Leow Wee Kheng

CS4243 Computer Vision and Pattern Recognition

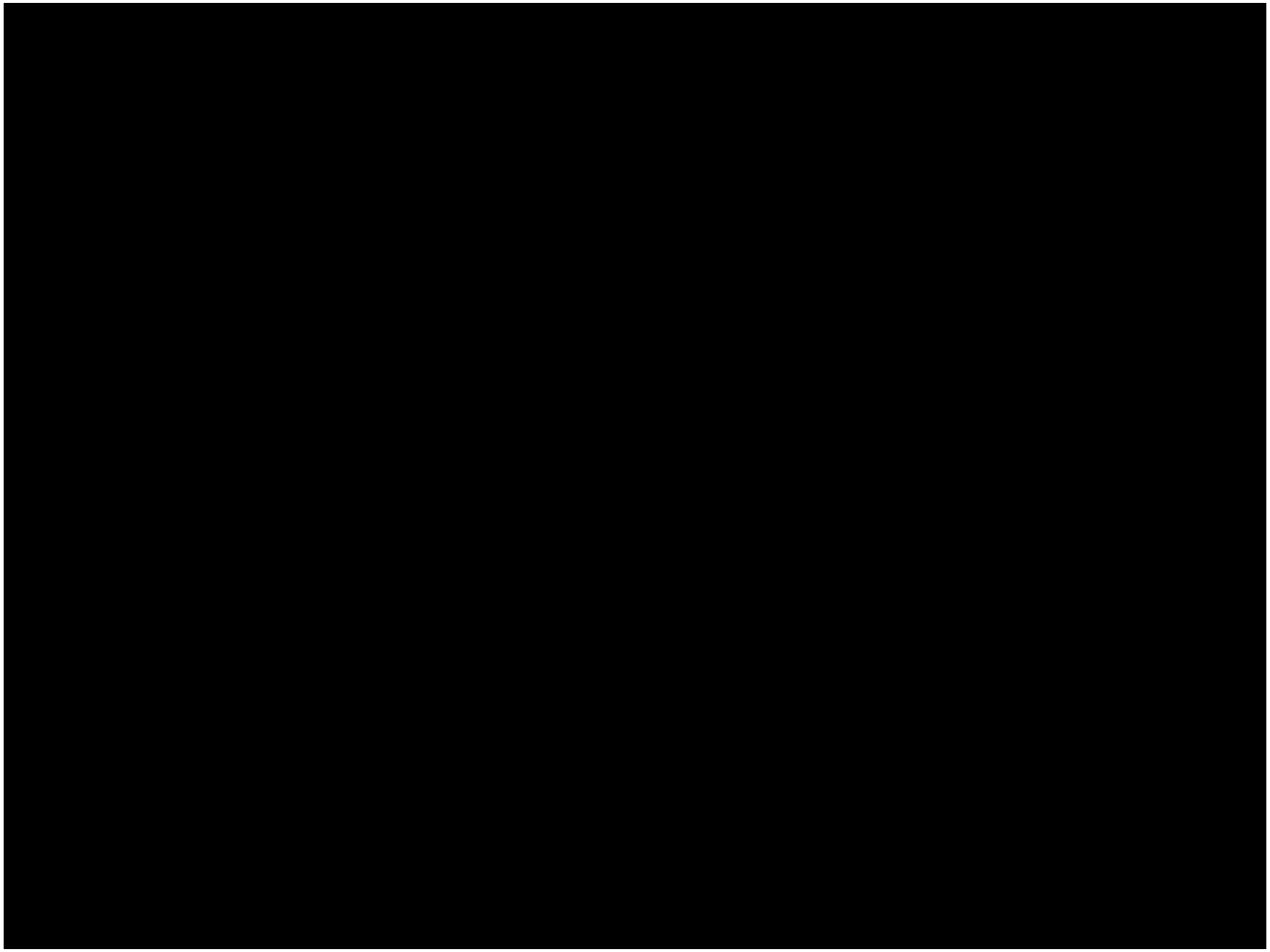
Image Warping and Morphing

Morphing is artistic

Morphing is captivating!

Morphing is dynamic!

Morphing is fun!



Morphing is fun!

OK, that was not really morphing.

But, you get the point.

Two kinds of morphing

- Image morphing
  - 2D

our focus

- Object morphing
  - 2D: like image morphing
  - 3D: more complex

Image Warping & Morphing

- Involves 3 things
  - Spatial transformation
  - Colour transfer
  - Continuous transition

image warping

Image Warping & Morphing

- Involves 3 things
  - Spatial transformation
  - Colour change
  - Continuous transition

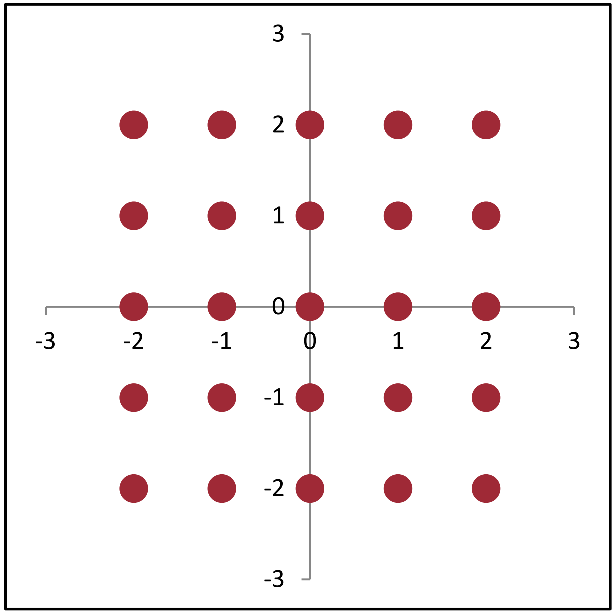
Spatial Transformation

- Simplest form: affine transformation

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

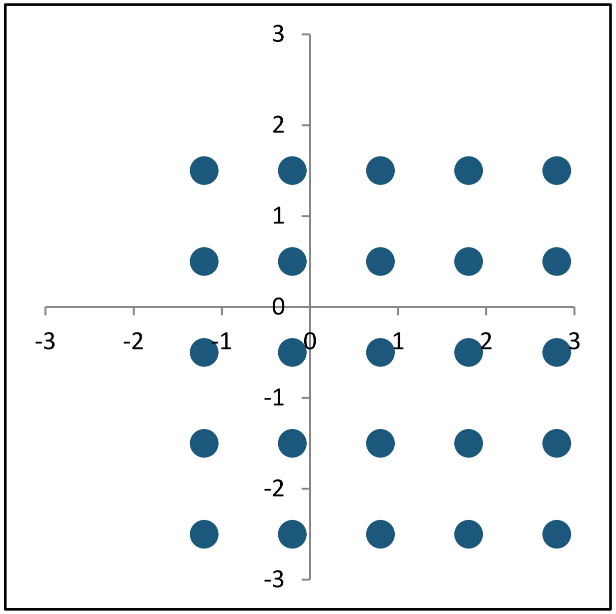
Affine Transformation

Source (x, y)



$$\begin{bmatrix} 1 & 0 & 0.8 \\ 0 & 1 & -0.5 \\ 0 & 0 & 1 \end{bmatrix}$$

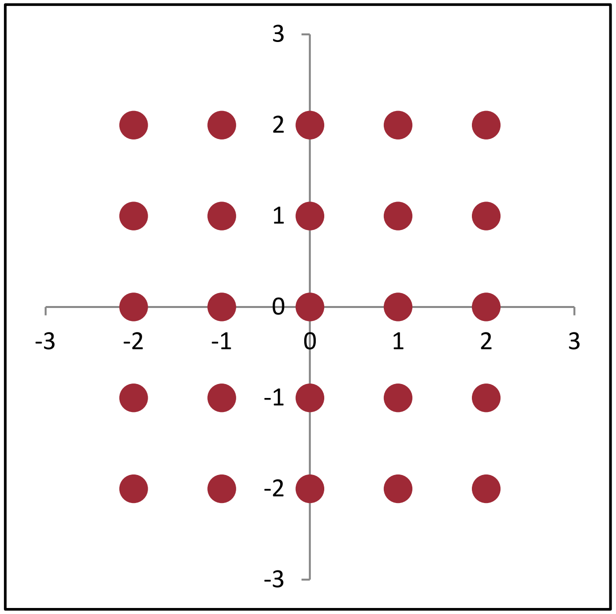
Target (u, v)



translation

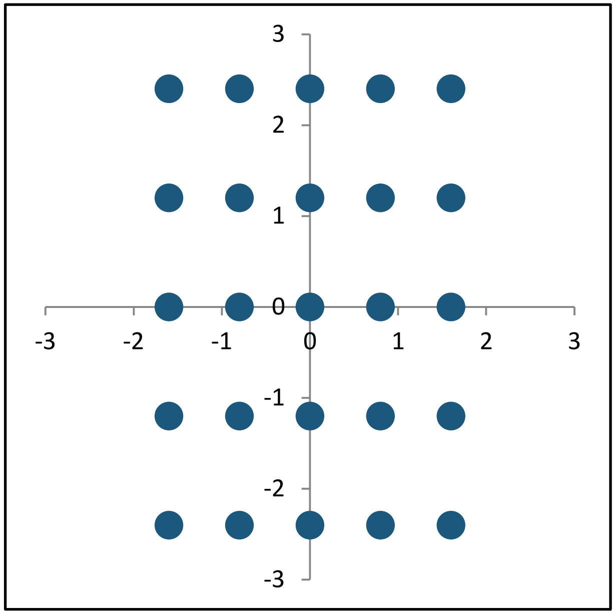
Affine Transformation

Source (x, y)



$$\begin{bmatrix} 0.8 & 0 & 0 \\ 0 & 1.2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

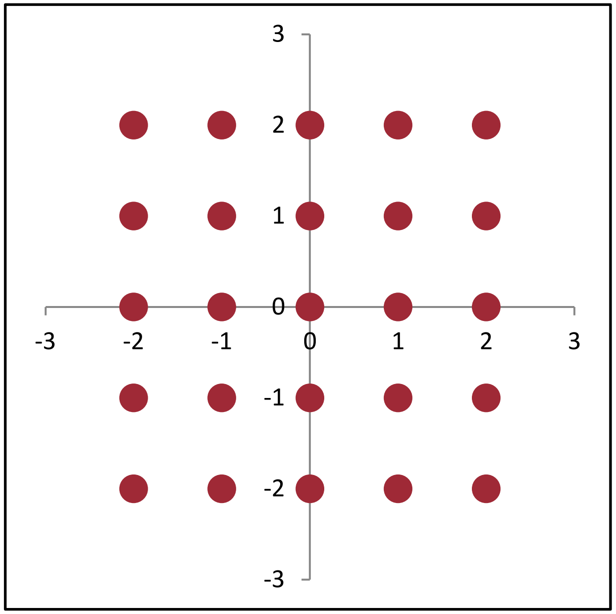
Target (u, v)



scaling

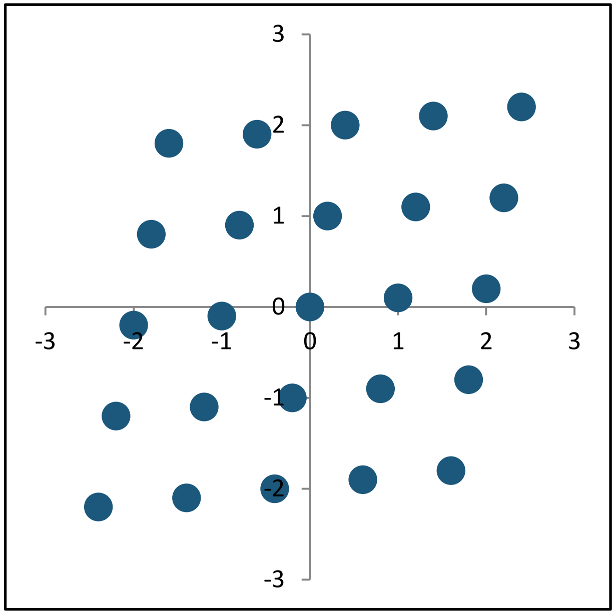
Affine Transformation

Source (x, y)



$$\begin{bmatrix} 1 & 0.2 & 0 \\ 0.1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Target (u, v)

