CS 4300
Computer Graphics

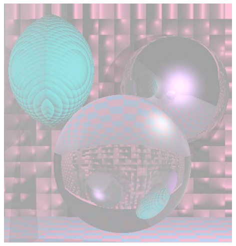
Prof. Harriet Fell

Fall 2012

Lecture 32 – November 19, 2012

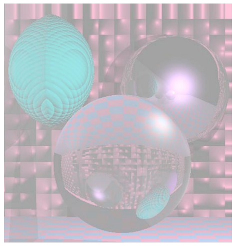
Today's Topics

- Morphing



Morphing History

- *Morphing* is turning one image into another through a seamless transition.
- Early films used cross-fading picture of one actor or object to another.
- In 1985, "Cry" by Godley and Crème, parts of an image fade gradually to make a smother transition.
- Early-1990s computer techniques distorted one image as it faded into another.
  - Mark corresponding points and vectors on the "before" and "after" images used in the morph.
  - E.g. key points on the faces, such as the contour of the nose or location of an eye
  - Michael Jackson's "Black or White" (1991)
    - » http://en.wikipedia.org/wiki/Morphing



Morphing History

- 1992 Gryphon Software's “Morph” became available for Apple Macintosh.
- For high-end use, “Elastic Reality” (based on Morph Plus) became the de facto system of choice for films and earned two Academy Awards in 1996 for Scientific and Technical Achievement.
- Today many programs can automatically morph images that correspond closely enough with relatively little instruction from the user.
- Now morphing is used to do cross-fading.

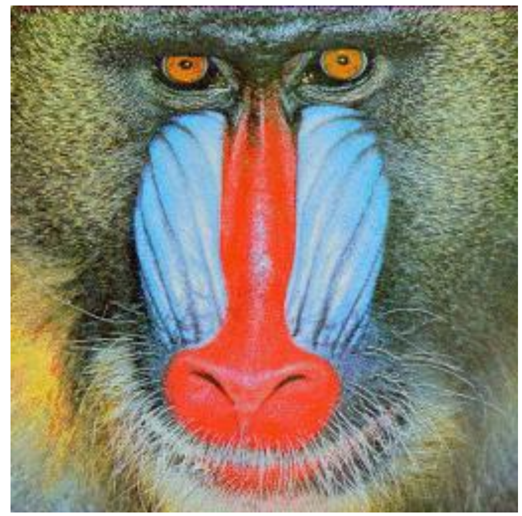
Harriet George Harriet…

Feature Based Image Metamorphosis

Thaddeus Beier and Shawn Neely 1992

- The morph process consists
  - warping two images so that they have the same "shape"
  - cross dissolving the resulting images
- cross-dissolving is simple
- warping an image is hard

