Deformable Image Registration

Adrien Bartoli
ALCoV – ISIT
Université d’Auvergne
Clermont-Ferrand, France

IIT, Genoa, December 2010
Video: paper

Video: fabric
The Image/Video Registration Problem

Template: an image that serves as a reference

Warp: a function that transfers pixels between images

\[ \mathcal{W} : \mathbb{R}^2 \times \mathbb{R}^p \rightarrow \mathbb{R}^2 \]

Warp visualization grid

Note: a parametric framework is used but what follows holds in the variational framework
Rigidity: an Important Prior

Image registration

3D reconstruction

SfM

Video: error image

2D homography
Deformable \approx \text{Global} + \text{Local}
The Deformable Surface Case

Prior knowledge: (piecewise) smoothness
Why is Registration Difficult?
Why is Registration Difficult?

- Appearance varies due to
  - Camera pose and internal setup
  - Scene lighting
  - Surface deformation
  - Motion and optics blur
  - Occlusions
  - ...

- Little prior knowledge on the transformations: piecewise smoothness
External and Self-Occlusions

Some pixels disappear
Smoothness preserved

Some pixels disappear
Smoothness partly preserved

Some pixels disappear
Smoothness preserved
External Occlusions
Self-Occlusions
An Optimization Problem

We do sequential registration and consider two images only.

The Region Of Interest (ROI) is a set of pixels in the template.

\[ \mathcal{W}(\cdot; u) \]

Cost function: measures the quality of the fit.

\[ \min_{u \in \mathbb{R}^p} \mathcal{E}(u; \lambda) \]

Hyperparameters: weight different terms.

Image data is not enough!
Statistical. The Active Appearance Model (AAM) is an example of a pre-trained model

Example of training images with variability in expression, pose and identity

[Statistical. Cootes et al, PAMI’01; Peyras et al, BMVC’08]

Physical. This simple template is an example of an un-trained model: only simple empirical/physical constraints are involved

+ surface smoothness (+ inextensibility) (+ zero gaussian curvature) + ...
Most commonly, an image is a rectangular array of pixels.

It is natural to see it as a function from $\mathbb{N}^2$ to $\mathbb{N}^c$.

\[ \mathcal{I} : \mathbb{N}^2 \rightarrow \mathbb{N}^c \]

$c$ is the number of channels, 1 for grey-level and 3 for color RGB images.

\[ \mathcal{I}((u \ v)) = (r \ g \ b) \quad \text{with} \quad (u \ v) \in \mathbb{N}^2 \quad \text{and} \quad (r \ g \ b) \in \mathbb{N}^3 \]
Using interpolation, an image can be evaluated at non integer coordinates

\[ \mathcal{I}(x, y) = (r, g, b) \quad \text{with} \quad (x, y) \in \mathbb{R}^2 \quad \text{and} \quad (r, g, b) \in \mathbb{N}^3 \]

Interpolation schemes range from nearest neighbour to spline-based

Bilinear interpolation is typical in computer vision
Lecture Roadmap

• Overview and general points
• **Parameterized warps**
• General points on estimation
• Feature-based estimation
• Pixel-based estimation
• Conclusions
What is a Warp?

• A point function between two images

• We use parameterized warps and thus

\[ \mathcal{W} : \mathbb{R}^2 \times \mathbb{R}^{2l} \rightarrow \mathbb{R}^2 \]
Driving Features

Interpolation between \( l \) driving features

\[ \mathbf{u}_k \leftrightarrow \mathbf{u}'_k \quad \text{with} \quad k = 1, \ldots, l \]

Source driving features \( \mathbf{u}_k \) known \quad \quad \quad Target driving features \( \mathbf{u}'_k \) unknown

We write \( \mathcal{W} \) as a function of point \( \mathbf{q} \) and \( \mathbf{U}' \):

\[ \mathbf{q}' = \mathcal{W}(\mathbf{q}; \mathbf{U}') \]

\[ \mathbf{U}' \overset{\text{def}}{=} \begin{pmatrix} \mathbf{u}'_1^\top \\ \vdots \\ \mathbf{u}'_l^\top \end{pmatrix} \in \mathbb{R}^{l \times 2} \text{ bundles all the target driving features} \]
LBE Warps (Linear Basis Expansion)