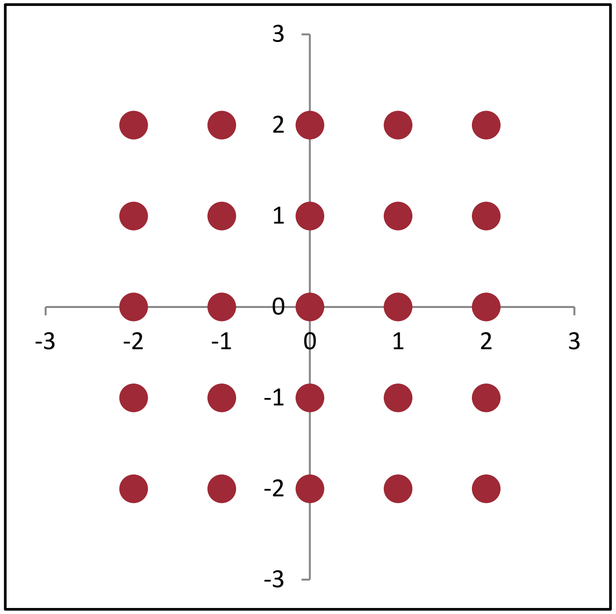


shearing

parallel lines remain parallel

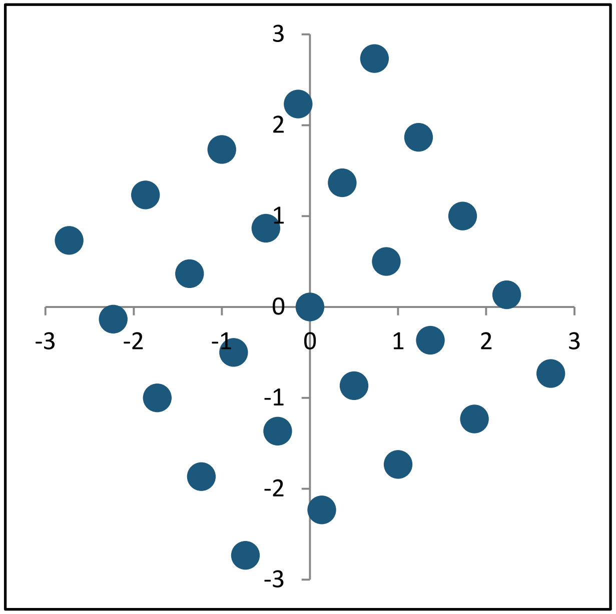
Affine Transformation

Source (x, y)



$$\begin{bmatrix} \cos & -\sin & 0 \\ \sin & \cos & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Target (u, v)



rotation

parallel lines remain parallel

Affine Transformation

- To solve for affine matrix:
  - For $i = 1,\ldots, n$, arrange into two matrix equations:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} = \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}$$

$$\begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a_{21} \\ a_{22} \\ a_{23} \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$$

  - Then, solve each equation using linear least square.

Perspective Transformation

- Generalisation of affine transformation:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

- Looks like a linear equation.
- But, perspective transformation is nonlinear.
- Will discuss further later.

Polynomial Transformation

- Described by polynomial equations

$$
\begin{aligned}
u &= \sum_k \sum_l a_{kl}\, x^k y^l \\
v &= \sum_k \sum_l b_{kl}\, x^k y^l
\end{aligned}
$$

- Example: 2nd-order polynomial, e.g., quadratic

$$
\begin{aligned}
u &= a_{20}x^2 + a_{02}y^2 + a_{11}xy + a_{10}x + a_{01}y + a_{00} \\
v &= b_{20}x^2 + b_{02}y^2 + b_{11}xy + b_{10}x + b_{01}y + b_{00}
\end{aligned}
$$

Quadratic Transformation

- In matrix form

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix} a_{20} & a_{02} & a_{11} \\ b_{20} & b_{02} & b_{11} \end{matrix}} & \begin{matrix} a_{10} & a_{01} & a_{00} \\ b_{10} & b_{01} & b_{00} \end{matrix} \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ xy \\ x \\ y \\ 1 \end{bmatrix}$$

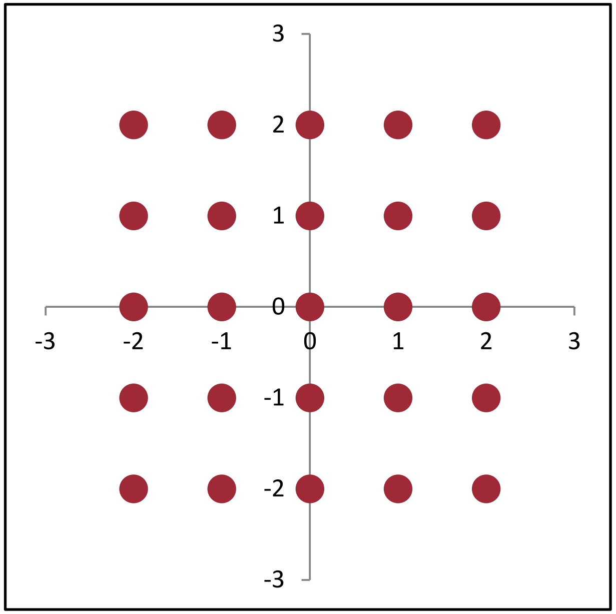
If these are zero,
what is the matrix?

- Solving the parameters is easy!
  - Just like for the affine case (exercise).

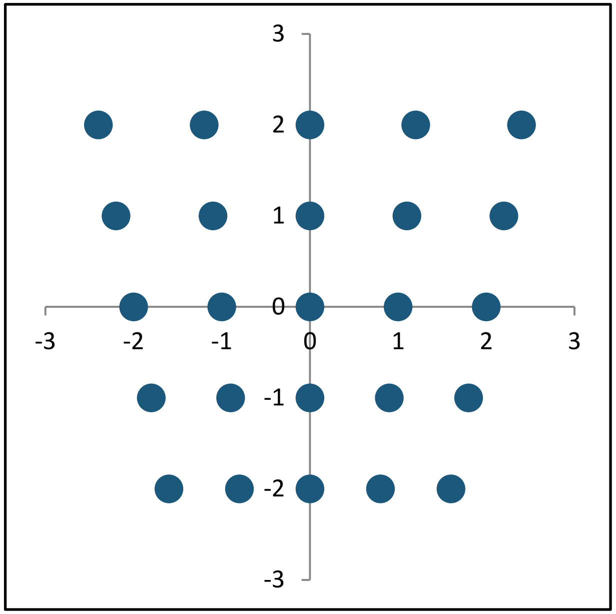
Quadratic Transformation

$$\begin{bmatrix} 0 & 0 & 0.1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Source (x, y)



Target (u, v)

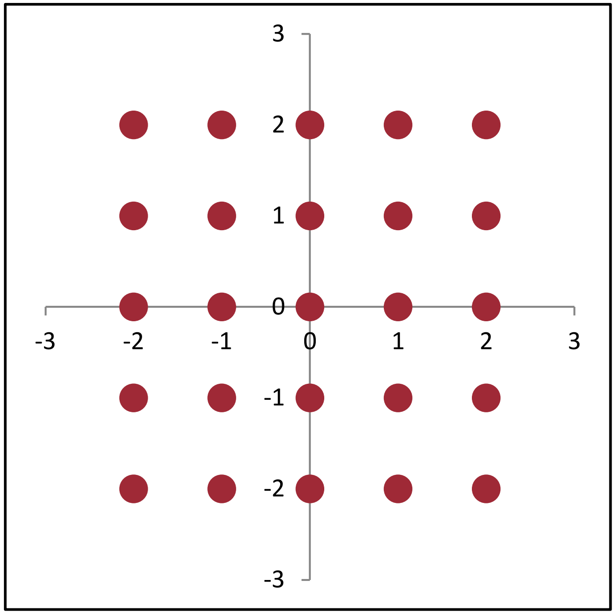


tapering / perspective distortion

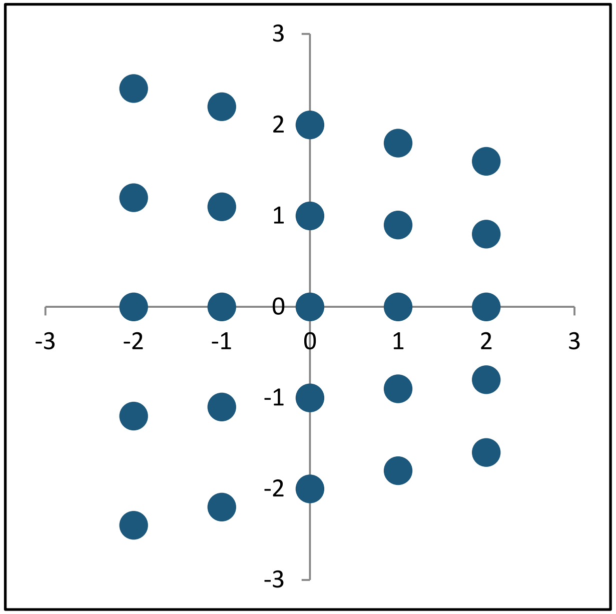
Quadratic Transformation

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -0.1 & 0 & 1 & 0 \end{bmatrix}$$

Source (x, y)



Target (u, v)

