $\boldsymbol{\alpha}(t)$, $t \in I \subset \mathbb{R}$ be a regular parametrized curve, $\phi : \tilde{I} \to I$ be a diffeomorphism of a real interval. The* reparametrization *of $\boldsymbol{\alpha}$ is the regular parametrized curve $\tilde{\boldsymbol{\alpha}}(\tilde{t}) := \boldsymbol{\alpha}(\phi(\tilde{t}))$.*
Reparametrizations determine an equivalence relation on the set of all regular parametrized curves. A class of this equivalence will be called a curve.

Note. Obviously, the range of all reparametrizations is the same. By notation $\langle \boldsymbol{\alpha} \rangle$ we will understand the range of $\boldsymbol{\alpha}$.

Note. Since

$$\left|\frac{\mathrm{d}\tilde{\boldsymbol{\alpha}}}{\mathrm{d}\tilde{t}}(\tilde{t})\right| = \left|\frac{\mathrm{d}\boldsymbol{\alpha}}{\mathrm{d}t}(\phi(\tilde{t}))\right| \left|\frac{\mathrm{d}\phi}{\mathrm{d}\tilde{t}}(\tilde{t})\right|,$$

we can see that the condition $\tilde{\boldsymbol{\alpha}}'(\tilde{t}) \neq 0$ is still satisfied.

Definition 1.3. *Let* $\boldsymbol{\alpha}(t),\ t \in I = (a, b)$ *be a parametrized curve.*
We say $\boldsymbol{\alpha}$ *is a* closed curve *if it satisfies:*

$$\begin{aligned}\boldsymbol{\alpha}(a) &= \boldsymbol{\alpha}(b),\\ \boldsymbol{\alpha}'(a) &= \boldsymbol{\alpha}'(b),\\ \boldsymbol{\alpha}''(a) &= \boldsymbol{\alpha}''(b).\end{aligned}$$

We say $\boldsymbol{\alpha}$ *is a* planar curve *if* $\langle \boldsymbol{\alpha} \rangle$ *lies in a plane* $T \subset \mathbb{R}^3$.

Definition 1.4. *Given a curve* $\boldsymbol{\alpha}(t),\ t \in (a, b)$ *and a differentiable real function* f *defined on* $\langle \boldsymbol{\alpha} \rangle$*, the integral of* f *over the curve* $\boldsymbol{\alpha}$ *is defined as:*

$$\int_{p \in \langle \boldsymbol{\alpha} \rangle} f(p)\,\mathrm{d}p = \int_a^b f(\boldsymbol{\alpha}(t))\,|\boldsymbol{\alpha}'(t)|\,\mathrm{d}t.$$

Definition 1.5. *The* length l *of a curve* $\boldsymbol{\alpha}(t),\ t \in (a, b)$ *is defined as:*

$$l = \int_{p \in \langle \boldsymbol{\alpha} \rangle} 1\,\mathrm{d}p = \int_a^b |\boldsymbol{\alpha}'(t)|\,\mathrm{d}t.$$

We can see that two points on a curve can be distinguished by how steeply the curve is turning in their neighbourhood (see Figure 1.1). This feature is called *curvature*.

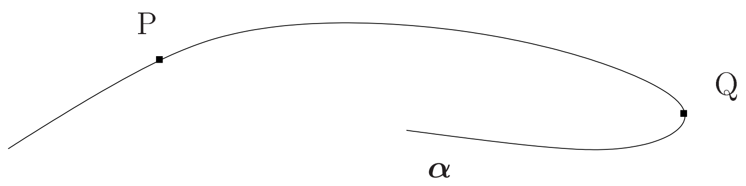


Figure 1.1: curve with two points with different curvature

Definition 1.6. *Given a curve* $\boldsymbol{\alpha}(t),\ t \in (a, b)$*, and a point* $t_1 \in I$*, we define the* curvature *of* $\boldsymbol{\alpha}$ *at* t_1 *as:*

$$\kappa(t_1) = \frac{|\boldsymbol{u}'(t_1)|}{|\boldsymbol{\alpha}'(t_1)|}.$$

Note. Notice that the length of a curve, the integral of a function over a curve and also the curvature do not change under reparametrization. It means that even though they are expressed by a concrete parametrization, they are properties of the curve.

Now we will describe two different ways of understanding the definition of curvature. Later we will use them to get the definition of curvature in the discrete setting.

Definition 1.7. *Let* $\boldsymbol{\alpha}(t)$, $t \in I = (a,b)$, *be a parametrized curve,* $\boldsymbol{\alpha}(t_1) = P$, $\kappa(t_1) \neq 0$, $\boldsymbol{\alpha}(t_2) = Q$, $\boldsymbol{\alpha}(t_3) = R$. *Consider a circle passing through* P,Q,R. *We define the* osculating circle *as a limiting position of the circle as* $t_2, t_3 \to t_1$.

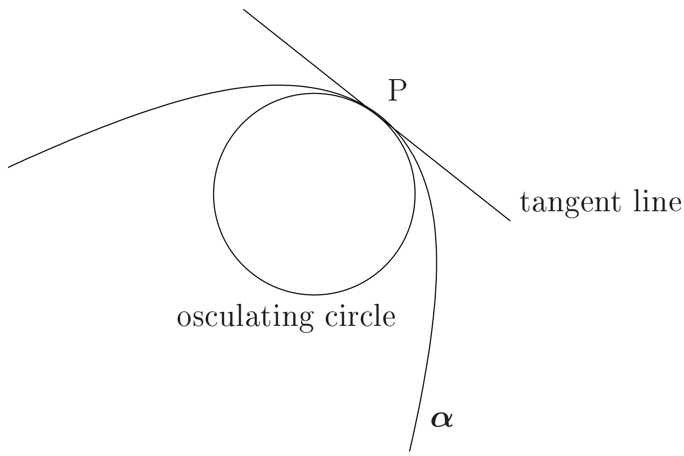


Figure 1.2: curve $\boldsymbol{\alpha}$ with an osculating circle and a tangent line at its point P

Curvature of the osculating circle is the same as the curvature of the curve at the corresponding point (Struik [1961, p. 14]). We know that the curvature of a circle with radius r is $\kappa_c = \frac{1}{r}$.

We will describe two different ways of expressing the radius of a circle using only local information.

1. The first one deals with the local change of tangent vectors. We will study how quickly unit tangent vectors are moving in the neighbourhood of a point. Looking at Figure 1.3,left, clearly:

$$r\,|\boldsymbol{u}_1 - \boldsymbol{u}_2| = |\boldsymbol{\alpha}(t_1) - \boldsymbol{\alpha}(t_2)|,$$

giving:

$$\frac{1}{r} = \frac{|\boldsymbol{u}_1 - \boldsymbol{u}_2|}{|\boldsymbol{\alpha}(t_1) - \boldsymbol{\alpha}(t_2)|}.$$

And the radius of an osculating circle, using only local information of the curve, can be computed as the limit:

$$\frac{1}{r} = \lim_{t_2 \to t_1} \left(\frac{|\boldsymbol{u}(t_2) - \boldsymbol{u}(t_1)|}{|\boldsymbol{\alpha}(t_2) - \boldsymbol{\alpha}(t_1)|}\right) = \frac{|\boldsymbol{u}'(t_1)|}{|\boldsymbol{\alpha}'(t_1)|}.$$

It brought us directly to the original definition of curvature (Definition 1.6):

$$\kappa(t_1) = \lim_{t_2 \to t_1} \left(\frac{|\boldsymbol{u}(t_2) - \boldsymbol{u}(t_1)|}{|\boldsymbol{\alpha}(t_2) - \boldsymbol{\alpha}(t_1)|}\right) = \frac{|\boldsymbol{u}'(t_1)|}{|\boldsymbol{\alpha}'(t_1)|}. \quad (1.1)$$

2. The second way deals with the change of angles of tangents.
   The radius of a circle can be expressed as:

$$\frac{1}{r} = \frac{\phi_{t_2}}{\int_{t_1}^{t_2} (|\boldsymbol{\alpha}'(s)|)\,\mathrm{d}s},$$

   where ϕ_{t_2} is the angle between the straight line connecting the center and $\boldsymbol{\alpha}(t_1)$ and the straight line connecting the center and $\boldsymbol{\alpha}(t_2)$ (see Figure 1.3,right). This angle is the same as the angle between the tangent vectors of $\boldsymbol{\alpha}$ at $\boldsymbol{\alpha}(t_1)$, $\boldsymbol{\alpha}(t_2)$.

   The radius of an osculating circle, using only local information of the curve can be computed as the limit:

$$\frac{1}{r} = \lim_{t_2 \to t_1} \left(\frac{\phi_{t_2}}{\int_{t_1}^{t_2} (|\boldsymbol{\alpha}'(s)|)\,\mathrm{d}s}\right).$$

   It brought us to a new definition of curvature:

$$\kappa(t_1) = \lim_{t_2 \to t_1} \left(\frac{\phi_{t_2}}{\int_{t_1}^{t_2} (|\boldsymbol{\alpha}'(s)|)\,\mathrm{d}s}\right), \tag{1.2}$$

   where ϕ_{t_2} is the angle described above.

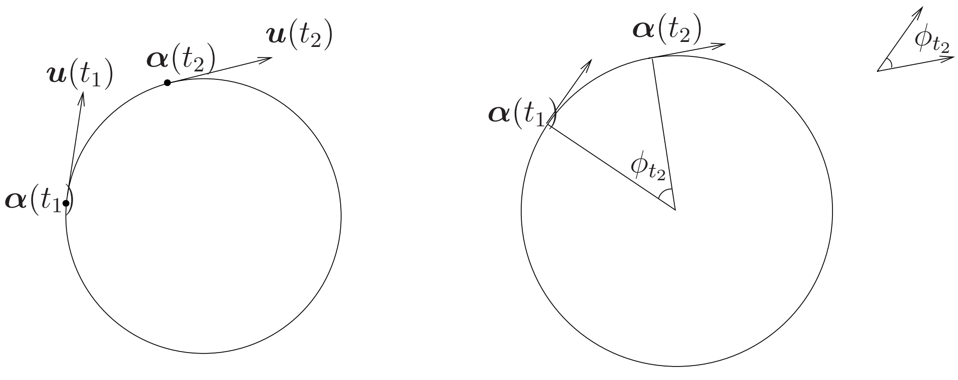


Figure 1.3: derivation of the definition of curvature

Proposition 1.1. *In the limit both expressions (1.1, 1.2) define the same curvature.*

Proof.
Looking at Figure 1.4,left we can see that:

$$|u(t_1) - u(t_2)| = 2\,sin(\frac{\phi_{t_2}}{2}) = \phi_{t_2} + \mathcal{O}(\phi_{t_2}^2),$$

which for the nominators of the expressions((1.1), (1.2)) implies:

$$\lim_{t_2 \to t_1} \frac{|u(t_1) - u(t_2)|}{\phi_{t_2}} = 1.$$

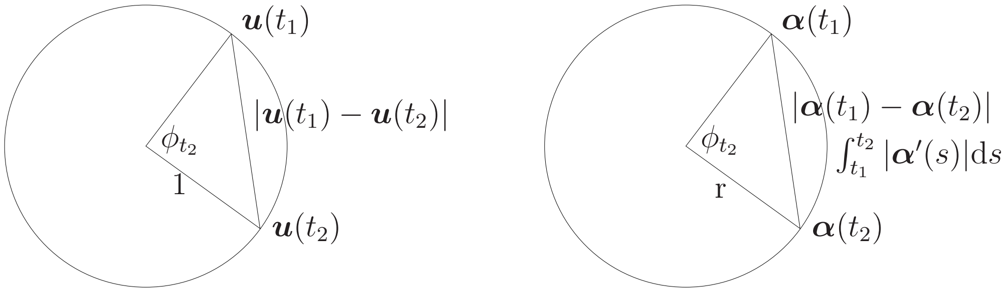


Figure 1.4: derivation of relation between the two definitions of curvature

Similarly for the denominators (see Figure 1.4,right):

$$\int_{t_1}^{t_2} |\boldsymbol{\alpha}'(s)|\mathrm{d}s = r\,\phi_{t_2},$$
$$|\boldsymbol{\alpha}(t_1) - \boldsymbol{\alpha}(t_2)| = r\,2\,sin(\frac{\phi_{t_2}}{2}) = r\,\phi_{t_2} + \mathcal{O}(\phi_{t_2}^2),$$

which means:

$$\lim_{t_2 \to t_1} \frac{\int_{t_1}^{t_2} |\boldsymbol{\alpha}'(s)|\mathrm{d}s}{|\boldsymbol{\alpha}(t_1) - \boldsymbol{\alpha}(t_2)|} = 1.$$

To finish this proof we show that in the limit the fraction of the expressions (1.1), (1.2) gives:

$$\lim_{t_2 \to t_1} \frac{\frac{|u(t_2)-u(t_1)|}{|\boldsymbol{\alpha}(t_2)-\boldsymbol{\alpha}(t_1)|}}{\frac{\phi_{t_2}}{\int_{t_1}^{t_2}(|\boldsymbol{\alpha}'(s)|)\,\mathrm{d}s}} = \lim_{t_2 \to t_1} \frac{|u(t_1) - u(t_2)|}{\phi_{t_2}} \cdot \lim_{t_2 \to t_1} \frac{\int_{t_1}^{t_2} |\boldsymbol{\alpha}'(s)|\mathrm{d}s}{|\boldsymbol{\alpha}(t_1) - \boldsymbol{\alpha}(t_2)|} = 1 \cdot 1 = 1.$$

❑

In a smooth setting we have two different definitions of curvature, which give us the same values for every point on a curve. They give rise to two different notions of curvature in the discrete setting, however, with different values.

