Optimizing Content-Preserving Projections for Wide-Angle Images

Leonardo Koller Sacht
leo-ks@visgraf.impa.br

Visgraf - IMPA

September 16th, 2009

Introduction

- Motivation;
- Previous Approaches;
- Main article: “Optimizing Content-Preserving Projections for Wide-Angle Images”, R. Carroll, M. Agrawala, A. Agarwala, SIGGRAPH 2009, August 2009;
- Extension directions;

Motivations

- Common cameras capture just a limitted field of view of the scene, while our eyes capture a much wider field with no obvious distortion;
- Representation of the scene, extrapolation of our perception.

Motivations

- Digital cameras became very popular;
- Development of the stitching techniques, equipment and software;
- Internet photo sharing sites.

The Viewing Sphere

- The scene observed from a fixed view point can be modeled as the unit sphere ($S^2 \subseteq \mathbb{R}^3$) on which each direction has an associated color;
- A good representation of the viewing sphere is the longitude/latitude rectangle $[-\pi, \pi] \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$:

$$\mathbf{r} : [-\pi, \pi] \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \rightarrow S^2$$

$$(\lambda, \phi) \mapsto (cos(\lambda)cos(\phi), sin(\lambda)cos(\phi), sin(\phi)),$$

The Equirectangular Format

Figure: Lauzert, french village

6,909 pictures and 904 members on: http://www.flickr.com/groups/equirectangular/

The problem

- The panorama problem can be stated as finding a map

$$\mathbf{u} : S \subseteq S^2 \rightarrow I \subseteq \mathbb{R}^2$$

$$(\lambda, \phi) \mapsto (u, v),$$

with desirable properties.
(λ, ϕ) states for longitude/latitude coordinates and (u, v) states for cartesian coordinates;

Perspective Projection

- $S \subseteq S^2$ is projected onto a tangent plane through lines emanating from the center of the sphere:
- $$P : \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \to \mathbb{R}^2$$
$$(\lambda, \phi) \mapsto (u, v) = \left(tan(\lambda), \frac{tan(\phi)}{cos(\lambda)}\right).$$

Equirectangular Image

Figure: Posters

Perspective Projection

Figure: 45 degree perspective

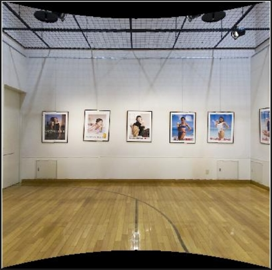


Figure: 60 degree perspective

Perspective Projection

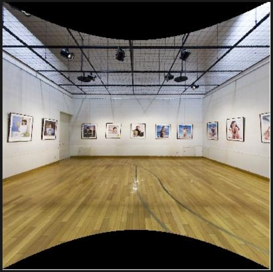


Figure: 90 degree perspective

Figure: 120 degree perspective

Stereographic Projection

- $S \subseteq S^2$ is projected onto a tangent plane through lines emanating from the pole oposite the point of tangency:
- $$S : [-\pi, \pi] \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \setminus \{(-\pi, 0)\} \to \mathbb{R}^2$$
$$(\lambda, \phi) \mapsto (u, v) = \left(\frac{2sin(\lambda)cos(\phi)}{cos(\lambda)cos(\phi) + 1}, \frac{2sin(\phi)}{cos(\lambda)cos(\phi) + 1} \right).$$

Stereographic Projection

Figure: 45 degrees

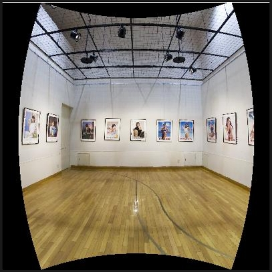


Figure: 90 degrees

Stereographic Projection

Figure: 135 degrees

Figure: 180 degrees

Mercator Projection

- Cylindrical projection (Snyder, p.37) designed to maintain conformality:
- $$M : [-\pi, \pi] \times \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \to \mathbb{R}^2$$
  $$(\lambda, \phi) \mapsto (u, v) = (\lambda, log(sec(\phi) + tan(\phi)))\,.$$

Mercator Projection

Figure: 90 × 45 degree

Figure: 180 × 90 degree

Figure: 270 × 135 degree

Article 1, 1995

- D. Zorin, A.H. Barr: *Correction of geometric perceptual distortion in pictures*, SIGGRAPH 95.


Figure: 92 degree photograph and its correction

Article 1, 1995

- **Structural features** (based on retinal projections)
  • The image of a surface should not be a point;
  • The image of a line segment shouldn't have loops. If it does it should be a point;
  • The image of a plane shouldn't have *twists* in it.

Article 1, 1995

- **Structural features** (based on retinal projections)
  • The image of a surface should not be a point;
  • The image of a line segment shouldn't have loops. If it does it should be a point;
  • The image of a plane shouldn't have *twists* in it.
- **Desirable properties**
  • Zero-curvature condition: Images of straight lines should be straight;
  • Direct view condition: Objects should look as they appear in the middle of a pespective;

Theoretical Results

- Any viewing transformation that satisfies the structural conditions can be decomposed as a perspective transformation followed by a transformation of the image plane;
- No viewing tranformation satisfies simultaneously both desirable conditions;
- The authors propose error functions to deal with them.