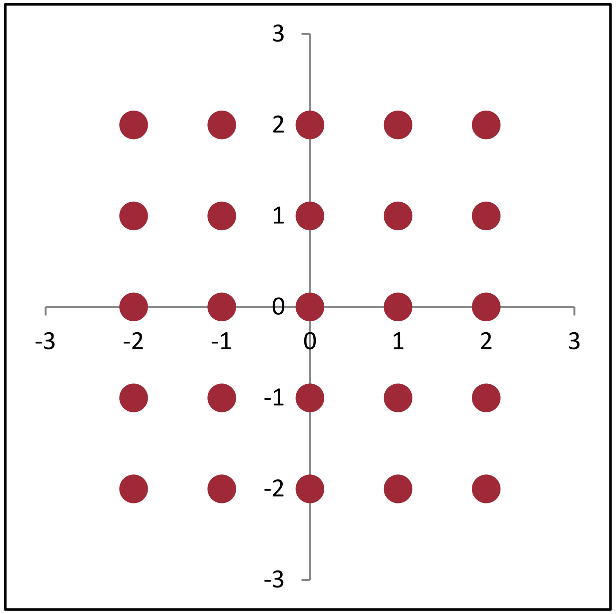


tapering / perspective distortion

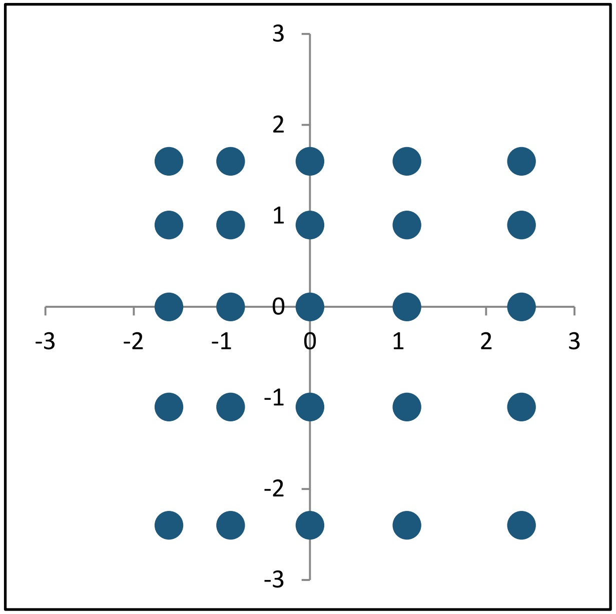
Quadratic Transformation

$$\begin{bmatrix} 0.1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -0.1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Source (x, y)



Target (u, v)

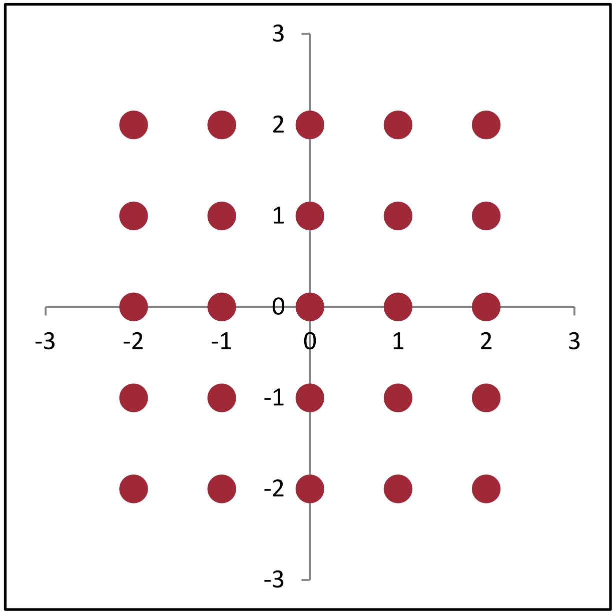


expansion / compression

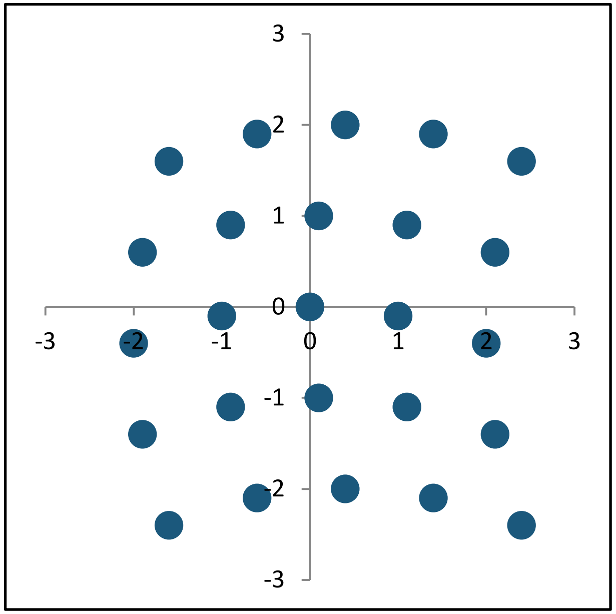
Quadratic Transformation

$$\begin{bmatrix} 0 & 0.1 & 0 & 1 & 0 & 0 \\ -0.1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Source (x, y)



Target (u, v)



warping

General Transformation

- General form of u and v

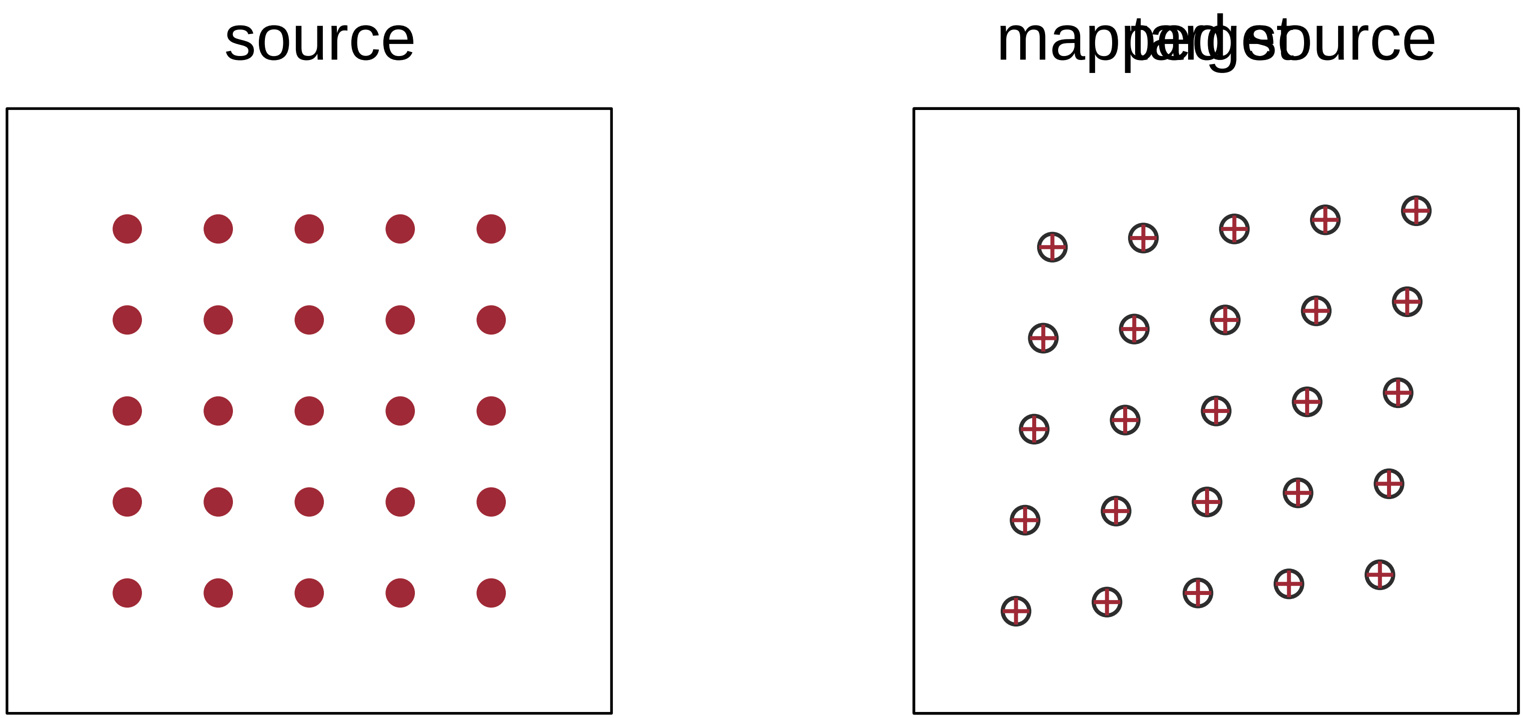
$$u = \sum_{k=1}^{K} \underbrace{a_k}_{\text{parameters}} \underbrace{h_k(x, y)}_{\text{kernel function}}$$

- Possible kernel functions:
  - Affine
  - Polynomial
  - Splines: B-splines, cubic splines, etc.
  - Thin-plate spline: model physical bending energy

Which kernel to use?

- General guidelines
  - Sufficient to capture transformation
  - Not overly complex

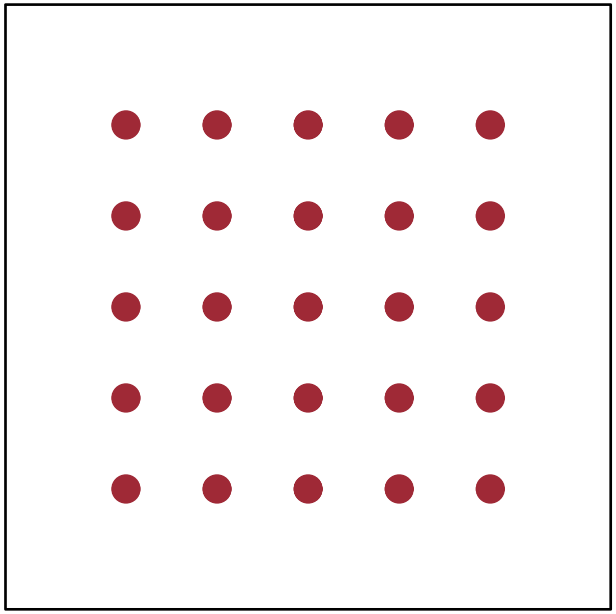
Example



- Fit affine transformation.
- Correct transformation, small error.