• An LBE warp is nonlinear in the source point coordinates $\mathbf{q}$ and linear in its parameter set

• Most parameterized warps are LBE

• Smoothness can be easily measured

• Property: an LBE warp can be Feature-Driven

$$\mathcal{W}(u_k; \mathbf{U}') = u'_k \quad \text{with} \quad k = 1, \ldots, l$$

• The complexity is related to $l$
Feature-Driven LBE Warps

\[ q' = \mathcal{W}(q; U') = \sum_{k=1}^{l} \nu_k(q) \, u'_k \]

The \( \nu_k : \mathbb{R}^2 \to \mathbb{R} \) are basis functions.

They are nonlinear and depend on the known \( U \).

Source driving features \( u_k \) known

Target driving features \( u'_k \) unknown
Classical Parameterizations

• Flow-Field (FF)
• Mesh Interpolation (MI)
• Radial Basis Functions (RBF)
• Tensor Product (TP)
• Moving Least Squares (MLS)
• ...
FF (Flow-Field)

• Every pixel displacement is parameterized
• This can be coupled with simple bilinear interpolation

\[ q' = \mathcal{W}(q; U') = q + \delta(q; u'_{k_1}, u'_{k_2}, u'_{k_3}, u'_{k_4}) \]

where \( u_{k_1}, \ldots, u_{k_4} \) are the four closest driving features to \( q \)

• \( l \) is the number of pixels
• Traditional parameterization in optical flow estimation and variational approaches [Horn et al, AI’81]
MI Warps (Mesh Interpolation)

- Mesh the driving features
- Use a simple parametric model over each facet
- Example: triangular mesh + affine model

\[ q' = \mathcal{W}(q; U') = \lambda_{q,1} u'_{k_1} + \lambda_{q,2} u'_{k_2} + \lambda_{q,3} u'_{k_3} \]

where \( u_{k_1}, u_{k,2} \) and \( u_{k_3} \) define the facet containing \( q \) and \( \lambda_{q,1}, \lambda_{q,2} \) and \( \lambda_{q,3} \) are constant weights

- \( l \) is the number of mesh vertices
- Used in image registration [Pilet et al, IJCV’08] and 3D reconstruction [Salzmann et al, PAMI’07]
RBF Warps (Radial Basis Function)

- An RBF depends on the distance between \( \mathbf{q} \) and each \( \mathbf{u}_k \)

\[
q' = \mathcal{W}(\mathbf{q}; \mathbf{U}') = \sum_{k=1}^{l} \rho(d^2(\mathbf{q}, \mathbf{u}_k)) \mathbf{u}_k'
\]

where \( \rho \) is the kernel

- \( \rho(x) = x \log x \) gives the TPS warp (Thin-Plate Spline) [Bookstein, PAMI’89] (linear reparameterization is then needed)
TP Warps (Tensor Product)

- A TP uses the $x$ and $y$ components of the distance between $\mathbf{q}$ and each $\mathbf{u}_k$

- The $\mathbf{u}_k$ must be on a regular grid

$$q' = \mathcal{W}(\mathbf{q}; \mathbf{U}') = \sum_{k_x=1}^{l_x} \sum_{k_y=1}^{l_y} \alpha_x(x) \alpha_y(y) \mathbf{u}_{k_x+(k_y-1)l_x}$$

with $\mathbf{q} = (x \ y)^\top$ and $l = l_x l_y$

- A common choice is the Uniform Cubic B-Spline (UCBS) [Rueckert et al, TMI’99]

- Also termed FFD (Free-Form Deformation)
Alternative Derivation for the TPS and UCBS Warps

\[ \mathbb{R} \times \mathbb{R}^{l_x} \rightarrow \mathbb{R} \]

Cubic B-Spline

\[ \mathbb{R}^2 \times \mathbb{R}^l \rightarrow \mathbb{R} \]

