PART I:  
OVERVIEW

DISCRETE DIFFERENTIAL 
GEOMETRY: 
AN APPLIED INTRODUCTION

Keenan Crane • CMU 15-458/858B • Fall 2017
Geometry is Coming…
Applications of DDG: Geometry Processing
Applications of DDG: Shape Analysis
Applications of DDG: Numerical Simulation
Applications of DDG: Discrete Models of Nature
What Will We Learn in This Class?

• First and foremost: how to think about shape…
  • …mathematically (differential geometry)
  • …computationally (geometry processing)

• Central Theme: link these two perspectives

• Why? Shape is everywhere!
  • computational biology, industrial design, computer vision, machine learning, architecture, computational mechanics, fashion, medical imaging…

• Flat images are old news :-)
Course Web Page

• All course information is spelled out in detail on course webpage:

   http://geometry.cs.cmu.edu/ddg

• All communication goes through this site (assignments, discussion, etc.)

• Register account from link at end of menu:

  META

  • Register
  • Log in
  • Entries RSS
  • Comments RSS
  • WordPress.org
Assignments

• **Derive** geometric algorithms from first principles (pen-and-paper)

• **Implement** geometric algorithms (coding)
  • Exterior calculus
  • Curvature
  • Smoothing
  • Parameterization
  • Distance computation
  • Vector field analysis
  • Direction Field Design

*(Pick 6 of 7!)*
Homework Submission

• All homework must be submitted digitally
• All source files in a single zip file called solution.zip
• All written exercises in a single PDF file called exercises.pdf
  • Either typeset (e.g., LaTeX) or scans/photos of written work.
  • Convert images to PDF using Preview (Mac) or imagetopdf.com
• Email to geometry.collective@gmail.com with requested string in the subject line (e.g., DDG17A1)
• Will receive written feedback via email as marked-up PDF
Late Policy

• Assignments due at 5:59:59pm on due date (Eastern time zone)
• Can use **five late days** throughout semester
  • Must indicate which late day you’re using (1–5) in email
  • If writing answers on paper, must also draw one of five Platonic solids:

1
2
3
4
5

• All subsequent late work will receive a zero!
Grade Breakdown

- **Assignments** – 78% (pick 6 out of 7*)
  - (13%) A1: Exterior Calculus
  - (13%) A2: Normals & Curvature
  - (13%) A3: Surface Fairing
  - (13%) A4: Vector Field Processing
  - (13%) A5: Surface Parameterization
  - (13%) A6: Geodesic Distance
  - (13%) A7: Direction Field Design

- **Final Exam** – 12%

- **Participation** – 10%
  - (5%) – in-class/web participation
  - (5%) – reading summaries/questions

“*I can’t give you a brain, but I can give you a diploma.*”

—Frank L. Baum

*Complete 7th assignment for up to 12% extra credit.*
What is Differential Geometry?

• **Language** for talking about *local properties of shape*
  - How fast are we traveling along a curve?
  - How much does the surface bend at a point?
  - etc.

• …and their connection to *global properties of shape*

• So-called “local-global” relationships.

• Modern language of geometry, physics, statistics, …

• Profound impact on scientific & industrial development in 20th century
What is Discrete Differential Geometry?

• Also a language describing local properties of shape

• *Infinity no longer allowed!*

• No longer talk about derivatives, infinitesimals…

• Everything expressed in terms of lengths, angles…

• Surprisingly little is lost!

• Faithfully captures many fundamental ideas

• Modern language of geometric computing

• Increasing impact on science & technology in 21st century.
Translate differential geometry into language suitable for computation.
How can we get there?

A common “game” is played in DDG to obtain discrete definitions:

1. Write down several **equivalent** definitions in the smooth setting.
2. Apply each smooth definition to an object in the discrete setting.
3. Determine which properties are captured by each resulting **inequivalent** discrete definition.

One often encounters a so-called “no free lunch” scenario: no single discrete definition captures *all* properties of its smooth counterpart.
Example: Discrete Curvature of Plane Curves

- **Toy example:** curvature of plane curves
- Roughly speaking: “how much it bends”
- First review smooth definition
- Then play The Game to get discrete definition(s)
- Will discover that no single definition is “best”
- *Pick the definition best suited to the application*
- **Today** we’re gonna quickly cover a lot of ground…
- Will start more slowly from the basics *next lecture*
Curvature of a Curve—Motivation
Curves in the Plane

• In the smooth setting, a **parameterized curve** is a map* taking each point in an interval \([0,L]\) of the real line to some point in the plane \(\mathbb{R}^2\):