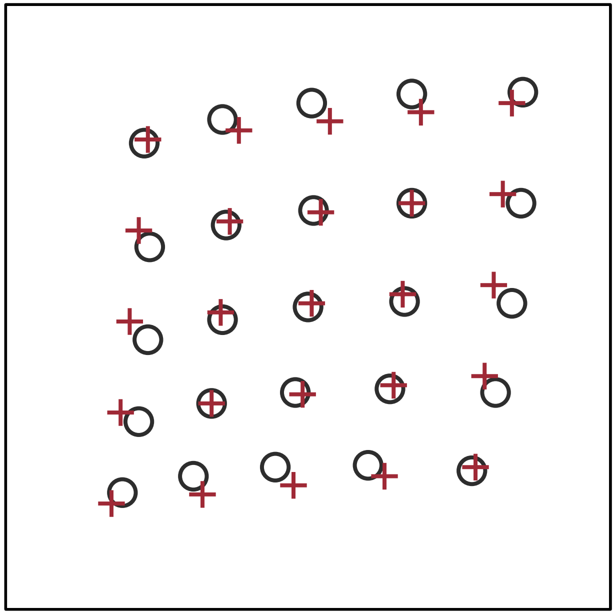
Example

source



mapped source

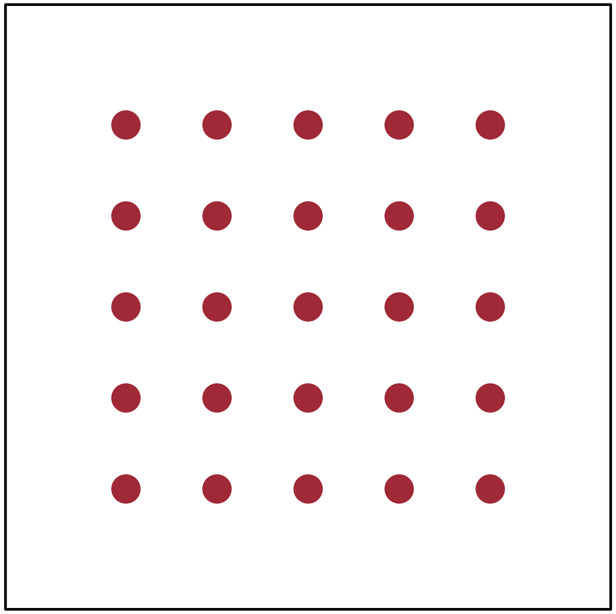
target



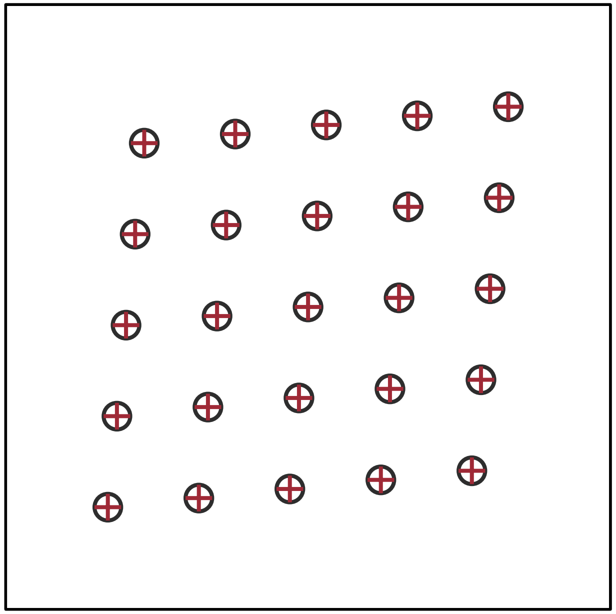
- Fit affine transformation.
- Incorrect transformation, large error.

Example

source



mapped source
target

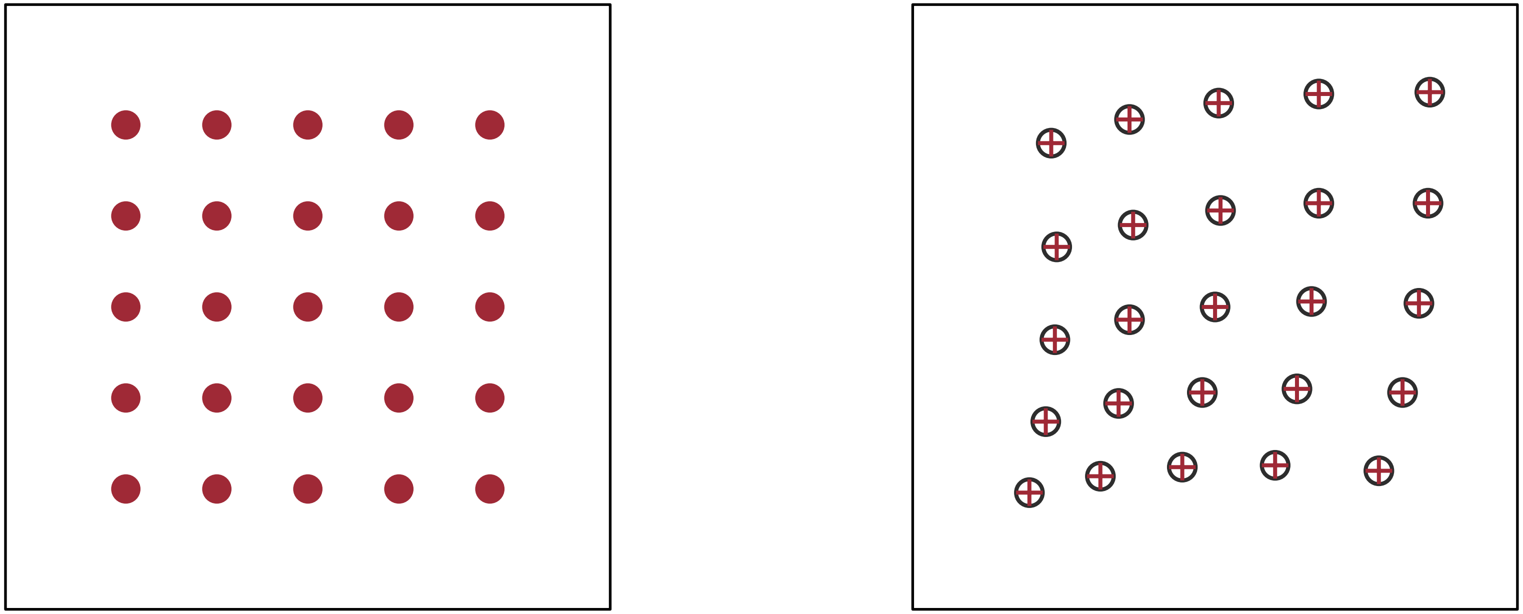


- Fit quadratic transformation.
- Looks correct, small error; but can over fit.

Example

source

target / mapped source



- Fit quadratic transformation.
- Correct transformation, small error.

Image Warping & Morphing

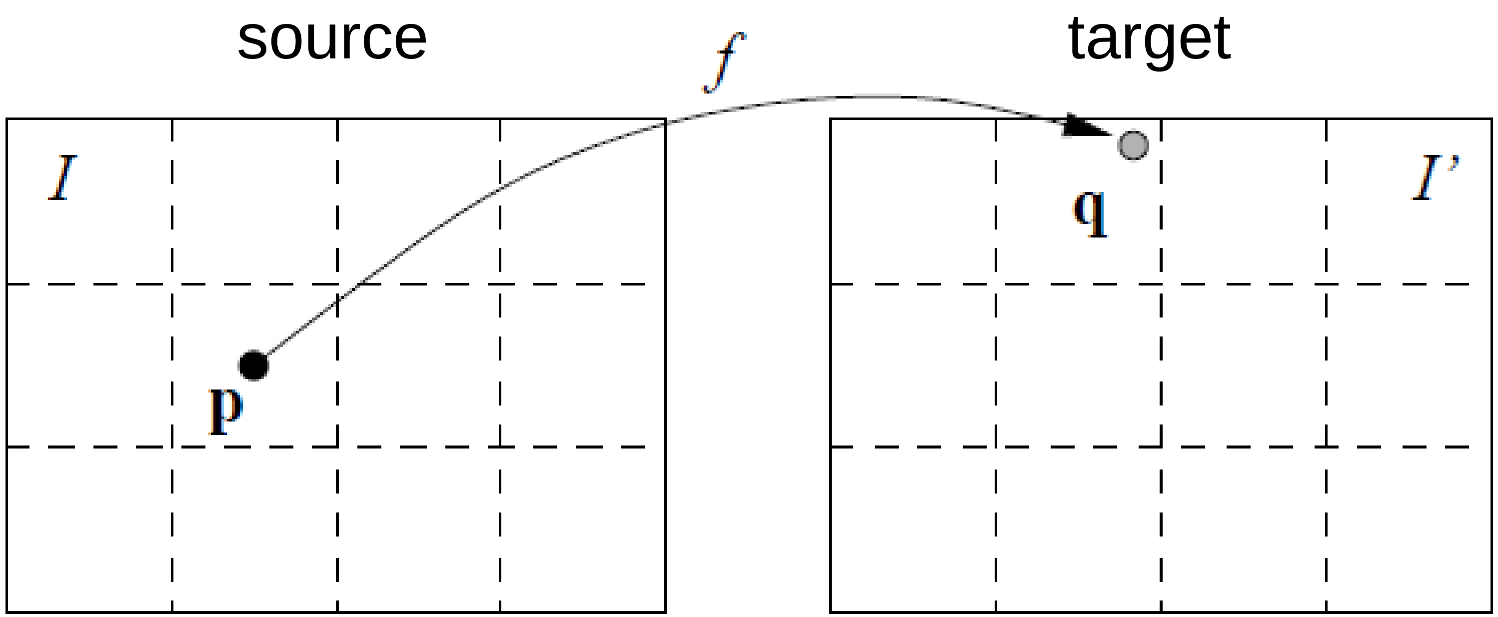
- Involves 3 things
  - Spatial transformation
  - Colour transfer
  - Continuous transition

Colour Transfer

- Determine colour in transformed image.
- Remember to use backward mapping plus bilinear interpolation

Colour Transfer

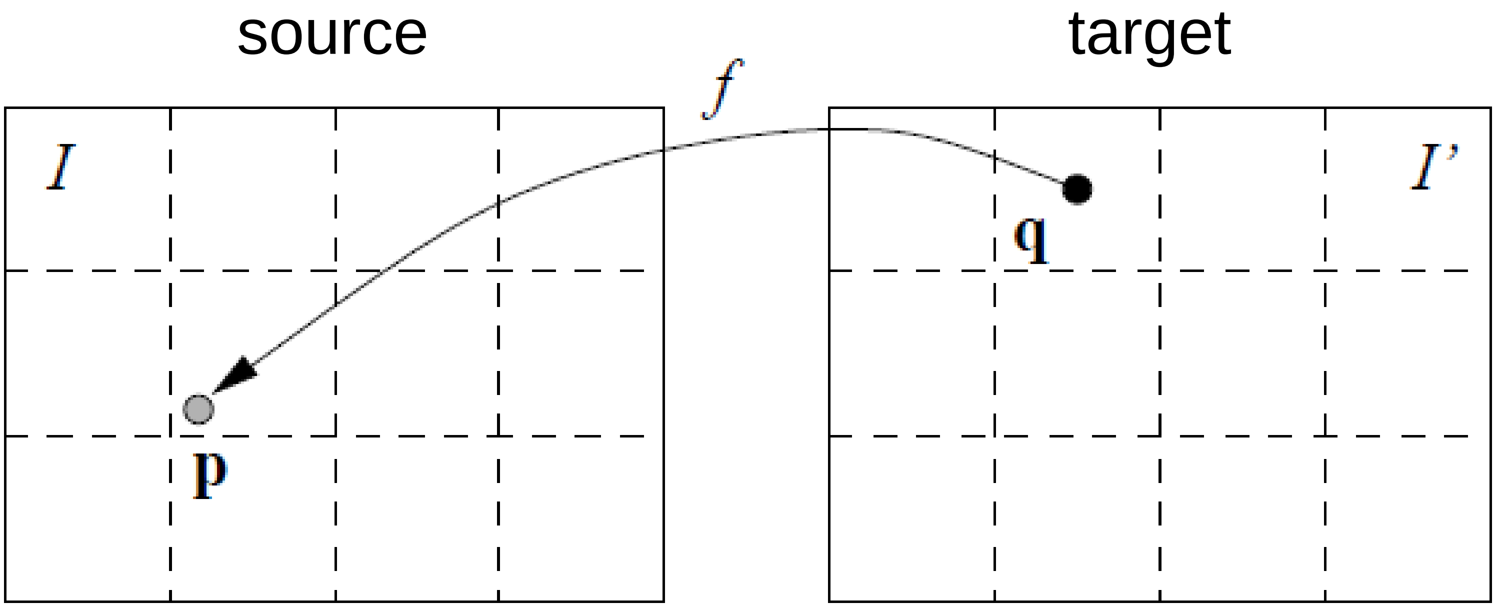
- Forward mapping



  - Copy colour of p in *I* to q in *I*'.
  - But q is at real-valued coordinates; problem.