Tensor Product 2.5D

The Thin-Plate Spline

Radial Basis 2.5D

\[ \mathbb{R}^2 \times \mathbb{R}^{2l} \rightarrow \mathbb{R}^2 \]

The UCBS Warp

The TPS warp
### Synthetic Comparison

<table>
<thead>
<tr>
<th>Sparsity</th>
<th>Number of unknowns</th>
<th>Continuity</th>
<th>Integral unsmoothness</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>++</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MI</td>
<td>++</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UCBS</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPS</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- The TPS has minimal bending energy [Duchon, RAIRO’71]
- Similar representational power [Brunet et al, IJCV’11]
- The UCBS (and FF) driving features must be on a regular grid
Measuring Unsmoothness

• Needed for estimation
• Use partial derivatives

\[ \mathcal{E}_s(U') \overset{\text{def}}{=} \int \left\| \frac{\partial^d \mathcal{W}}{\partial q^d} (q; U') \right\|_F^2 \, dq \text{ with usually } d = 2 \]

• For LBE warps discrete approximations are Linear Least Squares

\[ \mathcal{E}_s(U') \approx \| B U' \|_F^2 \]

(for the TPS and UCBS the integral has a closed-form)
Geometric Interpretation

- LBE ≈ Affine camera
- $(\mathbf{u}_k \in \mathbb{R}^{>1}) \approx$ Deformable surface

[Bartoli et al, IJCV’10]
Generalized Warps

- RBE (Rational Basis Expansion) \(\approx\) Perspective camera
- \((u_k \in \mathbb{R}) \approx\) Rigid surface

[Bartoli et al, IJCV’10 ; Brunet et al, BMVC’09]
Estimation of LBE Warps from $m = l$ Landmarks

- Landmarks are **exact** point matches

  $$q_j \leftrightarrow q'_j \quad \text{with} \quad j = 1, \ldots, m$$

- Example: 53 manually clicked points

- This is the *minimal* case: just pick the points matches as driving feature!
Estimation of LBE Warps from $m > l$ Landmarks

- This is the *redundent* case
- Minimize the transfer error and unsmoothness

$$
\min_{U' \in \mathbb{R}^{l \times 2}} \mathcal{E}_d(U') + \lambda \mathcal{E}_s(U')
$$

$$
\mathcal{E}_d(U') = \sum_{j=1}^{m} \| q'_j - \mathcal{W}(q_j; U') \|_2^2
$$

the solution is Linear Least Squares
An Example with the TPS Warp
Analyzing the Cost Function

\[
\min_{U' \in \mathbb{R}^l \times 2} \mathcal{E}_d(U') + \lambda \mathcal{E}_s(U')
\]

**Smoothing hyperparameter**
Controls the data fitness / warp smoothness trade-off (assumed known for now)

**Smoother**
Prior knowledge – corresponds to the prior in the Bayesian framework

**Data term**
Assesses the fit – corresponds to the likelihood in the Bayesian framework
Tweaking the Smoothness Parameter

Overfitting

About fine

Underfitting

Underfitting
Lecture Roadmap

- Overview and general points
- Parameterized warps
- General points on estimation
- Feature-based estimation
- Pixel-based estimation
- Conclusions
Cost Function with Multiple Terms

\[
\min_{U' \in \mathbb{R}^l \times 2} \mathcal{E}_d(U') + \lambda \mathcal{E}_s(U')
\]

• Data terms
  – Relate the warp to the image content
  – Feature-based vs pixel-based
  – Not enough to constrain the warp well enough

• Regularization terms
  – Measure closeness to priors
  – Piecewise smoothness
Abstracting or not abstracting the images?

