# Barycentric Coordinates



## Barycentric Coordinates

Interpolation

- given data at sites, interpolate smoothly and intuitively “in between”
- easy over simplices: linear
- more general shapes needed
  - morphing, shape deformation, attribute interpolation, physical modeling, and on and on and on



## Basic Principles

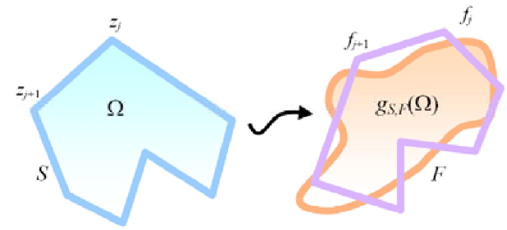
Data on boundary; extend

- as affine combination:

  if $p = \sum_{j=1}^{n} k_j(p) P_j$ then $f(p) = \sum_{j=1}^{n} k_j(p) f(P_j)$
- desirables
  - constant precision: $\sum k_j(x) = 1$
  - linear precision: $\sum k_j(x)\, x_j = x$
- convex: many choices; concave: few...



## Basic Setup: Cage




- from now on, we’ll focus on Weber/Ben-Chen/Gotsman’s method



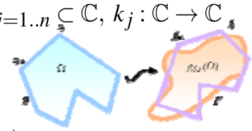
## Planar Case

Treat everything in complex plane

- tools from complex analysis...
- given source and target polygon

  $\{z_j\}_{j=1..n} \subset \mathbb{C}$, $\{f_j\}_{j=1..n} \subseteq \mathbb{C}$, $k_j : \mathbb{C} \to \mathbb{C}$

$g_{S,f}(z) = \sum k_j(z)\, f_j$



- desirables:

  $\sum k_j(z) = 1 \quad \sum k_j(z)\, z_j = z$



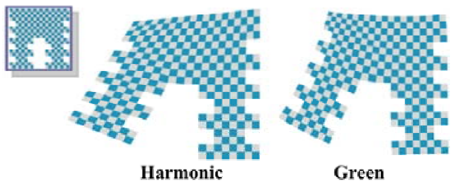
## Properties

Complex barycentric interpolation

- preserve similarities
- distinction with real coeffs?
  - preserve affine transformation iff

    $\sum \bar{k}_j(z)\, z_j = z$
  - not actually desirable... why?

## Real vs. Complex




- Affine transform not quite pleasing...



## Trade-off

Must give up something...

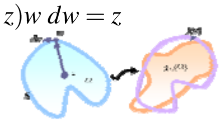
- not interpolating anymore

How to find such functions?

- study continuous setting

$$\oint_S k(w,z)\,dw = 1 \qquad \oint_S k(w,z)w\,dw = z$$

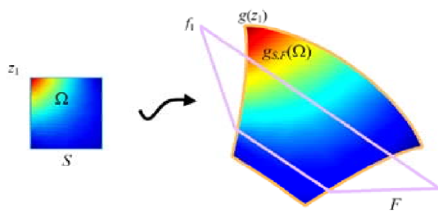
$$g_{S,f}(z) = \oint_S k(w,z)f(w)dw$$





## Not An Interpolation

Visualize one coordinate function






## Cauchy Formula

Holomorphic functions

- Cauchy kernel: $\mathbf{C(w,z)=\frac{1}{2\pi i}\frac{1}{w-z}}$

  $h(z) = \frac{1}{2\pi i}\oint_S C(w,z)h(w)dw$
- integral version of *mean value theorem*
  - recover value from average of boundary
- Cauchy coordinates: $\mathbf{g_{S,f}(z) = \frac{1}{2\pi i}\oint_S \frac{f(w)}{w-z}dw}$



## $g_{S,f}(z) = \frac{1}{2\pi i}\oint_S \frac{f(w)}{w-z}dw$

Only need data on boundary!

- apply to polygon

$$g_{S,f}(z) = \frac{1}{2\pi i}\sum \int_e \frac{f(w)}{w-z}dw$$

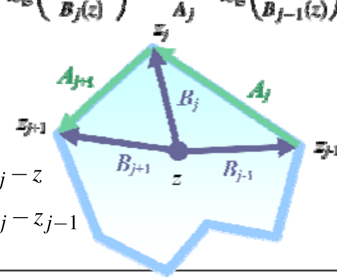
- define f linearly along edge
- grind out integrals...

$$g_{S,f}(z) = \sum C_j(z)f_j$$



## Resulting Formulas

$$C_j(z) = \frac{1}{2\pi i}\left(\frac{B_{j+1}(z)}{A_{j+1}}\log\left(\frac{B_{j+1}(z)}{B_j(z)}\right) - \frac{B_{j-1}(z)}{A_j}\log\left(\frac{B_j(z)}{B_{j-1}(z)}\right)\right)$$




$B_j(z) = z_j - z$

$A_j = z_j - z_{j-1}$

## Examples



## Properties

Best holomorphic function?