Colour Transfer

- Backward mapping



  - Map q in *I*' to p in *I*.
  - Use bilinear interpolation to determine colour of p.
  - Copy colour of p to q.

Image Warping Example

- With affine mapping

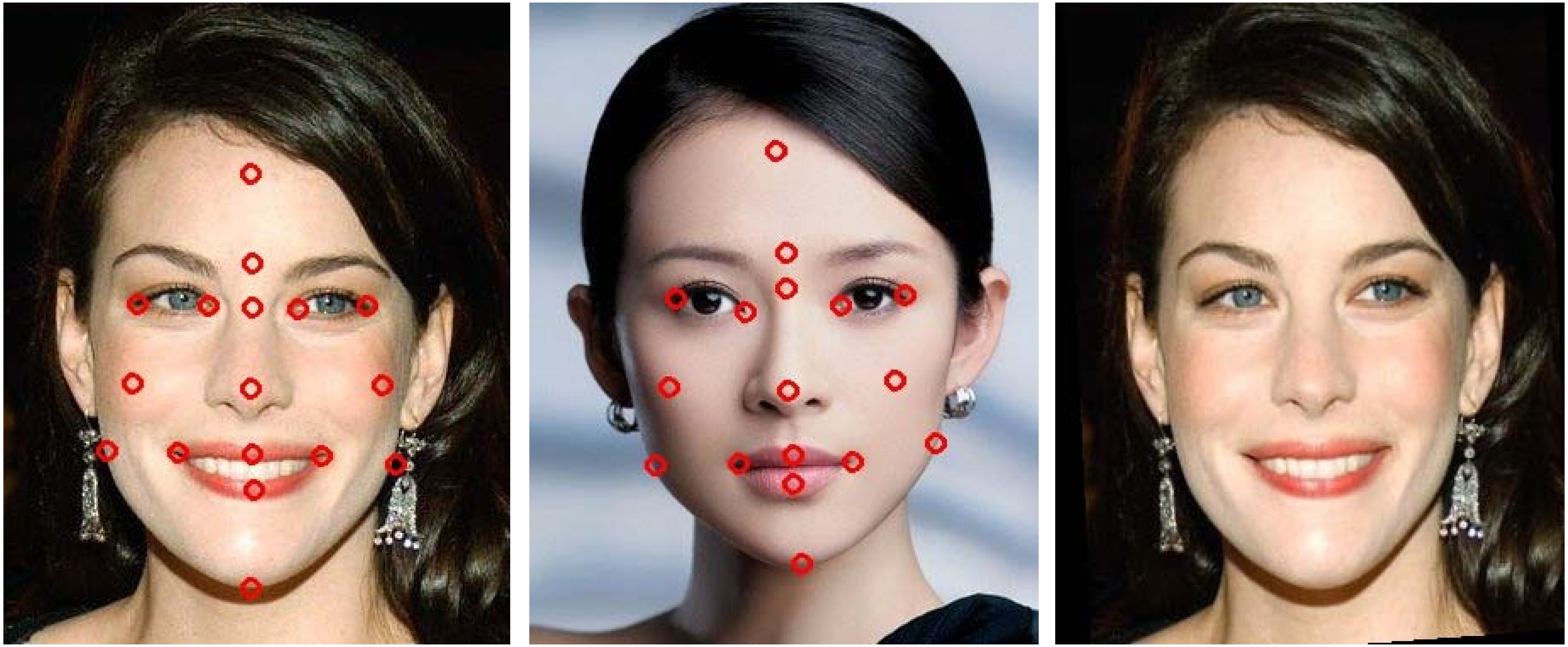


Image Warping Example

- With affine mapping

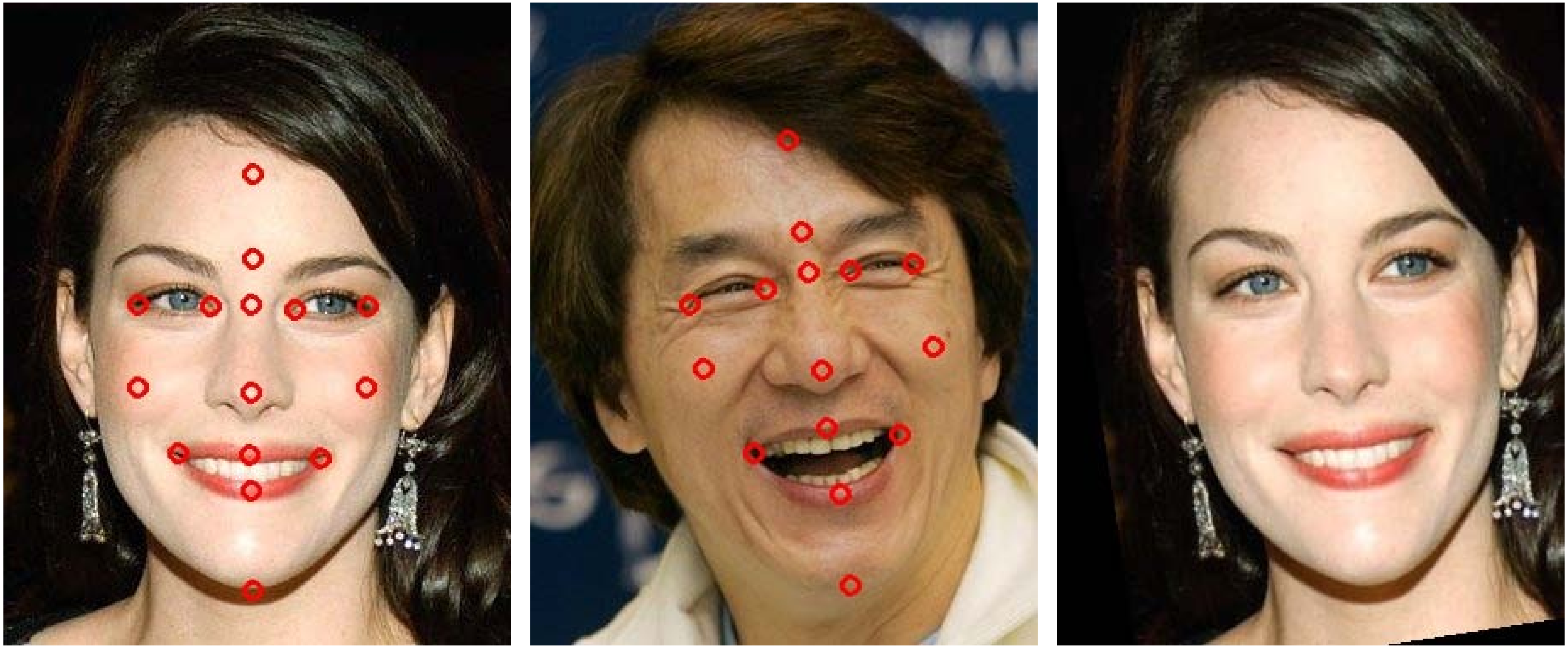


