Measuring Deformations and Illumination Changes in Images with Applications to Face Recognition

Ph.D. Defense

Anne Jorstad

Applied Mathematics & Statistics, and Scientific Computation
University of Maryland

Advisor: Professor David Jacobs, Department of Computer Science

July 16, 2012

The Deformation and Illumination Variation Problem

Current Methods for Handling Expression Variation

- Option 1: Direct Pixel Comparison
  (eg Principal Component Analysis)
  - Ignore pixels that change with expression,
    hope there are enough consistent pixels to determine identity

  - Ignore bottom half of face?

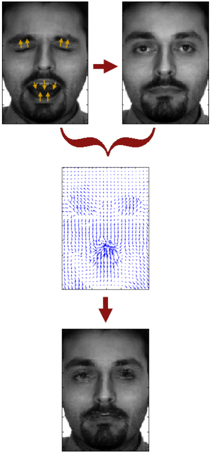
Current Methods for Handling Expression Variation

- Option 2: Match feature points, warp image
  (eg Active Appearance Models)
  - Find corresponding facial feature points, warp points on unknown face to align with their positions on a known face

- Understand expression variation:
  determine correspondences between every pixel in image pair

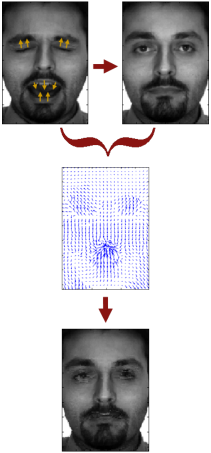
Using Dense Correspondences

- Design cost function to match pixels



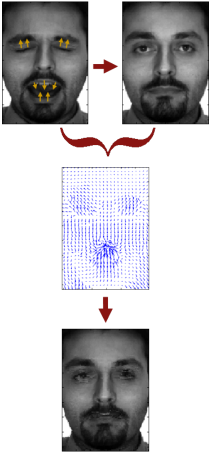
Using Dense Correspondences

- Design cost function to match pixels
- Search for lowest cost correspondences



Using Dense Correspondences

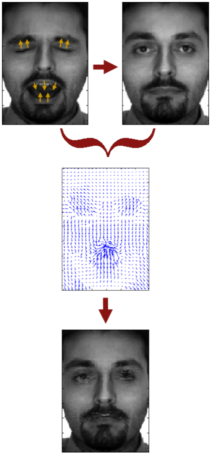
- Design cost function to match pixels
- Search for lowest cost correspondences
- Sum individual pixel correspondence costs for image pair matching cost



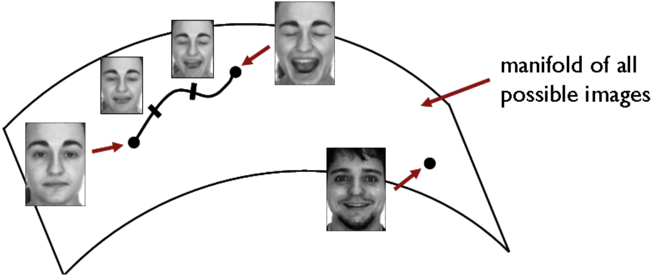
Using Dense Correspondences

- Design cost function to match pixels
- Search for lowest cost correspondences
- Sum individual pixel correspondence costs for image pair matching cost

- Can warp first face along correspondence vectors to generate face similar to second face



Motivation for My Contributions

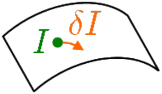


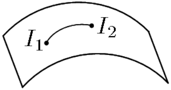
- Path through image manifold describes morphing between endpoint images
- Can we construct a metric that gives appropriate local structure to the manifold?

Riemannian Manifolds

- Riemannian manifold:
  - Differentiable surface
  - Neighborhood of each point on surface is locally Euclidean
  - Tangent space at each point has an inner product
    - distances and angles have meaning

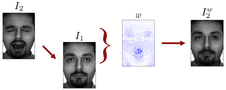
Riemannian Manifolds

- Riemannian manifold:
  - Differentiable surface
  - Neighborhood of each point on surface is locally Euclidean
  - Tangent space at each point has an inner product
    - distances and angles have meaning
- Given local structure by a metric
  - 


  - $d(I_1, I_2)$ = length of geodesic (shortest path) connecting points on the manifold 

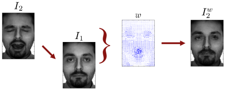
Outline of My Contributions

- Novel deformation- and lighting-insensitive local metric
  - Compute dense correspondence vector fields

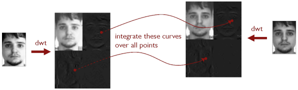


Outline of My Contributions

- Novel deformation- and lighting-insensitive local metric
  - Compute dense correspondence vector fields

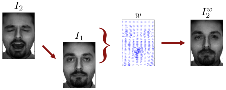


- Use metric to define Riemannian image manifold
  - Calculate wavelet-based point geodesics between illumination-variant images

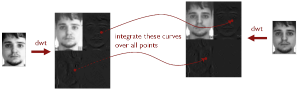


Outline of My Contributions

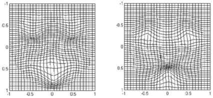
- Novel deformation- and lighting-insensitive local metric
  - Compute dense correspondence vector fields



- Use metric to define Riemannian image manifold
  - Calculate wavelet-based point geodesics between illumination-variant images

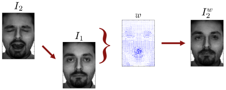


- Use diffeomorphisms on manifolds to model large deformations
  - Initial experiments to handle deformations and illumination changes together

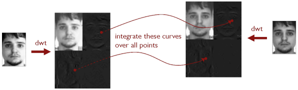


Outline of My Contributions

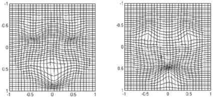
- Novel deformation- and lighting-insensitive local metric
  - Compute dense correspondence vector fields



- Use metric to define Riemannian image manifold
  - Calculate wavelet-based point geodesics between illumination-variant images



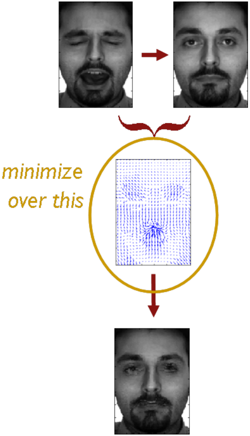
- Use diffeomorphisms on manifolds to model large deformations
  - Initial experiments to handle deformations and illumination changes together



A Deformation and Lighting Insensitive Metric

- For given image pair, minimize image matching cost

$$E(w) = \sum_{i,j} (1-\lambda) E_b(w) + \lambda E_r(w)$$

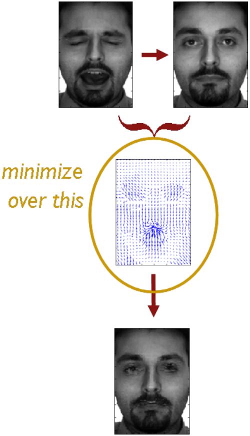


A Deformation and Lighting Insensitive Metric

- For given image pair, minimize image matching cost

$$E(w) = \sum_{i,j} (1-\lambda) E_b(w) + \lambda E_r(w)$$

w : correspondence vector field, where $w = \begin{bmatrix} u \\ v \end{bmatrix}$