Theorem 1.2 (Turning number theorem)**.** *Suppose $\boldsymbol{\alpha}(t)$, $t \in I = (a, b)$ is a closed planar curve. Then there exists an integer $l \in \mathbb{N}$ for which:*

$$\int_{p \in \boldsymbol{\alpha}} \kappa(p)\,\mathrm{d}p = 2\pi l.$$

Note. The integer l is called the *winding index* of curve and says how many times the curve turns around in counter-clockwise direction.

Proof. Detailed steps of this proof can be found in Shonkwiler [2005, p. 10]. ❑

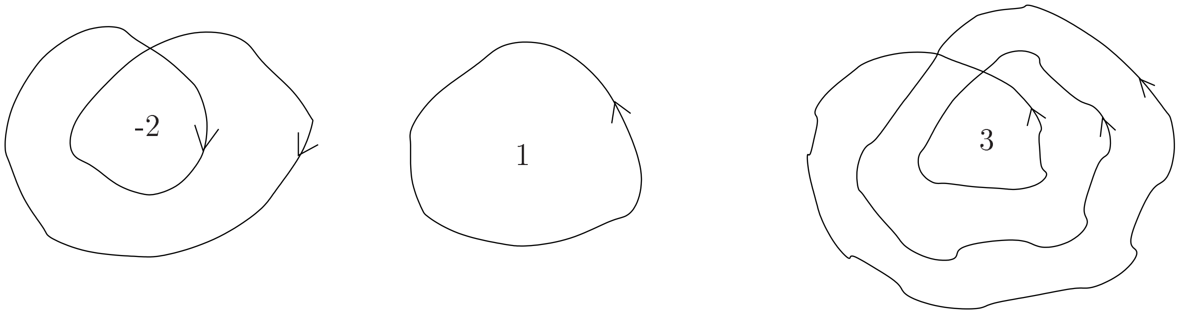


Figure 1.5: winding indices of some curves

1.2 Discrete Setting

In this section we will define analogue expressions for the discrete setting (in particular curve, tangent vectors, integral, length, curvature). We will follow the structure shown in Kazhdan [2009, Subject: Curves].

While applying differential geometric techniques in the discrete setting, we always have to keep two conditions in our mind:

- *Convergence*, which means that in the limit of a refinement sequence, discrete measures of new expressions agree with continuous measures.
- *Structure-preservation*, which means that for an arbitrary discrete curve, the discrete measure of new expressions obeys the important global theorems.

Definition 1.8. *A* discrete curve *is the union of line segments between an ordered set of vertices* $\{\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$*, such that* $\boldsymbol{v}_i \neq \boldsymbol{v}_{i+1}$ *for any index* $i \in \{0, \ldots, n-1\}$*. An* unit tangent *at any point on the edge* $\overline{\boldsymbol{v}_i \boldsymbol{v}_{i+1}}$ *is the unit vector pointing from* $\boldsymbol{v}_i$ *to* $\boldsymbol{v}_{i+1}$*:*

$$\boldsymbol{u}_i = \frac{\boldsymbol{v}_{i+1} - \boldsymbol{v}_i}{|\boldsymbol{v}_{i+1} - \boldsymbol{v}_i|}, \quad i \in 0, \ldots, n-1.$$

An example of a discrete curve is shown in Figure 1.6.

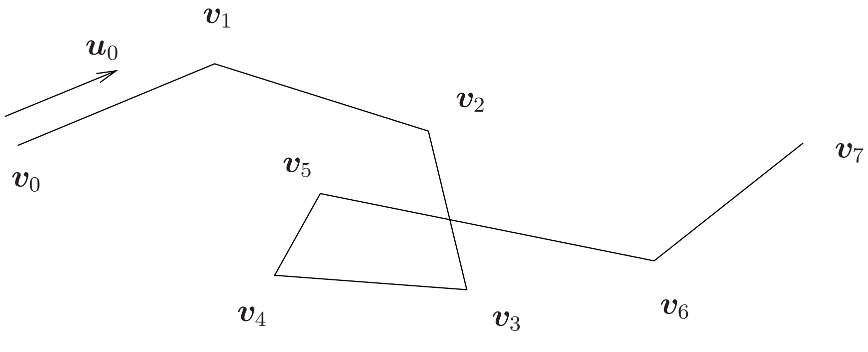


Figure 1.6: discrete curve

Definition 1.9. *We say a discrete curve is* closed*, if* $\boldsymbol{v}_0 = \boldsymbol{v}_n$*.*
We say a discrete curve is planar*, if*

$$\forall i \in \{0, \ldots, n-1\} : \overline{\boldsymbol{v}_i \boldsymbol{v}_{i+1}} \subset T \subset \mathbb{R}^2.$$

As for integrals of functions over discrete curves, we have to distinguish two types of functions: functions assigning values to edges and functions assigning values to vertices.

Definition 1.10. *Suppose we have a discrete curve.*
Given a function f assigning values f_i to edges $\overline{\boldsymbol{v}_i\boldsymbol{v}_{i+1}}$, we define the integral of f over the curve as:

$$F = \sum_{i=0}^{n-1} f_i |\boldsymbol{v}_{i+1} - \boldsymbol{v}_i|. \tag{1.3}$$

Given a function f assigning values f_i to vertices $\boldsymbol{v}_i$, we define the integral of f over the curve as:

$$F = \sum_{i=0}^{n} f_i \frac{|\boldsymbol{v}_{i+1} - \boldsymbol{v}_i| + |\boldsymbol{v}_i - \boldsymbol{v}_{i-1}|}{2}, \tag{1.4}$$

where $|\boldsymbol{v}_{n+1} - \boldsymbol{v}_n| = |\boldsymbol{v}_0 - \boldsymbol{v}_{-1}| = 0$.

According to the Definition 1.4 and Definition 1.5, the length of a smooth curve is obtained by integrating f = 1 over the curve. As for discrete curves, in terms of length we consider f as a function assigning values to edges. And we get:

Definition 1.11. *The* length *l of a discrete curve between vertices $\boldsymbol{v}_{i_1}$ and $\boldsymbol{v}_{i_2}$ is defined as:*

$$l = \sum_{j=i_1}^{i_2-1} 1\,|\boldsymbol{v}_{j+1} - \boldsymbol{v}_j| = \sum_{j=i_1}^{i_2-1} |\boldsymbol{v}_{j+1} - \boldsymbol{v}_j|,$$

which is the sum of the lengths of the line segments between them.

Both integrals (1.3, 1.4) conform the convergence condition:

Proposition 1.3. *In the limit of a refinement of an approximation of a smooth curve, discrete measure of integrals agree with continuous measure of integrals.*

Proof. Think of a discrete curve as an approximation of a smooth curve

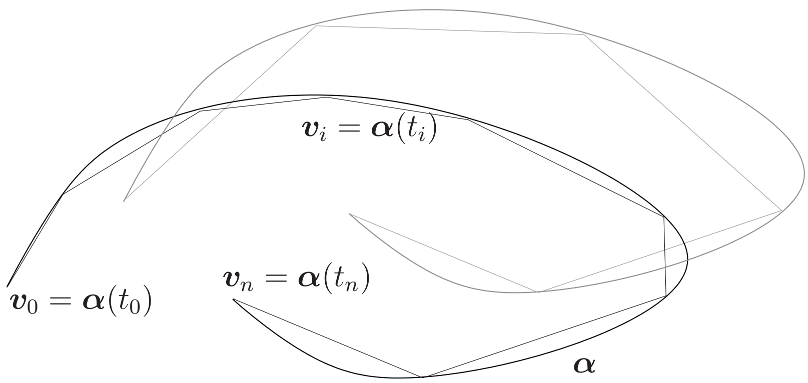


Figure 1.7: refinement of an approximation of a smooth curve

$\boldsymbol{\alpha} : (t_0, t_n) \to \mathbb{R}^3$ (see Figure 1.7):

$$\boldsymbol{\alpha}(t_i) = \boldsymbol{v}_i \quad i \in \{0, \ldots, n\}.$$

WOLOG:
$$\forall i \in \{0,\ldots,n-1\} \quad |t_{i+1}-t_i| = \frac{|t_n - t_0|}{n}.$$
We can express the integral of f over $\boldsymbol{\alpha}$ as:
$$\int_{p\in\langle\boldsymbol{\alpha}\rangle} f(p)\mathrm{d}p = \int_{t_0}^{t_1} f(\boldsymbol{\alpha}(t))\,|\boldsymbol{\alpha}'(t)|\mathrm{d}t \tag{1.5}$$
$$= \lim_{n\to\infty}\sum_{i=0}^{n-1} f(\boldsymbol{\alpha}(t_i))\,|\boldsymbol{\alpha}'(t_i)|\,\frac{|t_n-t_0|}{n}. \tag{1.6}$$
The expression $\lim_{n\to\infty}$ represents the limit of a refinement.
Let's think of f as a function assigning values f_i to edges $\overline{\boldsymbol{\alpha}(t_i)\boldsymbol{\alpha}(t_{i+1})}$. Then:
$$f(\boldsymbol{\alpha}(t_i)) = f_i,$$
$$|\boldsymbol{\alpha}'(t_i)| \approx \frac{|\boldsymbol{\alpha}(t_{i+1})-\boldsymbol{\alpha}(t_i)|}{|t_{i+1}-t_i|}.$$
Substituting to (1.5) we get:
$$\int_{p\in\langle\boldsymbol{\alpha}\rangle} f(p)\mathrm{d}p = \lim_{n\to\infty}\sum_{i=0}^{n-1} f_i\,|\boldsymbol{\alpha}(t_{i+1})-\boldsymbol{\alpha}(t_i)| = \lim_{n\to\infty}\sum_{i=0}^{n-1} f_i\,|\boldsymbol{v}_{i+1}-\boldsymbol{v}_i|.$$
Let's think of f as a function assigning values f_i to vertices $\boldsymbol{\alpha}(t_i)$. Then:
$$\begin{aligned} f(\boldsymbol{\alpha}(t_i)) &= f_i,\\ |\boldsymbol{\alpha}'(t_i)| &\approx \frac{|\boldsymbol{\alpha}(t_{i+1})-\boldsymbol{\alpha}(t_i)|}{|t_{i+1}-t_i|},\\ |\boldsymbol{\alpha}'(t_i)| &\approx \frac{|\boldsymbol{\alpha}(t_i)-\boldsymbol{\alpha}(t_{i-1})|}{|t_i-t_{i-1}|}. \end{aligned} \tag{1.7}$$
As an approximation of $|\boldsymbol{\alpha}'(t_i)|$ we take the average of two possible approximations displayed above (1.7):
$$|\boldsymbol{\alpha}'(t_i)| \approx \frac{1}{2}\,\frac{|\boldsymbol{\alpha}(t_{i+1})-\boldsymbol{\alpha}(t_i)| + |\boldsymbol{\alpha}(t_i)-\boldsymbol{\alpha}(t_{i-1})|}{\frac{|t_n-t_0|}{n}}.$$
Substituting to (1.5) we get:
$$\int_{p\in\langle\boldsymbol{\alpha}\rangle} f(p)\mathrm{d}p = \lim_{n\to\infty}\sum_{i=0}^{n-1} f_i\,\frac{|\boldsymbol{\alpha}(t_{i+1})-\boldsymbol{\alpha}(t_i)| + |\boldsymbol{\alpha}(t_i)-\boldsymbol{\alpha}(t_{i-1})|}{2}$$
$$= \lim_{n\to\infty}\sum_{i=0}^{n-1} f_i\,\frac{|\boldsymbol{v}_{i+1}-\boldsymbol{v}_i| + |\boldsymbol{v}_i-\boldsymbol{v}_{i-1}|}{2}.$$

❑

Now we need to define the curvature. Since while defining the curvature we look at the change of tangent vectors or angles between tangent vectors, we think of curvature as a function assigning values to vertices. As we mentioned above, there are two different ways:

1. The first way of defining the curvature, shown in the previous section (1.1) was the change in tangent vectors as we move through the vertex. We cannot use the same limit process as before, since a discrete curve is not smooth at its vertices:

$$\begin{aligned}\kappa_i &= \lim_{h\to 0}\Big(\frac{|\boldsymbol{u}_i - \boldsymbol{u}_{i-1}|}{|(1-h)\boldsymbol{v}_i + h\boldsymbol{v}_{i+1}| - |(1-h)\boldsymbol{v}_i + h\boldsymbol{v}_{i-1}|}\Big)\\ &= \lim_{h\to 0}\Big(\frac{|\boldsymbol{u}_i - \boldsymbol{u}_{i-1}|}{h\,|\boldsymbol{v}_{i+1} - \boldsymbol{v}_i|}\Big) = \infty.\end{aligned}$$

   But we can define the curvature as the change in tangent vectors divided by the distance of adjacent edge centres (see Figure 1.8):