Image Warping Example

- With affine mapping

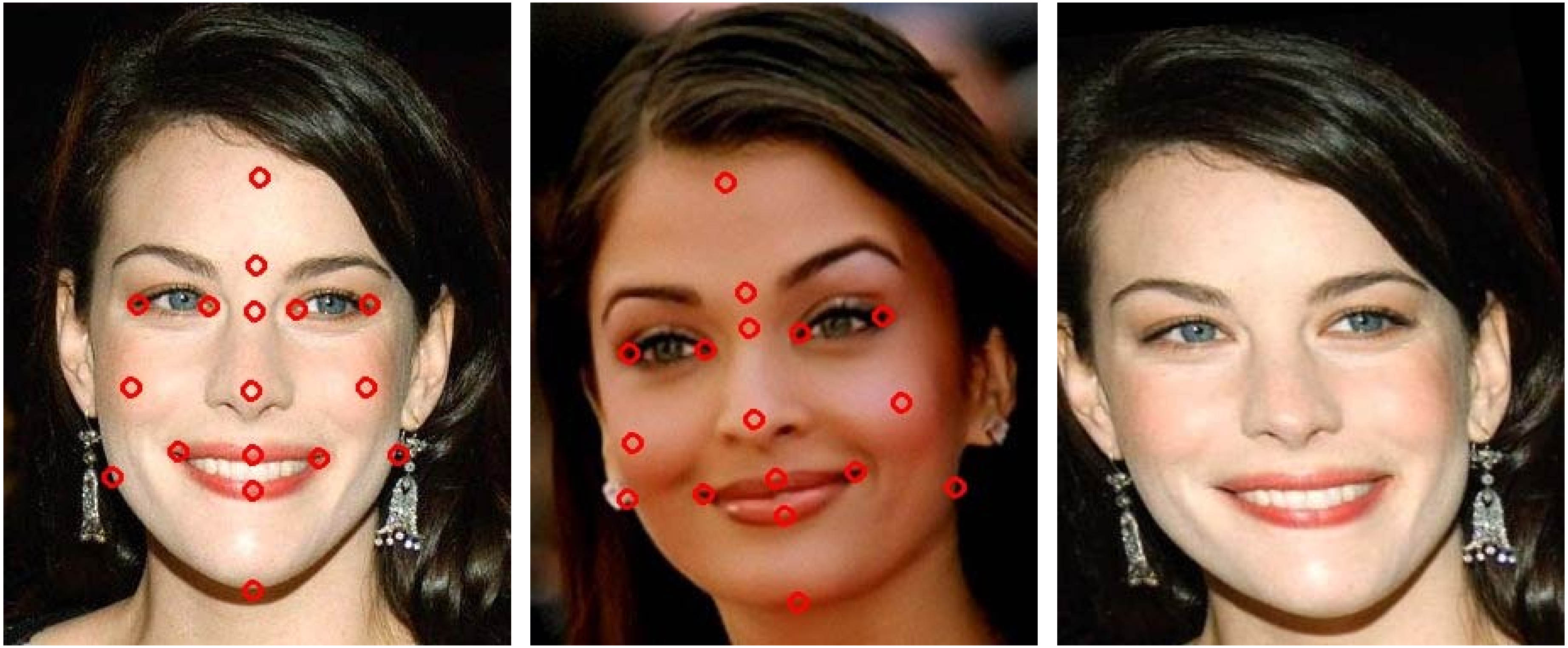


Image Warping Example

- With affine mapping

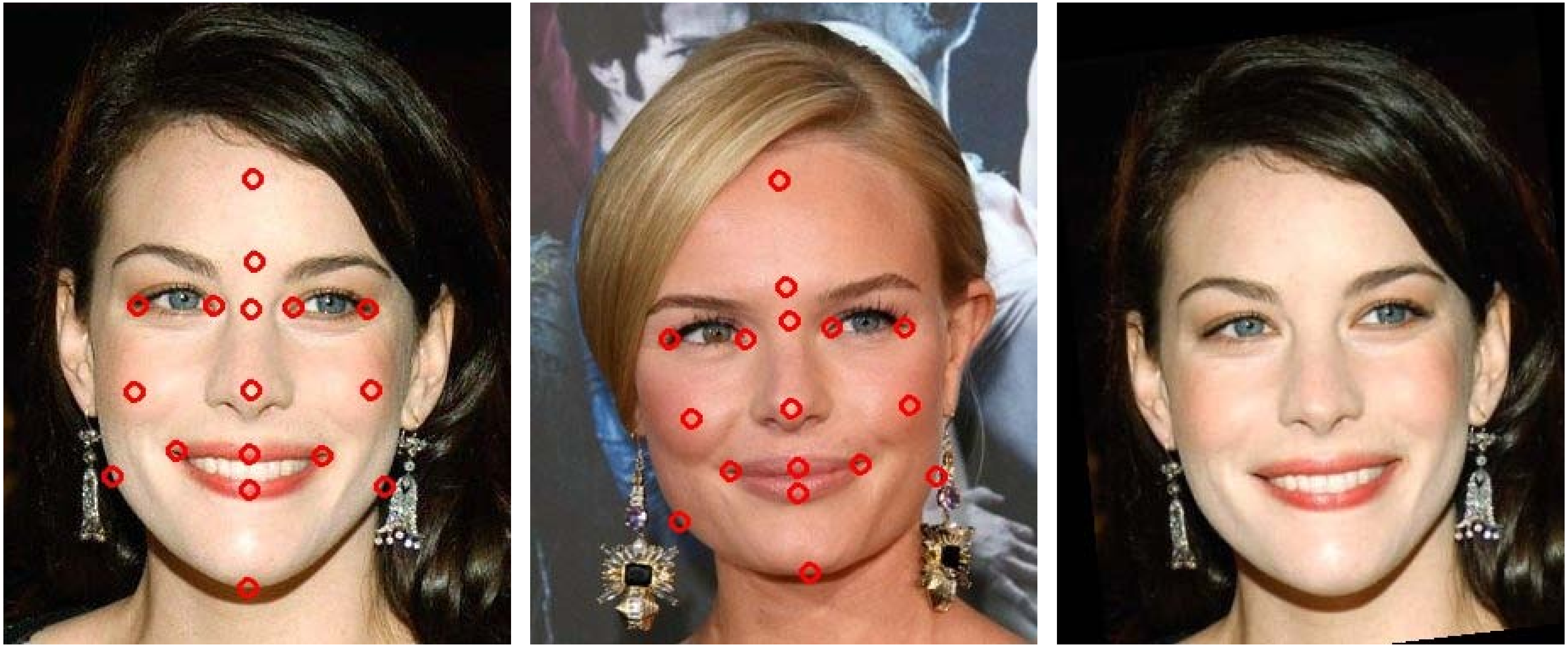
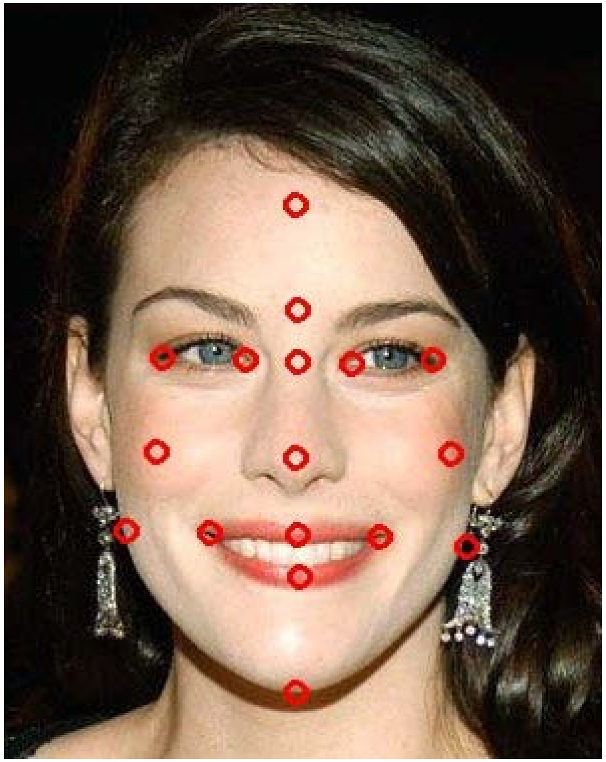
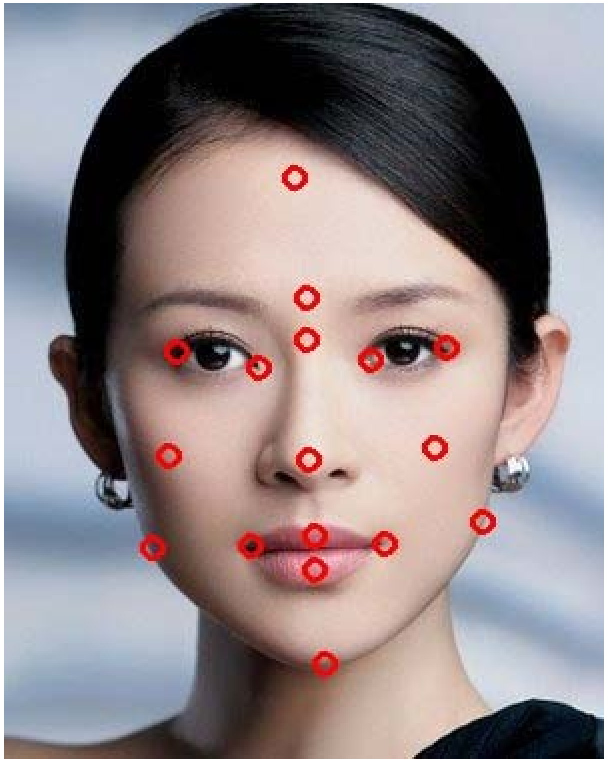


Image Warping Example

- With quadratic mapping

Why so much distortion?