Harriet & Mandrill



Harriet 276x293

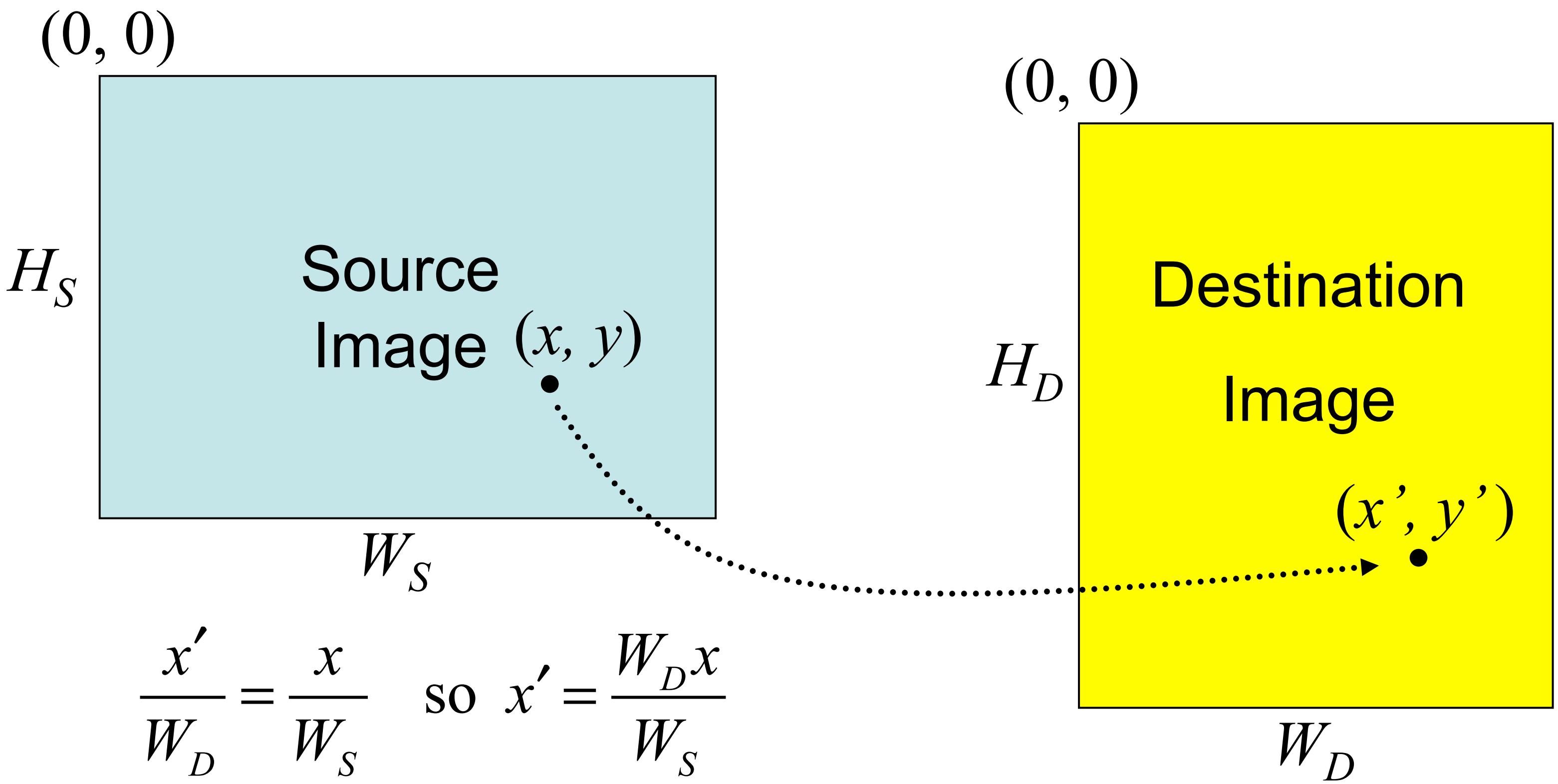
Mandrill 256x256

Warping an Image

There are two ways to warp an image:

- *forward mapping* - scan through source image pixel by pixel, and copy them to the appropriate place in the destination image.
  - some pixels in the destination might not get painted, and would have to be interpolated.
- *reverse mapping -* go through the destination image pixel by pixel, and sample the correct pixel(s) from the source image.
  - every pixel in the destination image gets set to something appropriate.