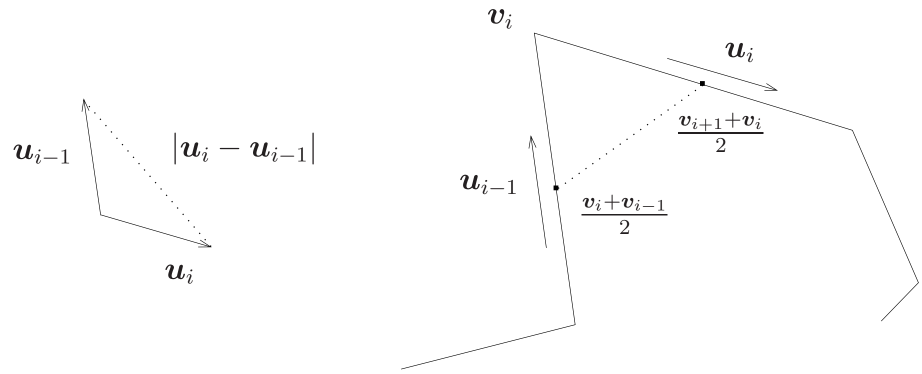


Figure 1.8: derivation of the first definition of discrete curvature

$$\kappa_i = \frac{|\boldsymbol{u}_{i+1} - \boldsymbol{u}_i|}{|[(\boldsymbol{v}_i + \boldsymbol{v}_{i+1})/2] - [(\boldsymbol{v}_{i-1} + \boldsymbol{v}_i)/2]|}. \qquad (1.8)$$

2. The second way of defining the curvature, also shown in the previous section (1.2) was the limit of the angle between tangent vectors of two points divided by their arclength as one approaches another. Again, we cannot directly use the limit process, but we can define it as the angle between tangent vectors divided by the length of the arc between adjacent edge centres (see Figure 1.9):

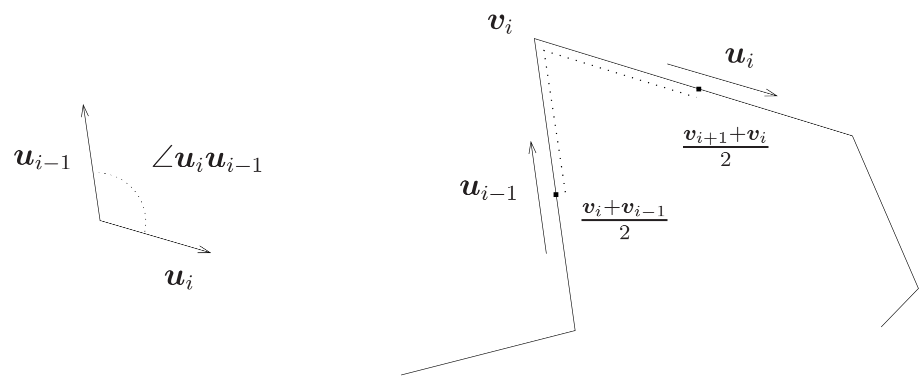


Figure 1.9: derivation of the second definition of discrete curvature

$$\kappa_i = \frac{\angle \boldsymbol{u}_i \boldsymbol{u}_{i-1}}{|(\boldsymbol{v}_{i+1} - \boldsymbol{v}_i)/2| + |(\boldsymbol{v}_i - \boldsymbol{v}_{i-1})/2|}. \tag{1.9}$$

Proposition 1.4. *Both of these definitions (1.8, 1.9) conform the convergence condition.*

Proof. Source: Kazhdan [2009, Subject: Curves]. ❑

Note. In the limit of a refinement sequence, both of the definitions of curvature come with the same value. However, in the discrete setting their values are not the same. So which one to choose? We will check, which one conforms the discrete form of Turning number theorem (Theorem 1.2):

Using the first definition we get:

$$\begin{aligned}\int_{p\in\boldsymbol{\alpha}} \kappa(p)\,\mathrm{d}p &= \sum_{i=0}^{n} \kappa_i \frac{|\boldsymbol{v}_{i+1} - \boldsymbol{v}_i| + |\boldsymbol{v}_i - \boldsymbol{v}_{i-1}|}{2} \\ &= \sum_{i=0}^{n} \frac{|\boldsymbol{u}_{i+1} - \boldsymbol{u}_i|}{|[(\boldsymbol{v}_i + \boldsymbol{v}_{i+1})/2] - [(\boldsymbol{v}_{i-1} + \boldsymbol{v}_i)/2]|} \frac{|\boldsymbol{v}_{i+1} - \boldsymbol{v}_i| + |\boldsymbol{v}_i - \boldsymbol{v}_{i-1}|}{2},\end{aligned}$$

which would very hardly come out as an integer multiple of 2π.
Using the second definition we get:

$$\begin{aligned}\int_{p\in\boldsymbol{\alpha}} \kappa(p)\,\mathrm{d}p &= \sum_{i=0}^{n} \kappa_i \frac{|\boldsymbol{v}_{i+1} - \boldsymbol{v}_i| + |\boldsymbol{v}_i - \boldsymbol{v}_{i-1}|}{2} \\ &= \sum_{i=0}^{n} \frac{\angle \boldsymbol{u}_i \boldsymbol{u}_{i+1}}{|(\boldsymbol{v}_{i+1} - \boldsymbol{v}_i)/2| + |(\boldsymbol{v}_i + \boldsymbol{v}_{i-1})/2|} \frac{|\boldsymbol{v}_{i+1} - \boldsymbol{v}_i| + |\boldsymbol{v}_i - \boldsymbol{v}_{i-1}|}{2} \qquad (1.10) \\ &= \sum_{i=0}^{n} \angle \boldsymbol{u}_i \boldsymbol{u}_{i+1},\end{aligned}$$

which is the sum of exterior angles over the vertices of a closed discrete curve. As for planar closed discrete curve it is always an integer multiple of 2π.

Definition 1.12. *We define the* curvature *of a discrete curve at the vertex* $\boldsymbol{v}_i$ *as:*

$$\kappa_i = \frac{\angle \boldsymbol{u}_i \boldsymbol{u}_{i-1}}{|(\boldsymbol{v}_{i+1} - \boldsymbol{v}_i)/2| + |(\boldsymbol{v}_i + \boldsymbol{v}_{i-1})/2|}.$$

Theorem 1.5 (Turning number theorem (discrete version))**.** *For every discrete planar closed curve there exists an integer* $l \in \mathbb{N}$ *for which:*

$$\sum_{j=2}^{n-1} \kappa_i \frac{|\boldsymbol{v}_{i+1} - \boldsymbol{v}_i| + |\boldsymbol{v}_i - \boldsymbol{v}_{i-1}|}{2} = 2\pi l.$$

Proof. As shown above (1.10):

$$\sum_{j=2}^{n-1} \kappa_i \frac{|\boldsymbol{v}_{i+1} - \boldsymbol{v}_i| + |\boldsymbol{v}_i - \boldsymbol{v}_{i-1}|}{2} = \sum_{j=2}^{n-1} \angle \boldsymbol{u}_i \boldsymbol{u}_{i+1},$$

which is the sum of exterior angles of a planar closed discrete curve. Every planar closed discrete curve C can be split into a set of closed simple planar discrete curves $\{C_i\}_{i=1}^{n}$, which are basically closed polygons.
Let $\boldsymbol{\alpha}_i$, $i = 1, \dots, n$ be the exterior angles of the split part, just like in the Figure 1.10.

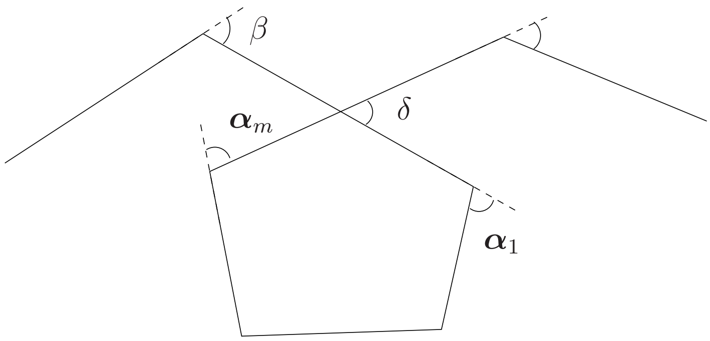


Figure 1.10: splitting general discrete curve into into a set of simple discrete curves

Since

$$\dots + \beta + \sum_{i=1}^{m} (\boldsymbol{\alpha}_i) + \dots = (\dots + \beta + (-\delta) + \dots) + (\sum_{i=1}^{m} (\boldsymbol{\alpha}_i) + \delta),$$

we can see that:

$$\sum_{i|\boldsymbol{\alpha}_i \in \text{Ext. a. of C}} \boldsymbol{\alpha}_i = \sum_{i=1}^{n} \sum_{j|\boldsymbol{\alpha}_j \in \text{Ext. a. of } C_i} \boldsymbol{\alpha}_j.$$

It means that the sum of all exterior angles of a planar closed discrete surface C is the sum of exterior angles of closed polygons C_i totalized together. We know that the sum of exterior angles of a closed polygon is $\pm 2\pi$. Therefore we get that the sum of all exterior angles of a closed planar discrete curve is an integer multiple of 2π.

❑

In the end we managed to define the fundamental expressions connected with curves in a discrete setting. All of them conform the convergence condition and when deciding between two options, we chose the one that preserved some global structures, in particular the Turning number theorem.

Chapter 2

Surfaces

As in the Chapter 1, we will start by formulating some important definitions and theorems concerning smooth surfaces. Then we will try to translate these ideas into the language of discrete setting.

2.1 Smooth Setting

Similarly to the smooth section in Chapter 1, we will formulate some basic definitions and theorems and then show some features that will be preserved while discretizing. As a source we take Souček [2012] compiling it with some ideas interpreted in Kazhdan [2009].

Definition 2.1. *Let* Ω *be an open subset of* $\mathbb{R}^2$*.*
A regular parametrized surface *in* $\mathbb{R}^3$ *is a smooth map*

$$\boldsymbol{\varphi} : \Omega \to \mathbb{R}^3, \text{ s.t.}$$

- $\forall (u, v) \in \Omega$ *the rank of the Jacobi matrix of* $\boldsymbol{\varphi}$ *at* $\boldsymbol{\varphi}(u, v)$ *is 2,*
- $\boldsymbol{\varphi}$ *is a homeomorphism of* Ω *onto* $S = \boldsymbol{\varphi}(\Omega)$*.*

Note. The condition that the rank of the Jacobi matrix of $\boldsymbol{\varphi}$ is 2 ensures us that at every point vectors:

$$\boldsymbol{\varphi}_u = (\frac{\partial \boldsymbol{\varphi}_1}{\partial u}, \frac{\partial \boldsymbol{\varphi}_2}{\partial u}, \frac{\partial \boldsymbol{\varphi}_3}{\partial u}),$$
$$\boldsymbol{\varphi}_v = (\frac{\partial \boldsymbol{\varphi}_1}{\partial v}, \frac{\partial \boldsymbol{\varphi}_2}{\partial v}, \frac{\partial \boldsymbol{\varphi}_3}{\partial v})$$

are linearly independent - they determine a plane in $\mathbb{R}^3$.

Definition 2.2. *Let* $\boldsymbol{\varphi}(u, v),\ (u, v) \in \Omega \subset \mathbb{R}^2$ *be a regular parametrized surface,* $\phi : \tilde{\Omega} \to \Omega$ *be a diffeomorphism of open subsets of* $\mathbb{R}^2$*. Then* $\tilde{\boldsymbol{\varphi}} := \boldsymbol{\varphi} \circ \phi$ *is a* reparametrization *of* $\boldsymbol{\varphi}$*.*

Theorem 2.1. *Let* $\boldsymbol{\varphi}$ *be a regular parametrized surface and* ϕ *a diffeomorphism then* $\tilde{\boldsymbol{\varphi}} := \boldsymbol{\varphi} \circ \phi$ *is a regular parametrized surface.*
Let $\boldsymbol{\varphi}$ *and* $\tilde{\boldsymbol{\varphi}}$ *be regular parametrized surfaces and their ranges are the same. Then there exists a diffeomorphism* ϕ *such that* $\tilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi} \circ \phi$*.*

Note. Under the term *transition map* we will understand the map ϕ.

Proof. This proof can be found in the textbook Souček [2012, p. 26]. ❑

Definition 2.3. *We say that $S \subset \mathbb{R}^3$ is a (smooth) surface, if for every point $p \in S$ there exists a neighbourhood $U \subset \mathbb{R}^3$ and a regular parametrized surface $\boldsymbol{\varphi} : \Omega \to S$, such that $V \cap S = \boldsymbol{\varphi}(\Omega)$.*

Definition 2.4. *Let $S \subset \mathbb{R}^3$ be a surface.*
We say that a vector $\boldsymbol{v}$ is a tangent vector *of S at a point $\boldsymbol{p}$ if: $\exists\, \boldsymbol{\alpha}(t)$ -curve, $\boldsymbol{\alpha}(t) \subset S\ \forall t,\ \boldsymbol{\alpha}(t_0) = \boldsymbol{p},\ \boldsymbol{\alpha}'(t_0) = \boldsymbol{v}$. The set of all tangent vectors at a point $\boldsymbol{p}$ is called the* tangent space *of S at a point $\boldsymbol{p}$.*

Note. If $\boldsymbol{\varphi}$ is a regular parametrized surface onto S then the tangent plane at $\boldsymbol{p} \in S,\ \boldsymbol{p} = \boldsymbol{\varphi}(u_0, v_0)$ is determined by vectors $\boldsymbol{\varphi}_u(u_0, v_0), \boldsymbol{\varphi}_v(u_0, v_0)$. The steps of this proof can be found in the textbook Souček [2012, p. 29].