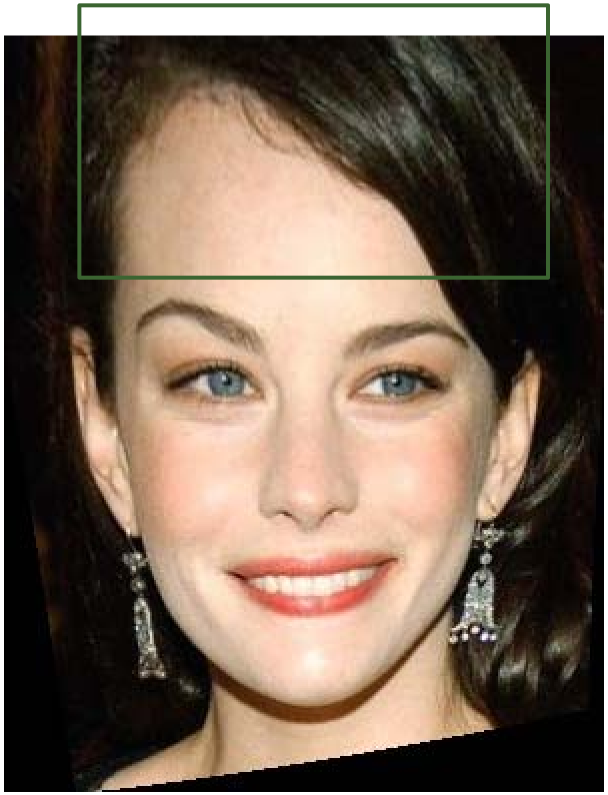


Image Warping Example

- With quadratic mapping

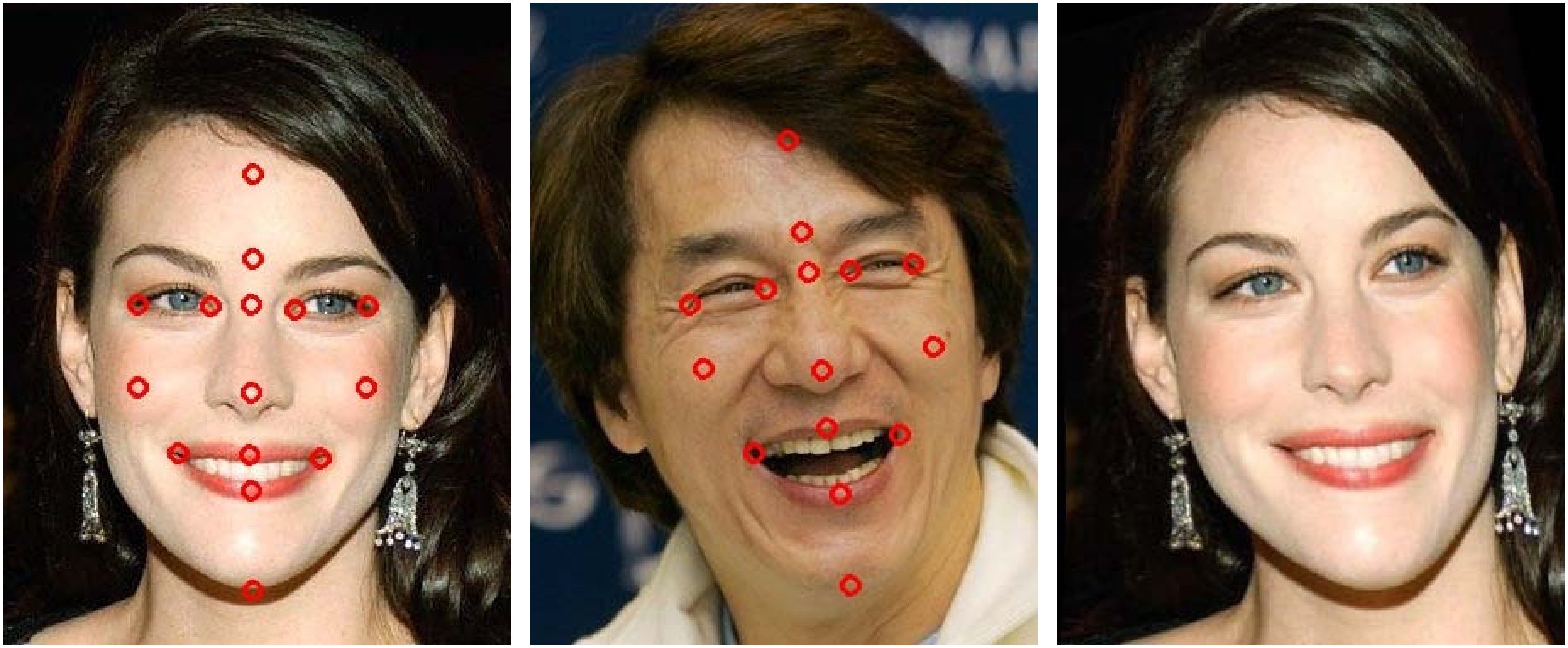


Image Warping Example

- With quadratic mapping

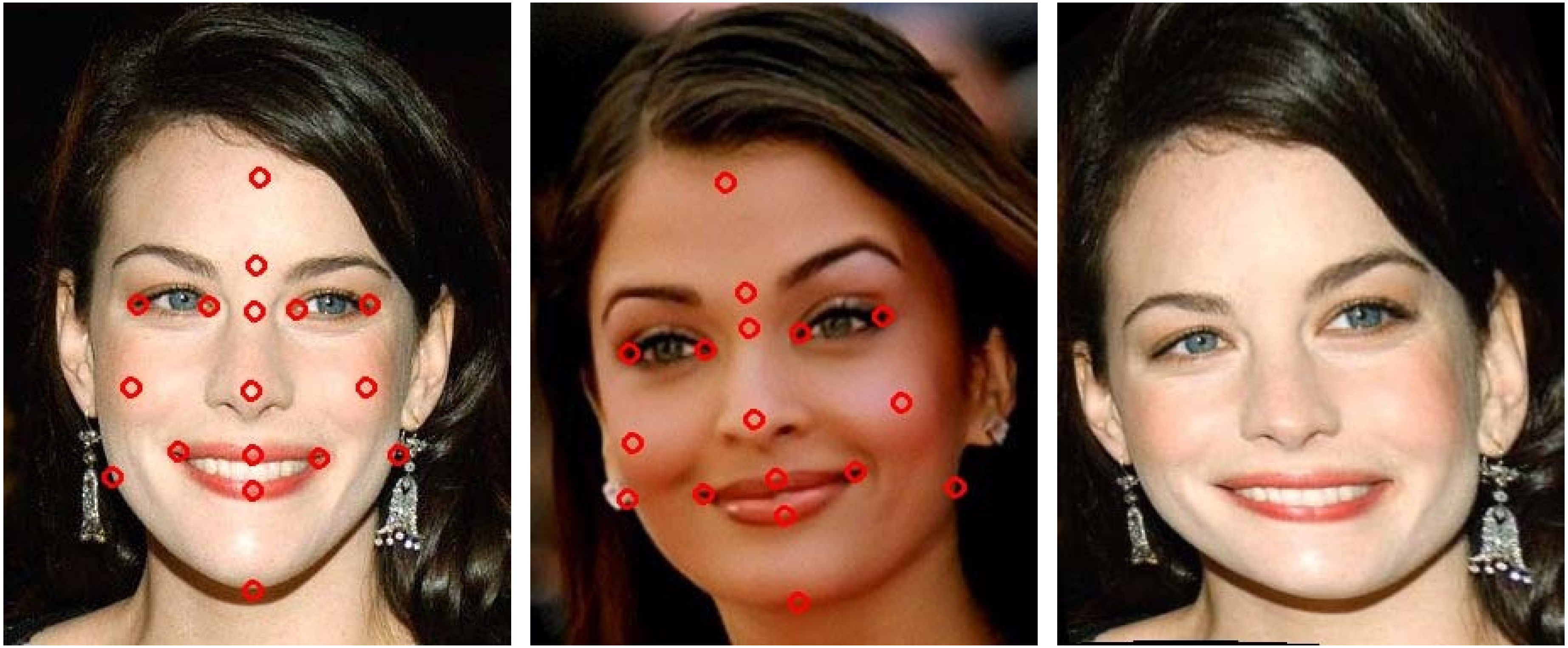
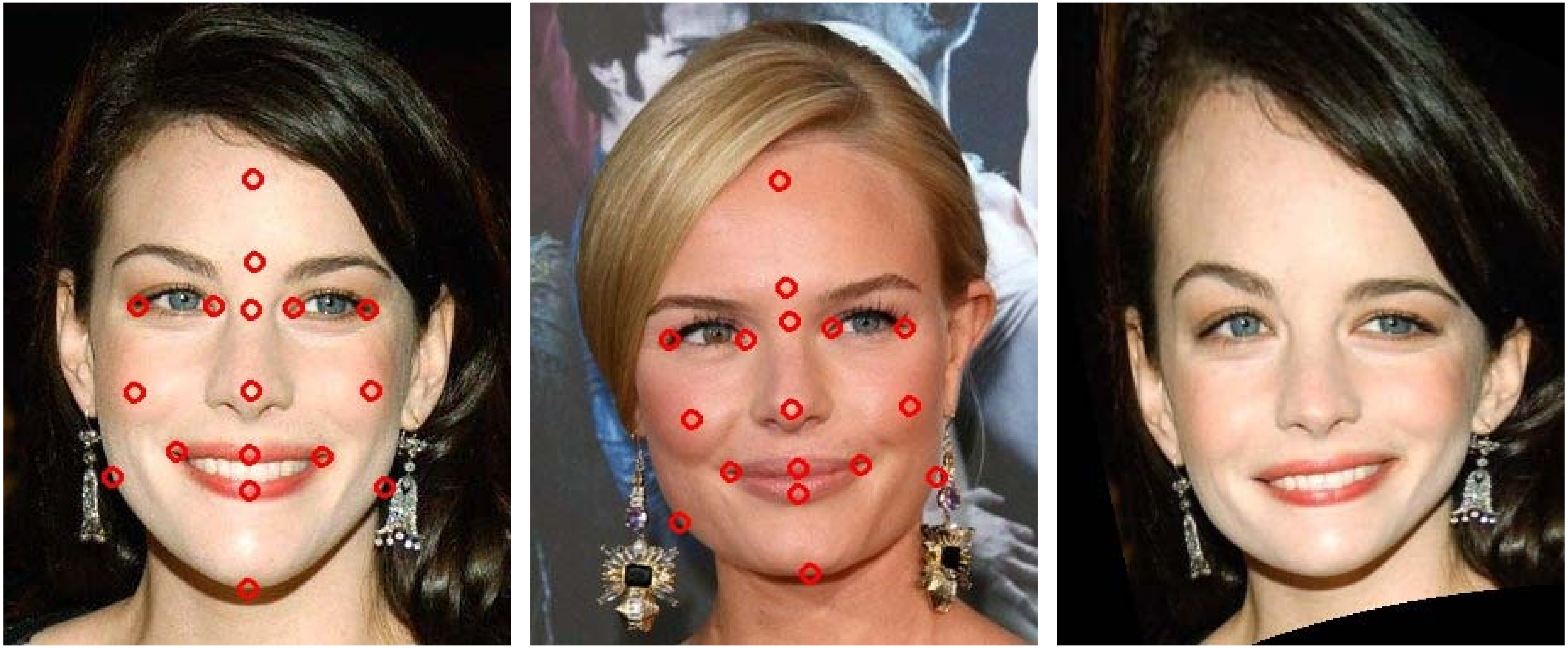


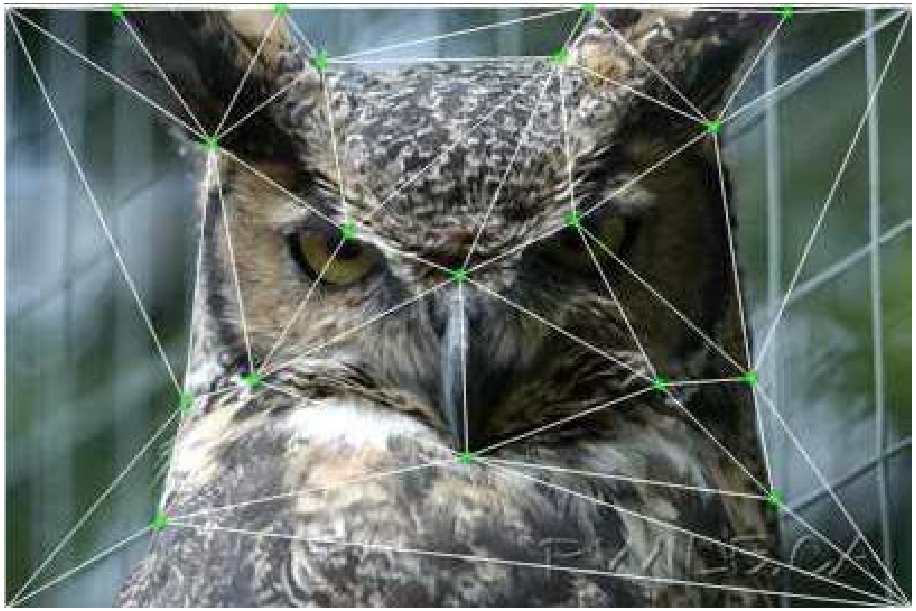
Image Warping Example

- With quadratic mapping



Local Transformation

- Divide image into regions.
- Warp each region by a different transform.
  - Ensure changes over boundaries are smooth.
- Achieve finer control.



Summary

- Need to mark good corresponding points.
- Lower-order function may not warp enough.
- Higher-order function can lead to distortions.
- Local transformations give finer control.

Image Warping & Morphing

- Involves 3 things
  - Spatial transformation
  - Colour transfer
  - Continuous transition

Image Morphing

- Basic ideas:
  - Want positions to change smoothly.
  - Want colours to change smoothly.