- An image is a lot of data
  \[ 3 \cdot 1000^2 = 3 \cdot 10^6 \text{ Bytes} \approx 3 \text{ MB} \]

- Is it worth keeping the whole image data for registration?
  \[ \rightarrow \text{Yes: Pixel-Based approaches (or ‘direct’)} \]

- Can we ‘abstract’ the images by visual cues?
  \[ \rightarrow \text{Yes: Feature-Based approaches} \]
Feature-Based versus Pixel-Based Approaches
Feature-Based versus Pixel-Based Approaches

Feature-based: transfer keypoints

Pixel-based: transfer the whole image
## Feature-Based versus Pixel-Based Approaches

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature-based</td>
<td>Wide</td>
<td>Lower than pixel-based</td>
</tr>
</tbody>
</table>
| Pixel-based       | • Small (from scratch)  
                     • Wide (if bootstrapped) | Higher than feature-based |

→ Bootstrap pixel-based by feature-based estimation
Lecture Roadmap

- Overview and general points
- Parameterized warps
- General points on estimation
- Feature-based estimation
- Pixel-based estimation
- Conclusions
Some Image Features
Basic Workflow – Example 1

Keypoint detection

Keypoint matching

LLS warp estimation
Basic Workflow – Example 2

1. Keypoint detection
2. Keypoint matching
3. LLS warp estimation
Robust Workflow – Example 1

Keypoint matching → Robust matching → LLS warp estimation
Robust Workflow – Example 2

Keypoint matching  →  Robust matching  →  LLS warp estimation
What are Keypoints?

• Also termed corner, interest, salient points,…

• Intuitively
  – Corners or junctions of contours
  – Locations where the image changes significantly

• They should be
  – Repeatable (stable over appearance variation)
  – Accurately localised
  – Discriminable
Detecting and Describing Keypoints

- Look into a small ‘window’ around each image pixel

- Lots of methods; to name just a few
  - Harris and variants [Mikolajczyk et al, IJCV’05]
  - SIFT [Lowe, IJCV’04], SURF [Bay et al, CVIU’08]
  - MSER [Matas et al, BMVC’02]

- Other approaches: parametric shape models, phase congruency, morphology, recognition [Ozuysal et al, PAMI’10], ...
Putative Matching

- Associated with each keypoint is a descriptor
- Simple matching: Winner-Takes-All
- There are always mismatches
Robust Estimation: Global Methods

• RANSAC not applicable (only for low-order global models)

• M-estimators
  – Inhibitate ‘large’ residuals
  – Replace the square loss by an M-estimator $\rho$
  – Estimation by annealing [Pilet et al, IJCV’08]
  – Estimation by combining with a pixel-based cost [Brunet et al, VMV’10]

• Global methods do not handle self-occlusions
Robust Estimation: Local Methods

• Clean up putative matches
  – Define neighborhood using Delaunay triangulation in the template
  – Select correct matches as those which are consistent with their neighbors
  – Consistency is measured using a low-order warp

• Guided matching
  – Re-define neighborhood using correct matches only
  – For each discarded match
    • Classify as correct if consistent with the correct matches
    • Update the neighborhood

[Pizarro et al, 3DPVT’10]
Robust Estimation Results
Robust Estimation Results

Inaccuracies are due to self-occlusions
Self-Occlusions

The global smoother $\mathcal{E}_s$ flattens out the whole warp.

Smoothing should be stronger along the self-occlusion boundary and/or not orthogonally!

Solution [Gay-Bellile et al, PAMI’10]: inhibitate ‘orthogonal smoothing’ with a shrinker.
Solution [Pizarro et al, 3DPVT’10]: overweight the smoother for self-occluded pixels.
Detecting Self-Occlusions

\[ J(q; u) \overset{\text{def}}{=} \frac{\partial W}{\partial q}(q; u) \]

be the warp's Jacobian matrix

The criterion for self-occlusion is
\[ \text{sign}(\lambda_1)\text{sign}(\lambda_2) \min(|\lambda_1|, |\lambda_2|) < 0.1 \]

where \( \lambda_1 \) and \( \lambda_2 \) are the eigenvalues of \( J(q; u) \)
Self-Occlusion Detection
Estimation

- Modify the smoothing term

\[ \tilde{E}_s(U') \defeq \int \kappa(q) \left\| \frac{\partial W}{\partial q^d}(q; U') \right\|_F^2 \, dq \]

where \( \kappa(q) \defeq K \mathcal{H}(q) + (1 - \mathcal{H}(q)) \)