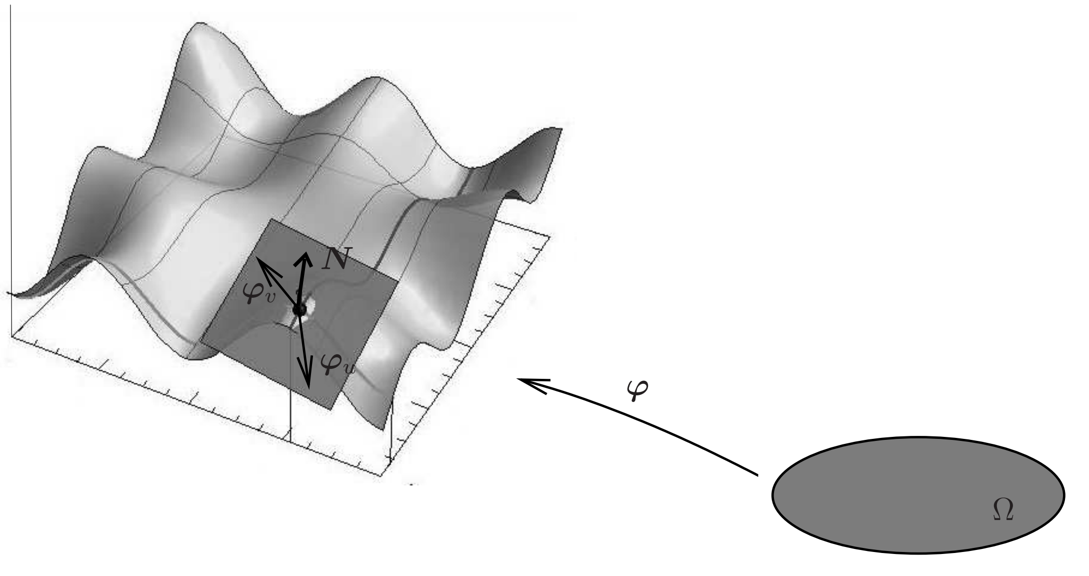


Figure 2.1: parametrized regular surface with a tangent space and a normal vector $\boldsymbol{N}$

Definition 2.5. *Let S be a surface. Then for every $\boldsymbol{p} \in S$ there exists a unit vector $\boldsymbol{N}$, s.t. $\boldsymbol{N}$ is perpendicular to the tangent space at $\boldsymbol{p}$. This vector is determined uniquely up to the sign.*
Let $\boldsymbol{\varphi}$ be a regular parametrized surface onto S. Then we define the normal vector *of S at $\boldsymbol{p} = \boldsymbol{\varphi}(u_0, v_0)$ as:*

$$\boldsymbol{N} = \frac{\boldsymbol{\varphi}_u \times \boldsymbol{\varphi}_v}{|\boldsymbol{\varphi}_u \times \boldsymbol{\varphi}_v|}\,(u_0, v_0).$$

We define the normal line *of S at a point $\boldsymbol{p} \in S$ as a line going through $\boldsymbol{p}$ and perpendicular to the tangent plane of $\boldsymbol{p}$.*

Note. The normal line is determined uniquely.
If $\boldsymbol{\varphi}$ and $\tilde{\boldsymbol{\varphi}}$ are two regular parametrized surfaces, both onto S, ϕ is their transition map, then they determine the same normal vector $\Leftrightarrow$ det $J_\phi > 0$. Such mappings are called *equally oriented.*

Definition 2.6. *Suppose f is a continuous function on a surface S, parametrized by a map $\boldsymbol{\varphi} : \Omega \to S$.*
The integral *of f over S is defined as:*

$$\int_S f(s)\,\mathrm{d}s := \int_\Omega f(\boldsymbol{\varphi}(u,v))\,|\boldsymbol{\varphi}_u(u,v) \times \boldsymbol{\varphi}_v(u,v)|\,\mathrm{d}u\,\mathrm{d}v.$$

Definition 2.7. *Given a surface S, parametrized by a map $\boldsymbol{\varphi} : \Omega \to S$, and an open set $\eta \subset \Omega$, the* area of $\boldsymbol{\varphi}(\eta)$ *is defined as:*

$$\int_\eta 1\,\mathrm{d}s = \int_{\boldsymbol{\varphi}^{-1}(\eta)} |\boldsymbol{\varphi}_u(u,v) \times \boldsymbol{\varphi}_v(u,v)|\,\mathrm{d}u\,\mathrm{d}v.$$

Note. An important fact is that all tangent plane, normal line and integrals are independent on reparametrization. They are features of the surface, not the mapping.
The proof of this statement and some other interesting facts can be found in Souček [2012].

Now we will study the curvature. It is a local information, it will be defined at every point. We know how to measure the curvature of a curve. But on surface there is an infinite number of curves going through a certain point.

Definition 2.8. *Let $S \subset \mathbb{R}^3$ be a surface, $\boldsymbol{p} \in S$, $\boldsymbol{N}$ be the normal vector and $\boldsymbol{w}$ be a tangent vector at $\boldsymbol{p}$.*
We define the normal curvature $\kappa_n(\boldsymbol{w})$ *of S at $\boldsymbol{p}$ in the direction of $\boldsymbol{w}$ as the curvature of the curve that is formed as an intersection of the surface S and the plane containing both $\boldsymbol{N}$ and $\boldsymbol{w}$.*

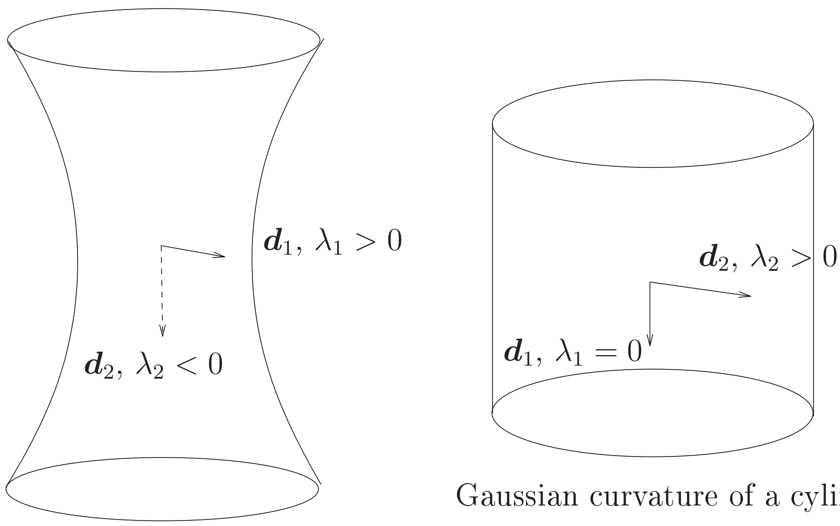
Note. $\kappa_n(\alpha\boldsymbol{w})$ remains the same for any $\alpha \in R$, $\alpha \neq 0$, therefore we say it is a curvature in the direction of $\boldsymbol{w}$.

Definition 2.9. *We define the* principal curvatures λ_1, λ_2 *as the minimum and the maximum of the normal curvature.*
We define the principal directions *as the corresponding directions of the principal curvatures.*

Note. The principal directions and curvatures show us in what direction and what is the value of the highest and the lowest bending at a concrete point $\boldsymbol{p} \in S$. The principal directions are always perpendicular to each other. The proof of this statement can be found in the textbook Souček [2012].

Definition 2.10. *We define the* Gaussian curvature *as the product of the principal curvatures:*

$$K = \lambda_1 \cdot \lambda_2.$$



Gaussian curvature of a cylinder = 0

Gaussian curvature of a hyperboloid < 0

Figure 2.2: principal directions $\boldsymbol{d}_1$, $\boldsymbol{d}_2$, principal curvatures λ_1, λ_2

We define the mean curvature *as the average of the principal curvatures:*

$$H = \frac{1}{2}(\lambda_1 + \lambda_2).$$

In the following we will present a relationship that will be crucial for deriving the discrete mean and Gaussian curvature.

Definition 2.11. *Let S be a surface, parametrized by a map $\boldsymbol{\varphi} : \Omega \to S$, $\boldsymbol{N}(u, v)$ be the normal vector of S at $\boldsymbol{\varphi}(u, v)$. A parametrized surface, offset from S by the distance of ε is defined as:*

$$\hat{\boldsymbol{\varphi}}(u, v) = \boldsymbol{\varphi}(u, v) + \varepsilon\, \boldsymbol{N}(u, v).$$

Theorem 2.2. *Let S be a smooth surface. Consider offsetting S by the distance of ε, s.t. $-\frac{1}{\varepsilon} < \lambda_1, \lambda_2$ (the principal curvatures). Measuring the area of the offset surface we get:*

$$A_\varepsilon = A + \varepsilon\, 2\,\mathcal{H} + \varepsilon^2\, \mathcal{K},$$

where A is the area of the original regular surface, A_ε is the area of the offset surface, $\mathcal{H}$ is the mean curvature integrated over S and $\mathcal{K}$ is the Gaussian curvature integrated over S.

Proof. Let $\boldsymbol{\varphi}(u, v)$ be the regular parametrization of S, $\boldsymbol{N}(u, v)$ be the normal vector of S and $\hat{\boldsymbol{\varphi}}(u, v)$ be an offset parametrized surface defined in Definition 2.11.

Now let $\hat{\boldsymbol{N}}(u, v)$ be the normal vector of $\hat{\boldsymbol{\varphi}}(u, v)$. The relation between normal vectors $\boldsymbol{N}(u, v)$ and $\hat{\boldsymbol{N}}(u, v)$ is given by Willmore [2012]:

$$||\hat{\boldsymbol{\varphi}}_u \times \hat{\boldsymbol{\varphi}}_v||\, \hat{\boldsymbol{N}} = ||\boldsymbol{\varphi}_u \times \boldsymbol{\varphi}_v||\, (1 + \varepsilon\, \lambda_1)(1 + \varepsilon\, \lambda_2)\, \boldsymbol{N}, \tag{2.1}$$

where λ_1 and λ_2 are the principal curvatures of $\boldsymbol{\varphi}(u, v)$. Using the Definition 2.10 we get:

$$||\hat{\boldsymbol{\varphi}}_u \times \hat{\boldsymbol{\varphi}}_v||\, \hat{\boldsymbol{N}} = ||\boldsymbol{\varphi}_u \times \boldsymbol{\varphi}_v||\, (1 + \varepsilon\, 2\, H + \varepsilon^2\, K)\, \boldsymbol{N}. \tag{2.2}$$

Taking a norm of (2.1) we get:

$$||\hat{\boldsymbol{\varphi}}_u \times \hat{\boldsymbol{\varphi}}_v|| = ||\boldsymbol{\varphi}_u \times \boldsymbol{\varphi}_v||\, |(1+\varepsilon\,\lambda_1)(1+\varepsilon\,\lambda_2)|.$$

Substituting to (2.1) we obtain:

$$\hat{\boldsymbol{N}} = \frac{(1+\varepsilon\,\lambda_1)(1+\varepsilon\,\lambda_2)}{|(1+\varepsilon\,\lambda_1)(1+\varepsilon\,\lambda_2)|}\,\boldsymbol{N}.$$

Since we suppose that $\lambda_1 > \lambda_2 > -\frac{1}{\varepsilon}$, we get:

$$\hat{\boldsymbol{N}} = \boldsymbol{N}.$$

Therefore, using (2.2) we obtain:

$$||\hat{\boldsymbol{\varphi}}_u \times \hat{\boldsymbol{\varphi}}_v|| = ||\boldsymbol{\varphi}_u \times \boldsymbol{\varphi}_v||\,(1+\varepsilon\,2\,H + \varepsilon^2\,K).$$

Finally, integrating over the domain of $\boldsymbol{\varphi}(u, v)$ we get:

$$A_\varepsilon = A + \varepsilon\,2\,\mathcal{H} + \varepsilon^2\,\mathcal{K},$$

❑

Now we will state an important theorem concerning Gaussian curvature that should be preserved after discretizing.

Theorem 2.3 (Gauss-Bonnet theorem)**.** *Let S be a compact smooth surface. Integrating the Gaussian curvature over S we get:*

$$\int_S K(s)\,\mathrm{d}s = 2\pi\chi,$$

where $\chi \in \mathbb{Z}$ is the Euler characteristic *of S.*

Proof. This proof is described in the textbook Souček [2012, p. 79].

❑

This theorem gives us some really surprising information. If someone bends or deforms the surface S, the total integral of the Gaussian curvature remains the same, even though the curvature locally changes.

2.2 Discrete Setting

This section will be devoted to the discretizing process of smooth surfaces. We will define analogue versions of concepts shown in the previous section, while trying to satisfy the convergence condition and structure-preservation condition presented in Section 1.2. Our source will be Kazhdan [2009] together with Sullivan [2002], Meyer et al. [2002], Polthier and Rossman [2002].