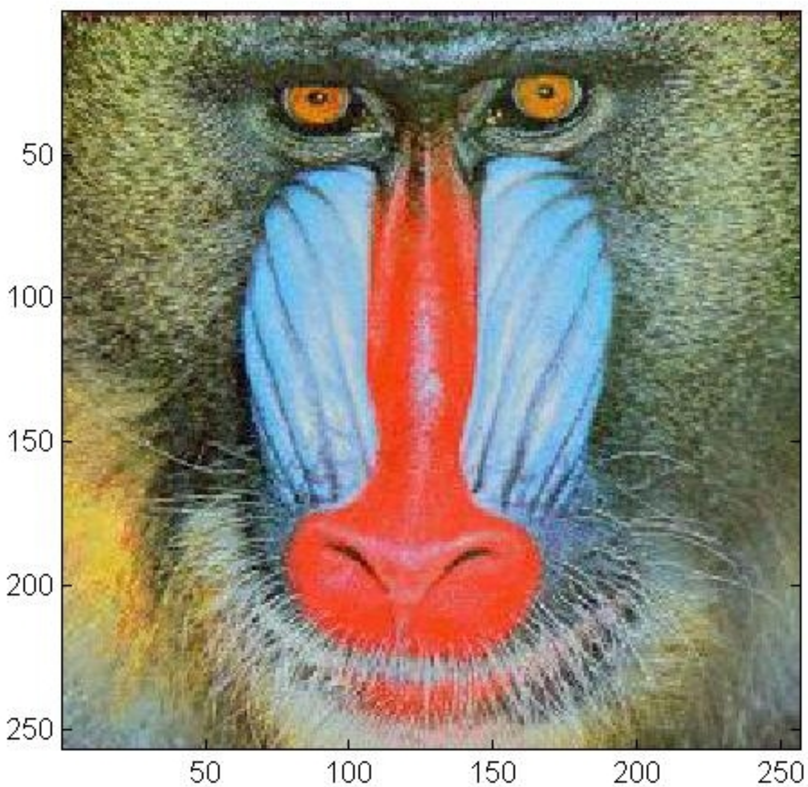
Forward Mapping

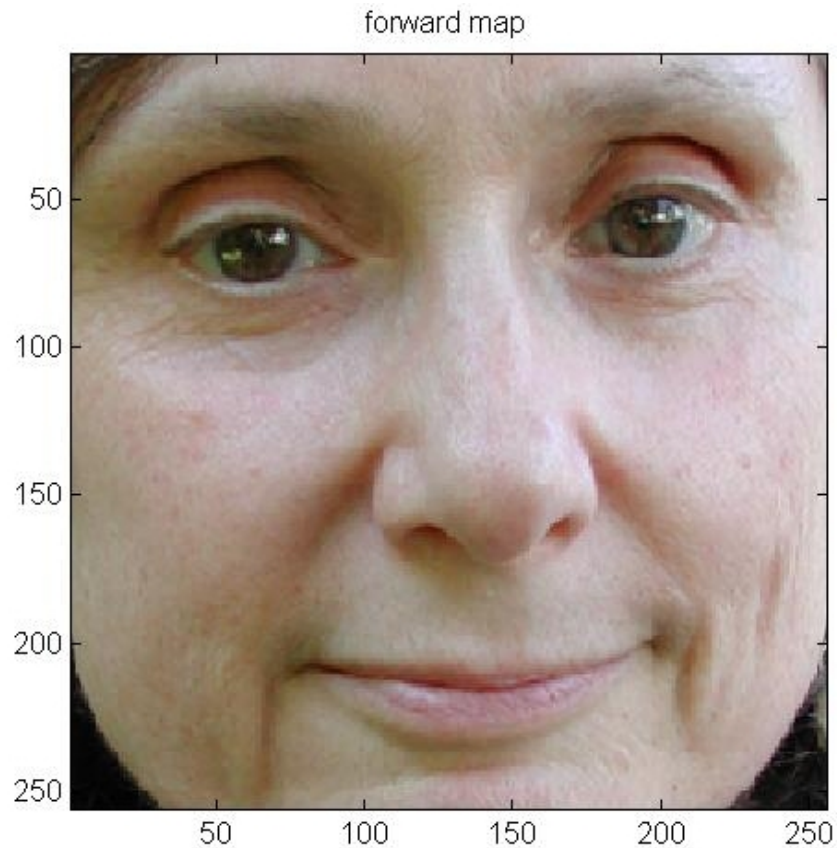


$$\frac{x'}{W_D} = \frac{x}{W_S} \quad \text{so} \quad x' = \frac{W_D x}{W_S}$$

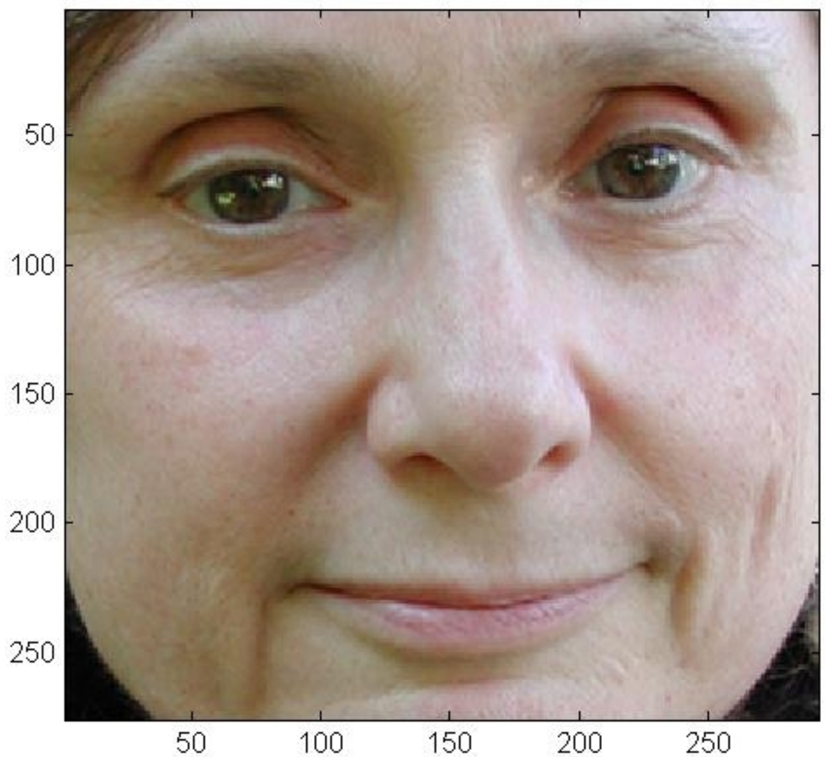
$$\frac{y'}{H_D} = \frac{y}{H_S} \quad \text{so} \quad y' = \frac{H_D y}{H_S}$$

