More Morphing

19 May 2015

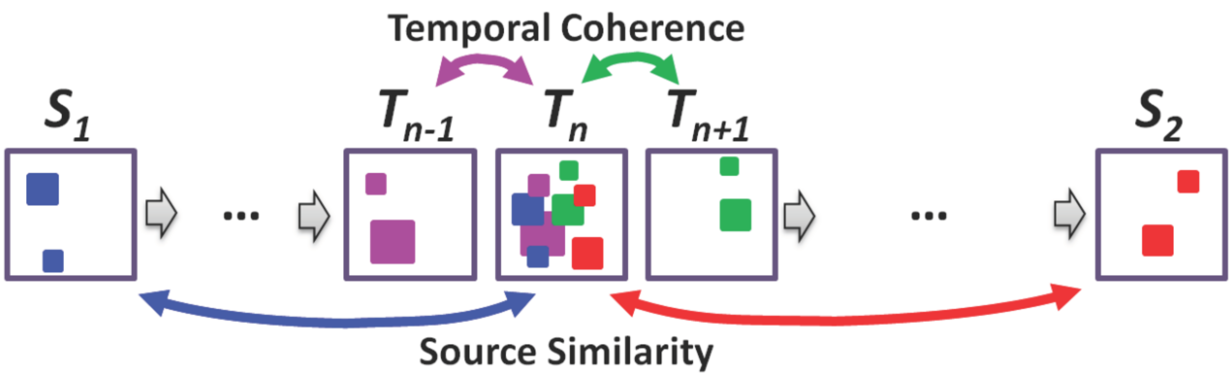
- Regenerative Morphing
  - *Schechtman et al., CVPR 2010*

- Automating Image Morphing using Structural Similarity on a Halfway Domain
  - *Liao et al., SIGGRAPH 2014*

Regenerative Morphing

- Bidirectional similarity
- Does not require manual correspondence
- Different parts of the scene move at different rates

Morphing as an example-based optimization



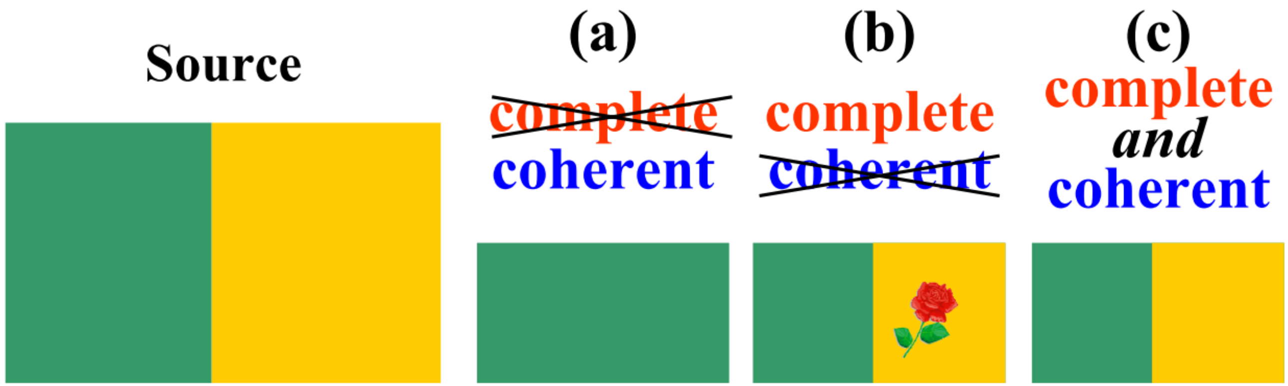
$$E_{Morph}(T_1, ..., T_{N-1}) =$$
$$E_{SourceSim}(T_1, ..., T_{N-1}) + \beta E_{TempCoher}(T_1, ..., T_{N-1})$$