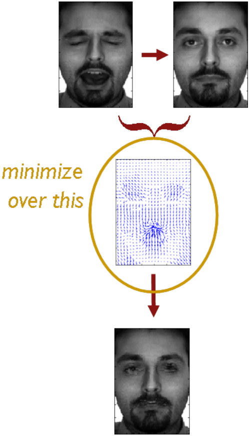
A Deformation and Lighting Insensitive Metric

- For given image pair, minimize image matching cost

$$E(w) = \sum_{i,j} (1-\lambda) E_b(w) + \lambda E_r(w)$$

w : correspondence vector field, where $w = \begin{bmatrix} u \\ v \end{bmatrix}$

E_b : pixel similarity cost



A Deformation and Lighting Insensitive Metric

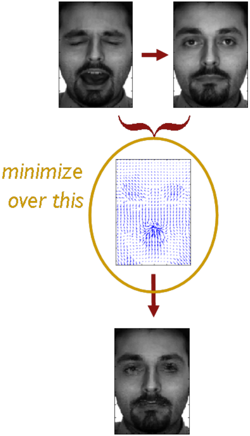
- For given image pair, minimize image matching cost

$$E(w) = \sum_{i,j} (1-\lambda) E_b(w) + \lambda E_r(w)$$

w : correspondence vector field, where $w = \begin{bmatrix} u \\ v \end{bmatrix}$

E_b : pixel similarity cost

E_r : correspondence field regularization cost



A Deformation and Lighting Insensitive Metric

- For given image pair, minimize image matching cost

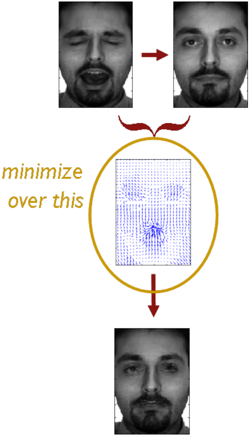
$$E(w) = \sum_{i,j} (1-\lambda) E_b(w) + \lambda E_r(w)$$

w : correspondence vector field, where $w = \begin{bmatrix} u \\ v \end{bmatrix}$

E_b : pixel similarity cost

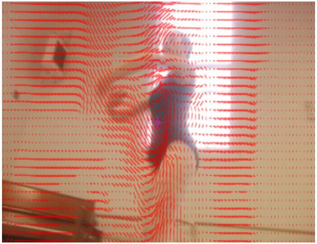
E_r : correspondence field regularization cost

λ : relative weighting constant



Optical Flow Summary

- Track rigid movement between images in video sequences, motion assumed to be small
- Traditionally based on intensity constraint equation:
  $$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t)$$
- Determine displacement vector $[u, v]$ from every pixel in the first image to the most similar pixel in the second $\rightarrow$ vector field



A Deformation and Lighting Insensitive Metric

- For given image pair, minimize image matching cost

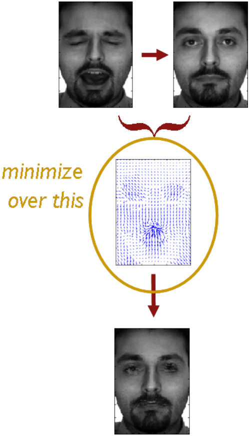
$$E(w) = \sum_{i,j} (1-\lambda) E_b(w) + \lambda E_r(w)$$

w : correspondence vector field, where $w = \begin{bmatrix} u \\ v \end{bmatrix}$

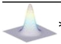
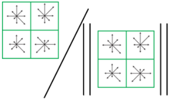
E_b : pixel similarity cost

E_r : correspondence field regularization cost

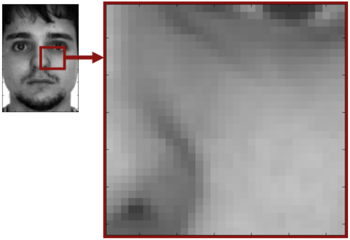
λ : relative weighting constant



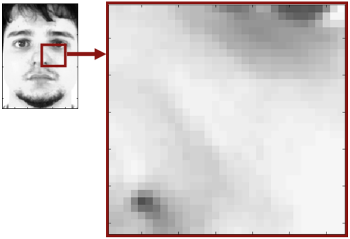
Pixel Similarity Cost: E_b

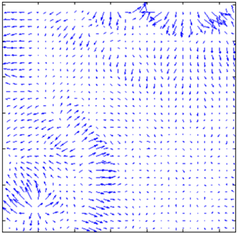
- Lighting-Insensitive Approaches
  - Self-Quotient Image: $\hat{I} = \frac{I}{\quad * I}$  $c = \|\hat{I}_2 - \hat{I}_1\|$
  - Gradient Direction: $\theta = \tan^{-1} \frac{I_y}{I_x}$ $c = \|\theta_2 - \theta_1\|$
  - SIFT Features: $h =$  $c = \|h_2 - h_1\|$
- Traditional Optical Flow: $c = \min\limits_{u,v} \|I_x u + I_y v + I_t\|$

Lighting Change at the Pixel Level

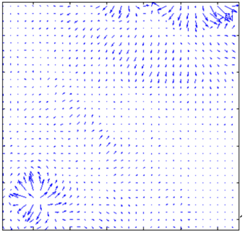


Intensity values are different.





Gradients of the intensity values still point in the same direction.



Direction of Gradient

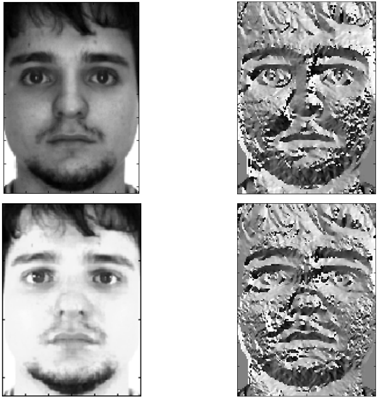
- At every pixel location in image I :

$$\nabla I = [I_x \;\; I_y]^T$$

$$\theta = \tan^{-1}\left(\frac{I_y}{I_x}\right)$$

- Direction θ ignores magnitudes

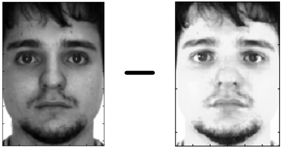
Image — Direction θ of Gradients



Direction of Gradient

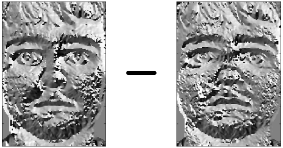
- Pixel intensity difference

$$d = \sum_{i,j} \|I_2(i,j) - I_1(i,j)\|$$



- Pixel gradient direction difference

$$d = \sum_{i,j} \|\theta_2(i,j) - \theta_1(i,j)\|$$



To handle $\theta \mod \pi$, actually use

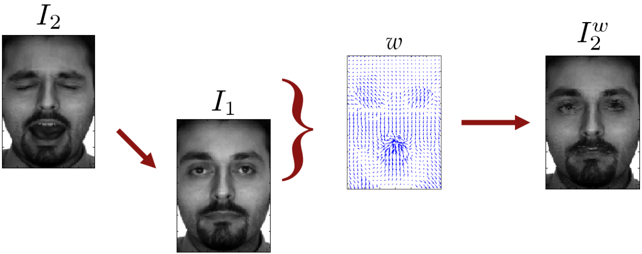
$$d = \sum_{i,j} \min\left(|\theta_2 - \theta_1|, \quad \pi - |\theta_2 - \theta_1|\right)$$