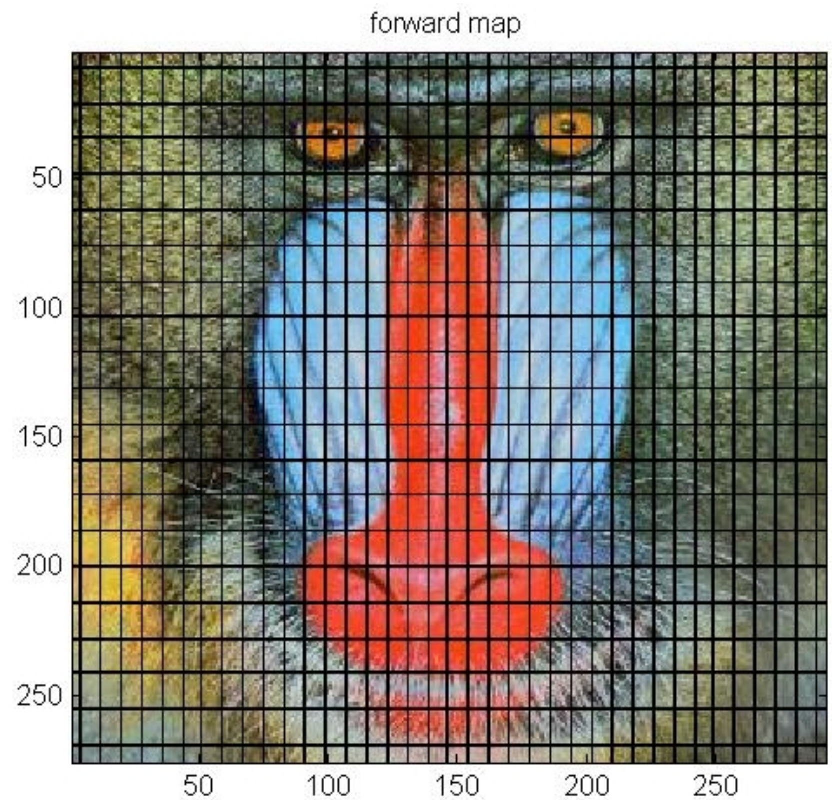
Forward Mapping
Harriet ➔ Mandrill



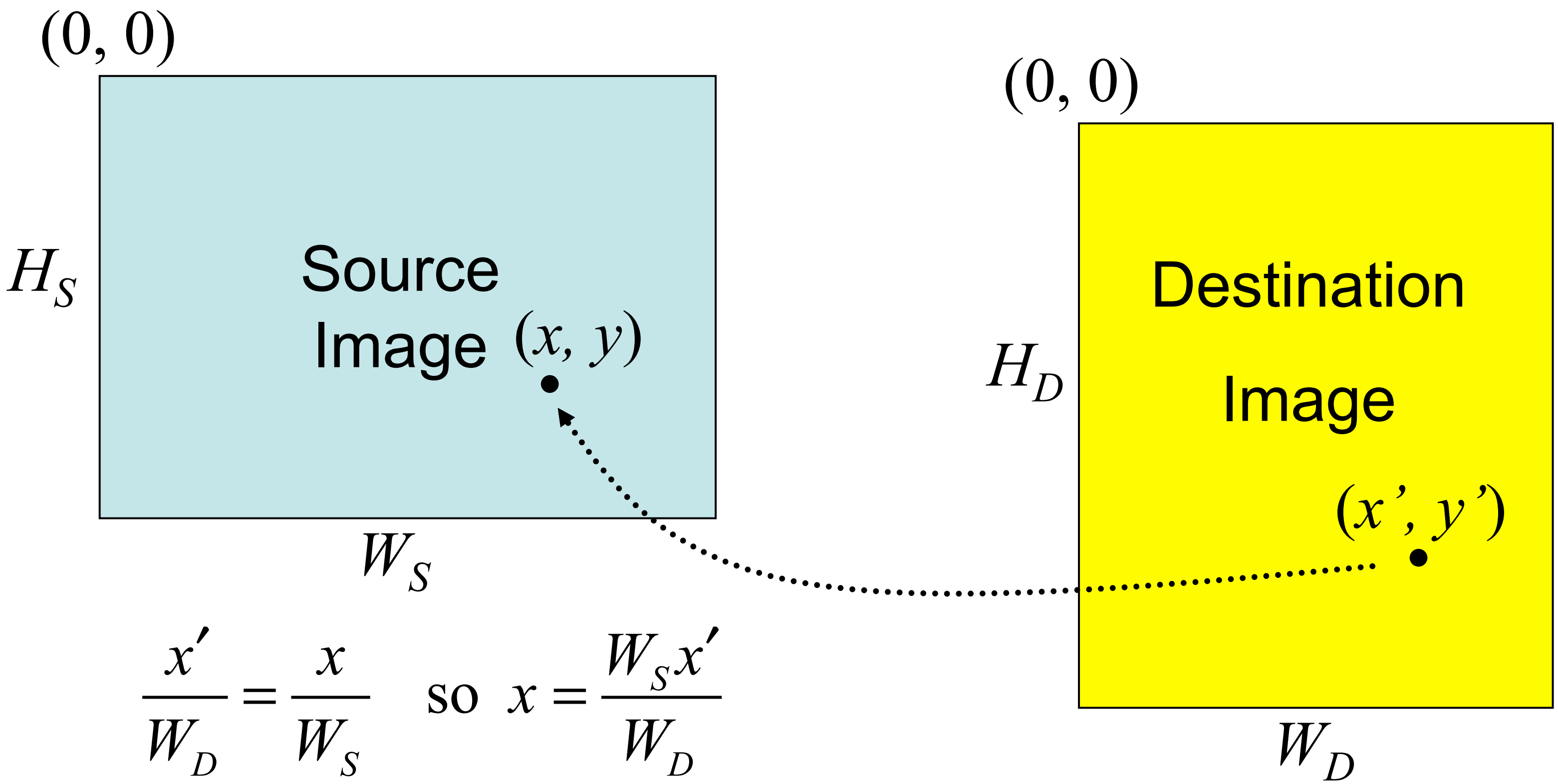


Forward Mapping
Mandrill → Harriet





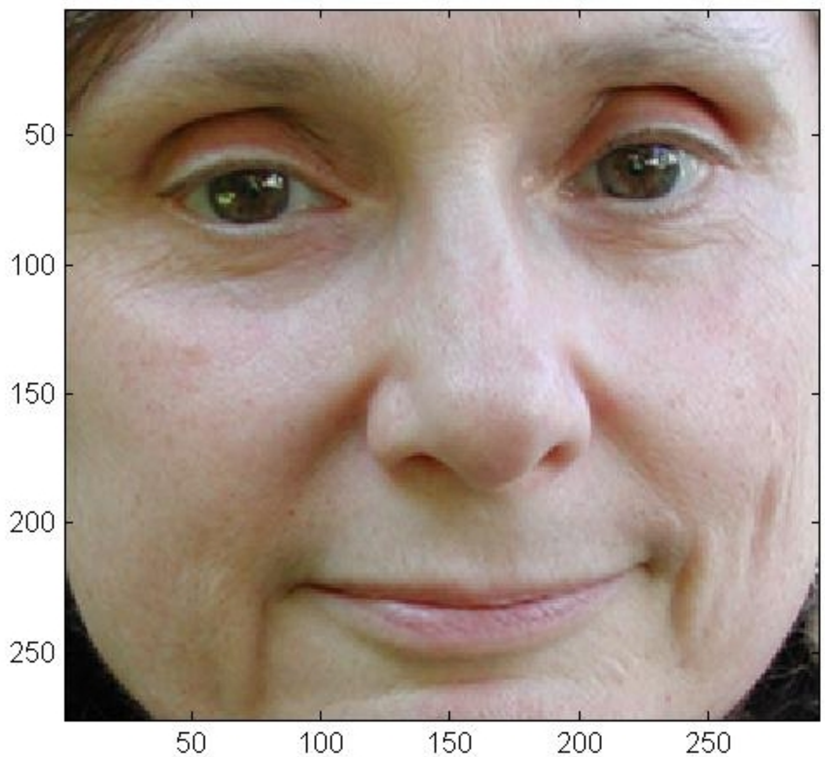
Inverse Mapping