Transformation of the image plane

$$S : I_1 \longrightarrow I_2$$

$$(r, \theta) \mapsto (\rho, \psi),$$

where

$$\rho = \lambda(\theta)\frac{r}{R} + (1 - \lambda(\theta))\frac{R(\sqrt{r^2+1}-1)}{r(\sqrt{R^2+1}-1)}$$

and

$$\psi = \theta.$$

Results

Figure: $\lambda = 1$, FOV $= 135$

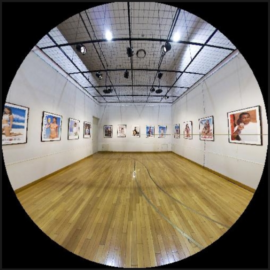


Figure: $\lambda = 0.5$, FOV $= 135$

Results

Figure: $\lambda = 0$, FOV $= 135$

Figure: Stereographic, FOV $=$ 135

Results

Figure: $\lambda(\theta) = \frac{cos(\theta)+2}{4}$, FOV = 135

Article 2, 2000

- M. Agrawala, D. Zorin, T. Munzner: *Artistic Multiprojection Rendering*, Eurographics Rendering Workshop 2000.

Figure: Multiprojection Still Life

Article 2, 2000

- The authors propose to use more than one perspective (coming from different view points) on the same image;
- A *master camera* is used for the scene, while the perspective of each object is changed to the perspective of a *local camera*.
- Approach created for artistic purposes (based on surrealist paintings), but may be applied to reduce distortions and improve representation of the objects.

S3D Project

Construction of multiperspective images from spherical panoramas:

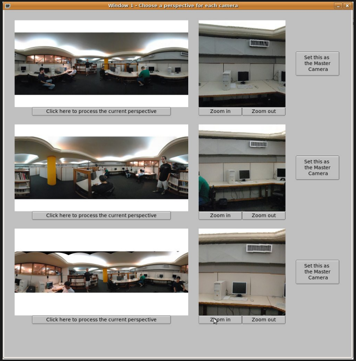


Figure: First window

Results

Figure: 90 degree perspective

Figure: Me with a local camera

For more: http://www.impa.br/ leo-ks/s3d

Article 3, 2005

- L. Zelnik-Manor, G. Peters, P. Perona: *Squaring the Circle in Panoramas*, ICCV 2005.

Figure: Multiplane Multiview projection

Article 3, 2005

- Multiple tangent planes to the sphere generating multiple perspective projections (from a *single* view point now);
- The planes intersect on least noticible regions;
- The distortions of the objects (split because of the seam or "naturally" distorded) are corrected using perspective projections centered on them.

Image Processing Project

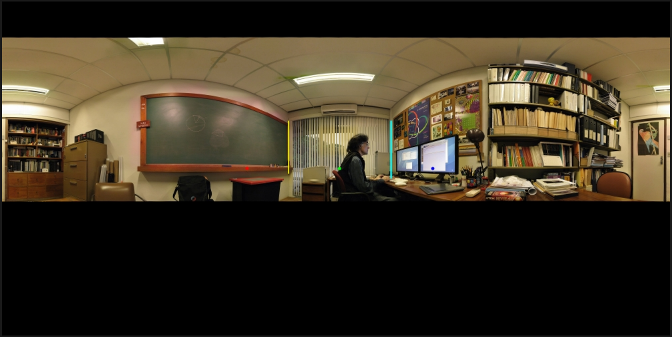


Figure: The user specifies walls and centers of projection

Results

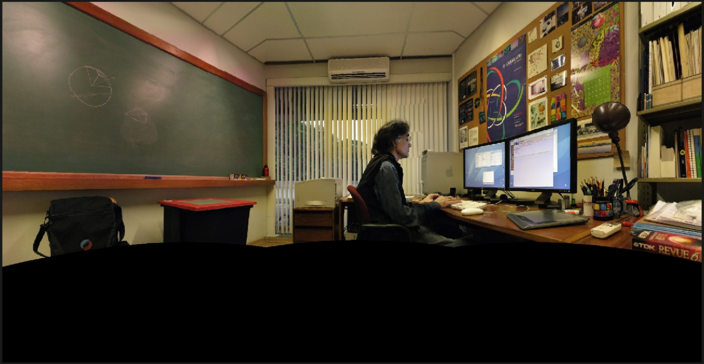


Figure: 200 degree panorama

Results

For more: http://www.impa.br/ leo-ks/image_processing

Summary

Based on previous approaches, we can conclude that a panorama should (as much as posible):

- Preserve the shape of the objects in the scene;

Summary

Based on previous approaches, we can conclude that a panorama should (as much as posible):

- Preserve the shape of the objects in the scene;
- Map straight lines in the scene to straight lines in the final result;

Summary

Based on previous approaches, we can conclude that a panorama should (as much as posible):