7x7 patch

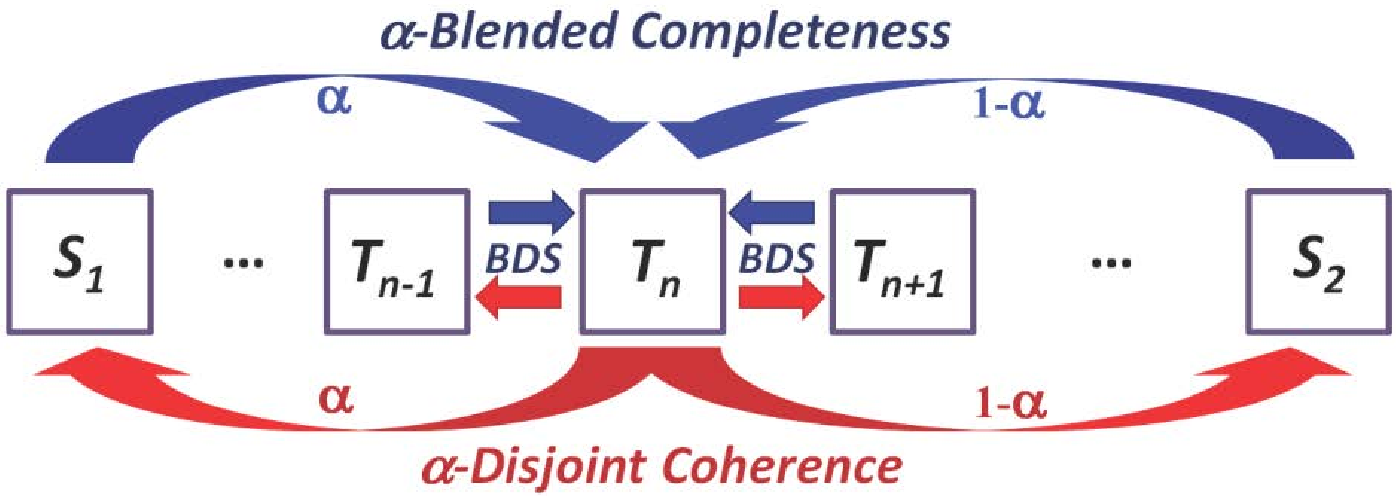
- Temporal coherence
  - Changes should be smooth
- Source similarity
  - Every region in every frame should be similar to some region in either of the sources

Bidirectional similarity

$$d_{BDS}(S,T) = \overbrace{\frac{1}{N_S}\sum_{s\subset S}\min_{t\subset T} D(s,t)}^{d_{Complete}(S,T)} + \overbrace{\frac{1}{N_T}\sum_{t\subset T}\min_{s\subset S} D(t,s)}^{d_{Coher}(S,T)}$$



Relative similarity with multiple sources



Alpha-blended bidirectional similarity

$$d_{\alpha BlendBDS}(T, S_1, S_2, \alpha) = \\ \alpha \cdot d_{BDS}(T, S_1) + (1-\alpha) \cdot d_{BDS}(T, S_2)$$

$$d_{\alpha BlendCohere}(T, S_1, S_2, \alpha) = \\ \alpha \cdot d_{Coher}(T, S_1) + (1-\alpha) \cdot d_{Coher}(T, S_2)$$

$$d_{\alpha BlendComplete}(T, S_1, S_2, \alpha) = \\ \alpha \cdot d_{Complete}(T, S_1) + (1-\alpha) \cdot d_{Complete}(T, S_2).$$

$$\boxed{\begin{gathered} d_{\alpha BlendComplete}(T, S_1, S_2, \alpha) = \\ \frac{\alpha}{N_{S_1}} \sum_{s_1 \subset S_1} \min_{t \subset T} D(s_1, t) + \frac{1-\alpha}{N_{S_2}} \sum_{s_2 \subset S_2} \min_{t \subset T} D(s_2, t) \end{gathered}}$$

disjoint

$$d_{\alpha BlendCoher}(T, S_1, S_2, \alpha) = \\ \frac{1}{N_T} \sum_{t \subset T} \left(\alpha \min_{s_1 \subset S_1} D(s_1, t) + (1-\alpha) \min_{s_2 \subset S_2} D(s_2, t) \right)$$

Alpha-disjoint coherence

$$d_{\alpha BlendCoher}(T, S_1, S_2, \alpha) =$$
$$\frac{1}{N_T} \sum_{t \subset T} \left(\alpha \min_{s_1 \subset S_1} D(s_1, t) + (1 - \alpha) \min_{s_2 \subset S_2} D(s_2, t) \right)$$

$$\boxed{\begin{gathered} d_{\alpha DisjCoher}(T, S_1, S_2, \alpha) = \\ \frac{1}{N_T} \sum_{t \subset T} \min \left(\min_{s_1 \subset S_1} D(s_1, t), \min_{s_2 \subset S_2} D(s_2, t) + D_{bias}(\alpha) \right) \end{gathered}}$$