Pixel Similarity Cost: Proposed Metric

- Our new lighting-insensitive pixel similarity cost:

$$E_b(w) = \sum_{i,j} \frac{\|\nabla \delta I\|^2}{\|\nabla I\|^2 + \epsilon^2}$$

where $\delta I = I_2^w - I_1$
I_2^w is I_2 warped along correspondence flow w

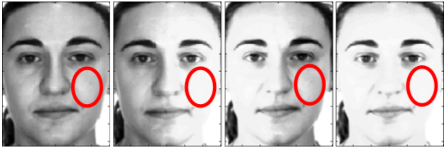


Pixel Similarity Cost: Proposed Metric

- Our new lighting-insensitive pixel similarity cost:

$$E_b(w) = \sum_{i,j} \frac{\|\nabla \delta I\|^2}{\|\nabla I\|^2 + \epsilon^2}$$

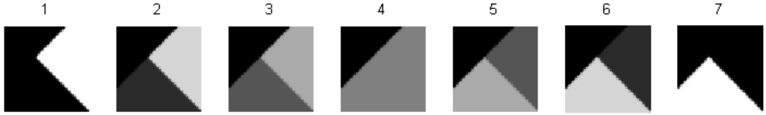
- Weights cost by $\|\nabla I\|^{-1}$



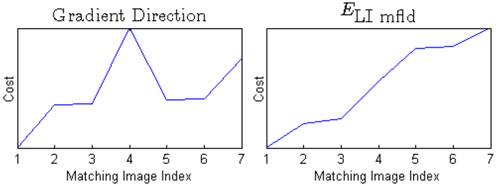
- Match smooth regions
  - where $\|\nabla I\|$ is small, want $\|\nabla \delta I\|$ to be very small
- Regions with larger $\|\nabla I\|$ can be matched in regions of larger $\|\nabla \delta I\|$

A New Lighting-Insensitive Pixel Similarity Cost

- New cost penalizes bigger changes, gradient direction does not
- Illustrative image sequence



Costs of each image compared to first image



A Deformation and Lighting Insensitive Metric

- For given image pair, minimize image matching cost

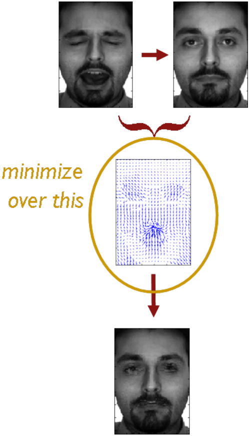
$$E(w) = \sum_{i,j} (1 - \lambda) E_b(w) + \lambda E_r(w)$$

w : correspondence vector field, where $w = \begin{bmatrix} u \\ v \end{bmatrix}$

E_b : pixel similarity cost

E_r : correspondence field regularization cost

λ : relative weighting constant



A Deformation and Lighting Insensitive Metric

- For given image pair, minimize image matching cost

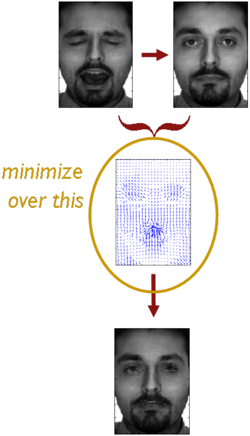
$$E(w) = \sum_{i,j} (1-\lambda) E_b(w) + \lambda E_r(w)$$

w : correspondence vector field, where $w = \begin{bmatrix} u \\ v \end{bmatrix}$

E_b : pixel similarity cost

E_r : correspondence field regularization cost

λ : relative weighting constant



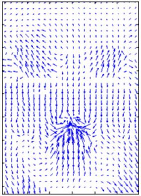
Correspondence Regularization Cost: E_r

- Popular Options
  - L_1-norm:

$$c = \|w\|_1 = \sum_{i,j} |u| + |v|$$

  - Horn and Schunck OF:

$$c = \min_{u,v} \sum_{i,j} \|\nabla u\|^2 + \|\nabla v\|^2$$



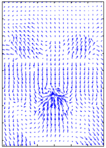
$$= w = \begin{bmatrix} u \\ v \end{bmatrix}$$

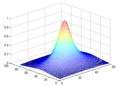
Correspondence Regularization Cost: Proposed Metric

- Our new regularization term:

$$E_r(\vec{w}) = \sum_{i,j} \|\vec{w}\|_g$$

where: $\vec{w} =$ correspondence vector field
$\|\vec{w}\|_g = \langle \vec{w}, \vec{\alpha} \rangle$
$\vec{w} = k * \vec{\alpha}$ (a convolution)
$k =$ gaussian-like filter

$\vec{w} =$ 

$k =$ 

Correspondence Regularization Cost: Proposed Metric

- Uses "Sobolev Gradients and Inner Products"
  - General Inner Product:

$$\langle \alpha, \beta \rangle_K = \langle \alpha, k * \beta \rangle_{\mathbb{R}^n}$$

  - Defines a gradient:

$$\delta f(\alpha) = \langle \nabla_K f(\alpha), \delta\alpha \rangle_K$$

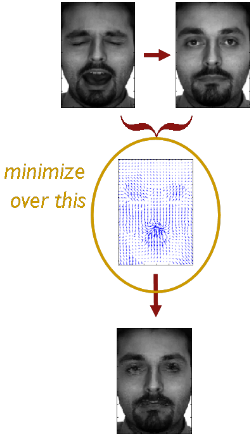
  - Smoother, superior rates of convergence

A Deformation and Lighting Insensitive Metric

- Minimize image matching cost

$$E(w) = \sum_{i,j} (1-\lambda) E_b(w) + \lambda E_r(w)$$

$$w_{\text{optimal}} = \arg\min_{w} E(w)$$



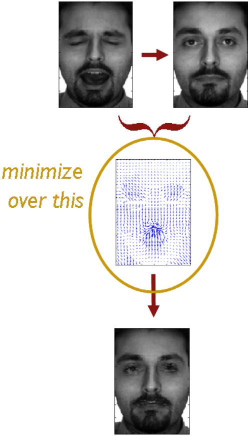
A Deformation and Lighting Insensitive Metric

- Minimize image matching cost

$$E(w) = \sum_{i,j} (1-\lambda) E_b(w) + \lambda E_r(w)$$

$$w_{\text{optimal}} = \arg\min_{w} E(w)$$

- Gradient Descent: $w^{k+1} = w^k - \gamma \nabla E(w^k)$
  - Requires $\nabla E_b(w)$ and $\nabla E_r(w)$
    - Can calculate these!