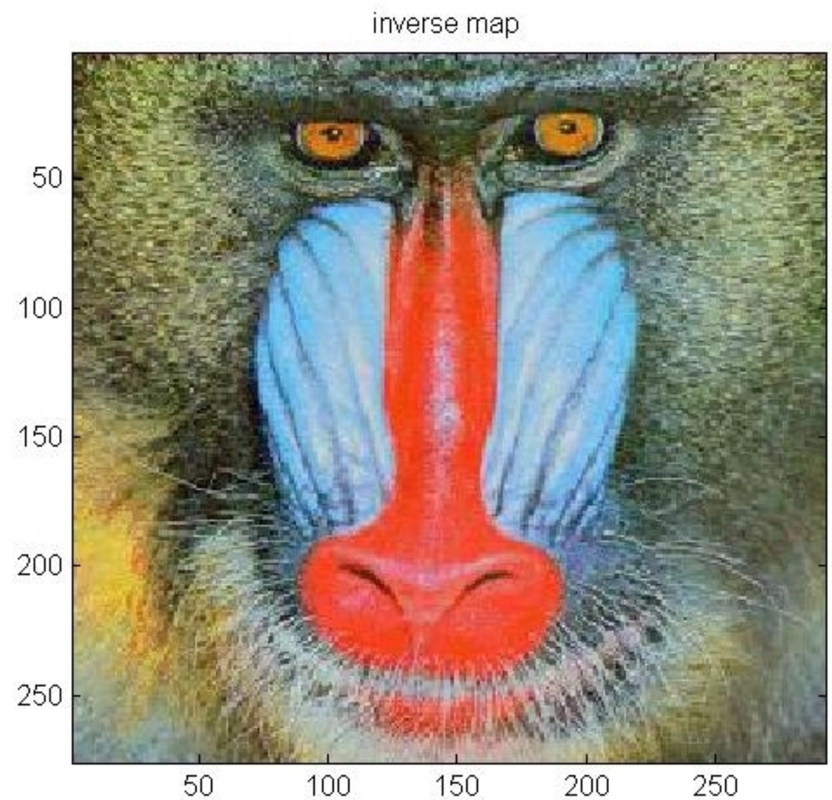


$$\frac{x'}{W_D} = \frac{x}{W_S} \quad \text{so} \quad x = \frac{W_S x'}{W_D}$$

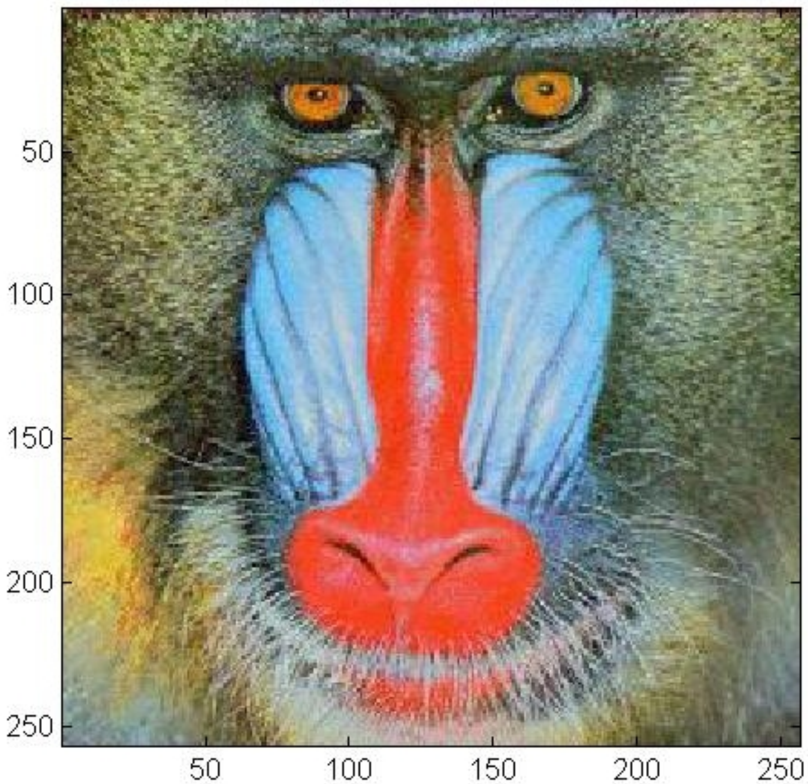
$$\frac{y'}{H_D} = \frac{y}{H_S} \quad \text{so} \quad y = \frac{H_S y'}{H_D}$$