(\( \mathcal{H} \) is the self-occlusion map and \( K \in \mathbb{R}^+ \))

- The warp is affine (and thus constant) in the self-occluded areas

- Still Linear Least Squares!

\[ \tilde{E}_s(U') \approx \left\| B_k U' \right\|_F^2 \]
Self-Occlusion Resistant Warp
Pros and Cons of using Features

■ Pros
  ■ Warp estimation insensitive to appearance change
  ■ Handles wide-baseline

■ Cons
  ■ Does not use all the data
  ■ Requires feature detection and matching
    (which is sensitive to appearance change)
Lecture Roadmap

• Overview and general points
• Parameterized warps
• General points on estimation
• Feature-based estimation
• **Pixel-based estimation**
• Conclusions
Pixel-Based Data Terms

Idea: warp the current image toward the template and compare them

Example: Sum of Squared Differences (SSD)

\[ E_d(u) = \sum_{q \in \mathbb{R}} \|T(q) - I(W(q; u))\|^2 \]

Current image \( I \)

Warped image \( W(\cdot; u) \)

Template \( T \)

Difference image
Matching the Colors

Explicit modeling of a photometric transformation $\mathcal{P} : \mathbb{R}^c \times \mathbb{R}^\bullet \rightarrow \mathbb{R}^c$

$$\mathcal{E}_d(u) = \sum_{q \in \mathcal{R}} \| \mathcal{T}(q) - \mathcal{P}(\mathcal{I}(\mathcal{W}(q; u_g)); u_p) \|^2$$

Invariance to lighting $\mathcal{Q} : \mathbb{R}^c \times \mathbb{R}^\bullet \rightarrow \mathbb{R}^{\bullet < c}$

$$\mathcal{E}_d(u) = \sum_{q \in \mathcal{R}} \| \mathcal{Q}(\mathcal{T}(q); u_p) - \mathcal{Q}(\mathcal{I}(\mathcal{W}(q; u_g)); u_p) \|^2$$
Explicit Photometric Transformation

Template

Current image

Videos of the difference image

No photometric model

Affine photometric model

[Bartoli, PAMI’08]
# The Photometric Transformation

We use **global transformations** (independent of the pixel location)

Let \( \mathbf{v} = \mathcal{I}(\mathbf{q}) \in \mathbb{R}^3 \) be some color

Some examples are:

<table>
<thead>
<tr>
<th>Single gain and bias</th>
<th>( \mathcal{B}_p(\mathbf{v}; \mathbf{u}_p) = \begin{pmatrix} \alpha &amp; \alpha &amp; \alpha \ \alpha \ \alpha \end{pmatrix} \mathbf{v} + \begin{pmatrix} \beta \ \beta \ \beta \end{pmatrix} )</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Multiple gains and biases</th>
<th>( \mathcal{B}_p(\mathbf{v}; \mathbf{u}_p) = \begin{pmatrix} \alpha_R &amp; \alpha_G &amp; \alpha_B \ \alpha_G \ \alpha_B \end{pmatrix} \mathbf{v} + \begin{pmatrix} \beta_R \ \beta_G \ \beta_B \end{pmatrix} )</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Full affine channel mixing</th>
<th>( \mathcal{B}<em>p(\mathbf{v}; \mathbf{u}<em>p) = \begin{pmatrix} \alpha</em>{RR} &amp; \alpha</em>{RG} &amp; \alpha_{RB} \ \alpha_{GR} &amp; \alpha_{GG} &amp; \alpha_{GB} \ \alpha_{BR} &amp; \alpha_{BG} &amp; \alpha_{BB} \end{pmatrix} \mathbf{v} + \begin{pmatrix} \beta_R \ \beta_G \ \beta_B \end{pmatrix} )</th>
</tr>
</thead>
</table>

There also exist **non-global transformations** (depend on the pixel location)

- Account for cast shadows, highlight, . . .
- We prefer to resort to **light invariant images**
Light-Invariant Registration