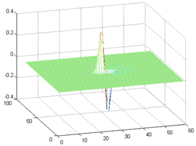


A Deformation and Lighting Insensitive Metric

- Calculate cost between new image and all known images
- Identity of new image = identity of image returning lowest cost

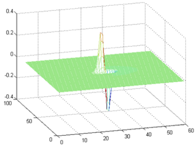
Implementation Notes of Interest

- For more robustness:
  - finite differences $\rightarrow$ broader gradient-like filters h_x and h_y

$h_x =$ 

Implementation Notes of Interest

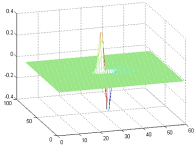
- For more robustness:
  - finite differences $\rightarrow$ broader gradient-like filters h_x and h_y

$h_x =$ 

- $\nabla I \rightarrow \nabla I_1$ in $E_b(w)$ denominator

Implementation Notes of Interest

- For more robustness:
  - finite differences $\rightarrow$ broader gradient-like filters h_x and h_y

$h_x =$ 

- $\nabla I \rightarrow \nabla I_1$ in $E_b(w)$ denominator

- Optimize algorithm for computation speed
  - Use FFTS: accept periodic boundary conditions, weight boundary points less
  - Minimization: ~ 1 second per image pair in Matlab

Use Machine Learning to Improve Results

- Here "machine learning" = black box

  in: data from image pair
  out: classification ("same person" or "different person")

- Based on already known data

Use Machine Learning to Improve Results

- At every pixel, have cost vector $\vec{E}_{xy} = [\ \underbrace{\boxed{E^u_{b_{xy}} \quad E^v_{b_{xy}}}}_{\text{components of pixel similarity cost}} \ \underbrace{\boxed{E^u_{r_{xy}} \quad E^v_{r_{xy}}}}_{\text{components of regularization cost}}\ ]$

Use Machine Learning to Improve Results

- At every pixel, have cost vector $\vec{E}_{xy} = [\ \boxed{E^u_{b_{xy}} \quad E^v_{b_{xy}}}\ \boxed{E^u_{r_{xy}} \quad E^v_{r_{xy}}}\ ]$

  components of pixel similarity cost

  components of regularization cost

- Learn typical correspondence patterns between image pairs (same-person vs different-person)
  - Maximum liklihood estimation (4D Gaussian at each pixel)
  - Naïve Bayes: assume pixel independence (clearly false)

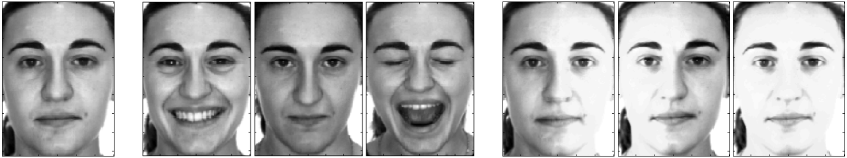
$$P_{\text{same}} = \prod_{x,y} P_{\text{same}}\left(E^u_{b_{xy}}, E^v_{b_{xy}}, E^u_{r_{xy}}, E^v_{r_{xy}}\right)$$

  - Final Similarity Cost:

$$S(I_1, I_2) = \frac{P_{\text{same}}(\vec{E}(w))}{P_{\text{diff}}(\vec{E}(w))}$$

A Deformation and Lighting Insensitive Metric

- Tested on the AR Face Database



  - 100 people
  - 7 images per person:
    - neutral (is the gallery)
    - 3 expression variations
    - 3 lighting variations

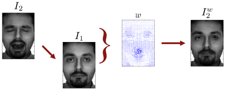
Results published in CVPR 2011

Method	*Expression*	*Lighting*	*Overall*
Proposed Framework with image differencing	84.0%	8.7%	46.3%
Significant Jet Point	80.8%	91.7%	86.3%
Binary Edge Feature and MI	78.5%	97.0%	87.8%
Gradient Direction	86.0%	96.0%	91.0%
Elastic Shape-Texture Matching	98.3%*	97.2%	97.8%*
Elastic Local Reconstruction	99.2%*	98.6%	98.9%*
Proposed Method	**89.6%**	**98.9%**	**94.3%**
Pixel Level Decisions	98.0%	94.0%	96.0%

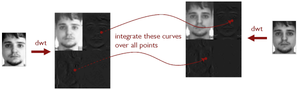
A. Jorstad, D. Jacobs, A. Trouvé. "A Deformation and Lighting Insensitive Metric for Face Recognition Based on Dense Correspondences." Computer Vision and Pattern Recognition (CVPR), pp. 2353-2360, Jun. 2011.

Outline of My Contributions

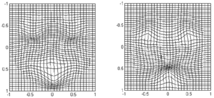
- Novel deformation- and lighting-insensitive local metric
  - Compute dense correspondence vector fields



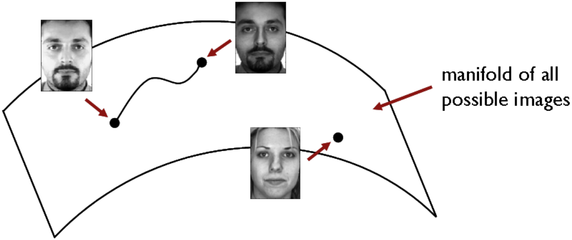
- Use metric to define Riemannian image manifold
  - Calculate wavelet-based point geodesics between illumination-variant images



- Use diffeomorphisms on manifolds to model large deformations
  - Initial experiments to handle deformations and illumination changes together



Consider the Image Manifold



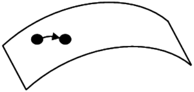
- Face deforms along path through time
- Image similarity = length of geodesic (shortest path)
  connecting them on the image manifold

Geodesics on the Image Manifold

- Identity determined by shortest geodesic
- Benefits
  - Image changes introduced gradually through time $\rightarrow$ robust
  - Metric can be adjusted for various problems
  - Elegant mathematical solution to otherwise messy problem
- But how to calculate geodesics??

Geodesics on the Image Manifold

- Require a local metric to give structure to the manifold



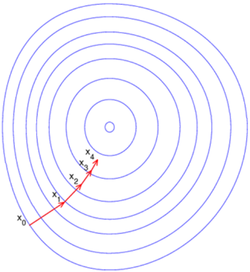
- Use our new pixel similarity cost: $E = \sum_{i,j} \frac{\|\nabla \delta I\|^2}{\|\nabla I\|^2 + \epsilon^2}$
  - Geodesic $I(t)$ parameterized by time t
  - $\delta I(t) = I(t + \delta t) - I(t)$
- Consider *only lighting change* (for now)

Finding The Shortest Path

- Geodesic path $= \arg\min_{I(t)} \int_0^1 E(I(t))dt \; = \; \arg\min_{I(t)} F(I(t))$

Finding The Shortest Path

- Geodesic path $= \arg\min_{I(t)} \int_0^1 E(I(t))dt \; = \; \arg\min_{I(t)} F(I(t))$
- Gradient Descent Method
  - Input: starting point $I(t)^0$; function $F(I(t))$; gradient $\nabla F(I(t))$
  - Iterate: $I(t)^{k+1} = I(t)^k - \gamma \nabla F(I(t)^k)$
  - Output: a local minimum of $F(I(t))$



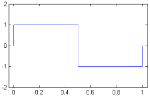
Finding The Shortest Path

- Path $I(t)$ on image manifold is $M \times N \times T$: HUGE!
  (image length x width x # steps discretizing geodesic)
- Nontrivial ∇E calculations
  - ∇I at each point depends locally on its neighbors
  - $\nabla E(I, \nabla I)$ involves many interdependent terms