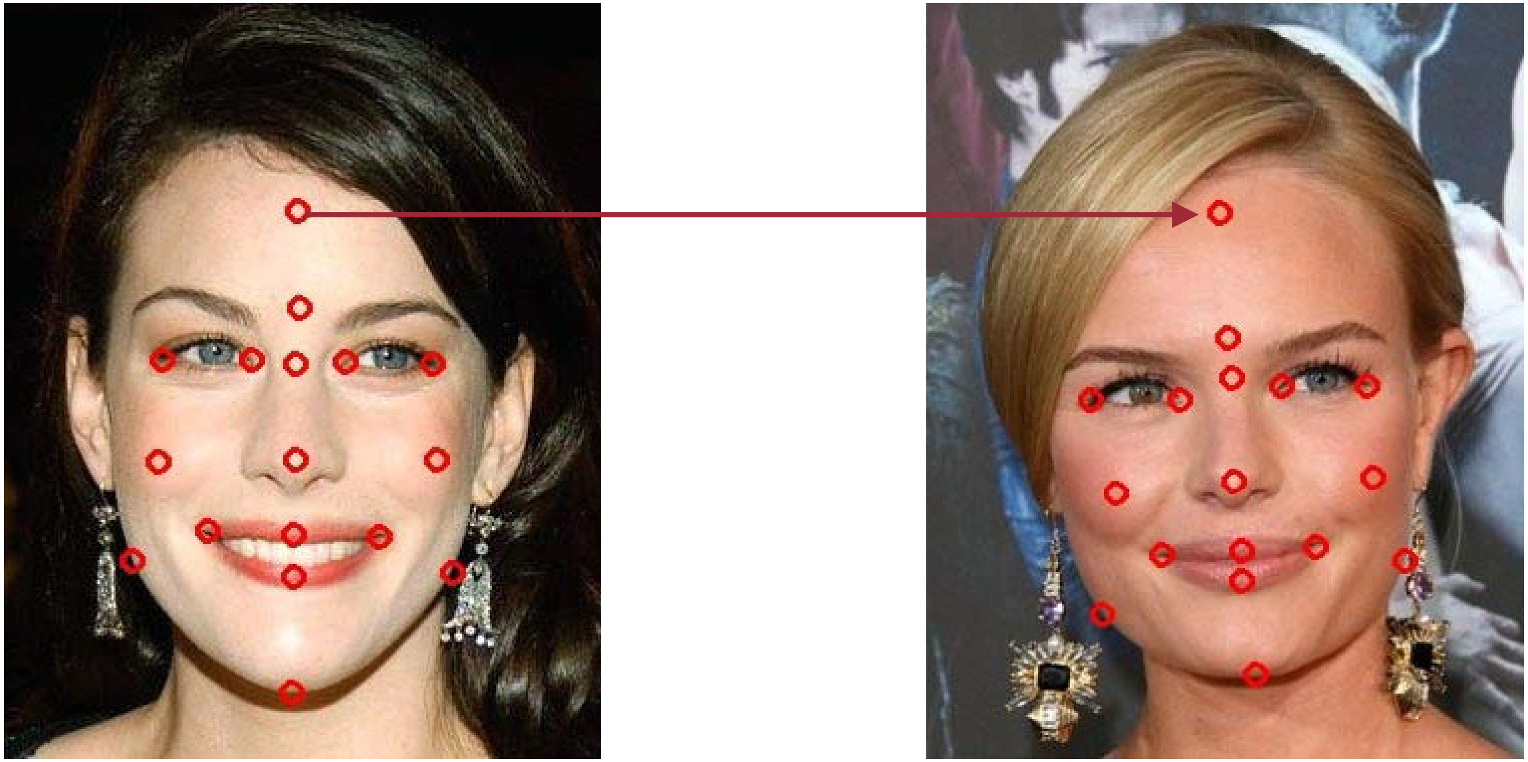
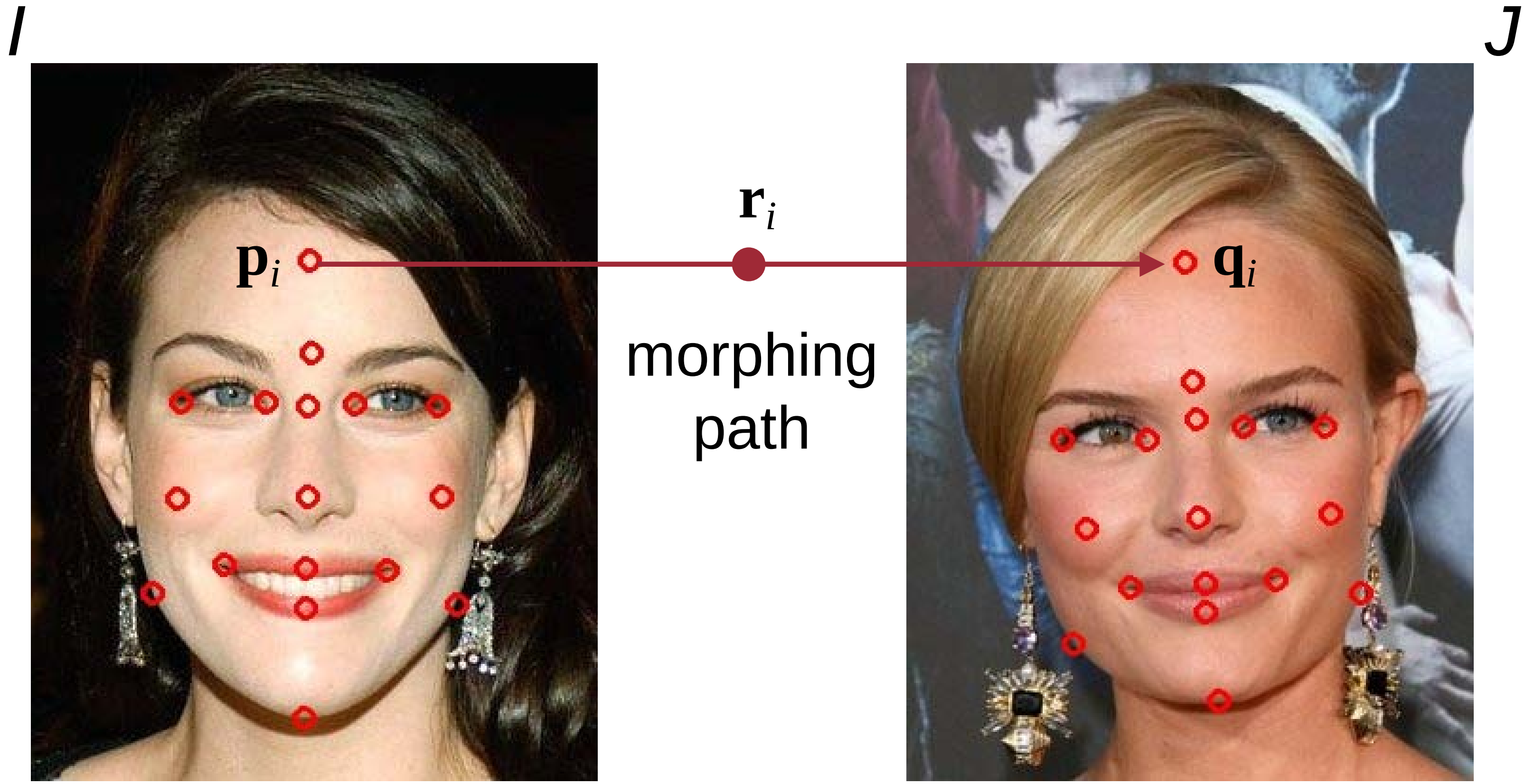


Image Morphing

- Simplest transition: linear

$$\mathbf{r}_i(t) = (1 - t)\,\mathbf{p}_i + t\,\mathbf{q}_i \qquad 0 \le t \le 1$$



- Morphing path is actually like this:

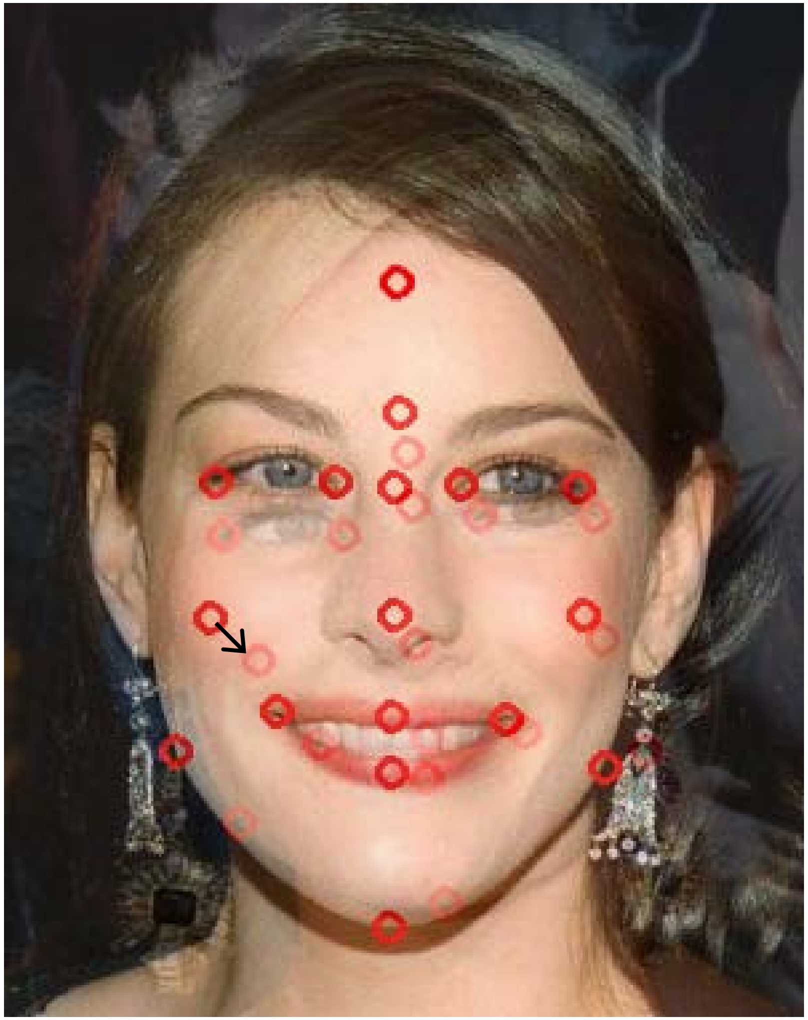
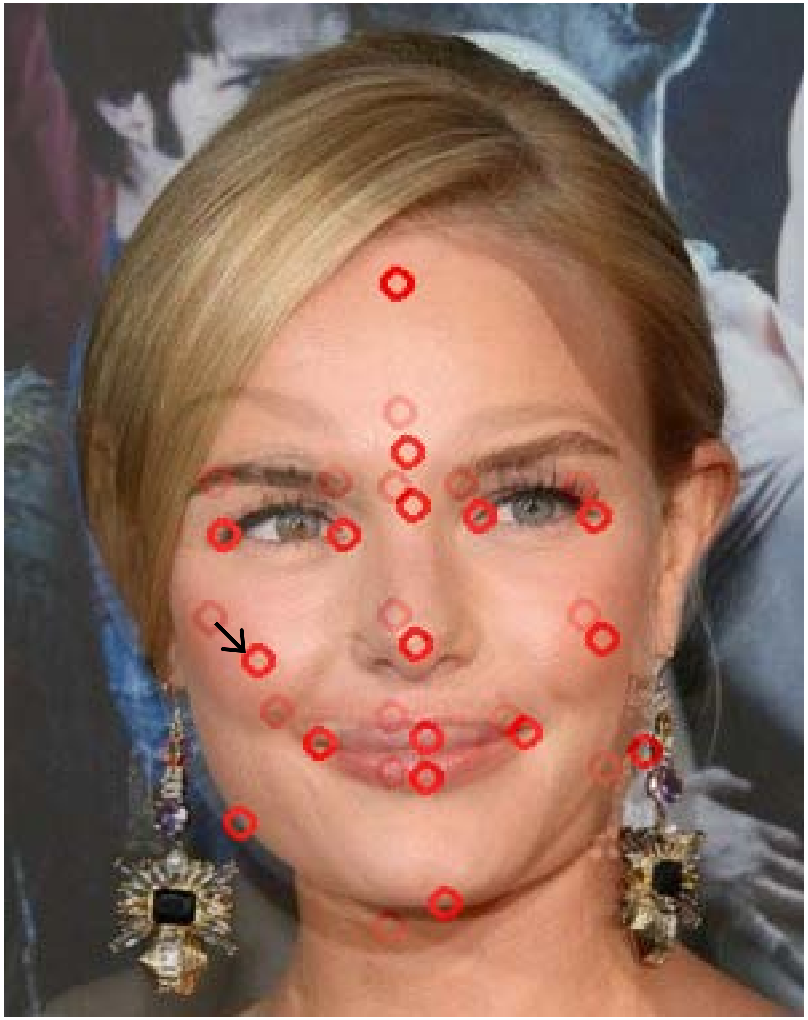
 

Image Morphing

- Set time step $\Delta t = 1$ / number of frames.
- Repeat for $0 \leq t \leq 1$
  - Warp I to $I(t)$ by mapping $\mathbf{p}_i$ to $\mathbf{r}_i$(t).
  - Warp J to $J(t)$ by mapping $\mathbf{q}_i$ to $\mathbf{r}_i$(t).
  - Blend $I(t)$ and $J(t)$ into $M(t)$

$$M(t) = (1 - t)\, I(t) + t\, J(t)$$

  - Save $M(t)$ into a video.

Example

Anything peculiar?

Sample faces

Summary

- For seamless morphing
  - Match all corresponding features.
  - Need accurate transformation.
- For smooth morphing
  - Use smaller time step.

Further Reading

- More sophisticated image warping: [Wolberg90].
- More sophisticated image morphing: [Lee96]

References

- S.-Y. Lee and S. Y. Shin. Warp generation and transition control in image morphing. In *Interactive Computer Animation*. Prentice Hall, 1996.
- G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1990.