The morphing objective function

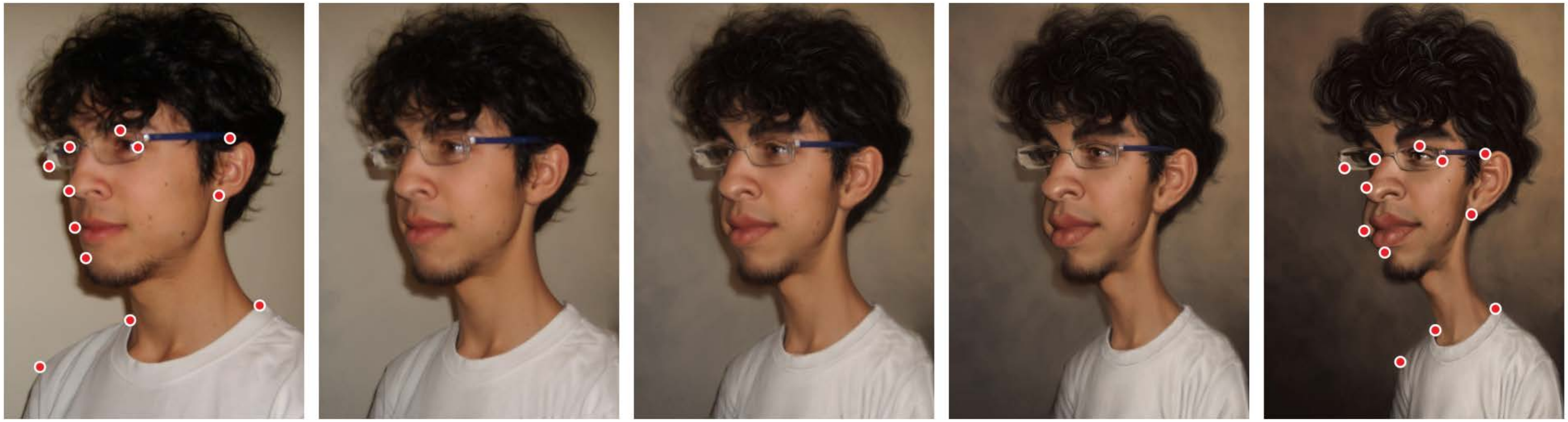
$$E_{Morph} = E_{TempCoher} + \beta E_{SourceSim}$$

$$E_{TempCoher}(\{T_n\}_1^{N-1}) = \sum_{n=1}^{N-1} d_{\alpha BlendBDS}(T_n, T_{n-1}, T_{n+1}, 0.5)$$

$$E_{SourceSim}(\{T_n\}_1^{N-1}) = \sum_{n=1}^{N-1} d_{\alpha BlendComplete}(T_n, S_1, S_2, \tfrac{n}{N}) + d_{\alpha DisjCoher}(T_n, S_1, S_2, \tfrac{n}{N})$$