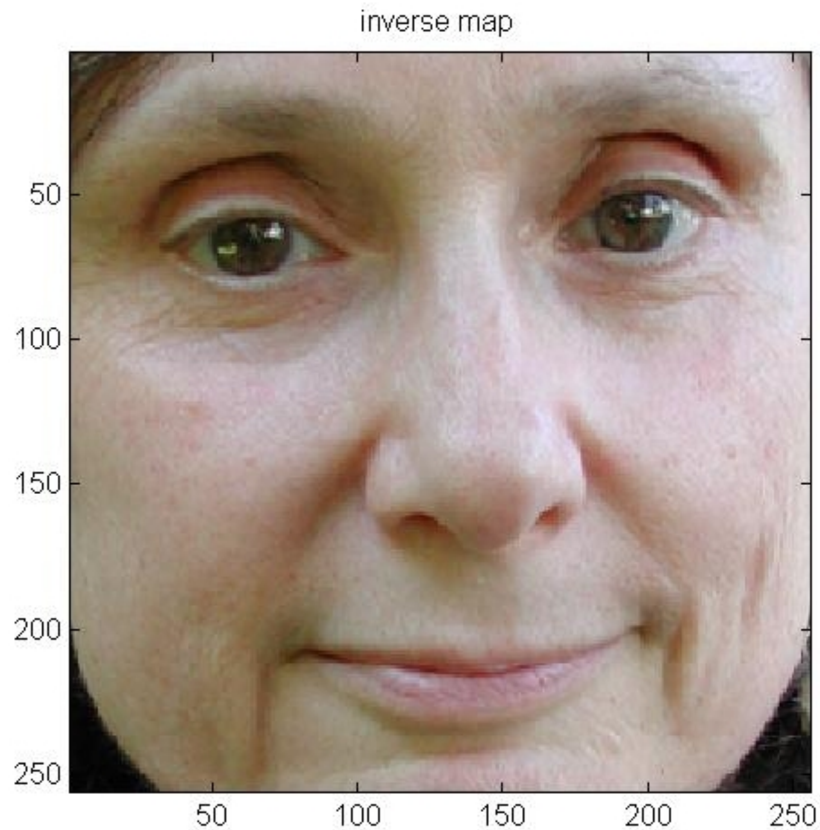
Inverse Mapping
Mandrill → Harriet



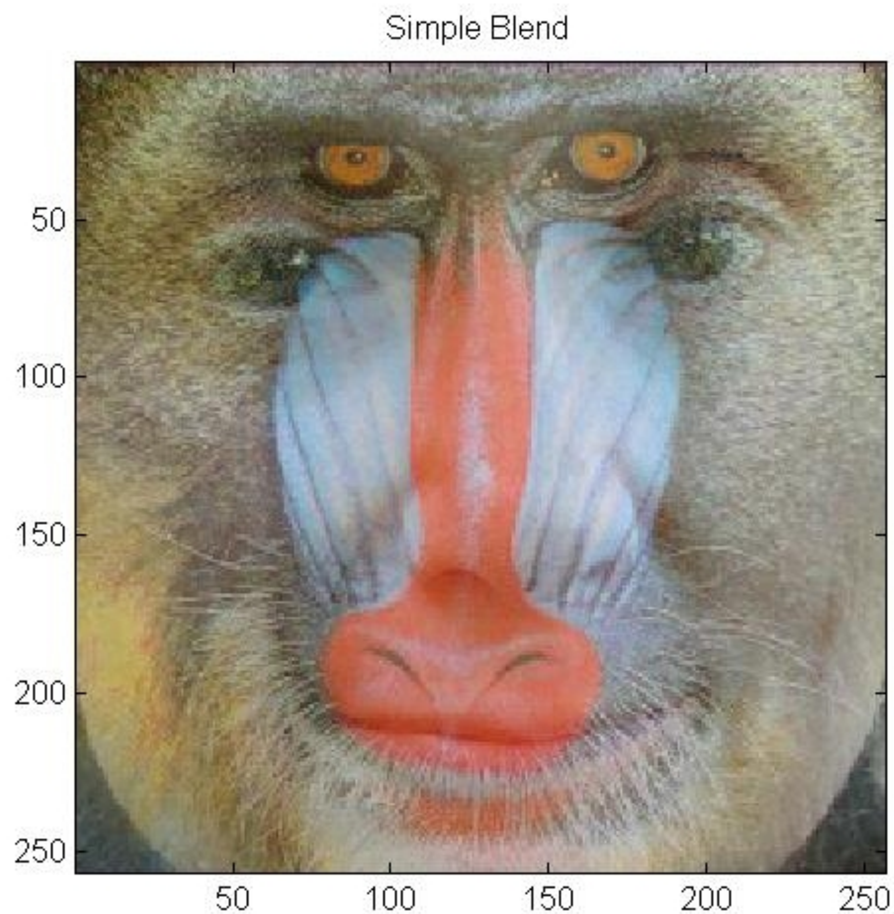


Inverse Mapping
Harriet → Mandrill

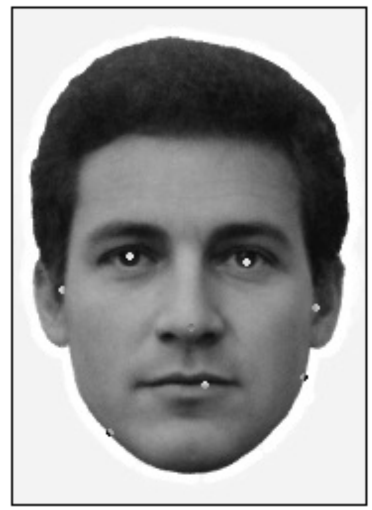




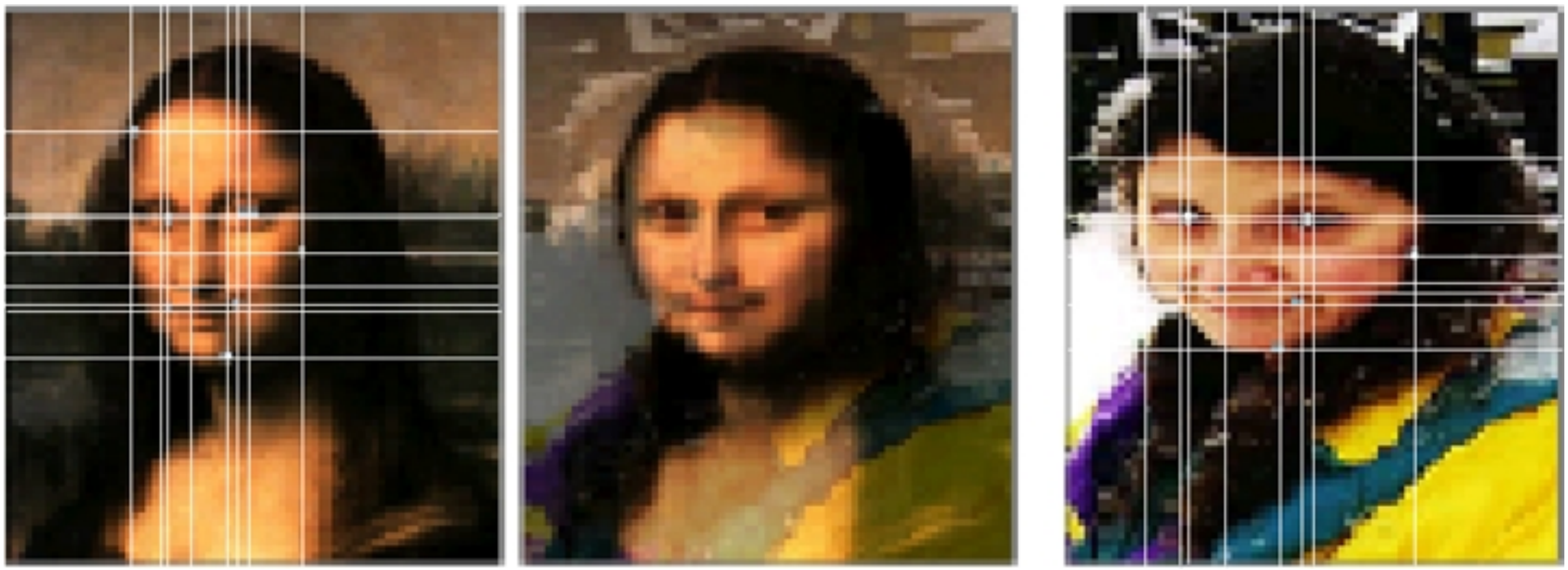
(harrietINV + mandrill)/2



Matching Points

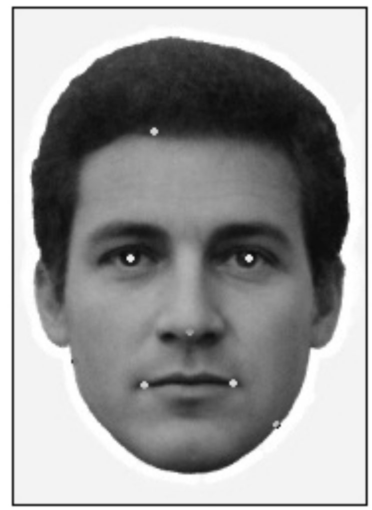
Matching Ponts