- Preserve the shape of the objects in the scene;
- Map straight lines in the scene to straight lines in the final result;
- Respect the structural conditions in order to imitate retinal projections;

Summary

Based on previous approaches, we can conclude that a panorama should (as much as posible):

- Preserve the shape of the objects in the scene;
- Map straight lines in the scene to straight lines in the final result;
- Respect the structural conditions in order to imitate retinal projections;
- Have orientation constancy;

Summary

Based on previous approaches, we can conclude that a panorama should (as much as posible):

- Preserve the shape of the objects in the scene;
- Map straight lines in the scene to straight lines in the final result;
- Respect the structural conditions in order to imitate retinal projections;
- Have orientation constancy;
- Vary scale and orientation smoothly;

Summary

Based on previous approaches, we can conclude that a panorama should (as much as posible):

- Preserve the shape of the objects in the scene;
- Map straight lines in the scene to straight lines in the final result;
- Respect the structural conditions in order to imitate retinal projections;
- Have orientation constancy;
- Vary scale and orientation smoothly;
- Not depend on some particular structure of the scene;

Main article, 2009

- R. Carroll, M. Agrawala, A. Agarwala: *Optimizing Content-Preserving Projections for Wide-Angle Images*, SIGGRAPH 2009.

Figure: A 270 × 135 degree result

Interface

Demonstration video:
http://vis.berkeley.edu/papers/capp/projection-sig09.mov

Discretization

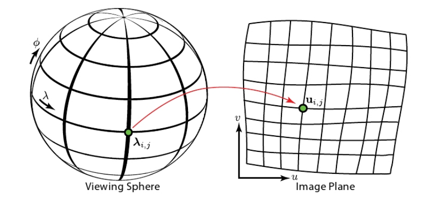


Figure: We look for a map from the equirectangular discretized domain $\tilde{\lambda}_{i,j} = (\lambda_{i,j}, \phi_{i,j})$ to the plane where the corresponding points are $\mathbf{u}_{i,j} = (u_{i,j}, v_{i,j})$.

- V = set of all vertices (i, j) that fall in the specified field of view.

Shape of the objects

- Objects as if they were looked directly;

Shape of the objects

- Objects as if they were looked directly;
- $\mathbf{u} : S^2 \rightarrow \mathbb{R}^2$ should behave locally as a rotation followed by a homothety of the tangent plane (a *conformal* mapping);

Shape of the objects

- Objects as if they were looked directly;
- $\mathbf{u} : S^2 \to \mathbb{R}^2$ should behave locally as a rotation followed by a homothety of the tangent plane (a *conformal* mapping);
- Consider the differential north and east vectors (image of an orthonormal basis of the tangent plane by derivative of $\mathbf{u}$):

$$\mathbf{h} = \begin{pmatrix} \frac{\partial u}{\partial \phi}(p) \\ \frac{\partial v}{\partial \phi}(p) \end{pmatrix}, \mathbf{k} = \begin{pmatrix} \frac{1}{cos(\phi)} \frac{\partial u}{\partial \lambda}(p) \\ \frac{1}{cos(\phi)} \frac{\partial v}{\partial \lambda}(p) \end{pmatrix}.$$

Cauchy Riemann Equations

- It's enough that

$$\mathbf{h} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \mathbf{k},$$

which leads to the CR equations:

$$\frac{\partial u}{\partial \phi}(p) = -\frac{\partial v}{\partial \lambda}(p)\frac{1}{cos(\phi)}, \frac{\partial v}{\partial \phi}(p) = \frac{\partial u}{\partial \lambda}(p)\frac{1}{cos(\phi)}.$$

Cauchy Riemann Equations

- It's enough that

$$\mathbf{h} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \mathbf{k},$$

which leads to the CR equations:

$$\frac{\partial u}{\partial \phi}(p) = -\frac{\partial v}{\partial \lambda}(p)\frac{1}{cos(\phi)}, \frac{\partial v}{\partial \phi}(p) = \frac{\partial u}{\partial \lambda}(p)\frac{1}{cos(\phi)}.$$

- For a detailed approach to obtain these equations, check the Appendix of this presentation.

Examples

- The stereographic projection is a conformal mapping;
- Setting $u(\lambda, \phi) = \lambda$, we have $\dfrac{\partial u}{\partial \lambda} = 1$ and due to second CR equation:

$$\frac{\partial v}{\partial \phi} = \frac{1}{cos(\phi)} \Rightarrow v(\lambda, \phi) = log(sec(\phi) + tan(\phi)),$$

which is the Mercator projection.