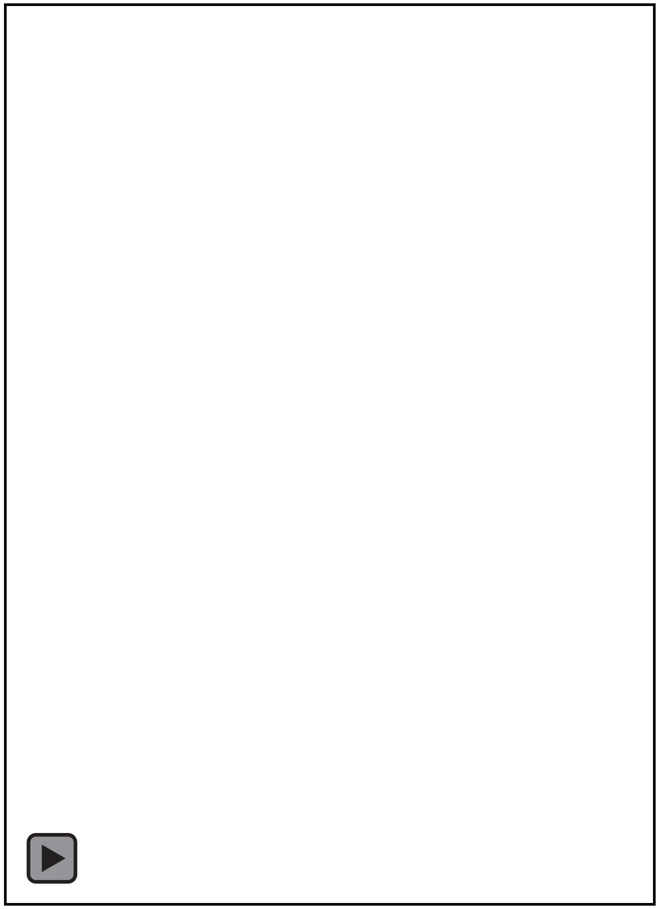
Results

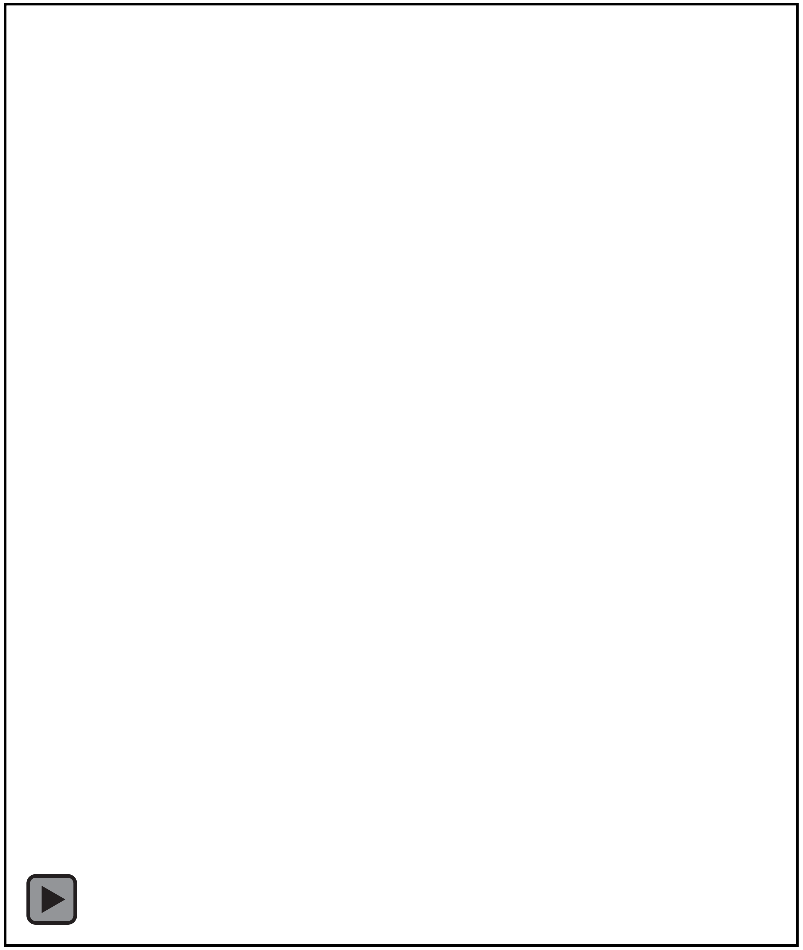
Structural Similarity on a Halfway Domain



- Optimization based
- Automatically aligning image structures
- A small number of point correspondences
- Quadratic motion paths