Template → SSD registration → Light-invariant registration → Current image

Videos of the difference image

[Pizarro et al, SCIA’07, CVPR’08]
Mutual Information

- A pixel-based image similarity measure
- Invented by Shannon in 1948
- Handles complex relationships between the intensities
- Requires no a priori photometric model
- Ignoring some constants, MI is approximated by the joint entropy of the two images
- A good survey is [Pluim et al., TMI ’03]
Pros and Cons of using Pixel Values

- **Pros**
  - Uses all the data
  - More accurate (than using features) in some cases
  - No matching step required

- **Cons**
  - Sensitive to appearance change
  - Small baseline only (with some exceptions)
Iterative Versus Direct Methods

• **Iterative methods**
  – Need an initial estimate
  – Update the estimate until convergence
  – Are generic and flexible
  – May get trapped in a local minimum

• **Direct methods**
  – Are generally quite *ad hoc*
  – Find the global minimum

For a recent MRF-based direct method, see [Glocker et al, MIA’08]
Example

$$\min_u \mathcal{E}(u; \lambda)$$

SSD + first order smoothing

$$\mathcal{E}(u; \lambda) = \sum_{q \in \mathcal{R}} \| \mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u)) \|^2 + \lambda \sum_{q \in \mathcal{R}} \left\| \frac{\partial \mathcal{W}}{\partial q} \right\|^2$$

Nonlinear Least Squares

Linear Least Squares
Gauss-Newton Optimization

Additive update rule: \( u \leftarrow u + \delta \) – iterate updates until convergence

The problem (at each iteration) becomes:
\[
\min_{\delta} \mathcal{E}(u + \delta; \lambda)
\]

The regularization term is LLS:
\[
\mathcal{E}_s(u + \delta) = \| B\delta + Bu \|^2
\]

The data term is NLS; Gauss-Newton approximation:
\[
\mathcal{E}_d(u + \delta) = \sum_{q \in \mathcal{R}} \| \mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u + \delta)) \|^2
\]

\[
\approx \sum_{q \in \mathcal{R}} \left\| \mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u)) - \frac{\partial \mathcal{I}}{\partial q}(\mathcal{W}(q; u)) \frac{\partial \mathcal{W}}{\partial u}(q; u) \delta \right\|^2
\]

- The difference of error image
- The warped image
- Warped spatial derivatives of the image
- Derivatives of the warp wrt the parameters

Relates the parameters to the warped image
Example

Video overlaid with the visualization grid

Video: retexturing
Examples

Videos: original, visualization grid and retexturing

[Tiilikainen et al, BMVC’08]
Video: deformation retargetting and retexturing
What About Occlusions?

Video: failure in case of an external occlusion
Robustification with an M-Estimator

Idea: inhibit the data terms that get ‘too large’

The square loss function is replaced by an M-estimator \( \rho \)

\[
\mathcal{E}_d(u) = \sum_{q \in \mathcal{R}} \left\| \mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u)) \right\|^2
\]

\[
\mathcal{E}_d(u) = \sum_{q \in \mathcal{R}} \rho(\mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u)))
\]
Robustifying

<table>
<thead>
<tr>
<th>Basic</th>
<th>Data term</th>
<th>Smoother</th>
</tr>
</thead>
<tbody>
<tr>
<td>A pixel of interest</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Classical robustification methods

<table>
<thead>
<tr>
<th>Robustified</th>
<th>Data term</th>
<th>Smoother</th>
</tr>
</thead>
<tbody>
<tr>
<td>A visible pixel of interest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>An occluded pixel of interest</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Iteratively Reweighted Least Squares (IRLS)

\[ \mathcal{E}_d(u) = \sum_{q \in \mathcal{R}} \rho(\mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u))) \]

\[ = \sum_{q \in \mathcal{R}} w(||\mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u)||)||\mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u))||^2 \]

with \( w \) the weight function associated to the M-estimator \( \rho \)

IRLS trick: alternate weight and parameter estimation

\[ \mathcal{E}_d(u + \delta) \approx \sum_{q \in \mathcal{R}} w(||\mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u)||)||\mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u + \delta))||^2 \]

Plugging the Gauss-Newton approximation gives:

\[ \approx \sum_{q \in \mathcal{R}} w(||\mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u)||)||\mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u)) - J(q; u)\delta||^2 \]

which is weighted Linear Least Squares problem
Example

Video: visualization grid and template with outlying pixels in blue
What About Self-Occlusions?