*Continuous, differentiable, smooth…
Curves in the Plane—Example

• As an example, we can express a circle as a parameterized curve \( \gamma \):

\[
\gamma : [0, 2\pi) \rightarrow \mathbb{R}^2; \ s \mapsto (\cos(s), \sin(s))
\]
Discrete Curves in the Plane

- Special case: a **discrete curve** is a *piecewise linear* parameterized curve, *i.e.*, it is a sequence of **vertices** connected by straight line segments:

Shorthand: $\gamma_i := \gamma(s_i)$
A simple example is a curve comprised of two segments:

\[ \gamma(s) := \begin{cases} 
(s,0), & 0 \leq s \leq 1, \\
(1,s), & 1 < s \leq 2 
\end{cases} \]
Informally, a vector is tangent to a curve if it “just barely grazes” the curve.

More formally, the unit tangent (or just tangent) of a parameterized curve is the map obtained by normalizing its first derivative:

\[ T(s) := \frac{d}{ds} \gamma(s) / \left\| \frac{d}{ds} \gamma(s) \right\| \]

If the derivative already has unit length, then we say the curve is arc-length parameterized and can write the tangent as just

\[ T(s) := \frac{d}{ds} \gamma(s) \]
Tangent of a Curve — Example

Let’s compute the unit tangent of a circle:

\[ \gamma : [0, 2\pi) \rightarrow \mathbb{R}^2; \ s \mapsto (\cos(s), \sin(s)) \]

\[ \frac{d}{ds} \gamma(s) = (-\sin(s), \cos(s)) \]

\[ \cos^2(s) + \sin^2(s) = 1 \]

\[ \Rightarrow T = \frac{d}{ds} \gamma(s) \]
Informally, a vector is normal to a curve if it "sticks straight out" of the curve.

More formally, the unit normal (or just normal) can be expressed as a quarter-rotation \( \mathcal{J} \) of the unit tangent in the counter-clockwise direction:

\[
N(s) := \mathcal{J} T(s)
\]

In coordinates \((x,y)\), a quarter-turn can be achieved by* simply exchanging \(x\) and \(y\), and then negating \(y\):

\[
(x, y) \xrightarrow{\mathcal{J}} (-y, x)
\]

*Why does this work?
Normal of a Curve—Example

• Let’s compute the unit normal of a circle:

\[ \gamma : [0, 2\pi) \to \mathbb{R}^2; \ s \mapsto (\cos(s), \sin(s)) \]

\[ T(s) = (-\sin(s), \cos(s)) \]
\[ N(s) = J T(s) = (-\cos(s), -\sin(s)) \]

Note: could also adopt the convention \( N = -J T \).
(Just remain consistent!)
Curvature of a Plane Curve

• Informally, curvature describes “how much a curve bends”

• More formally, the curvature of an arc-length parameterized plane curve can be expressed as the rate of change in the tangent:

\[ \kappa(s) := \langle N(s), \frac{d}{ds} T(s) \rangle \]
\[ = \langle N(s), \frac{d^2}{ds^2} \gamma(s) \rangle \]

**Key Idea**
Curvature is a second derivative.

*Here the angle brackets denote the usual dot product, i.e., \( \langle (a, b), (x, y) \rangle := ax + by. \)
Curvature: From Smooth to Discrete

**Key Idea**
Curvature is a *second derivative*.

\[ \kappa = \langle \mathcal{I} \frac{d}{ds} \gamma, \frac{d^2}{ds^2} \gamma \rangle \]

Can we directly apply this definition to a discrete curve?

**Smooth**

No! Will get either zero or “\( \infty \)”. Need to think about it another way…
What is Discrete Curvature?

**Key Idea**

Curvature is a *second derivative*.

Can we directly apply this definition to a discrete curve?

**Smooth**

Curvature is finite.

**Discrete**

Curvature can be either zero or “∞”. Need to think about it another way…
When is a Discretization “Good?”

• How will we know if we came up with a good definition?

• Many different criteria for “good”:
  
  • satisfies (some of the) same properties/theorems as smooth curvature
  
  • converges to smooth value as we refine our curve
  
  • efficient to compute / solve equations
  
  • …

\[ \int \kappa \, ds \in 2\pi \mathbb{Z} \]

Complex Ta = gamma[i] - gamma[i-1];
Complex Tb = gamma[i+1] - gamma[i];
double kappa = (Tb*Ta.inv()).arg();
Curvature, Revisited

• In the **smooth** setting, there are several other **equivalent** definitions of curvature.

• **IDEA:** perhaps some of these definitions can be applied directly to our discrete curve!
Turning Angle

- Our initial definition of curvature was the rate of change of the tangent in the normal direction.