Discretization

Using simple forward differences on each quad:

$$u_{i,j+1} - u_{i,j} = -\frac{v_{i+1,j} - v_{i,j}}{cos(\phi_{i,j})}, v_{i,j+1} - v_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{cos(\phi_{i,j})}$$

$$\Rightarrow u_{i,j+1} - u_{i,j} + \frac{v_{i+1,j} - v_{i,j}}{cos(\phi_{i,j})} = 0, \frac{u_{i+1,j} - u_{i,j}}{cos(\phi_{i,j})} - (v_{i,j+1} - v_{i,j}) = 0$$

Multiplying by $cos(\phi_{i,j})$, we get

$$(v_{i+1,j} - v_{i,j}) + cos(\phi_{i,j})(u_{i,j+1} - u_{i,j}) = 0,$$

$$(u_{i+1,j} - u_{i,j}) - cos(\phi_{i,j})(v_{i,j+1} - v_{i,j}) = 0.$$

Conformality Energy

$$E_C = \sum_{(i,j)\in V} w_{i,j}^2 \left((v_{i+1,j} - v_{i,j}) + cos(\phi_{i,j})(u_{i,j+1} - u_{i,j})\right)^2 +$$

$$+ \sum_{(i,j)\in V} w_{i,j}^2 \left((u_{i+1,j} - u_{i,j}) - cos(\phi_{i,j})(v_{i,j+1} - v_{i,j})\right)^2 ,$$

or

$$E_C = \| C\mathbf{x} \|^2,$$

where C will be called *Conformality Matrix*.

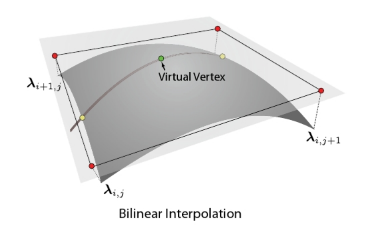
Straight Lines

Optimizing Content-Preserving Projections for Wide-Angle Images

Introduction
Motivation
Previous Approaches
Main article
Results
Extension Ideas
Appendix

Straight Lines

- $L = \{\text{lines marked by the user}\}$;
- $L \supseteq L_f = \{\text{fixed orientation lines}\}$;
- Let $l \in L$ and consider $V_l = \{(i,j)\}$ the set of all quads intersected by l.



Straight Lines

- For each $(i, j) \in V_l$ define a virtual vertex $\Lambda_{i,j}$ at the midpoint of the line-quad intersection;
- Project the corresponding quad vertices $(\tilde{\lambda}_{i,j}, \tilde{\lambda}_{i+1,j}, \tilde{\lambda}_{i+1,j+1}, \tilde{\lambda}_{i,j+1})$ on the tangent plane through $\Lambda_{i,j}$ and obtain their bilinear interpolation coefficients (a, b, c, d);
- Define a virtual output vertex $\mathbf{u}^l_{i,j} = a\mathbf{u}_{i,j} + b\mathbf{u}_{i+1,j} + c\mathbf{u}_{i+1,j+1} + d\mathbf{u}_{i,j+1}$.
- Define other two output virtual vertices $\mathbf{u}_{start}$ and $\mathbf{u}_{end}$.

Straight Lines

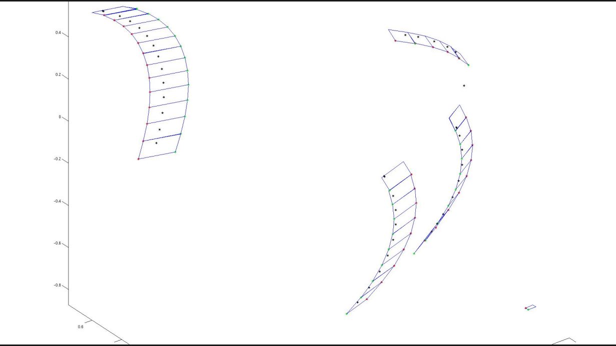


Figure: Projected Quads

Line Energy

- We want to minimize the distance of the $\mathbf{u}_{i,j}$'s to the line connecting $\mathbf{u}_{start}$ to $\mathbf{u}_{end}$;
  The distance of a point $\mathbf{u}$ to this line is

$$(\mathbf{u} - \mathbf{u}_{start})^T \mathbf{n}(\mathbf{u}_{start}, \mathbf{u}_{end}),$$

where

$$\mathbf{n}(\mathbf{u}_{start}, \mathbf{u}_{end}) = R_{90} \frac{\mathbf{u}_{end} - \mathbf{u}_{start}}{\| \mathbf{u}_{end} - \mathbf{u}_{start} \|}.$$

Line Energy

- We want to minimize the distance of the $\mathbf{u}_{i,j}$'s to the line connecting $\mathbf{u}_{start}$ to $\mathbf{u}_{end}$;
  The distance of a point $\mathbf{u}$ to this line is

$$(\mathbf{u} - \mathbf{u}_{start})^T \mathbf{n}(\mathbf{u}_{start}, \mathbf{u}_{end}),$$

  where

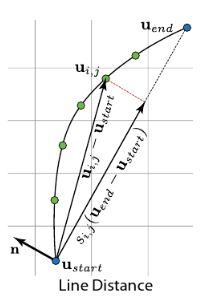
$$\mathbf{n}(\mathbf{u}_{start}, \mathbf{u}_{end}) = R_{90} \frac{\mathbf{u}_{end} - \mathbf{u}_{start}}{\| \mathbf{u}_{end} - \mathbf{u}_{start} \|}.$$