Rectangular Transforms



Halfway Blend

Image1

Image2



(1-t)Image1 + (t)Image2

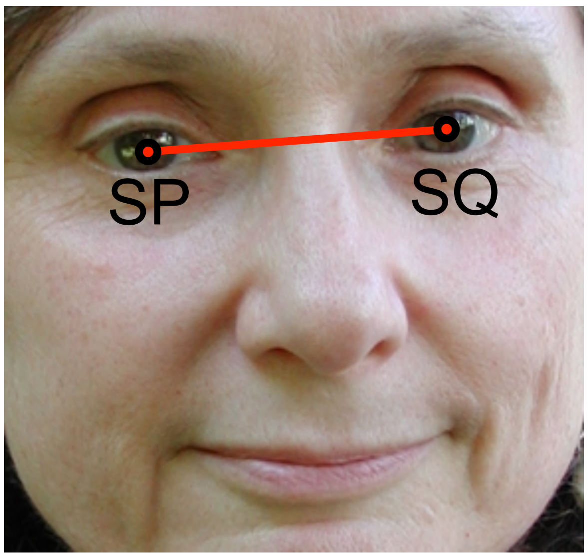
T = .5

Caricatures
Extreme Blends

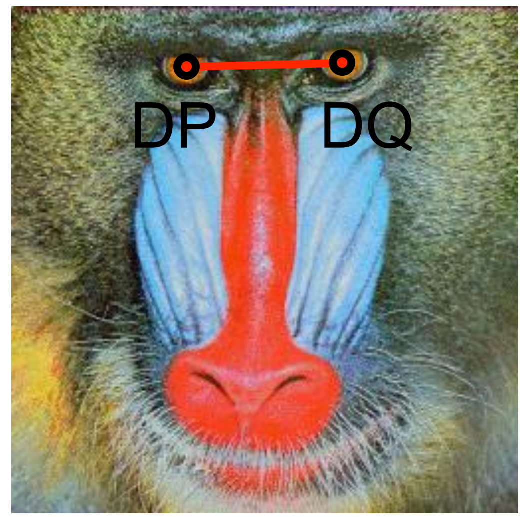
t = 1.5

Harriet & Mandrill Matching Eyes

Match the endpoints of a line in the source with the endpoints of a line in the destination.