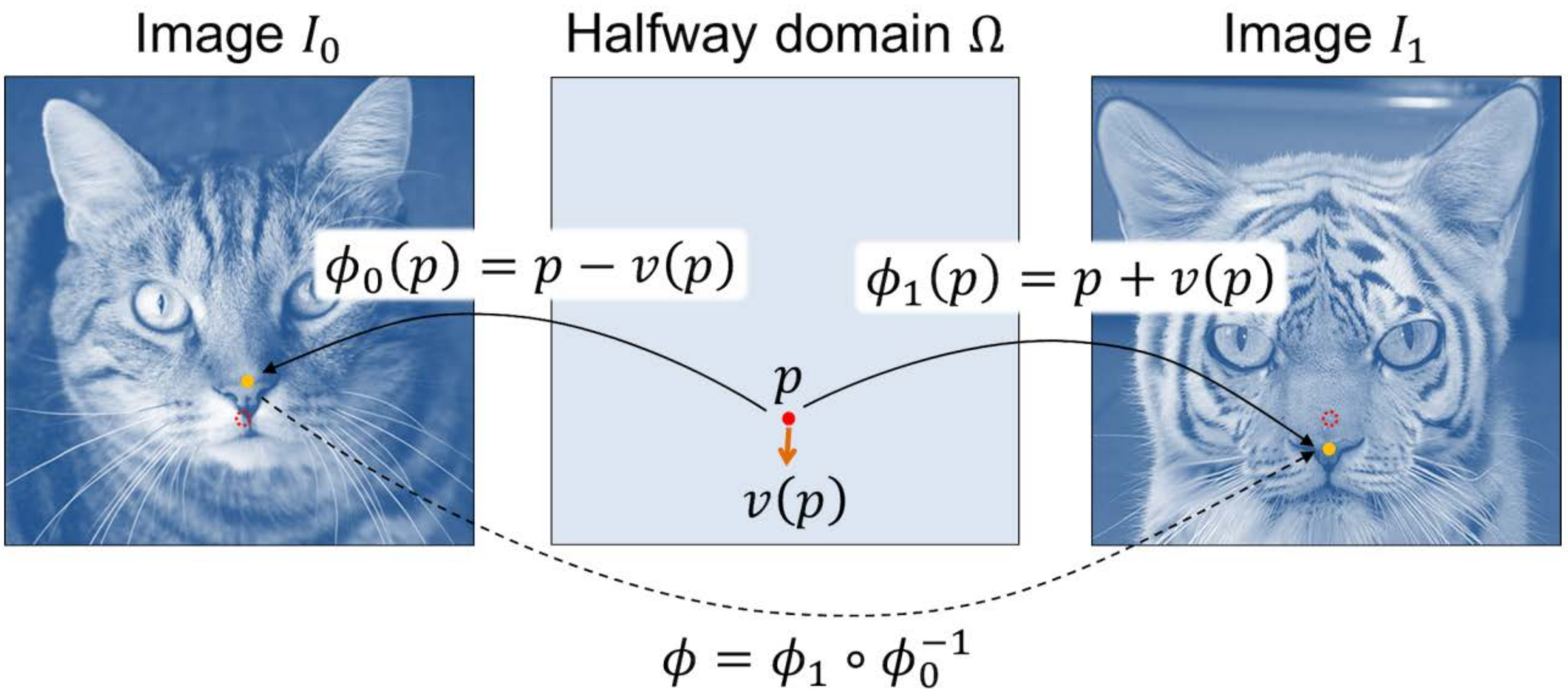
Examples



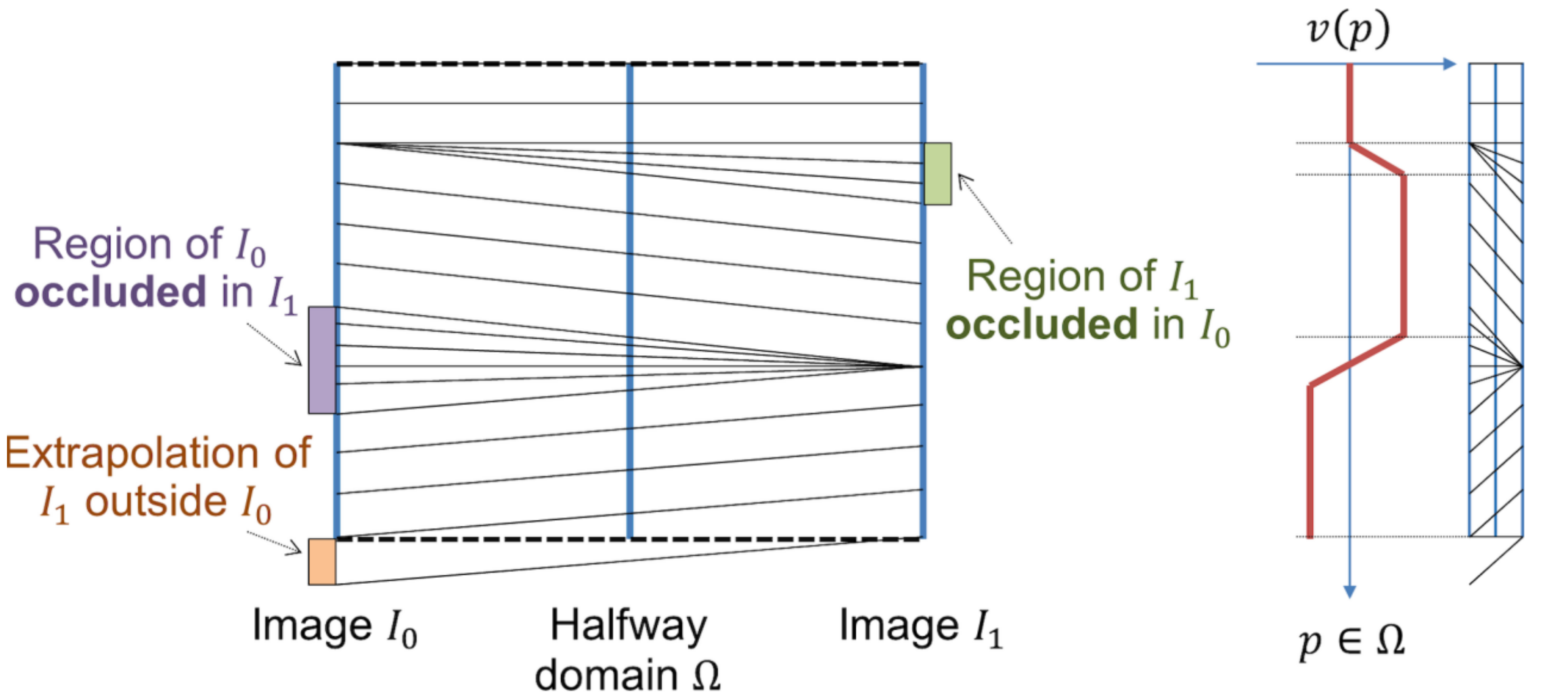


Halfway Parameterization

- Inter-image map



Continuous vector field under simple occlusion



Correspondence optimization

- Energy function

$$E = \sum_{p \in \Omega} E(p), \quad \text{with}$$

$$E(p) = E_{SIM}(p) + \underset{\substack{\uparrow \\ 0.001}}{\lambda} E_{TPS}(p) + \underset{\substack{\uparrow \\ 100}}{\gamma} E_{UI}(p).$$