- Line energy for line l:

$$E_l = \sum_{(i,j) \in V_l} \left((\mathbf{u}_{i,j} - \mathbf{u}_{start})^T \mathbf{n}(\mathbf{u}_{start}, \mathbf{u}_{end}) \right)^2 .$$

Avoiding Nonlinearity

- The distance of a point $\mathbf{u}$ can also be expressed as the difference of $(\mathbf{u} - \mathbf{u}_{start})$ and its projection onto the unit vector on the direction of $(\mathbf{u}_{end} - \mathbf{u}_{start})$:



Avoiding Nonlinearity

- Which leads to another expression for E_l:

$$E_l = \sum_{(i,j)\in V_l} \| (\mathbf{u}_{i,j} - \mathbf{u}_{start}) - s(\mathbf{u}_{i,j}, \mathbf{u}_{start}, \mathbf{u}_{end})(\mathbf{u}_{end} - \mathbf{u}_{start}) \|^2$$

where

$$s(\mathbf{u}_{i,j}, \mathbf{u}_{start}, \mathbf{u}_{end}) = \frac{(\mathbf{u} - \mathbf{u}_{start})^T(\mathbf{u}_{end} - \mathbf{u}_{start})}{\| \mathbf{u}_{end} - \mathbf{u}_{start} \|^2}.$$

Avoiding Nonlinearity

- Now we have two ways to minimize E_l:
- Setting the line directions and minimizing

$$E_{lo} = \sum_{(i,j) \in V_l} \left((\mathbf{u}_{i,j} - \mathbf{u}_{start})^T \mathbf{n} \right)^2 ,$$

or

- Setting the projection coefficients $s_{i,j}$ and minimizing

$$E_{ld} = \sum_{(i,j) \in V_l} \| (\mathbf{u}_{i,j} - \mathbf{u}_{start}) - s_{i,j}(\mathbf{u}_{end} - \mathbf{u}_{start}) \|^2,$$

- Observe that $E_{lo} = \| (LO)\mathbf{x} \|^2$ and $E_{ld} = \| (LD)\mathbf{x} \|^2$.

Line energy minimization

- Initialize $s_{i,j}$ as being the arc length between $\lambda_{i,j}$ and λ_{start} on the sphere and minimize E_{ld}.
- With the new values for $\mathbf{u}$ calculate the normals for each line:

$$\mathbf{n}(\mathbf{u}_{start}, \mathbf{u}_{end}) = R_{90} \frac{\mathbf{u}_{end} - \mathbf{u}_{start}}{\| \mathbf{u}_{end} - \mathbf{u}_{start} \|}$$

  and minimize E_{lo}.
- With the new values for $\mathbf{u}$ calculate

$$s_{i,j} = \frac{(\mathbf{u}_{i,j} - \mathbf{u}_{start})^T (\mathbf{u}_{end} - \mathbf{u}_{start})}{\| \mathbf{u}_{end} - \mathbf{u}_{start} \|^2}$$

  and minimize E_{ld};
- Repeat the process until its convergence.

Smoothness of the projection

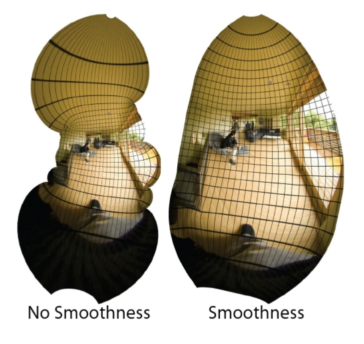


Figure: Smoothness correction

Smoothness

- We ask that the differential north vector $\mathbf{h}$ does not change too much across the projection;

Smoothness

- We ask that the differential north vector **h** does not change too much across the projection;
- In a least squares sense, we want that

$$\frac{\partial \mathbf{h}}{\partial(\lambda,\phi)} = \begin{pmatrix} \frac{\partial^2 u}{\partial\phi\partial\lambda} & \frac{\partial^2 u}{\partial\phi^2} \\ \frac{\partial^2 v}{\partial\phi\partial\lambda} & \frac{\partial^2 v}{\partial\phi^2} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

Smoothness Energy

- Approximating second derivatives by finite differences, weighting again by $cos(\phi_{i,j})$ and $w_{i,j}$ and using the forbenius matrix norm, we get the *Smoothness Energy*:

$$E_s = \sum_{(i,j)\in V} w_{i,j}^2 cos^2 \phi_{i,j} \left| \begin{matrix} u_{i,j+1} - 2u_{i,j} + u_{i,j-1} \\ v_{i,j+1} - 2v_{i,j} + v_{i,j-1} \\ u_{i+1,j+1} - u_{i+1,j} - u_{i,j+1} + u_{i,j} \\ v_{i+1,j+1} - v_{i+1,j} - v_{i,j+1} + v_{i,j} \end{matrix} \right|^2$$

- $E_s = \| S\mathbf{x} \|^2$, where S will be called *smoothness matrix*.