Definition 2.12. *A discrete surface (triangular mesh) is a collection of three kinds of mesh elements: vertices, edges, convex triangles, all set in $\mathbb{R}^3$. The vertex is characterized by its coordinates (vector $\subset \mathbb{R}^3$). Edges are line segments that connect two vertices. Triangle is a part of plane that is bounded by three edges.*
This collection must satisfy two conditions:

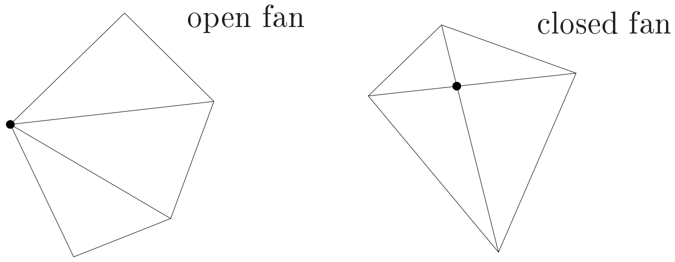


Figure 2.3: types of fans

- *each edge is incident to only one or two faces,*
- *the triangles incident to a vertex form a closed or an open fan.*

Note. More generally, instead of triangles one could consider polygons. Since every polygon can be split into triangles, we define it just as triangles, without loss of generality.
We will denote by V the set of vertices, by E the set of edges, by T the set of triangles.

Definition 2.13. *A* closed discrete surface *is a discrete surface, where always the triangles incident to a vertex form a closed fan. (See Figure 2.4,left)*

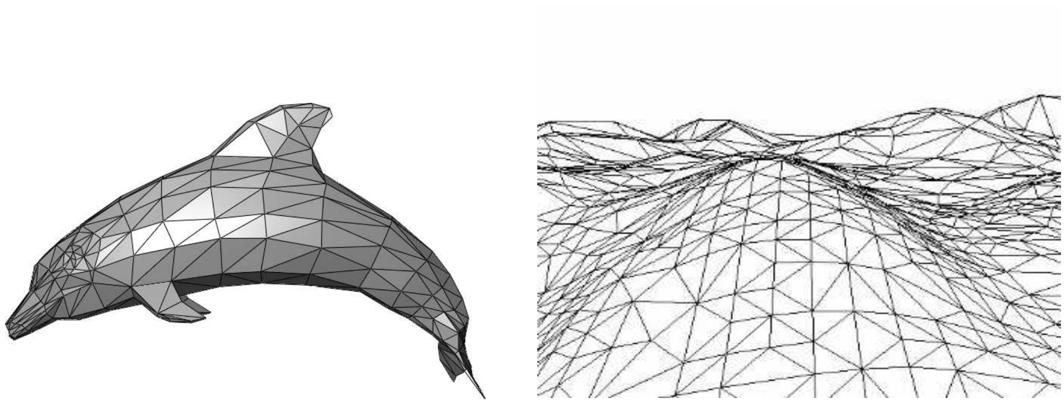


Figure 2.4: discrete surfaces (meshes)

Definition 2.14. *We define a* normal vector *of a triangle $t \in T$ as a unit vector perpendicular to the triangle t.*

Definition 2.15. Voronoi tessellation *is a tessellation of a discrete surface into a number of regions. Each triangle is divided into 3 regions, all meeting in the circumcenter, and separated by perpendicular bisectors.* The Voronoi region *of a vertex $\boldsymbol{v}$ is the union of all regions created by Voronoi tessellation that meet at the vertex $\boldsymbol{v}$. (See Figure 2.5)*

As for integrals of a function f over a discrete surface we have to differ three types of functions: functions defined on T, functions defined on E and functions defined on V.

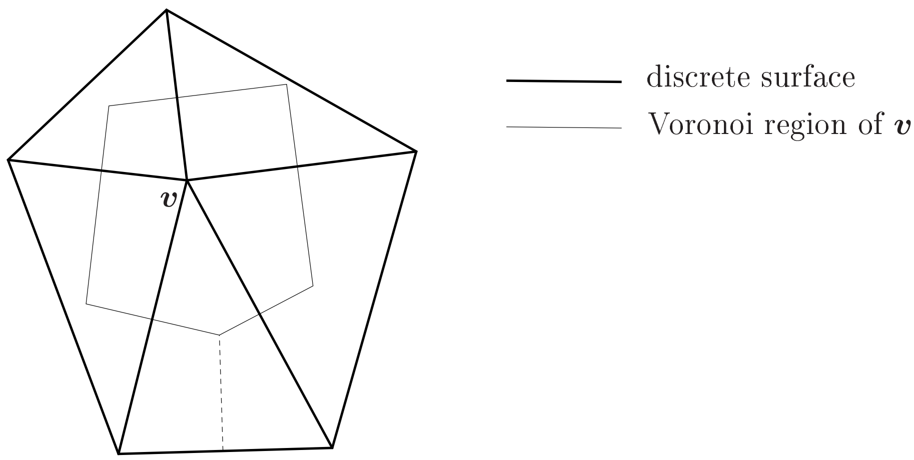


Figure 2.5: Voronoi region of a vertex $\boldsymbol{v}$

Definition 2.16. *Let S be a discrete surface.*
Given a function g assigning values g_t to triangles $t \in T$, we define the integral of g over the surface S as:

$$F = \sum_{t \in T} g_t \, A_t,$$

where A_t is the area of the triangle t.
Given a function g assigning values $g_{\boldsymbol{e}}$ to edges $\boldsymbol{e} \in E$, we define the total value of $g_{\boldsymbol{e}}$ as:

$$G_{\boldsymbol{e}} = g_{\boldsymbol{e}} \, |\boldsymbol{e}|,$$

where $|\boldsymbol{e}|$ is the length of $\boldsymbol{e}$, and the integral of g over the surface S as:

$$F = \sum_{\boldsymbol{e} \in E} G_{\boldsymbol{e}}.$$

Given a function g assigning values $g_{\boldsymbol{v}}$ to vertices $\boldsymbol{v} \in V$, we define the integral of g over the surface S as:

$$F = \sum_{\boldsymbol{v} \in V} g_{\boldsymbol{v}} \, A_{Voronoi_{\boldsymbol{v}}},$$

where $A_{Voronoi_{\boldsymbol{v}}}$ is the area of the Voronoi region of the vertex $\boldsymbol{v}$.

To define the curvature of a discrete surface, firstly we will have to look in the smooth case and then make an analogue for the discrete version.

In the following part we will try to express the curvature of a discrete surface. In the smooth case we defined normal and principal curvatures. Defining such curvature in the discrete setting would be ambiguous. Therefore we will try to define the discrete Gaussian and discrete mean curvature directly. We will follow the concept illustrated in the previous section.

Definition 2.17. *Let M be a convex discrete surface. The offset of M by the distance of ε is defined as the union of all points that are outside of the convex object created by M, distant from M of ε.*

Consider offsetting a convex discrete surface by a distance of ε. In order to compute the area of this object, we decompose it into three parts.

1. The triangles, whose area is exactly the same as the area of the original discrete surface.

2. The cylindrical arcs, corresponding to the edges of the original discrete surface. Since the height of each of these cylinders is the length of the edge they belong to $|\boldsymbol{e}|$ and the radius is ε, we compute the area of each cylindrical arc as:
$$S = \varepsilon\, \gamma_{\boldsymbol{e}}\, |\boldsymbol{e}|,$$
where $\gamma_{\boldsymbol{e}}$ is the angle between the normals of the two corresponding faces. It means that if t_1, t_2 are triangles forming the edge $\boldsymbol{e}$ and $\boldsymbol{N}_1, \boldsymbol{N}_2$ are their normals, $\gamma_{\boldsymbol{e}}$ satisfies:
$$\cos \gamma_{\boldsymbol{e}} = \langle \boldsymbol{N}_1, \boldsymbol{N}_2 \rangle$$

3. The spherical caps, belonging to the vertices of the original discrete surface. Since the radius of this sphere is ε, we compute the area of each spherical cap as:
$$A_s = \varepsilon^2\, \gamma_{\boldsymbol{v}},$$
where $\gamma_{\boldsymbol{v}}$ is the solid angle formed by the normals of the faces meeting at vertex $\boldsymbol{v}$.

Summing everything up we get that the area of the offset surface is:
$$A_\varepsilon = A + \varepsilon\, 2 \sum_{\boldsymbol{e} \in Edges} \frac{\gamma_{\boldsymbol{e}}}{2} |\boldsymbol{e}| + \varepsilon^2 \sum_{\boldsymbol{v} \in Vertices} \gamma_{\boldsymbol{v}}.$$

Therefore we can associate the mean curvature of $\boldsymbol{e}$ with: $\gamma_{\boldsymbol{e}}/2$ and the Gaussian curvature of $\boldsymbol{v}$ with the solid angle $\gamma_{\boldsymbol{v}}$ divided by the area of the Voronoi region of $\boldsymbol{v}$. Now we will try to answer the question of what exactly the solid angle $\gamma_{\boldsymbol{v}}$ is.

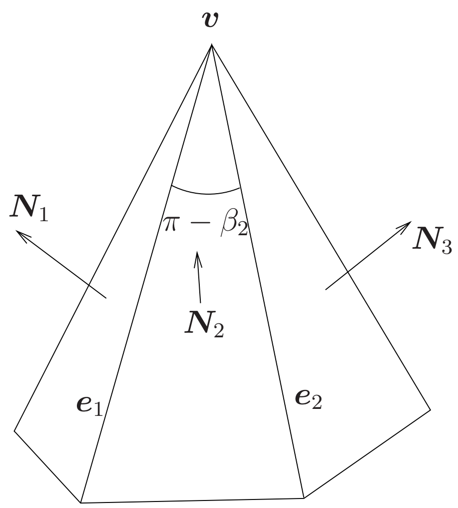


Figure 2.6: deriving the value of the solid angle $\gamma_{\boldsymbol{e}}/2$

Let $t_1, \ldots, t_n$ be the triangles (polygons) that meet at the vertex $\boldsymbol{v}$, $\boldsymbol{N}_i$ be the normal of the i-th triangle, $\boldsymbol{e}_i$ be the edge between t_i and t_{i+1} ($\boldsymbol{e}_n$ between t_n

and t_1) and α_i be the angle between $\boldsymbol{e}_{i-1}$ and $\boldsymbol{e}_i$, intersecting at $\boldsymbol{v}$, $i = 1, \ldots, n$ (see Figure 2.6). The value of a solid angle is equal to the area of the segment on a unit sphere, centred at the angle's vertex. In our case it is the area of a spherical polygon going through $\boldsymbol{N}_1, \ldots, \boldsymbol{N}_n$. Let $\beta_1, \beta_2, \ldots, \beta_n$ be the interior angles of this polygon. We know that on a sphere, the area of a polygon with interior angles $\beta_1, \beta_2, \ldots, \beta_n$ is equal to (Kazhdan [2009, Subject: Surfaces]):

$$A_p = \pi\,(2 - n) + \sum_{i=1}^{n} \beta_i.$$

But what is the relation between β_i and α_i? An arc on a sphere is an intersection of the sphere and a plane going through the center. β_i is an angle between two arcs - in fact it is the angle between two planes, one going through the center and including vectors $\boldsymbol{N}_{i-1}$ and $\boldsymbol{N}_i$, the other going through the center and including vectors $\boldsymbol{N}_i$ and $\boldsymbol{N}_{i+1}$. The angle between these planes is π minus the angle between the planes' normals. But since $\boldsymbol{e}_i$ is perpendicular to both $\boldsymbol{N}_i$ and $\boldsymbol{N}_{i+1}$ and $\boldsymbol{e}_{i-1}$ is perpendicular to $\boldsymbol{N}_{i-1}$ and $\boldsymbol{N}_i$, the angle between the planes' normals is in fact the angle between $\boldsymbol{e}_i$ and $\boldsymbol{e}_{i-1}$: α_i. So we get:

$$\beta_i = \pi - \alpha_i.$$

And since the value of $\gamma_{\boldsymbol{v}}$ is the area of a spherical polygon going through $\boldsymbol{N}_1, \ldots, \boldsymbol{N}_n$, we get:

$$\begin{aligned}\gamma_{\boldsymbol{v}} &= \pi(2 - n) + \sum_{i=1}^{n} \beta_i \\ &= \pi(2 - n) + \sum_{i=1}^{n} (\pi - \alpha_i) \\ &= 2\pi - \sum_{i=1}^{n} \alpha_i.\end{aligned}$$

Now we can see that:
Integrated discrete mean curvature over discrete surface S is equal to:

$$\sum_{\boldsymbol{e} \in Edges} \frac{\gamma_{\boldsymbol{e}}}{2}\,|\boldsymbol{e}|.$$

Integrated discrete Gaussian curvature over discrete surface S is equal to:

$$\sum_{\boldsymbol{v} \in Vertices} (2\pi - \sum_{i=1}^{n} \alpha_i).$$

Definition 2.18. *Let S be a discrete surface.*
We define the mean curvature *as a function assigning values to edges:*

$$H_{\boldsymbol{e}} = \frac{\gamma_{\boldsymbol{e}}}{2}, \quad \boldsymbol{e} \in E,$$

where $\gamma_{\boldsymbol{e}}$ is the angle between the normals of the two faces that meet at the edge $\boldsymbol{e}$.
We define the Gaussian curvature *as a function assigning values to vertices:*

$$K_{\boldsymbol{v}} = \frac{2\pi - \sum_{t \in T | t \cap \boldsymbol{v} \neq \emptyset} \angle_t \boldsymbol{v}}{A_{Voronoi_{\boldsymbol{v}}}}, \quad \boldsymbol{v} \in V,$$

where $\angle_t \boldsymbol{v}$ is the interior angle at vertex $\boldsymbol{v}$ of a triangle t containing $\boldsymbol{v}$.