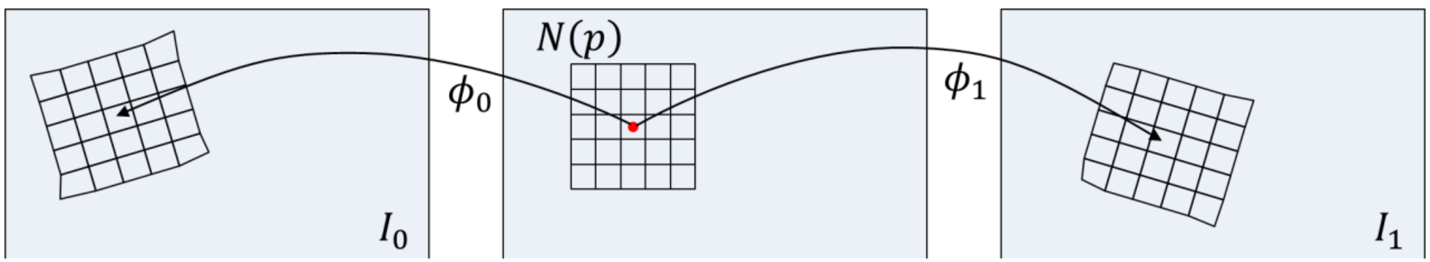
- Similarity, smoothness, user-specified

Similarity energy

- Modified structural similarity index (SSIM)

$$E_{SIM}(p) = -\frac{1}{wh}\,\mathrm{SIM}(N_0, N_1), \quad \text{with}$$

$$N_0 = I_0\big(\phi_0(N(p)\big) \quad \text{and} \quad N_1 = I_1\big(\phi_1(N(p)\big)$$



$N_0 = I_0\big(\phi_0(N(p))\big)$ $N_1 = I_1\big(\phi_1(N(p))\big)$

Modified SSIM (no luminance term)

- In terms of the means, variances, and covariances of the pixel values

$$\mathrm{SIM}(N_0, N_1) = c(N_0, N_1) \cdot s(N_0, N_1), \quad \text{with}$$

contrast
$$c(N_0, N_1) = \frac{2\sigma_{N_0}\sigma_{N_1} + C_2}{\sigma_{N_0}^2 + \sigma_{N_1}^2 + C_2} \quad \text{and}$$

structure
$$s(N_0, N_1) = \frac{|\sigma_{N_0 N_1}| + C_3}{\sigma_{N_0}\sigma_{N_1} + C_3}.$$