Spatially-varying weighting

- For each vertex we associate weigths that vary the strength of conformality and smoothness constraints;

Spatially-varying weighting

- For each vertex we associate weigths that vary the strength of conformality and smoothness constraints;
- **Line endpoint weights:** $w^L_{i,j}$ are computed to be higher near line endpoints. For each line endpoint, a gaussian centered on the endpoint is computed and for each vertex $w^L_{i,j}$ is the sum over all gaussians that are positive at (i, j);

Spatially-varying weighting

- For each vertex we associate weigths that vary the strength of conformality and smoothness constraints;
- **Line endpoint weights:** $w^L_{i,j}$ are computed to be higher near line endpoints. For each line endpoint, a gaussian centered on the endpoint is computed and for each vertex $w^L_{i,j}$ is the sum over all gaussians that are positive at (i,j);
- **Salience weights:** $w^S_{i,j}$ are set to be higher in regions with greater color variation;

Spatially-varying weighting

- **Face weights:** $w_{i,j}^F$ are computed to be higher near faces. The Mercator projection is applied to the input image and faces are detected on the Mercator image using Viola and Jones face detector. Then a gaussian is calculated on the Mercator image and its values are warped back to the equirectangular grid;

Spatially-varying weighting

- **Face weights:** $w^F_{i,j}$ are computed to be higher near faces. The Mercator projection is applied to the input image and faces are detected on the Mercator image using Viola and Jones face detector. Then a gaussian is calculated on the Mercator image and its values are warped back to the equirectangular grid;
- **Total weight:**

$$w_{i,j} = 2w^L_{i,j} + 2w^S_{i,j} + 4w^F_{i,j} + 1.$$

Total energy and optimization

- **Total energy:**

$$E = w_c^2 E_c + w_s^2 E_s + w_l^2 \left(\sum_{l \in L \setminus L_f} E_l + \sum_{l \in L_f} E_{lo} \right).$$

Total energy and optimization

- **Total energy:**

$$E = w_c^2 E_c + w_s^2 E_s + w_l^2 \left(\sum_{l \in L \setminus L_f} E_l + \sum_{l \in L_f} E_{lo} \right).$$

- We alternate between minimizing

$$E_d = w_c^2 E_c + w_s^2 E_s + w_l^2 \left(\sum_{l \in L \setminus L_f} E_{ld} + \sum_{l \in L_f} E_{lo} \right),$$

and

$$E_o = w_c^2 E_c + w_s^2 E_s + w_l^2 \left(\sum_{l \in L} E_{lo} \right).$$

Total energy and optimization

- The authors use $w_c = 1$, $w_s = 12$ and $w_l = 1000$ for all results.
- Both energy terms are sums of linear squares: $\| A\mathbf{x} \|^2$. For example,

$$E_o = \| A\mathbf{x} \|^2,$$

where

$$A = \begin{pmatrix} w_c C \\ w_s S \\ w_l LO \end{pmatrix},$$

where C, S and LO are the conformality, smoothness and fixed orientation line matrices.

Nontrivial solutions for (min $\| A\mathbf{x} \|^2$)

- We look for a minimizer $\mathbf{x}$ s.t. $\| \mathbf{x} \|^2 = 1$;
  $E(\mathbf{x}) = \| A\mathbf{x} \|^2 = \mathbf{x}^T A^T A\mathbf{x}$;

Nontrivial solutions for $(\min \| A\mathbf{x} \|^2)$

- We look for a minimizer $\mathbf{x}$ s.t. $\| \mathbf{x} \|^2 = 1$; $E(\mathbf{x}) = \| A\mathbf{x} \|^2 = \mathbf{x}^T A^T A\mathbf{x}$;
- $A^T A$ is symmetric positive semidefinite what implies that it has a orthonormal basis of eigenvectors $\mathbf{e}_i$ $(i = 1, \ldots, q)$ with eigenvalues $0 \leqslant \lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_q$.

Nontrivial solutions for $(\min \| A\mathbf{x} \|^2)$

- We look for a minimizer $\mathbf{x}$ s.t. $\| \mathbf{x} \|^2 = 1$; $E(\mathbf{x}) = \| A\mathbf{x} \|^2 = \mathbf{x}^T A^T A \mathbf{x}$;
- $A^T A$ is symmetric positive semidefinite what implies that it has a orthonormal basis of eigenvectors $\mathbf{e}_i$ $(i = 1, \dots, q)$ with eigenvalues $0 \leqslant \lambda_1 \leqslant \lambda_2 \leqslant \dots \leqslant \lambda_q$.
- Thus any unit vector can be written as

$$\mathbf{x} = \mu_1 \mathbf{e}_1 + \dots + \mu_q \mathbf{e}_q,$$

where $\mu_1^2 + \dots + \mu_q^2 = 1$.

Nontrivial solutions for (min $\| A\mathbf{x} \|^2$)

- Thus

$$E(\mathbf{x}) - E(\mathbf{e}_1) = \mathbf{x}^T(A^TA)\mathbf{x} - \mathbf{e}_1^T(A^TA)\mathbf{e}_1 =$$

$$= \lambda_1^2\mu_1^2 + \ldots + \lambda_q^2\mu_q^2 - \lambda_1^2 \geqslant \lambda_1^2(\mu_1^2 + \ldots + \mu_q^2 - 1) = 0.$$

- The eigenvector $\mathbf{e}_1$ associated to the minimum eigenvalue λ_1 is the minimizer and $\min E = \min_{\|x\|^2=1} \| Ax \|^2 = \lambda_1^2$.