- Equivalently, we can measure the rate of change of the angle the tangent makes with the horizontal:

$$\kappa(s) = \langle N(s), \frac{d}{ds} \gamma(s) \rangle$$

$$\kappa(s) = \frac{d}{ds} \varphi(s)$$
• Still can’t evaluate curvature at vertices of a discrete curve \((at \text{ what rate does the angle change?})\)

• But let’s consider the \textit{integral} of curvature along a short segment:

\[
\int_a^b \kappa(s) \, ds = \int_a^b \left( \frac{d}{ds} \varphi(s) \right) \, ds = \varphi(b) - \varphi(a)
\]

• Instead of \textit{derivative} of angle, we now just have a \textit{difference} of angles.

• This definition works for our discrete curve!
Discrete Curvature (Turning Angle)

• This formula gives us our first definition of discrete curvature, as just the exterior angle at the vertex of each curve:

\[ \theta_i := \varphi_{i,i+1} - \varphi_{i-1,i} \quad \text{(exterior angle)} \]

\[ \kappa_i^A := \theta_i \in (-\pi, \pi) \quad \text{(discrete curvature)} \]

• Here we encounter another theme from discrete differential geometry: the quantities we want to work with are often more naturally expressed as \textit{integrated} rather than \textit{pointwise} quantities.
Length Variation

• Are there other ways to get a definition for discrete curvature?
• Well, here’s a useful fact about curvature from the smooth setting:

  The fastest way to decrease the length of a curve is to move it in the normal direction, with speed proportional to curvature.

• Intuition: in flat regions, moving the curve doesn’t change its length; in curved regions, the change in length (per unit length) is large:
Length Variation

• More formally, consider an arbitrary variation of the curve. I.e., suppose we have another curve* \( \eta : [0, L] \to \mathbb{R}^2 \). One can show that

\[
\frac{d}{d\varepsilon}|_{\varepsilon=0} \text{length}(\gamma + \varepsilon\eta) = -\int_{0}^{L} \langle \eta(s), \kappa(s)N(s) \rangle \, ds
\]

• Therefore, the motion that most quickly decreases length is \( \eta = \kappa N \).

*Technical note: must go to zero at endpoints (i.e., pass through the origin).
Gradient of Length for a Line Segment

- This all becomes much easier in the discrete setting: just take the gradient of length with respect to vertex positions.

- First, a warm-up exercise. Suppose we have a single line segment:

\[ \ell := |b - a| \]

- How can we move the point \( b \) to most quickly increase its length?

\[ \nabla_b \ell = (b - a) / \ell \]
Gradient of Length for a Discrete Curve

• To find the motion that most quickly increases the total length $L$, we now just have to sum the contributions of each segment:

• Using some simple trigonometry, we can also express the length gradient in terms of the exterior angle $\theta_i$ and the angle bisector $N_i$:

$$\nabla_{\gamma_i} L = 2 \sin(\frac{\theta_i}{2}) N_i$$
Discrete Curvature (Length Variation)

• How does this help us define discrete curvature?

• Recall that in the smooth setting, the gradient of length is equal to the curvature times the normal.

• Hence, our expression for the discrete length variation provides a definition for the discrete curvature times the discrete normal.

$$\kappa_i^B N_i := 2 \sin(\theta_i/2) N_i$$
Let’s recap what we’ve done so far. We considered starting points that are **equivalent** in the smooth setting…

1. turning angle
2. length variation

…which led to two **inequivalent** definitions of curvature in the discrete setting:

1. \( \kappa^A_i := \theta_i \)
2. \( \kappa^B_i := 2 \sin(\theta_i/2) \)

For *small* angles, they agree. But in general, which one is “better”? And are there more possibilities? Let’s keep going…
Steiner Formula

- Steiner’s formula is closely related to our last approach: it says that if we move at a constant speed in the normal direction, then the change in length is proportional to curvature:

\[ \text{length}(\gamma + \varepsilon N) = \text{length}(\gamma) - \varepsilon \int_0^L \kappa(s) \, ds \]

- The intuition is the same as before: for a constant-distance normal offset, length will change in curved regions but not flat regions:
Discrete Normal Offsets

• How do we apply normal offsets in the discrete case?
• The first problem is that *normals* are not defined at vertices!
• We can at very least offset individual edges along their normals:

![Diagram showing offset of edges]

- $\varepsilon N$

• The question now is: how can we connect the normal-offset segments to get the final normal-offset curve?
Discrete Normal Offsets

- There are then several natural ways to connect segments:
  (A) along a circular arc of radius $\varepsilon$
  (B) along a straight line
  (C) extend edges until they intersect
- If we now compute the total length of the connected curves, we get (after some work…):

$$\text{length}_A = \text{length}(\gamma) - \varepsilon \sum_i \theta_i$$
$$\text{length}_B = \text{length}(\gamma) - \varepsilon \sum_i 2 \sin(\theta_i / 2)$$
$$\text{length}_C = \text{length}(\gamma) - \varepsilon \sum_i 2 \tan(\theta_i / 2)$$
Discrete Curvature (Steiner Formula)

- Since Steiner’s formula says that the change in length is proportional to curvature, we get yet another definition for curvature by comparing the original and normal-offset lengths.