A Solution: Work in Wavelet Space

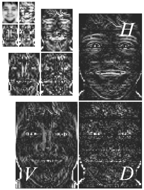
- Observe: Haar wavelets are basically derivative filters

1-D Haar filter:



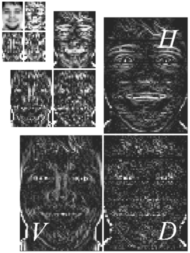
- Project into wavelet basis, orthogonality effectively removes local interdependencies

2-D wavelet decomposition for 3 scales:



Wavelets Summary

- Orthonormal functions
  - $f(t) = \sum_{m,n} \langle f, \psi_{m,n} \rangle \cdot \psi_{m,n}(t)$
  - Allows local analysis of a function according to scale
- 2-D discrete wavelet transform (dwt) output (1 scale):
  - Filtered function in H, V, D directions
  - Downsampled function for next coarser scale
- Orthogonal basis functions $\implies$ changing one coefficient does not affect any other locations and scales: each term of decomposition is independent

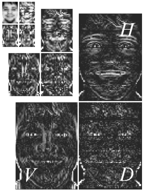


A Solution: Work in Wavelet Space

- Approximate lighting metric in Haar wavelet space:

$$E(I) = \sum_{i,j} \frac{\|\nabla \delta I\|^2}{\|\nabla I\|^2 + \epsilon^2} \rightsquigarrow E(H,V) = \sum_{m,n} \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2}$$

$$H(m,n) \approx \frac{\partial I}{\partial x}, \; V(m,n) \approx \frac{\partial I}{\partial y}$$

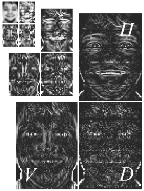


A Solution: Work in Wavelet Space

- Approximate lighting metric in Haar wavelet space:

$$E(I) = \sum_{i,j} \frac{\|\nabla \delta I\|^2}{\|\nabla I\|^2 + \epsilon^2} \rightsquigarrow E(H,V) = \sum_{m,n} \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2}$$

$$H(m,n) \approx \frac{\partial I}{\partial x}, \quad V(m,n) \approx \frac{\partial I}{\partial y}$$



- All points now independent!
  - Before, ∇I depended on neighbors
  - In Haar wavelet domain, messy *inter*dependent terms become simple *in*dependent terms

$$\min \int_0^1 \sum_{m,n} \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2} dt = \min \sum_{m,n} \int_0^1 \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2} dt$$

independent! can minimize over these separately!

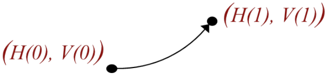
A Solution: Work in Wavelet Space

- To solve explicitly, convert

$$\underset{H(t),\ V(t)}{\arg\min} \int_0^1 \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2} dt$$

(the lighting metric converted to wavelet space)

into an ODE defining geodesic paths
(a boundary value problem)



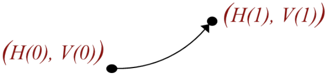
A Solution: Work in Wavelet Space

- To solve explicitly, convert

$$\underset{H(t),\ V(t)}{\arg\min} \int_0^1 \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2} dt$$

(the lighting metric converted to wavelet space)

into an ODE defining geodesic paths (a boundary value problem)



- Recall Euler-Lagrange
  - Given functional

$$J(f) = \min_{f(t)} \int_0^1 F(t, f(t), f'(t))dt$$

  - The function $f(t)$ that minimizes $J(f)$ is described by the equation

$$\frac{\partial F}{\partial f} - \frac{d}{dt}\frac{\partial F}{\partial f} = 0$$

A Solution: Work in Wavelet Space

- The 2-D Euler-Lagrange Equations convert

$$\underset{H(t),\ V(t)}{\arg\min} \int_0^1 \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2} dt$$

into the ODEs

$$H(H'^2 + V'^2) - 2H'(HH' + VV') + H''(H^2 + V^2) = 0$$
$$V(H'^2 + V'^2) - 2V'(HH' + VV') + V''(H^2 + V^2) = 0$$

where the geodesic path $= [H(t), V(t)]$

A Solution: Work in Wavelet Space

- The 2-D Euler-Lagrange Equations convert

$$\underset{H(t),\ V(t)}{\arg\min} \int_0^1 \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2} dt$$

into the ODEs

$$H(H'^2 + V'^2) - 2H'(HH' + VV') + H''(H^2 + V^2) = 0$$
$$V(H'^2 + V'^2) - 2V'(HH' + VV') + V''(H^2 + V^2) = 0$$

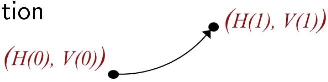
where the geodesic path $= [H(t), V(t)]$

- Computational Algorithm:
  - Solve for $H'',\ V''$ algebraically from above
  - Solve system of first order equations numerically (a BVP)

$$\vec{x} = [H\ \ H'\ \ V\ \ V'] \qquad \frac{d}{dt}\vec{x} = [\vec{x}(2)\ \ H''\ \ \vec{x}(4)\ \ V'']$$

A Solution: Work in Wavelet Space

- Key Idea:
  - Given $H(0), V(0), H(1), V(1)$, numerical integration produces minimum cost at each point location



  - No optimization routine required!!
  - Sum up independent integrals for overall cost

$$\sum_{m,n} \int_0^1 \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2} dt$$

A Solution: Work in Wavelet Space

- Key Idea:
  - Given $H(0), V(0), H(1), V(1)$, numerical integration produces minimum cost at each point location

  - No optimization routine required!!
  - Sum up independent integrals for overall cost

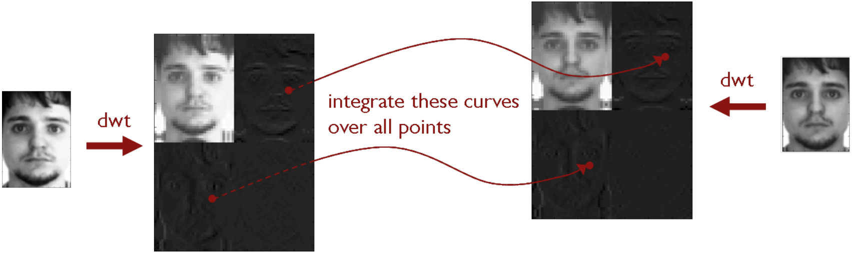
$$\sum_{m,n} \int_0^1 \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2} dt$$

- Interesting side note: numeric integration in polar space more stable

$$\underset{r(t),\ \theta(t)}{\arg\min} \int_0^1 \frac{r'^2 + r^2\theta'^2}{r^2 + \theta^2 + \epsilon^2} dt$$

A Solution: Work in Wavelet Space

- Algorithm considering lighting variation alone:
  - Assume aligned images
  - Convert to Haar wavelet basis at one scale (dwt)
  - For each pair of corresponding points,
    - Integral of BVP solution curve = cost to match corresponding coefficients
  - Sum costs of all point pairs for final cost



Add in Deformations...