Results

- The results are obtained by making bilinear interpolation of the projected vertices;
- Many results and comparisons at: http://vis.berkeley.edu/papers/capp;
- The results seem like conformal projections (Mercator and Stereographic), but with lines straightened;
- User marks from 5 to 40 lines, with an average of 20;
- The images that came from 180^o fisheye lens were computed with about 80,000 vertices on the viewing sphere and took approximately 1 minute;

What I did until now

- Simple interface to mark lines (and their orientation) on equirectangular images;

What I did until now

Matlab implementations of:

- Conformality matrix construction, depending on the size of the mesh and field of view;
- Smoothnes matrix construction, depending on the size of the mesh and field of view;
- Fixed line orientation matrix construction, depending on the size of the mesh, field of view and list of points marked on the equirectangular image (output from previous interface);
- Energy minimization using just vertical and horizontal line constraints.

Result: 180×120 degree

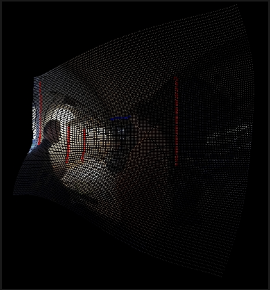


Figure: $w_c = 1$, $\mathbf{w}_s = 2.5$ and $w_l = 0.5$

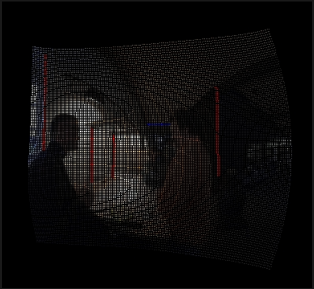


Figure: $w_c = 1$, $\mathbf{w}_s = 6$ and $w_l = 0.5$

Result: 180×120 degree

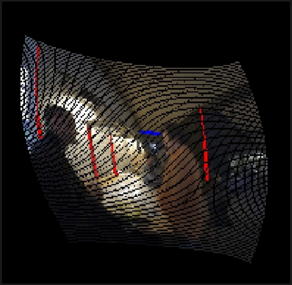


Figure: $w_c = 1$, $w_s = 1$ and $\mathbf{w}_l = 0.02$

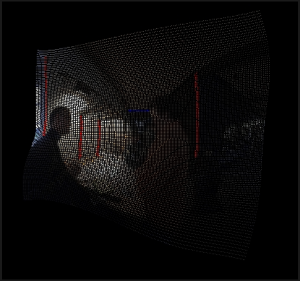


Figure: $w_c = 1$, $w_s = 1$ and $\mathbf{w}_l = 0.2$

Extension Ideas

- Automatic detection of lines (straight lines in panoramas are not always intuitive to find);
- Use orientations of the lines from some known mapping (perspective or Zelnik-Manor's);
- Treat other objects that are not faces (include in the interface or detect automatically);
- Use of gigapixel images to obtain high-quality results (GigaPan);

Panoramic Videos

- Lady Bug;
- Transition from one frame to another has to be smooth;
- Calculate the mapping for new frames for just some points (which leads to min $\| A\mathbf{x} - \mathbf{b} \|^2$);
- Start with simpler cases and combine these cases to obtain a global approach;
- The development of these ideas can result in new possibilities of filming;

Thanks!

References

[1] Agrawala, M., Zorin D., and Munzer, T. 2000. *Artistic Multiprojection Rendering*. In Eurographics Rendering Workshop 2000.
[2] Carrol, R., Agrawala M., and Agarwala, A. 2009. *Optimizing Content-Preserving Projections for Wide-Angle Images*. In Proc. SIGGRAPH 2009, 43:1-43:9.
[3] Do Carmo, M. P. 2005. *Geometria Diferencial de Curvas e Superficies*. Sociedade Brasileira de Matemtica.

References

[4] Snyder, J. P. 1987. *Map projections - A Working Manual: U.S. Geological Survey Prof. Paper 1395*. U.S. Gov. Printing Office.
[5] Zelnik-Manor, L., Peters, G., and Perona, P. 2005. *Squaring the Circle in panoramas*. In IEEE International Conference in Computer Vision (ICCV), vol. 2, 1292-1299.
[6] Zorin, D., and Barr, A.H. 1995. *Correction of geometric perceptual distortion in pictures*. In Proc. SIGGRAPH, 257-264.