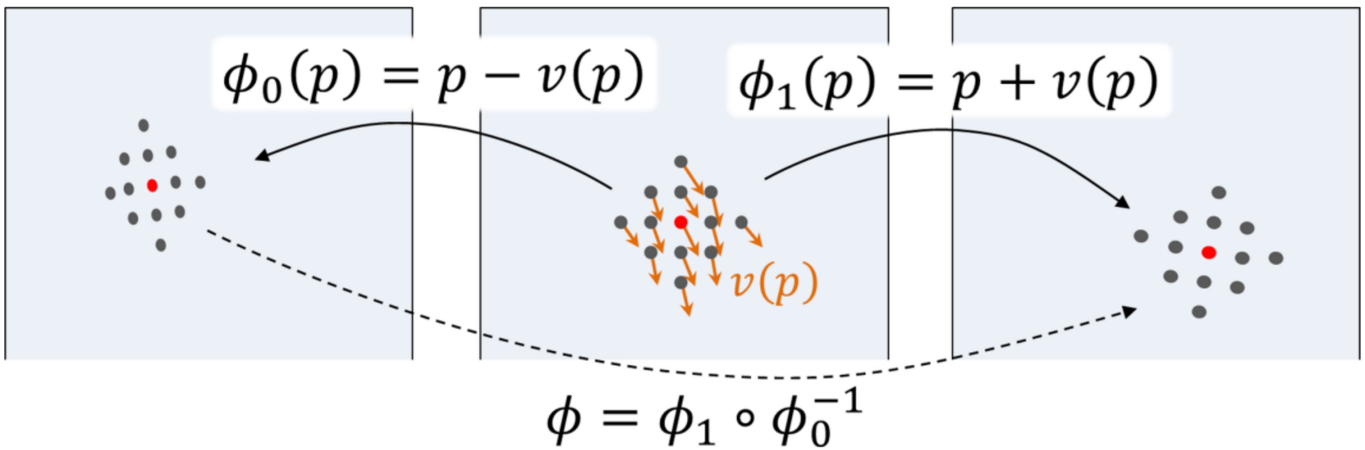
allows swapping

Smoothness energy

› Thin-plate spline (TPS) energy

$$E_{TPS}(p) = \mathrm{TPS}\big(v_x(p)\big) + \mathrm{TPS}\big(v_y(p)\big), \quad \text{where}$$

$$\mathrm{TPS}\big(v_x(p)\big) = \Big(\frac{\partial^2 v_x(p)}{\partial {p_x}^2}\Big)^2 + 2\Big(\frac{\partial^2 v_x(p)}{\partial p_x p_y}\Big)^2 + \Big(\frac{\partial^2 v_x(p)}{\partial {p_y}^2}\Big)^2$$

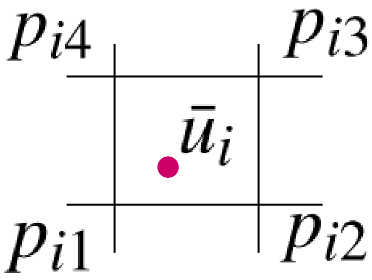


UI energy

$$\bar{u}_i = (u_i^0 + u_i^1)/2 \qquad v_{u_i} = (u_i^1 - u_i^0)/2$$

control points in two images

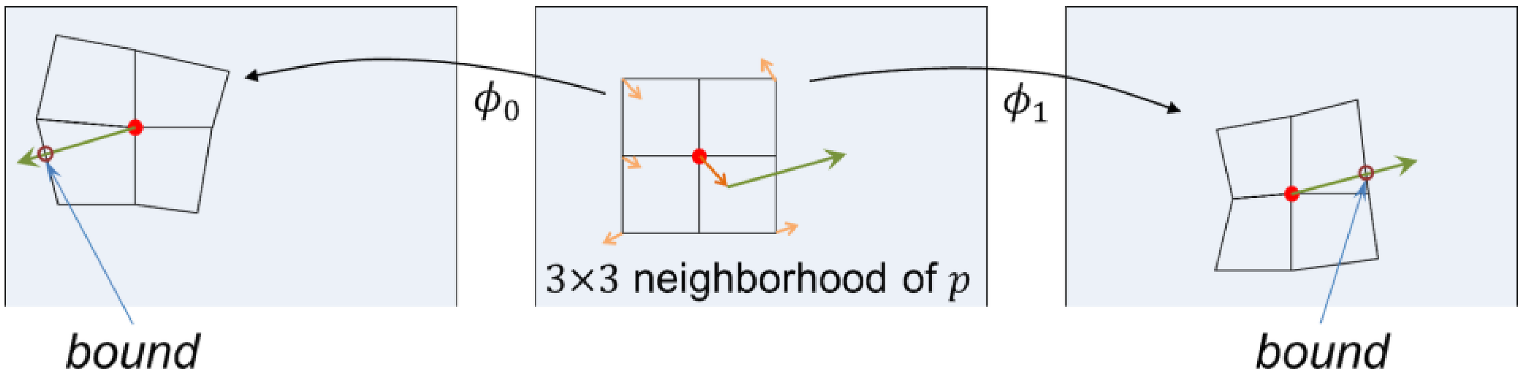
$$\sum_{j=1}^{4} b(p_{ij}, \bar{u}_i)\, p_{ij} = \bar{u}_i.$$



$$E_{UI}(p) = \frac{1}{wh} \sum_{i,j \,|\, p_{ij}=p} b(p_{ij}, \bar{u}_i) \left\| v(p_{ij}) - v_{u_i} \right\|^2.$$

Optimization

- Multiresolution solver
  - Coarse to fine
  - Up-sampling the solved vector field v
- Iterative optimization at each level
  - Finite difference for energy gradient
  - Golden-section search