- it doesn't interpolate; is it "best"?
  - closest to given boundary data
- stick with $C_j$ but use "virtual" poly.

$$g_U(z) = \sum C_j(z) u_j$$

  - minimize functional to find best poly.

$$\min_U E_S(g_U)$$



## Szegö Coordinates

Optimize "fit"

$$E_S(g_U) = \oint_S |g_U(w) - f(w)|^2 \, dw$$

- need $C_j$ on boundary...
- define through limit
- do point collocation (sample boundary)

$$E_S(g_U) = |Cu - f_S|^2$$

matrix with sampled $C_j$ as columns

LSQ problem



## Solution

Pseudo inverse

$$u = C^+ f_S = (C^* C)^{-1} C^* f_S$$

- size is number of vertices (small)

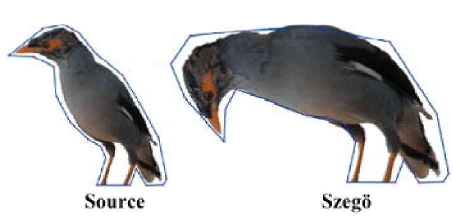
$$g_{S,f}^{\text{Szegö}}(z) = \sum G_j(z) f_j$$

- letting H be sampling operator:

$$f_S = HF \qquad u = C^+ f_S = MF \qquad G_j(z) = \sum_{j=1}^{j=n} C_i(z) M_{i,j}$$

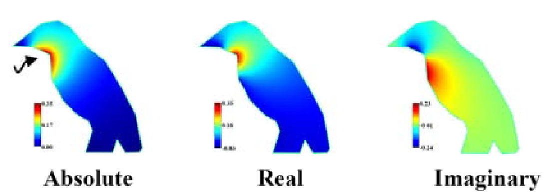


## Example

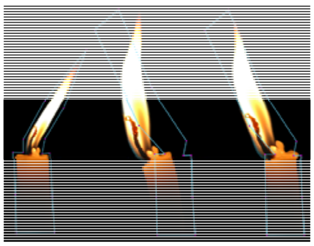





## Visualization



# Comparison



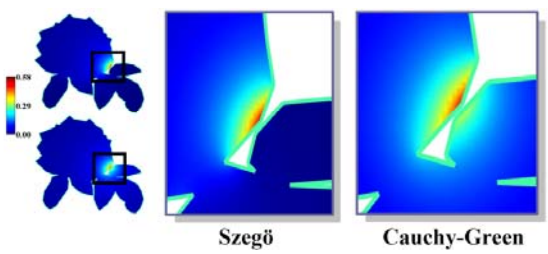


# Another Comparison

Cauchy-Green vs Szego



# Visualization






# Point-to-Point

Simplify UI

- just specify landmarks
- underconstrain (typically)
- add fairness constraint

$$E_S^{\text{smooth}}(g) = \oint_S |g''(w)|^2 dw$$

$$E_S^{\text{pts}}(g) = \sum_{i=1}^{i=p} |g(r_i) - f_i|^2$$



# P2P Coordinates

Joint minimization

$$E_S^{\text{P2P}}(g) = E_S^{\text{pts}}(g) + \lambda^2 E_S^{\text{smooth}}(g)$$

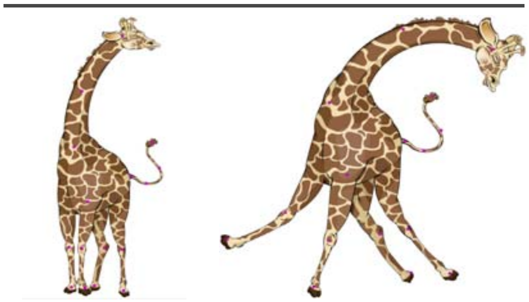
- point collocation...

$$E_S^{\text{P2P}}(g) = \|C_u - f_s\|^2 + \|\lambda D\|^2 \qquad N = \begin{pmatrix} C \\ \lambda D \end{pmatrix}^+_{..,1,\ldots,p}$$

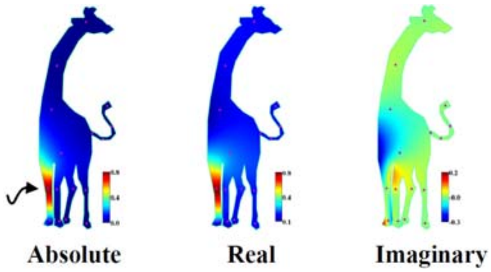
$$g_{S,F}^{\text{P2P}}(z) = \sum_{i=1}^{p} P_i(z) f_i \qquad P_i(z) = \sum_{j=1}^{n} C_j(z) N_{j,i}$$



# Example



# Visualization






# Example



# Video