- Consider variations in facial expression



- Add in first three scales of wavelet decomposition

$$\text{geodesic path} = \sum_{m,n} \sum_{s} \underset{H(t),\ V(t)}{\arg\min}\ \lambda_s \int_0^1 \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2} dt$$

Add in Deformations...

- Consider variations in facial expression

- Add in first three scales of wavelet decomposition

$$\text{geodesic path} = \sum_{m,n} \sum_{s} \operatorname*{arg\,min}_{H(t),\ V(t)} \lambda_s \int_0^1 \frac{\delta H^2 + \delta V^2}{H^2 + V^2 + \epsilon^2} dt$$

- Why would this help?
  - Coarse scales ignore fine details
  - Small deformations within the support of each wavelet basis function are handled together

A Fast Algorithm

- Fast Algorithm:
  - Pre-calculate curve lengths for all possible input (discretized)
  - Store in lookup table
    - 3-D: $(r_1, r_2, \Delta\theta)$
  - Generate 40x40x40 table:
    - 1.5 hours
- Runtime:
  - Convert images to polar wavelet coefficients
  - Look up value of each point position pair in table; sum
  - Compare two 5000-pixel images:
    - 0.0013 seconds!

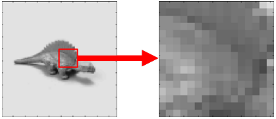
Results to be published in ECCV 2012

Method	*Time (sec)*	*Expression*	*Lighting*	*Overall*
Image Differencing	3.1×10^{-5}	83.0%	9.0%	46.0%
Normalized Cross-Correlation	7.2×10^{-3}	84.0%	59.3%	71.7%
Significant Jet Point	–	80.8%	91.7%	86.3%
Binary Edge Feature and MI	–	78.5%	97.0%	87.8%
Gradient Direction	3.8×10^{-4}	85.0%	95.3%	90.2%
E_{DLI} (previous method)	1.0×10^{0}	89.6%	98.9%	94.3%
Proposed Method	1.3×10^{-3}	**93.7%**	**96.7%**	**95.2%**
Pixel Level Decisions	5.6×10^{-4}	98.0%	94.0%	96.0%
Proposed Method thresholded	1.3×10^{-3}	**97.3%**	**97.0%**	**97.2%**

A. Jorstad, D. Jacobs, A. Trouvé. "A Fast Illumination and Deformation Insensitive Image Comparison Algorithm Using Wavelet-Based Geodesics." European Conference on Computer Vision (ECCV), Oct. 2012.

Template Matching

- Proof of concept on NORB dataset
- Given template:

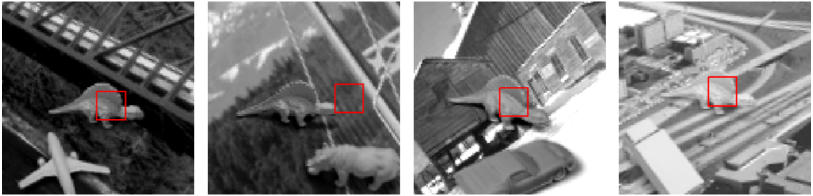


- Search for most similar patch at every location in cluttered images
  - 6 lighting conditions
  - 3 azimuth (pose) variations ($\approx$ deformations)
  - Compare 11,664 positions in each of 36 images

Template Matching

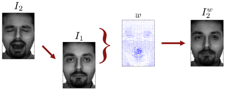
- Proof of concept results:
  - "correct" if best location within 8 pixels of true location

Method	*Localization Accuracy*
Normalized Cross-Correlation	25.0%
Gradient Direction	61.1%
Proposed Method	**80.6%**

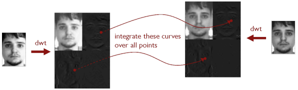


Outline of My Contributions

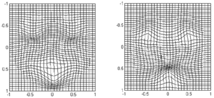
- Novel deformation- and lighting-insensitive local metric
  - Compute dense correspondence vector fields



- Use metric to define Riemannian image manifold
  - Calculate wavelet-based point geodesics between illumination-variant images



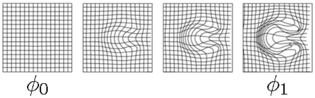
- Use diffeomorphisms on manifolds to model large deformations
  - Initial experiments to handle deformations and illumination changes together

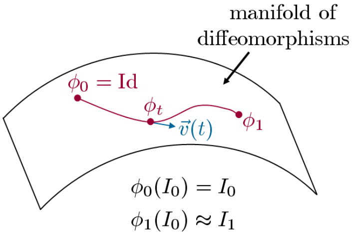


Diffeomorphisms in Computer Vision

- Diffeomorphism: a smooth, invertible bijective mapping between points on manifolds
  - Images can deform gradually through time

ϕ_0 = uniform mesh
ϕ_t = diffeomorphism at time t
$\vec{v}(t)$ = vector field defining direction of deformation at time t

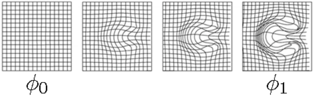




Diffeomorphisms in Computer Vision

- Diffeomorphism: a smooth, invertible bijective mapping between points on manifolds
  - Images can deform gradually through time

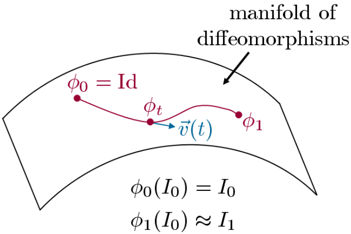
$$
\begin{aligned}
\phi_0 &= \text{uniform mesh} \\
\phi_t &= \text{diffeomorphism at time } t \\
\vec{v}(t) &= \text{vector field defining direction of deformation at time } t
\end{aligned}
$$



- "Compute diffeomorphism" $=$ find $\vec{v}(t)$ at every t

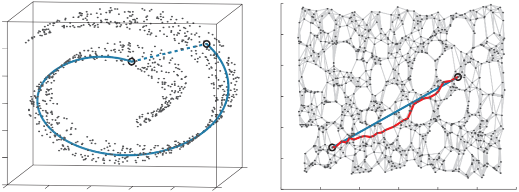
$$E(\vec{v}) = \int_0^1 \|\vec{v}(t)\| dt \quad (+\lambda d(\phi_1(I_0), I_1))$$

$$\hat{\vec{v}} = \arg\min_{\vec{v}} E(\vec{v})$$

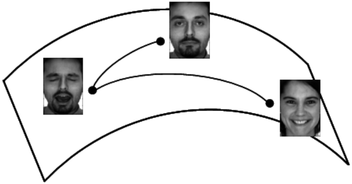


Diffeomorphisms in Computer Vision

- Analytic geometry, not manifold learning
- Manifold learning

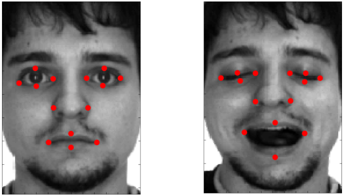


- Our work



Explicitly Handle Deformations: an Initial Foray

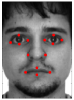
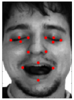
- Following Bookstein Splines-based work of
  - V. Camion, L. Younes. “Geodesic Interpolating Splines.” EMMCVPR, 2001.
  - C. Twining, S. Marsland, C. Taylor. “Measuring Geodesic Distances on the Space of Bounded Diffeomorphisms.” BMVC, 2002.
- Fix sparse point correspondences



- Automatic facial feature point detection algorithms
  - L. Ding, A.M. Martinez. “Features Versus Context: An Approach for Precise and Detailed Detection and Delineation of Faces and Facial Features.” PAMI, 2010.
  - P. Belhumeur, D. Jacobs, D. Kriegman, N. Kumar. “Localizing Parts of Faces Using a Consensus of Exemplars.” CVPR, 2011.