Quadratic motion paths

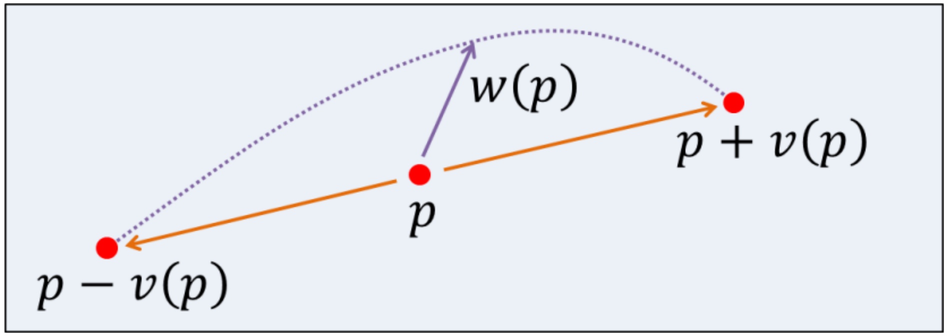
- Cf. linear interpolation

$$q_0 = \phi_0(p) \Rightarrow q_1 = \phi_1(p)$$

$$q_\alpha = p + (2\alpha - 1)v(p)$$

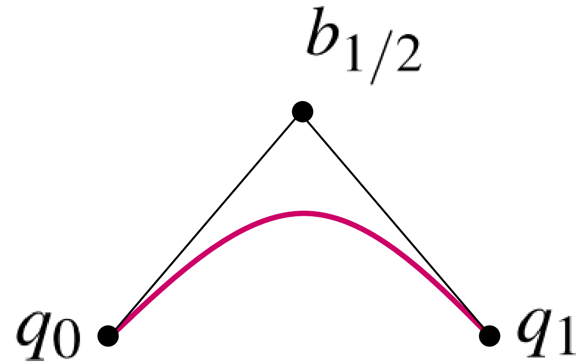
time interval [0, 1]

- Quadratic motion paths



Additional vector $w(p)$

- Control point $b_{1/2} = p + 2w(p)$
  - Quadratic Bezier curve

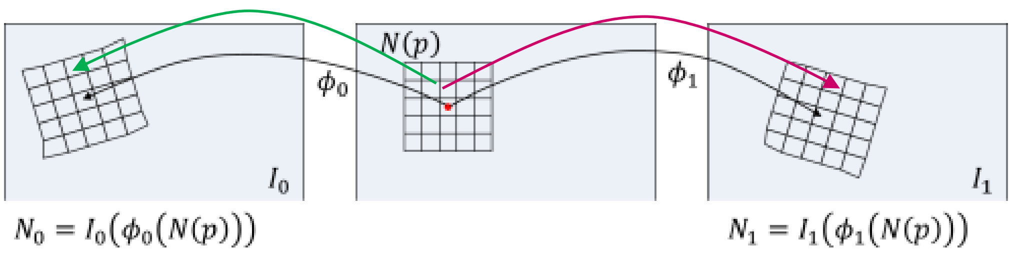


$$q_\alpha = (1-\alpha)\big((1-\alpha)q_0 + \alpha b_{1/2}\big) + \alpha\big((1-\alpha)b_{1/2} + \alpha q_1\big)$$
$$= p + (2\alpha - 1)v(p) + 4\alpha(1-\alpha)w(p).$$

$$q_{1/2} = p + w(p)$$

Computing $w(p)$

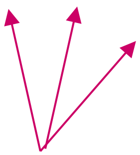
- Pair of neighbors in the halfway domain p_i, p_j



$$
\begin{aligned}
d_0(p_i, p_j) &= \phi_0(p_j) - \phi_0(p_i) \\
&= p_j - p_i - \big(v(p_j) - v(p_i)\big), \\
d_1(p_i, p_j) &= p_j - p_i + \big(v(p_j) - v(p_i)\big).
\end{aligned}
$$

Minimizing $E(w) = E_D(w) + \beta E_R(w)$ for $w(p)$

rotation and scaling



$$\tilde{d}_{1/2}(p_i, p_j) = \sqrt{\|d_0(p_i, p_j)\| \, \|d_1(p_i, p_j)\|} \; \hat{d}_s(p_i, p_j),$$

$$\text{with} \quad d_s(p_i, p_j) = \hat{d}_0(p_i, p_j) + \hat{d}_1(p_i, p_j) \quad \text{and} \quad \hat{u} = u / \|u\|.$$

- Actually obtained from the quadratic path

$$d_{1/2}(p_i, p_j) = p_j - p_i + (w(p_j) - w(p_i)).$$

- Deformation energy

$$E_D(w) = \sum_{p_i, p_j} \left\| d_{1/2}(p_i, p_j) - \tilde{d}_{1/2}(p_i, p_j) \right\|^2.$$

- Resting energy

$$E_R(w) = \sum_{p_i \text{ s.t. } \|v(p_i)\| < 1} \left(1 - \|v(p_i)\|\right) \|w(p_i)\|^2.$$

small v