Video: failure in case of a self-occlusion
Folds and loops change the sign of the directional warp derivative along the normal to the occlusion boundary.
Example

Video with self-occlusion: visualization grid and template with self-occluded pixels in white
Example

Video with self-occlusion: visualization grid and template with self-occluded pixels in white and external occluded pixels in blue
Example

Video with self-occlusion: visualization grid and template with self-occluded pixels in white
Retexturing the Surface
Example

Video: retexturing
+ Mickey

Video: retexturing
Comparison Self-Occlusion Approach Versus Simple Robust Optimization

Video: self-occlusion reasoning versus robust method; Visualization grid and template with occluded and self-occluded pixels
Field of View

\[ \mathcal{E}_d(u) = \sum_{q \in \mathcal{R}} \| \mathcal{T}(q) - \mathcal{I}(\mathcal{W}(q; u)) \|^2 \]

What if \( \mathcal{W}(q; u) \notin \mathcal{I} \)?

Solution in [Brunet et al, VMV’10]: registration without region of interest
<table>
<thead>
<tr>
<th>Source image (pattern)</th>
<th>Target image</th>
<th>Warped target</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our approach</strong></td>
<td><img src="image1" alt="Source Image" /></td>
<td><img src="image2" alt="Target Image" /></td>
</tr>
<tr>
<td><strong>Rectangular Roll (large margin)</strong></td>
<td><img src="image4" alt="Source Image" /></td>
<td><img src="image5" alt="Target Image" /></td>
</tr>
<tr>
<td><strong>Rectangular Roll (small margin)</strong></td>
<td><img src="image7" alt="Source Image" /></td>
<td><img src="image8" alt="Target Image" /></td>
</tr>
</tbody>
</table>
Conclusions

• Linear Basis Expansion warps very practical (LLS, generic)
• Estimation by threading feature-based and pixel-based
• Both are sensitive to occlusions and self-occlusions
• We assumed all hyperparameters known (M-estimator threshold, number of driving features, smoothing weight, ...) – how to find them in practice?
Take-Home Messages

• **Overview** « deformable image registration is a difficult problem due to appearance variation and weak prior knowledge. »

• **Parametric warps** « a warp is a function that models the inter-image displacement field; there exist various parametric warp models. Most warps can be modeled by Linear Basis Expansion. Estimation from landmarks is Linear Least Squares. »

• **Estimation** « two complementary approaches: feature- and pixel-based. The feature-based approach detects and matches features (usually keypoints) and then estimates the warp. It handles wide baseline and can bootstrap the pixel-based approach which is generally more accurate. The pixel-based approach directly estimates the warps by comparing the pixels’ values. »
• **Overview and general points**
  – The deformable image registration problem
  – The appearance may vary strongly
  – Priors: statistical (e.g., AAMs), physical (e.g., rigidity, smoothness)
  – Problem statement using optimization
  – Image modeling

• **Parametric warps**
  – Definition
  – Smoothness and piecewise smoothness, self-occlusions
  – The Flow-Field
  – Mesh Interpolation warps
  – Radial Basis Function warps
  – Tensor-Product warps
  – Generalized warps
  – Estimation from landmarks

• **General points on estimation**
  – Feature-based vs pixel-based
  – Bootstrapping

• **Feature-based estimation**
  – Estimation from clean point matches
  – Keypoint detection and matching
  – Robustness
  – Self-occlusions

• **Pixel-based estimation**
  – Basic iterations
  – Photometric variations
  – Occlusions and self-occlusions
Deformable Image Registration

Adrien Bartoli
ALCoV – ISIT
Université d’Auvergne
Clermont-Ferrand, France

IIT, Genoa, December 2010