Explicitly Handle Deformations: an Initial Foray

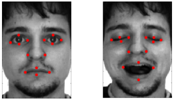
- Fix sparse point correspondences

  - Define a diffeomorphism between images restricted at these points
  - Free variables = paths connecting these points through time

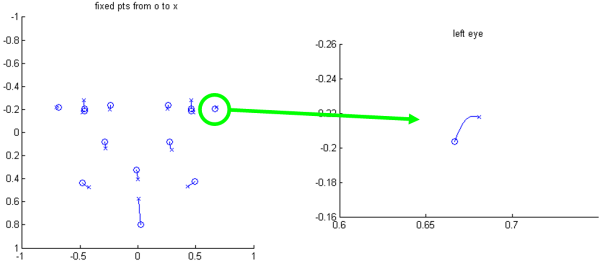
Explicitly Handle Deformations: an Initial Foray

- Fix sparse point correspondences



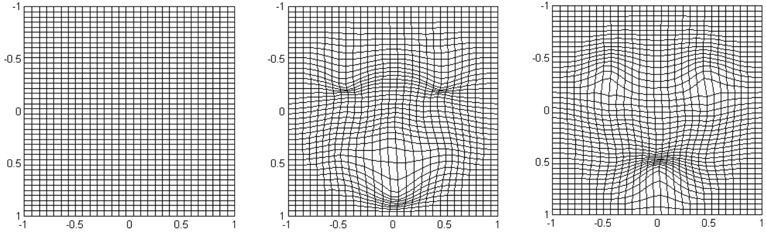
  - Define a diffeomorphism between images restricted at these points
  - Free variables = paths connecting these points through time

- Calculate geodesic paths between fixed points



Explicitly Handle Deformations: an Initial Foray

- Spline interpolation $\rightarrow$ diffeomorphism through all point pairs



clamped plate spline = 0 on unit circle

- Cost of diffeomorphism depends on smoothness/regularization of diffeomorphic paths

Explicitly Handle Deformations: an Initial Foray

- Define spline form: $\vec{v}(t, x(t)) = \sum_{i=1}^{N} \alpha_i(t) G(x(t), x_i(t))$

  x : points in space
  α : weighting coefficients
  $G(x_1, x_2)$: Greens function for L
  (defines form of spline by relating points)

Explicitly Handle Deformations: an Initial Foray

- Define spline form: $\vec{v}(t, x(t)) = \sum_{i=1}^{N} \alpha_i(t) G(x(t), x_i(t))$

  x : points in space
  α : weighting coefficients
  $G(x_1, x_2)$: Greens function for L
  (defines form of spline by relating points)

Green's Functions

Linear differential operator L has Green's function $G(x, y)$ satisfying $LG(x, y) = \delta(x - y)$.
Useful for solving $Lu(x) = f(x)$:

$$\begin{aligned}
\int LG(x,y)f(y)dy &= \int \delta(x-y)f(y)dy \\
&= f(x) \\
Lu(x) = f(x) &= \int LG(x,y)f(y)dy \\
&= L\int G(x,y)f(y)dy \\
\rightarrow u(x) &= \int G(x,y)f(y)dy
\end{aligned}$$

Explicitly Handle Deformations: an Initial Foray

- Define spline form: $\vec{v}(t, x(t)) = \sum_{i=1}^{N} \alpha_i(t) G(x(t), x_i(t))$

  x : points in space
  α : weighting coefficients
  $G(x_1, x_2)$: Greens function for L
  (defines form of spline by relating points)

- The regularization term (N correspondences):

$$E(\vec{v}(t, x(t))) = \sum_{i=1}^{N} \int_0^1 \|L\vec{v}\|^2 dt$$

Green's Functions

Linear differential operator L has Green's function $G(x, y)$ satisfying $LG(x, y) = \delta(x - y)$.
Useful for solving $Lu(x) = f(x)$:

$$\int LG(x,y) f(y) dy = \int \delta(x-y) f(y) dy = f(x)$$

$$Lu(x) = f(x) = \int LG(x,y) f(y) dy = L \int G(x,y) f(y) dy$$

$$\rightarrow u(x) = \int G(x,y) f(y) dy$$

Explicitly Handle Deformations: an Initial Foray

- Define spline form: $\vec{v}(t, x(t)) = \sum_{i=1}^{N} \alpha_i(t) G(x(t), x_i(t))$

  x : points in space
  α : weighting coefficients
  $G(x_1, x_2)$: Greens function for L
  (defines form of spline by relating points)

- The regularization term (N correspondences):

$$E(\vec{v}(t, x(t))) = \sum_{i=1}^{N} \int_0^1 \|L\vec{v}\|^2 dt = \sum_{i=1}^{N} \int_0^1 \langle \vec{v}, L^2 \vec{v} \rangle dt$$
$$= \sum_{i=1}^{N} \int_0^1 \alpha_i(t) \cdot \vec{v}(t, x(t)) dt$$

Green's Functions

Linear differential operator L has Green's function $G(x, y)$ satisfying $LG(x, y) = \delta(x - y)$.
Useful for solving $Lu(x) = f(x)$:

$$\begin{aligned}
\int LG(x, y) f(y) dy &= \int \delta(x - y) f(y) dy \\
&= f(x) \\
Lu(x) = f(x) &= \int LG(x, y) f(y) dy \\
&= L \int G(x, y) f(y) dy \\
\rightarrow u(x) &= \int G(x, y) f(y) dy
\end{aligned}$$

Explicitly Handle Deformations: an Initial Foray

- Define spline form: $\vec{v}(t, x(t)) = \sum_{i=1}^{N} \alpha_i(t) G(x(t), x_i(t))$

  x : points in space
  α : weighting coefficients
  $G(x_1, x_2)$: Greens function for L
  (defines form of spline by relating points)

- The regularization term (N correspondences):

$$E(\vec{v}(t, x(t))) = \sum_{i=1}^{N} \int_0^1 \|L\vec{v}\|^2 dt = \sum_{i=1}^{N} \int_0^1 \langle \vec{v}, L^2 \vec{v} \rangle dt$$
$$= \sum_{i=1}^{N} \int_0^1 \alpha_i(t) \cdot \vec{v}(t, x(t)) dt$$

- Minimize $E \Leftrightarrow$ solve for geodesic $\vec{v}(t)$

Green's Functions

Linear differential operator L has Green's function $G(x, y)$ satisfying $LG(x, y) = \delta(x - y)$.
Useful for solving $Lu(x) = f(x)$:

$$\int LG(x, y) f(y) dy = \int \delta(x - y) f(y) dy$$
$$= f(x)$$
$$Lu(x) = f(x) = \int LG(x, y) f(y) dy$$
$$= L \int G(x, y) f(y) dy$$
$$\rightarrow u(x) = \int G(x, y) f(y) dy$$