Conformal Mappings

- Given $p = (\lambda, \phi) \in S^2$, $\left\{ \frac{\partial \mathbf{r}}{\partial \lambda}(p), \frac{\partial \mathbf{r}}{\partial \phi}(p) \right\}$ is the basis of T_pS^2 associated to $\mathbf{r}$, where $\mathbf{r}$ is the longitude/latitude parametrization of the sphere;

Conformal Mappings

- Given $p = (\lambda, \phi) \in S^2$, $\left\{ \frac{\partial \mathbf{r}}{\partial \lambda}(p), \frac{\partial \mathbf{r}}{\partial \phi}(p) \right\}$ is the basis of T_pS^2 associated to $\mathbf{r}$, where $\mathbf{r}$ is the longitude/latitude parametrization of the sphere;
- $\frac{\partial \mathbf{r}}{\partial \lambda}(p) = (-sin(\lambda)cos(\phi), cos(\lambda)cos(\phi), 0)^T$ and $\frac{\partial \mathbf{r}}{\partial \phi}(p) = (-cos(\lambda)sin(\phi), -sin(\lambda)sin(\phi), cos(\phi))^T$ are orthogonal;

Conformal Mappings

- **Definition:** A dipheomorphism $\mathbf{u} : S \to \overline{S}$ is a *conformal mapping* if $\forall p \in S$ and $\forall v_1, v_2 \in T_pS$ holds

$$< d\mathbf{u}_p(v_1), d\mathbf{u}_p(v_2) >_{\mathbf{u}(p)} = \theta^2(p) < v_1, v_2 >_p,$$

where θ^2 is a differentiable function that never vanishes on S.

Conformal Mappings

- **Definition:** A dipheomorphism $\mathbf{u} : S \to \overline{S}$ is a *conformal mapping* if $\forall p \in S$ and $\forall v_1, v_2 \in T_pS$ holds

$$< d\mathbf{u}_p(v_1), d\mathbf{u}_p(v_2) >_{\mathbf{u}(p)} = \theta^2(p) < v_1, v_2 >_p,$$

  where θ^2 is a differentiable function that never vanishes on S.
- In our case, $S \subseteq S^2$ and $\overline{S} \subseteq \mathbb{R}^2$.

Conformal Mappings

- **Definition:** A dipheomorphism $\mathbf{u} : S \to \overline{S}$ is a *conformal mapping* if $\forall p \in S$ and $\forall v_1, v_2 \in T_pS$ holds

$$< d\mathbf{u}_p(v_1), d\mathbf{u}_p(v_2) >_{\mathbf{u}(p)} = \theta^2(p) < v_1, v_2 >_p,$$

  where θ^2 is a differentiable function that never vanishes on S.
- In our case, $S \subseteq S^2$ and $\overline{S} \subseteq \mathbb{R}^2$.
- Conformal mappings preserve angles (Do Carmo, p.270).

Conformal Mappings

Consider the unit vectors

$$\mathbf{h} = \frac{\partial \mathbf{r}}{\partial \phi}(p) = (0,1)_{\{\frac{\partial \mathbf{r}}{\partial \lambda}(p), \frac{\partial \mathbf{r}}{\partial \phi}(p)\}}$$

(differential north vector) and

$$\mathbf{k} = \frac{1}{cos(\phi)}\frac{\partial \mathbf{r}}{\partial \lambda}(p) = \left(\frac{1}{cos(\phi)}, 0\right)_{\{\frac{\partial \mathbf{r}}{\partial \lambda}(p), \frac{\partial \mathbf{r}}{\partial \phi}(p)\}}$$

(differential east vector).

Conformal Mappings

- **Obs.:** $d\mathbf{u}_p(w) = \begin{pmatrix} \frac{\partial u}{\partial \lambda}(p) & \frac{\partial u}{\partial \phi}(p) \\ \frac{\partial v}{\partial \lambda}(p) & \frac{\partial v}{\partial \phi}(p) \end{pmatrix} w_{\{\frac{\partial \mathbf{r}}{\partial \lambda}(p), \frac{\partial \mathbf{r}}{\partial \phi}(p)\}}$.

Conformal Mappings

- **Obs.:** $d\mathbf{u}_p(w) = \begin{pmatrix} \frac{\partial u}{\partial \lambda}(p) & \frac{\partial u}{\partial \phi}(p) \\ \frac{\partial v}{\partial \lambda}(p) & \frac{\partial v}{\partial \phi}(p) \end{pmatrix} w_{\{\frac{\partial \mathbf{r}}{\partial \lambda}(p), \frac{\partial \mathbf{r}}{\partial \phi}(p)\}}$.
- It's enough that $d\mathbf{u}_p(\mathbf{h}) = R_{90} d\mathbf{u}_p(\mathbf{k})$:

$$\begin{pmatrix} \dfrac{\partial u}{\partial \phi}(p) \\ \dfrac{\partial v}{\partial \phi}(p) \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \dfrac{1}{cos(\phi)} \dfrac{\partial u}{\partial \lambda}(p) \\ \dfrac{1}{cos(\phi)} \dfrac{\partial v}{\partial \lambda}(p) \end{pmatrix}$$

Cauchy Riemann Equations

Spherical version:

$$\frac{\partial u}{\partial \phi}(p) = -\frac{\partial v}{\partial \lambda}(p)\frac{1}{cos(\phi)}$$

and

$$\frac{\partial v}{\partial \phi}(p) = \frac{\partial u}{\partial \lambda}(p)\frac{1}{cos(\phi)}.$$