In fact, we get three definitions: two we’ve seen before and one we haven’t:

\[
\begin{align*}
\kappa_i^A & := \theta_i \\
\kappa_i^B & := 2 \sin\left(\frac{\theta_i}{2}\right) \\
\kappa_i^C & := 2 \tan\left(\frac{\theta_i}{2}\right)
\end{align*}
\]
Osculating Circle

• One final idea is to consider the osculating circle, which is (roughly speaking) the circle that best approximates a curve at a given point $p$:

• More precisely, if we consider a circle passing through the point $p$ itself and two equidistant points $a$ and $b$ to the “left” and “right” (resp.), the osculating circle is the limit as $a$ and $b$ approach $p$.

• The curvature is then the reciprocal of the radius: $\kappa(p) = \frac{1}{r(p)}$
Discrete Curvature (Osculating Circle)

• A natural idea, then, is to consider the circumcircle of three consecutive points on a discrete curve (i.e., the circle touching all three points):

• We then get a fourth definition of discrete curvature:

\[
\kappa_i^D := \frac{1}{r_i} = \frac{2 \sin(\theta_i)}{w_i}
\]
A Tale of Four Curvatures

• Starting with four equivalent definitions of smooth curvature, we ended up with four inequivalent definitions for discrete curvature:

So… which one should we use?
Pick the Right Tool for the Job

• **Answer:** pick the right tool for the job!

• For a given application, which properties are most important to us? Which quantities do we want to preserve? How much computation are we willing to do? Etc.

• *E.g.*, in a physical simulation we might care about preserving energy but not momentum—or momentum, but not energy.

• Ok… but what about curvature?
Toy Example: Curvature Flow

• A simple example is *curvature flow*, where a closed curve moves in the normal direction with speed proportional to curvature:
  \[ \frac{d}{dt} \gamma(s, t) = \kappa(s, t)N(s, t) \]

• Shows up in many places (finding silhouettes in images, annealing in metals, closed geodesics on manifolds…)

• Some key properties:
  • (TOTAL) Total curvature remains constant throughout the flow.
  • (DRIFT) The center of mass does not drift from the origin.
  • (ROUND) Up to rescaling, the flow is stationary for circular curves.
Discrete Curvature Flow—No Free Lunch

- We can approximate curvature flow by repeatedly moving each vertex a little bit in the direction of the discrete curvature normal:
  \[ \gamma_i^{t+1} = \gamma_i^t + \tau k_i N_i \]

- But no choice of discrete curvature simultaneously captures all three properties of the smooth flow*:

*In fact, it’s impossible!
No Free Lunch—Other Examples

• Beyond this “toy” problem, the no free lunch scenario is quite common in discrete differential geometry.

• Many examples (e.g., Whitney-Graustein / Kirchoff analogy for curves; conservation of energy, momentum, and symplectic form for conservative time integrators; discrete Laplace operators...)

• At a more practical level: The Game played in DDG often leads to new & unexpected ways of formulating geometric algorithms. (E.g., faster, simpler, clearer guarantees, ...)

• Will see much more of this as the course continues!
Food for Thought: Convolution?

• Hot topic lately: convolutional neural nets (CNNs)
• Does DDG perspective help with discretization of convolution?
  • (Especially on geometric domains?)
• What properties are important to preserve?
  • **Basic ones:** (bi-)linearity, commutativity, associativity, distributivity, …
  • **Product rule:** \((f * g)' = f' * g = f * g'\)
  • **Convolution theorem:** \(F\{f * g\} = F\{f\} \cdot F\{g\}\)
• Differential geometry: *model kernel as Green’s function of differential operator*
First Reading Assignment

• Overview article from Notices of the AMS:

“A Glimpse into Discrete Differential Geometry”
Crane & Wardetzky (2017)
http://geometry.cs.cmu.edu/glimpse.pdf

• Written for broad mathematics audience

• Quite mathematical for non-math majors!

• Don’t sweat the details

• Try to get the high-level story

• Think of it like learning a foreign language…
DISCRETE DIFFERENTIAL GEOMETRY:
AN APPLIED INTRODUCTION
Keenan Crane • CMU 15-869(J) • Spring 2016

Thanks!