Note. For other possibilities of defining the Gaussian curvature, see Xu and Xu [2009].

Now, let's see if this discrete version of the Gaussian curvature conforms a discrete version of Gauss-Bonnet theorem (Theorem 2.3).

Theorem 2.4 (Discrete Gauss-Bonnet theorem)**.** *Let S be a closed discrete surface.*
Integrating the Gaussian curvature over S we get:

$$\int_S K_{\boldsymbol{v}} = 2\pi\chi,$$

where χ is the Euler characteristic of S.

Note. In the discrete setting the Euler characteristic χ is defined as:

$$\chi = |V| - |E| + |T|.$$

Proof. As a source of this proof we took Kazhdan [2009, Subject: Surfaces].

$$\begin{aligned}
\int_S K_{\boldsymbol{v}} &= \sum_{\boldsymbol{v}\in V} K_{\boldsymbol{v}}\, A_{Voronoi_{\boldsymbol{v}}} \\
&= \sum_{\boldsymbol{v}\in V} (2\pi - \sum_{t\in T|t\cap\boldsymbol{v}\neq\emptyset} \angle_t \boldsymbol{v}) \\
&= \sum_{\boldsymbol{v}\in V} (2\pi) - \sum_{\boldsymbol{v}\in V} \sum_{t\in T|t\cap\boldsymbol{v}\neq\emptyset} (\angle_t \boldsymbol{v}) \\
&= 2\pi|V| - \sum_{t\in T} \sum_{\boldsymbol{v}\in V|t\cap\boldsymbol{v}\neq\emptyset} (\angle_t \boldsymbol{v}) \\
&= 2\pi|V| - \sum_{t\in T} (\pi) \\
&= 2\pi|V| - \pi|T| \\
&\overset{*}{=} 2\pi|V| - \pi(2|E| - 2|T|) \\
&= 2\pi(|V| - |E| + |T|) \\
&= 2\pi\chi.
\end{aligned}$$

In $(*)$ we used a relation that says: $3\,|T| = 2\,|E|$, Since every triangle is bounded by three edges and every edge belongs to two triangles.

❑

We know that the Euler characteristic is a topological invariant. It means that however we pull or push some vertices, the total sum of all Gaussian curvatures remains the same.

Chapter 3

Mean curvature flow

We say that a set of surfaces is evolving under mean curvature flow if the length of the normal component of the velocity by which a point on the surface moves equals the mean curvature of the surface.

This chapter will be divided into Smooth setting, Discrete setting and Implementation. In the section Smooth setting we will show a direct relation between mean curvature flow and area minimization. In the section Discrete setting we will try to derive an analogue process: discrete mean curvature flow and show that one of its features is also the minimization of area of a discrete surface. It can be used for smoothing (denoising) a discrete surface. This application will be implemented in the last section.

3.1 Smooth Setting

In this section we define the mean curvature vector and mean curvature flow. We present some theorems that explain why mean curvature flow can be used for minimization of area of a surface. As a source we use Kazhdan [2009], Ambrosio and Soner [2004].

Definition 3.1. *We define the mean curvature vector* $\boldsymbol{H}$ *as:*

$$\boldsymbol{H} = H\,\boldsymbol{N},$$

H *stands for the mean curvature,* $\boldsymbol{N}$ *stands for the normal vector.*

Note. For a convex surface the mean curvature vector points inwards the convex object.

Definition 3.2. *A family of surfaces* S_t *is a* smooth flow *if there exists a smooth, one to one deformation map* $\boldsymbol{\phi}(.,t) : S_0 \to S_t$ *such that the Jacobian* $J_{\boldsymbol{\phi}(x,t)}$ *is not singular,* $\boldsymbol{\phi}(\boldsymbol{x},0) = \boldsymbol{x}$ *and* $\boldsymbol{\phi}_t(\boldsymbol{x},t)$ *is perpendicular to* S_t *at* $\boldsymbol{\phi}(\boldsymbol{x},t)$ *for any* $\boldsymbol{x} \in S_0$*,* $t \in [0,T]$*. A smooth flow is* mean curvature flow *if*

$$\boldsymbol{\phi}_t(\boldsymbol{x},t) = \boldsymbol{H}(\phi(\boldsymbol{x},t),t) \quad \forall\, \boldsymbol{x} \in S_0,\ t \in [0,T],$$

where $\boldsymbol{H}(y,t)$ *is the mean curvature vector of* S_t *at* $\boldsymbol{y}$*.*

Note. In other words, a family of surfaces is evolving under mean curvature flow if the length of the normal component of the velocity by which a point on the surface moves equals the mean curvature of the surface.

Now we need to show that **H** points in the direction where the area of a surface decreases most.

Theorem 3.1. *Let $g \in C_0^1(\mathbb{R}^3, \mathbb{R}^3)$ be a vector field, defining:*

$$\Phi(\boldsymbol{x}) := \boldsymbol{x} + \tau\, g(\boldsymbol{x}), \qquad S_\tau := \Phi_\tau(S),$$

for $|\tau| << 1$. Let $A(S_\tau)$ be the area of the surface S_τ.
Looking at the derivative of $\tau \to A(S_\tau)$ at $\tau = 0$ we get:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}(A(S_\tau))_{\tau=0} = -\int_S g\, \boldsymbol{H}\, \mathrm{d}s. \tag{3.1}$$

Proof. The proof and some other information can be found in Ambrosio and Soner [2004, p. 2].

❑

The formula 3.1 shows that the surface area decreases if the surface moves in the direction of the mean curvature vector **H**.

Corollary. Let S be a surface. Evolving S under mean curvature flow is evolving S in a way that the surface area decreases most rapidly out of all possible smooth flows.

Proof. This proof can be found in Garcke [2013, p. 3-5].

❑

We finally proved the connection between mean curvature flow and area minimization. If $\varphi : \Omega \to S$ is a regular surface, we decrease the area of S by shifting each point by vector $\varepsilon\,\mathbf{H}$. An example of this process is shown in Figure 3.1.

Figure 3.1: mean curvature flow on an approximation of a surface looking like a cup

3.2 Discrete setting

In this section we will present a discrete version of mean curvature flow. The discrete mean curvature vector will be defined and we will prove that the basic structures are still preserved. From this section will directly arise the algorithm for smoothing (denoising) discrete surfaces. Let Kazhdan [2009], Sullivan [2002], Crane [2013] be our principal sources of information.

As shown in Sullivan [2002], the mean curvature vectors supported along the edges and vertices are defined as the average of integrated mean curvature vectors over some regions including related edges and vertices. Since we do not characterize such integrating methods in our thesis, we just come up with the result.

Definition 3.3. *The mean curvature vector supported along an edge* $\boldsymbol{e}$ *is defined as:*

$$\boldsymbol{H_e} := \frac{1}{2}(\boldsymbol{e} \times \boldsymbol{N}_2 - \boldsymbol{e} \times \boldsymbol{N}_1),$$

where $\boldsymbol{N}_i$ *is the normal of triangle* T_i *and* T_1*,* T_2 *are triangles both including the edge* $\boldsymbol{e}$*.*

Note. In Chapter 2 we defined the total mean curvature of edge $\boldsymbol{e}$ as: $|\boldsymbol{e}|\,\frac{\gamma_e}{2}$, where $\gamma_{\boldsymbol{e}}$ is such an angle, that $\cos\gamma_{\boldsymbol{e}} = \langle \mathbf{N}_1, \mathbf{N}_2 \rangle$. As for mean curvature vectors, notice that $|\mathbf{H}_{\boldsymbol{e}}| = |\boldsymbol{e}| \sin\frac{\gamma_e}{2}$. Concerning the smooth version, in particular the Definition 3.1, we would like these values to be the same. However, in the discrete setting such relation is not true. Some information about this problem are mentioned in Sullivan [2002].

Even though the mean curvature and mean curvature vectors are understood to be supported along edges, the discrete mean curvature flow is an algorithm that moves the vertices. Therefore we define the mean curvature vectors supported on vertices, too.

Definition 3.4. *The mean curvature vector supported on a vertex* $\boldsymbol{v}_i$ *is defined as:*

$$\boldsymbol{H}_{\boldsymbol{v}_i} = \frac{1}{2\,A_{Voronoi_{\boldsymbol{v}_i}}} \sum_{\boldsymbol{e}|\boldsymbol{v}_i \in \boldsymbol{e}} \boldsymbol{H_e}, \tag{3.2}$$

where the edge $\boldsymbol{e}$ *is taken as a vector pointing from* $\boldsymbol{v}_i$ *to its neighbour.*

By further computation and using Proposition 3.2 we get that:

$$\mathbf{H}_{\boldsymbol{v}_i} = \frac{1}{4\,A_{Voronoi_{\boldsymbol{v}_i}}} \sum_{j|\boldsymbol{v}_j \in Nbr(\boldsymbol{v}_i)} (\cot\gamma_{ij} + \cot\beta_{ij})(\mathbf{v}_j - \mathbf{v}_i), \tag{3.3}$$

where γ_{ij}, β_{ij} denote the two angles opposite the edge $\mathbf{v}_j - \mathbf{v}_i$.

Note. We take the edges as vectors pointing from $\mathbf{v}_i$ to its neighbours so that the mean curvature vectors of vertices of a convex discrete surface point inwards (just like in the smooth case). The meaning of the constants $2\,A_{Voronoi_{\boldsymbol{v}_i}}$ in (3.2) is explained in Meyer et al. [2002]. Very roughly speaking, the 2 stands for the two vertices belonging to the edge, the $A_{Voronoi_{\boldsymbol{v}_i}}$ is to normalize the integration over Voronoi region that is expressed by $\sum_{\boldsymbol{e}|\boldsymbol{v}_i \in \boldsymbol{e}} \mathbf{H}_{\boldsymbol{e}}$.

Theorem 3.2. *Let* $0, p, q$ *be a triangle, such that* $\angle 0pq, \angle 0qp \leq \frac{\pi}{2}$*. The vector perpendicular to* $\boldsymbol{p}$*-*$\boldsymbol{q}$ *with length* $|\boldsymbol{p}$*-*$\boldsymbol{q}|$ *can be expressed as:*

$$\boldsymbol{u} = \cot(\angle 0pq)\,\boldsymbol{q} + \cot(\angle 0qp)\,\boldsymbol{p}.$$

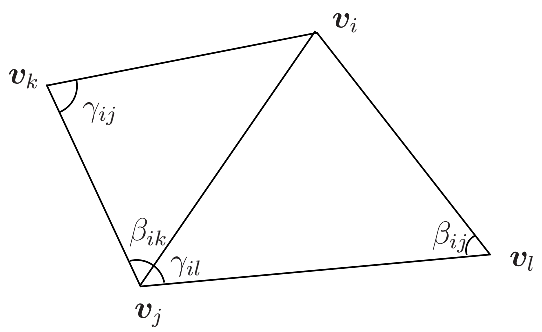


Figure 3.2: vertex with its three neighbouring vertices

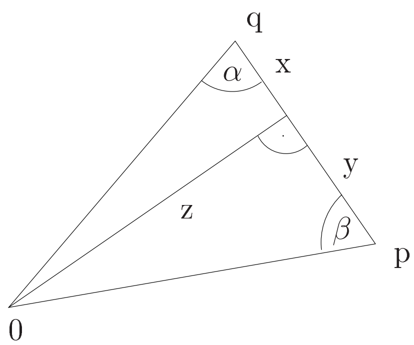


Figure 3.3: triangle with its height

Proof. Firstly we will prove that the inner product of **u** and **p-q** is equal to 0. Under x, y, z, α, β we will understand the length of sides and angles shown in Figure 3.3.

$$\begin{aligned}
\mathbf{u} \cdot (\mathbf{p\text{-}q}) &= (\cot(\angle\beta)\,\mathbf{q} + \cot(\angle\alpha)\,\mathbf{p}) \cdot (\mathbf{p\text{-}q}) \\
&= \frac{y}{z}|\mathbf{q}|^2 - \frac{y}{z}|\mathbf{p}||\mathbf{q}|\cos(\pi - (\alpha+\beta)) + \frac{x}{z}|\mathbf{p}||\mathbf{q}|\cos(\pi - (\alpha+\beta)) - \frac{x}{z}|\mathbf{p}|^2 \\
&= \frac{y}{z}(z^2 + x^2) - \frac{y}{z}(z^2 - xy) + \frac{x}{z}(z^2 - xy) + xz - \frac{x}{z}(z^2 + y^2) = 0.
\end{aligned}$$

Now we need to show that $|\mathbf{u}|^2 = |\mathbf{p\text{-}q}|^2 = (x+y)^2$:

$$\begin{aligned}
|\mathbf{u}|^2 &= (\cot(\beta)\,\mathbf{q} + \cot(\angle\alpha)\,\mathbf{p})^2 \\
&= (\frac{y^2}{z^2}(x^2 + z^2) + 2\frac{y}{z}\frac{x}{z}(z^2 - xy) + \frac{x^2}{z^2}(z^2 + y^2)) \\
&= (x^2 + 2xy + y^2).
\end{aligned}$$

❑

The discrete mean curvature vectors that determine the discrete mean curvature flow are defined. Now we show that the discrete mean curvature flow decreases the surface area.