Preliminary Results

- Diffeomorphism defined by
  - 14 points
  - 2-D (x,y)
  - 10 time steps

  $\Longrightarrow$ 280 independent variables

Preliminary Results

- Diffeomorphism defined by
  - 14 points
  - 2-D (x,y)
  - 10 time steps

  $\Longrightarrow$ 280 independent variables
- Point-based diffeomorphisms *not* ideal for faces:

Preliminary Results

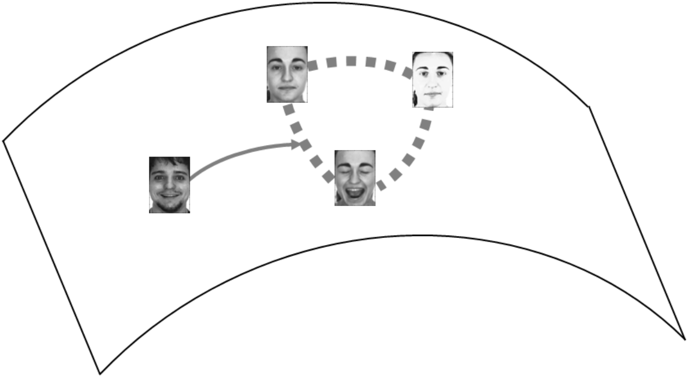
- Scream case (the hard one) identification accuracy

  - Dense correspondence vector field method:
    - 80%
  - Wavelet-based point geodesics method:
    - 81%
  - Diffeomorphisms with wavelet-based point geodesic costs:
    - 82%

(All pre-learning)

Generating Intermediate Images

- Given diffeomorphic path, can generate intermediate images between any two given images
- Compare unknown image to all intermediate images



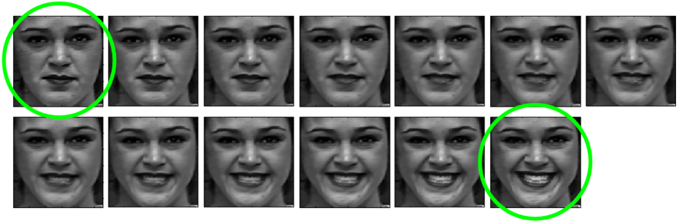
Generating Intermediate Images

- Cohn-Kanade AU-Coded Facial Expression Database: compare known intermediate images to those generated from two extremes

True:

Generated:

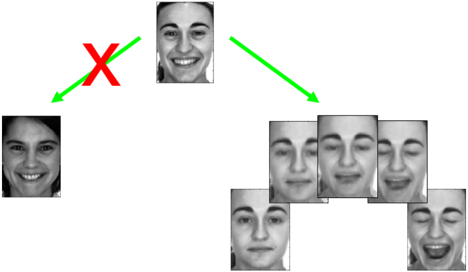
input images



Generating Intermediate Images

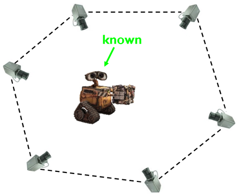
- 323 expression sequences
  - Compare(known images, generated images)
  - 80% of each known sequence matched more closely to generated images than to the input extremes

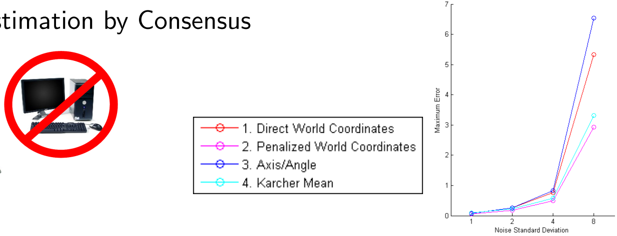
- Avoid common problem:



Other Research from Johns Hopkins Applied Physics Lab

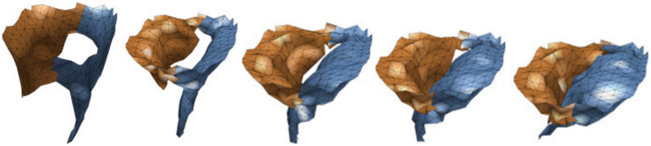
- Model-Based Pose Estimation by Consensus





- Patient-Specific Modeling and Analysis of the Mitral Valve Using 3D-TEE

$$\text{minimize } \textit{valve configuration energy} = \phi^X + \phi^E + \phi^T + \phi^C + \phi^K$$

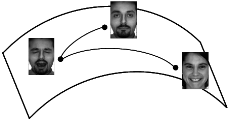


Future Work

- Explicitly handle occlusions
- To make diffeomorphisms more realistic: interpolate using splines with basis functions specific to faces, or consider all points as independent variables
- Apply stronger optimization schemes to be able to calculate more accurate geodesics through diffeomorphisms
- Combine with methods designed to handle pose variation
- Use in conjunction with machine learning algorithms to learn diffeomorphic patterns specific to faces

Conclusions

- Can effectively model both deformations and illumination changes using geodesics and diffeomorphisms on manifolds.
- These models capture more information than simple correspondence fields.
- It is encouraging that face recognition results using these methods are so strong, considering no face-specific knowledge is considered.
- Using these methods in conjunction with machine learning algorithms specific to faces should provide very robust methods for face recognition across changes in lighting and expression.



Publications

- A. Jorstad, D. Jacobs, A. Trouvé. **"A Fast Illumination and Deformation Insensitive Image Comparison Algorithm Using Wavelet-Based Geodesics."** European Conference on Computer Vision (ECCV), Oct. 2012.
- A. Jorstad, D. Jacobs, A. Trouvé. ''**A Deformation and Lighting Insensitive Metric for Face Recognition Based on Dense Correspondences."** Computer Vision and Pattern Recognition (CVPR), pp. 2353-2360, Jun. 2011.
- A. Jorstad, D. DeMenthon, I. Wang, P. Burlina. **"Distributed Consensus on Camera Pose."** Image Processing, IEEE Transactions on, vol. 19, pp. 2396-2407, Sep. 2010.
- P. Burlina, C. Sprouse, D. DeMenthon, A. Jorstad, R. Juang, F. Contijoch, T. Abraham, D. Yuh, E. McVeigh. **"Patient-Specific Modeling and Analysis of the Mitral Valve Using 3D-TEE."** Information Processing in Computer-Assisted Interventions (IPCAI), First International Conference, Jun. 2010; Lecture Notes in Computer Science, vol. 6135/2010, pp. 135-146, 2010.
- P. Burlina, C. Sprouse, D. DeMenthon, A. Jorstad, F. Contijoch, E. McVeigh, R. Juang, T. Abraham, D. Yuh. **"Individualized Cardiothoracic Surgical Planning using Computer Aided 3D Modeling and Image Analysis."** American Medical Association and the IEEE Engineering in Medicine and Biology Society Conference on Medical Technology, Mar. 2010.
- A. Jorstad, P. Burlina, I. Wang, D. Lucarelli, D. DeMenthon. **"Model-Based Pose Estimation by Consensus."** Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2008), pp. 569-574, Dec. 2008.