Theorem 3.3. *The discrete mean curvature vector assigned to a vertex points in the opposite direction of the discrete area gradient assigned to the related vertex.*

Proof. Since the area gradient is a local information, we consider the area of triangles including related vertex.

Firstly we compute the area gradient of a triangle 0, p, q with respect to vertex 0. The question is: What direction should we move the origin in order to maximally increase triangle's area?

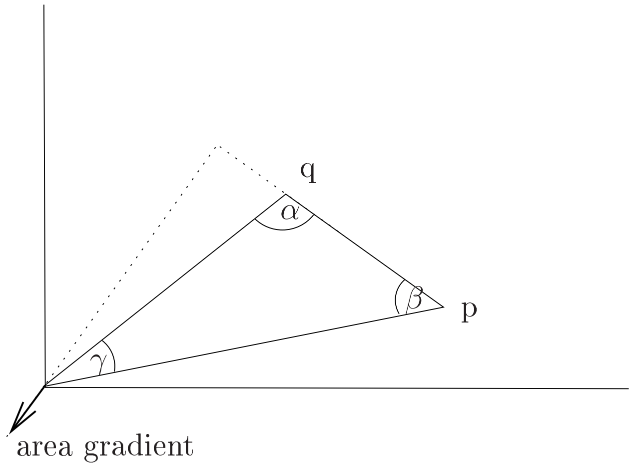


Figure 3.4: triangel with its area gradient

Since the area of a triangle is computed as:

$$S = \frac{1}{2}|h||\mathbf{p}\text{-}\mathbf{q}|,$$

where h is the height of the triangle, perpendicular to **p-q**.
So, in order to maximally increase the area of this triangle, we move the third vertex in the direction of h. Offsetting the vertex by ε we get a triangle with the area of:

$$S = \frac{1}{2}(|h| + \varepsilon)|\mathbf{p}\text{-}\mathbf{q}|,$$

so the change is:

$$\varepsilon\frac{1}{2}|\mathbf{p}\text{-}\mathbf{q}|.$$

It implies that the triangle area gradient is the vector perpendicular to **p-q** with the length of $|\mathbf{p}\text{-}\mathbf{q}|/2$, pointing to 0. Thanks to the Proposition 3.2 we know that the triangle area gradient displayed in Figure 3.4 can be expressed as:

$$\frac{1}{2}\left(\cot(\angle\beta)\,\mathbf{q} + \cot(\angle\alpha)\,\mathbf{p}\right).$$

By summing the area gradients of all triangles including vertex $\mathbf{v}_i$ we get that the area gradient of $\mathbf{v}_i$ is:

$$\bigtriangledown_{area}(\mathbf{v}_i) = \frac{1}{2}\sum_{j|\boldsymbol{v}_j\in Nbr(\boldsymbol{v}_i)} (\cot\gamma_{ij} + \cot\beta_{ij})(\mathbf{v}_i - \mathbf{v}_j),$$

where γ_{ij} and β_{ij} are opposite angles of edge $\mathbf{v}_i - \mathbf{v}_j$ (See Figure 3.2).
It means that the discrete mean curvature vector is pointing in the opposite direction of the area gradient.

$\square$

We finally managed to show that the discrete version of mean curvature vector has the area minimization feature and also preserves some other global theorems. The discrete mean curvature flow is basically shifting each vertex $\mathbf{v}_i$ by vector $\varepsilon \mathbf{H}_{v_i}$. In the next section we will see that it produces some really nice denoising results.

3.3 Implementation

In this section we will come up with a MATLAB code that minimizes the total area and removes the noise from discrete surfaces. Our code will be able to cope only with closed triangulated surfaces.

3.3.1 Algorithm

According to earlier computation we can express the discrete mean curvature vector at vertex v_i as:

$$\mathbf{H}_{v_i} = \frac{1}{4\,A_{Voronoi_{v_i}}} \sum_{j|v_j \in Nbr(v_i)} (\cot \gamma_{ij} + \cot \beta_{ij})(\mathbf{v}_j - \mathbf{v}_i), \tag{3.4}$$

where γ_{ij}, β_{ij} denote the two angles opposite the edge $\mathbf{v}_j - \mathbf{v}_i$.

In our implementation we will compute the discrete mean curvature vector for every vertex of a discrete surface and then move it by an ε-multiple of the vector in order to minimize the surface area or remove the noise ("smoothed" our surface). The ε can not be too large because it might lead to a numerical blow-up. Repeating this process for several times will make the discrete surface successively smoother.

The area of the Voronoi region will be approximated by one third of the sum of all triangles including related vertex.

From now one let's call this matrix the *curvature operator*:

$$\mathbf{K}_{ij} = \frac{(\cot \gamma_{ij} + \cot \beta_{ij})}{2}, \tag{3.5}$$

where γ_{ij}, β_{ij} denote the two angles opposite the edge $\mathbf{v}_j - \mathbf{v}_i$. Obviously this matrix will be a sparse matrix, filled only for those (i, j) where there exists an edge between v_i, v_j.

3.3.2 Input and Output

Our area minimizing process is implemented as a function that takes a mesh and produces a new smoother mesh. Discrete surfaces are standardly stored in simple text formats that specify vertex positions as their (x,y,z) coordinates and triangles as lists of vertex indices.

Example. .
v 0 0 0
v 1 0 0
v 0.5 1 0
v 0.5 0.5 1
f 1 2 4
f 2 3 4
f 1 3 4
f 1 2 3

During our process we store mesh in two matrices:
V ... dense $|V| \times 3$ matrix that represents vertex positions ($|V|$ is the number of vertices)
T ... dense $|T| \times 3$ matrix that represents the indices of 3 vertices forming each triangle ($|T|$ is the number of triangles).

$$\{V_{ij}\} = \text{the j-th coordinate of the i-th vertex}$$
$$\{T_{ij}\} = \text{index of the j-th vertex that is one of the corners of the i-th triangle}$$

As for our example (3.3.2)

$$V = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.5 & 0.5 & 1 \end{pmatrix},$$
$$T = \begin{pmatrix} 1 & 2 & 4 \\ 2 & 3 & 4 \\ 1 & 3 & 4 \\ 1 & 2 & 3 \end{pmatrix}.$$

The Output will be a mesh represented by two matrices V and T, V representing new vertex positions, T remains unchanged.

3.3.3 Process

In this subsection we will describe three main codes used in implementing the discrete mean curvature operator. The codes and a deeper description can be found in Appendix A.

In our implementation the process of minimizing surface area is realized by a function *step(filename)*:

```
function [V,T] = step ( filename ).
```

INPUT:
filename ... path to a mesh file stored in a simple text format

OUTPUT:
V ... dense $|V| \times 3$ matrix of new vertex positions

T ... dense $|T| \times 3$ matrix of triangles stored as indices into vertex list

Function *step(filename)* loads a mesh, stores it in matrices V and T, initiates another function $mean\_curv\_vectors(V, T)$, which produces the discrete mean curvature vectors for all vertices. Function *step* moves each vertex in the direction of its mean curvature vector with a small step size. This process iterates several times. The step size and the number of iterates is set before.

Function that creates the discrete mean curvature vectors is called $mean\_curv\_vectors(V, T)$:

```
function[vectors] = mean_curv_vectors( V,T )
```

INPUT:
V ... dense $|V| \times 3$ matrix of vertex positions
T ... dense $|T| \times 3$ matrix of triangle stored as indices into vertex list

OUTPUT:
vectors ... dense |V|x3 matrix of discrete mean curvature vectors of all vertices

Function $mean\_curv\_vectors(V, T)$ uses function $build\_K(V, T)$, which produces the curvature operator and the vector of areas of adjacent triangles to each vertex. After that it creates the discrete mean curvature vector to each vertex by multiplying the curvature operator by vector $(v_j - v_i)$ and dividing by $2\, A_{Voronoi_{v_i}}$.

Function *build_K(V,T)* produces the curvature operator and the vector of triangle areas:

```
function [K,areas] = build_K( V,T )
```

INPUT:
V ... dense $|V| \times 3$ matrix of vertex positions
T ... dense $|T| \times 3$ matrix of triangle stored as indices into vertex list

OUTPUT:
K ... sparse |V|x|V| matrix - curvature operator areas ... dense $1 \times |V|$ matrix of areas of triangles including the concrete vertex, i-th column is the adjacent area of the i-th vertex

Function *build_K(V,T)* creates the curvature operator defined in (3.5). When making this matrix, it is easiest to iterate over each triangle and compute its contribution to each vertex. This function uses other functions, namely *cotangent(a,b,c)* and *area_of_triangle(a,b,c)*. They will be displayed in Appendix, too.

3.3.4 Results

We will show some examples of the results of minimizing area process.

Cube

At the beginning we have a cube, each edge with the length of 1, displayed in Figure 3.5. After applying the process with the number of iterations: 15, size of step: 0.006, we get a new discrete surface, displayed in Figure 3.6.

Noisy Torus

This is an example of a noise-removing process.

At the beginning we have a noisy torus, displayed in Figure 3.7. After applying the process with the number of iterations: 4, size of step: 0.002, we get a new, smoother discrete surface, displayed in Figure 3.8. After applying the process with the number of iterations: 15, size of step: 0.002, we get an even more smooth discrete surface, displayed in Figure 3.9.

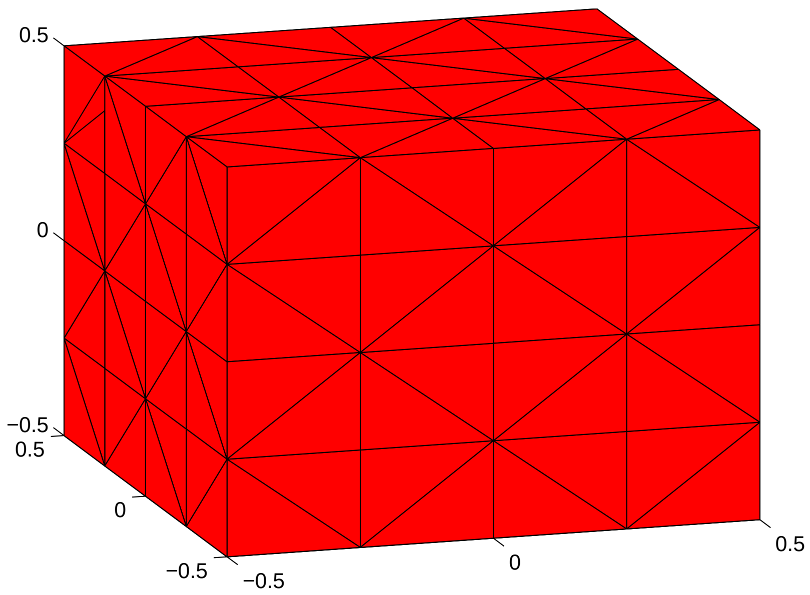


Figure 3.5: cube before the beginning of the process

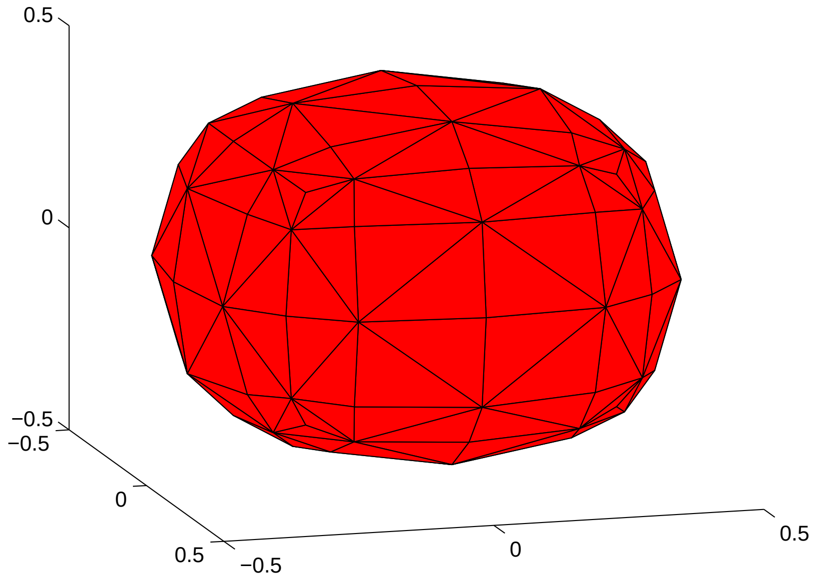


Figure 3.6: cube after 15 iterations

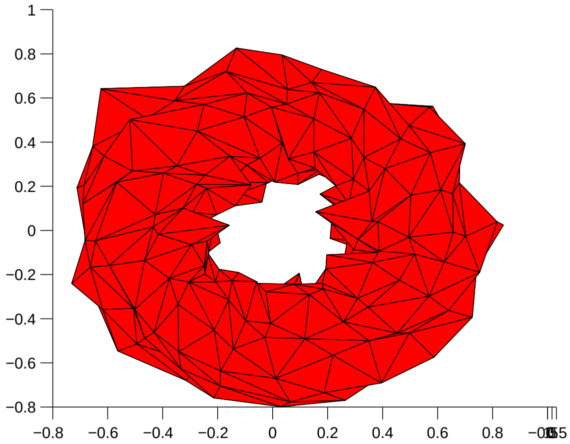


Figure 3.7: noisy torus before the beginning of the process

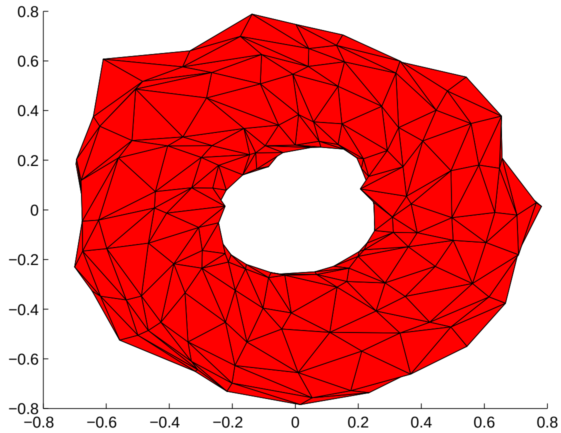


Figure 3.8: noisy torus after 4 iterations

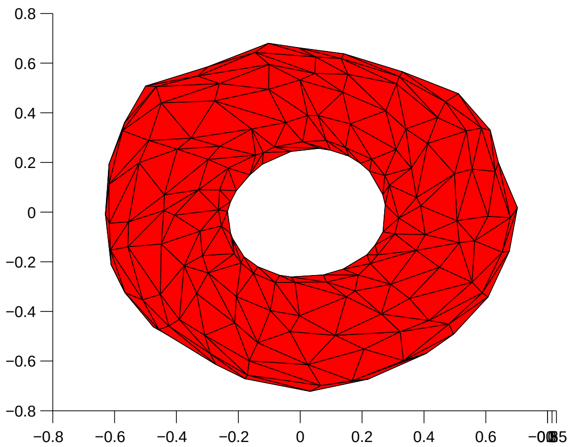


Figure 3.9: noisy torus after 15 iterations

Conclusion

We defined some examples of the important expressions from Differential geometry in the Discrete differential geometry. Even though some of them are not as consistent as one would like (discrete mean curvature, mean curvature vector), the preservation of some other global features (Turning number theorem, Gauss-Bonnet theorem) ensures us that there really is a point in finding the analogues of expressions from classic Differential geometry.

The implementation of Discrete mean curvature flow shows us that it can be used for denoising of discrete surfaces. Since the main motivation for introducing Discrete differential geometry was the practical usage, this thesis can be thought of as the theoretical basis for some other applications.

Bibliography

L. Ambrosio and H. M. Soner. Flow by mean curvature of surfaces of any codimension. 2004. URL http://www.math.ethz.ch/~hmsoner/pdfs/Flow-by-mean-curvature-flow.pdf. Date of access: 25.7.2014.

K. Crane. Digital geometry processing with discrete exterior calculus. 2013. URL http://www.geometry.caltech.edu/pubs/DGPDEC_SIG13.pdf. Date of access: 25.7.2014.

H. Garcke. Curvature driven interface evolution. 2013. URL http://www.uni-regensburg.de/Fakultaeten/nat_Fak_I/Mat8/lst/PREPRINT.pdf. Date of access: 25.7.2014.

M. Kazhdan. Advanced topics in computer graphics: Discrete differential geometry. 2009. URL http://www.cs.jhu.edu/~misha/Fall09/. Date of access: 25.7.2014.

M. Meyer, M. Desbrun, P. Schroder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. 2002. URL http://multires.caltech.edu/pubs/diffGeoOps.pdf. Date of access: 25.7.2014.

K. Polthier and W. Rossman. Discrete constant mean curvature surfaces and their index. 2002. URL http://polthier.info/articles/index/marylin_Preprint.pdf. Date of access: 25.7.2014.

A. Pressley. *Elementary Differential Geometry*. Second Edition. Springer London, 2010. ISBN 978-1-84882-890-2.

C. Shonkwiler. Differential geometry hw 1. 2005. URL http://www.math.uga.edu/~clayton/courses/660/660_1.pdf. Date of access: 25.7.2014.

V. Souček. Geometrie v $\mathbb{R}^3$, 4. semestr. 2012. URL http://www.karlin.mff.cuni.cz/~soucek/kpl_11_2012.pdf. Date of access: 25.7.2014.

D. J. Struik. *Lectures on Classical Differential Geometry*. Second Edition. Dover Publications, 1961. ISBN 000-0486656098.

J. M. Sullivan. Curvatures of smooth and discrete surfaces. 2002. URL `http://torus.math.uiuc.edu/jms/Papers/dscrv.pdf`. Date of access: 25.7.2014.

T. J. Willmore. *An Introduction To Differential Geometry*. Dover Publications, New York, 2012. ISBN 0486486184.

Z. Xu and G. Xu. Discrete schemes for gaussian curvature and their convergence. 57:1187 – 1195, 2009. URL `http://www.cc.ac.cn/06research_report/0611.pdf`. Date of access: 25.7.2014.

List of Figures

Appendix

Here we present the MATLAB codes - implemented discrete mean curvature flow with their detailed description.

`function [V,T] = step ( filename ).`

INPUT:
filename ... path to a mesh file stored in a simple text format

OUTPUT:
V ... dense $|V| \times 3$ matrix of new vertex positions
T ... dense $|T| \times 3$ matrix of triangles stored as indices into vertex list

Function *step(filename)* loads a mesh, stores it in metrices V and T, initiates another function $mean\_curv\_vectors(V, T)$ and its product stores in a matrix *vectors*. Function $mean\_curv\_vectors(V, T)$ produces discrete mean curvature vectors for each vertex. After obtaining these vectors, function *step* moves each vector in the direction of its mean curvature vector with a step size of *eps*. This process iterates nr_iter-times. The value of *eps* and nr_iter is set before.

MATLAB code:

```
function [V,T] = step (filename)
% function [V,T] = step( filename )
%
% Computes new vertex positions for a smoothed mesh.
%
% INPUT:
%    filename ... path to mesh file
%
% OUTPUT:
%    V ... dense |V|x3 matrix of new vertex positions
%    T ... dense |T|x3 matrix of triangles stored as indices into vertex list
%

% loads a mesh
[V,T] = mesh_read(filename);

% size of a step taken in the mean curvature vector direction
eps = 0.01;

% number of iterations
nr_iter = 30;

for i = 1:nr_iter
        % mean curvature vectors for all vertices
    vectors = mean_curv_vectors(V,T);

        % moving in the area gradient direction with the step of eps
    V = V + eps*vectors;
end
end
```

Function that creates the discrete mean curvature vectors is called $mean\_curv\_vectors(V,T)$:

```
function[vectors] = mean_curv_vectors( V,T )
```

INPUT:
V ... dense $|V| \times 3$ matrix of vertex positions
T ... dense $|T| \times 3$ matrix of triangles stored as indices into vertex list

OUTPUT:
vectors ... dense |V|x3 matrix of discrete mean curvature vectors of all vertices

Function $mean\_curv\_vectors(V,T)$ uses function $build\_K(V,T)$ to create the curvature operator. It is stored in a matrix K. Areas of adjacent triangels to each vertex are stored in a vector *areas*. After that it creates the mean curvature vector to each vertex, stores them in matrix *vectors*.

```
function[vectors] = mean_curv_vectors(V,T)
% function [vectors] = mean_curv_vectors( V,T )
%
% Computes the discrete mean curvature vector
% for each vertex in  mesh represented by V,T.
%
% INPUT:
%    V ... dense |V|x3 matrix of vertex positions
%    T ... dense |T|x3 matrix of triangles stored as indices into vertex list
%
% OUTPUT:
%    vectors ... dense |V|x3 matrix of discrete mean curvature vectors
%                                 of all vertices
%

% used to store mean curvature vectors
K = sparse(length(V),length(V));

% areas(i) - area of all triangles meeting at vertex V(i,:)
areas = zeros(1,length(V));

%computing curvature operator and areas of triangles
[K,areas] = build_K(V,T);

% used to store current vectors V(i,:)-V(j,:)
U = zeros(length(V),3);

% mean curvature vectors
vectors = zeros(length(V),3);

%computing mean curvature vectors
for i=1:length(V)
   for j = 1:length(V)
       U(j,:) = V(j,:)-V(i,:);
       vectors(i,:) = K(i,:)*U;
   end
end

for i = 1:length(V)
    vectors(i,:) = vectors(i,:)/(2*areas(i)/3);
end
```

Function *build_K(V,T)* produces the curvature operator:

```
function [K,areas] = build_K( V,T )
```

INPUT:
V ... dense $|V| \times 3$ matrix of vertex positions
T ... dense $|T| \times 3$ matrix of triangles stored as indices into vertex list

OUTPUT:
K ... sparse |V|x|V| matrix of curvature operator areas ... dense $1 \times |V|$ matrix of areas of triangles including the concrete vertex, i-th column is the adjacent area of the i-th vertex

Function *build_K(V,T)* creates the curvature operator. It is represented by a $|V| \times |V|$ sparse matrix, defined in 3.5. When filling K it is easiest to iterate over each triangle and compute its contribution to each vertex. This function uses other functions, namely *cotangent(a,b,c)* and *area_of_triangle(a,b,c)*. They will be displayed in this appendix, later.

```
function [K,areas] = build_K( V,T )
% function [K,areas] = build_K( V,T )
%
% Computes the curvature operator.
%
% INPUT:
%    V ... dense |V|x3 matrix of vertex positions
%    T ... dense |T|x3 matrix of triangles stored as indices into vertex list
%
% OUTPUT:
%    K ... sparse |V|x|V| matrix of curvature operator
%    areas ... dense 1x|V| matrix of areas of triangles adjacent to each vertex
%

% used to store the curvature operator
K = sparse(length(V),length(V));

% areas(i) - area of all triangles meeting at vertex V(i,:)
areas = zeros(1,length(V));

for i = 1:length(T)
        % current vertices, forming the i-th triangle, indices into vertex list
    tmp_vert = T(i,:);

    % computing the area of current triangle(triangle)
    S = area_of_triangle(norm(V(tmp_vert(2),:) - V(tmp_vert(3),:)),
                norm(V(tmp_vert(3),:) - V(tmp_vert(1),:)),
                norm(V(tmp_vert(2),:) - V(tmp_vert(1),:)));

    % computing cotangent of the angle at current vertex nr. 1
    cotg = cotangent(norm(V(tmp_vert(1),:) - V(tmp_vert(3),:)),
                norm(V(tmp_vert(2),:) - V(tmp_vert(1),:)),
                norm(V(tmp_vert(2),:) - V(tmp_vert(3),:)),S);
    K(tmp_vert(2),tmp_vert(3)) = K(tmp_vert(2),tmp_vert(3)) + cotg/2;
    K(tmp_vert(3),tmp_vert(2)) = K(tmp_vert(2),tmp_vert(3));

    % computing cotangent of the angle at current vertex nr. 2
    cotg = cotangent(norm(V(tmp_vert(1),:) - V(tmp_vert(2),:)),
                norm(V(tmp_vert(2),:) - V(tmp_vert(3),:)),
                norm(V(tmp_vert(1),:) - V(tmp_vert(3),:)),S);
    K(tmp_vert(1),tmp_vert(3)) = K(tmp_vert(1),tmp_vert(3)) + cotg/2;
    K(tmp_vert(3),tmp_vert(1)) = K(tmp_vert(1),tmp_vert(3));

    % computing cotangent of the angle at current vertex nr. 3
    cotg = cotangent(norm(V(tmp_vert(1),:) - V(tmp_vert(3),:)),
                norm(V(tmp_vert(2),:) - V(tmp_vert(3),:)),
                norm(V(tmp_vert(1),:) - V(tmp_vert(2),:)),S);
    K(tmp_vert(1),tmp_vert(2)) = K(tmp_vert(1),tmp_vert(2)) + cotg/2;
    K(tmp_vert(2),tmp_vert(1)) = K(tmp_vert(1),tmp_vert(2));

    %storage of the sum of all triangles adjacent to each vertex
    areas(tmp_vert(1)) = areas(tmp_vert(1)) + S;
```

```
    areas(tmp_vert(2)) = areas(tmp_vert(2)) + S;
    areas(tmp_vert(3)) = areas(tmp_vert(3)) + S;
end
end
```

Two more minor codes used in creating the curvature operator, we will not write more information about them, since they are quite trivial.

```
function [cotg] = cotangent(a,b,c,S)
% function [cotg] = cotangent(a,b,c,S)
%
% Computes the cotangent of the angle opposite the side c.
%
% INPUT:
%    a,b,c ... lengths of sides forming a triangle
%    S ... area of the triangle formed by sides a,b,c
%
% OUTPUT:
%   cot ... cotangent of the angle opposite the side c.

cotg = (a^2 + b^2 - c^2)/(4*S);
end
```

```
function [S] = area_of_triangle(a,b,c)
% function [S] = area_of_triangle(a,b,c)
%
% Computes the area of the triangle formed by sides a,b,c.
%
% INPUT:
%    a,b,c ... lengths of sides forming a triangle
%
% OUTPUT:
%    S ... area of the triangle formed by sides a,b,c


s = (a+b+c)/2;
S = sqrt(s*(s-a)*(s-b)*(s-c));
end
```