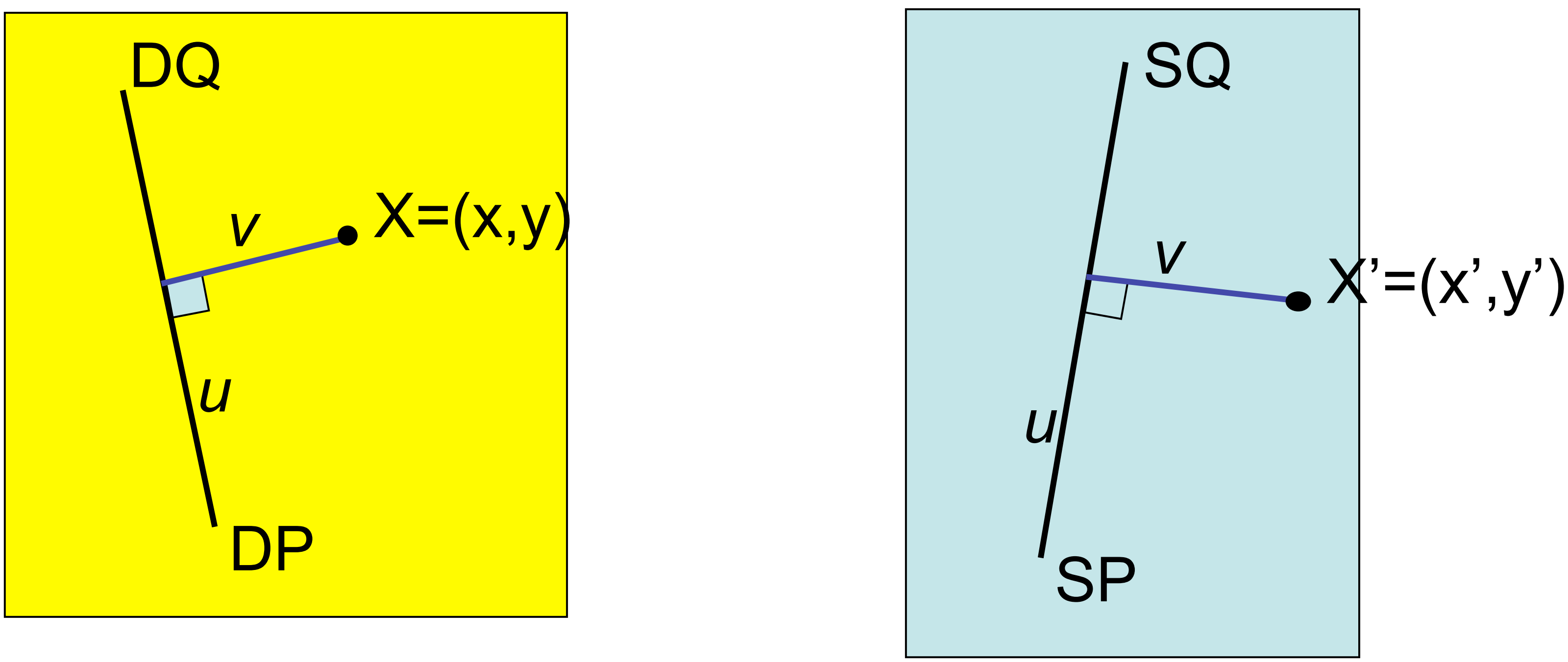
Harriet 276x293



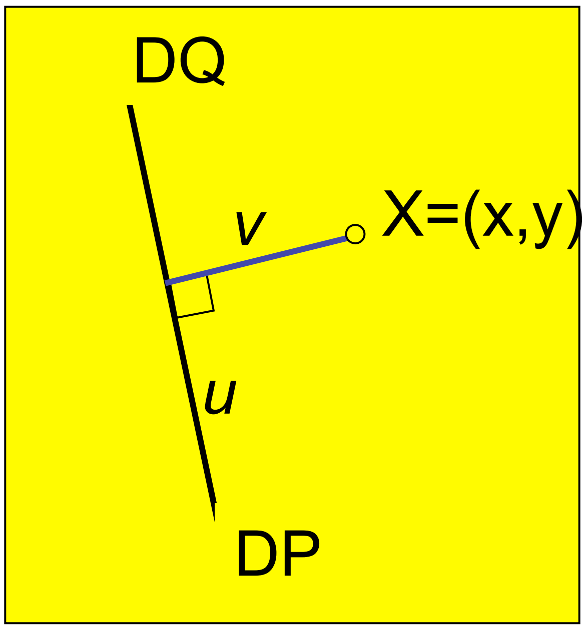
Mandrill 256x256

Line Pair Map

The *line pair map* takes the source image to an image the same size as the destinations and take the line segment in the source to the line segment in the destination.



Finding *u* and *v*



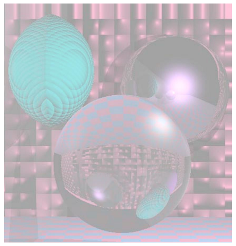
$$u = \frac{(X - DP) \cdot (DQ - DP)}{\|DQ - DP\|^2}$$

$$v = \frac{(X - DP) \cdot perp(DQ - DP)}{\|DQ - DP\|}$$

$$X' = SP + u \times (SQ - SP) + \frac{v \times perp(SQ - SP)}{\|SQ - SP\|}$$

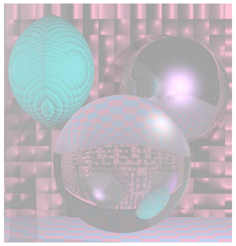
u is the proportion of the distance from DP to DQ.

v is the distance to travel in the perpendicular direction.



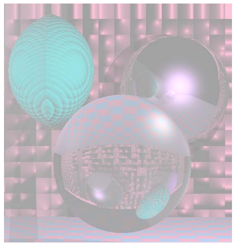
linePairMap.m header

```
% linePairMap.m
% Scale image Source to one size DW, DH with line pair
mapping
function Dest = forwardMap(Source, DW, DH, SP, SQ, DP, DQ);
% Source is the source image
% DW is the destination width
% DH is the destination height
% SP, SQ are endpoints of a line segment in the Source [y, x]
% DP, DQ are endpoints of a line segment in the Dest [y, x]
```



linePairMap.m body

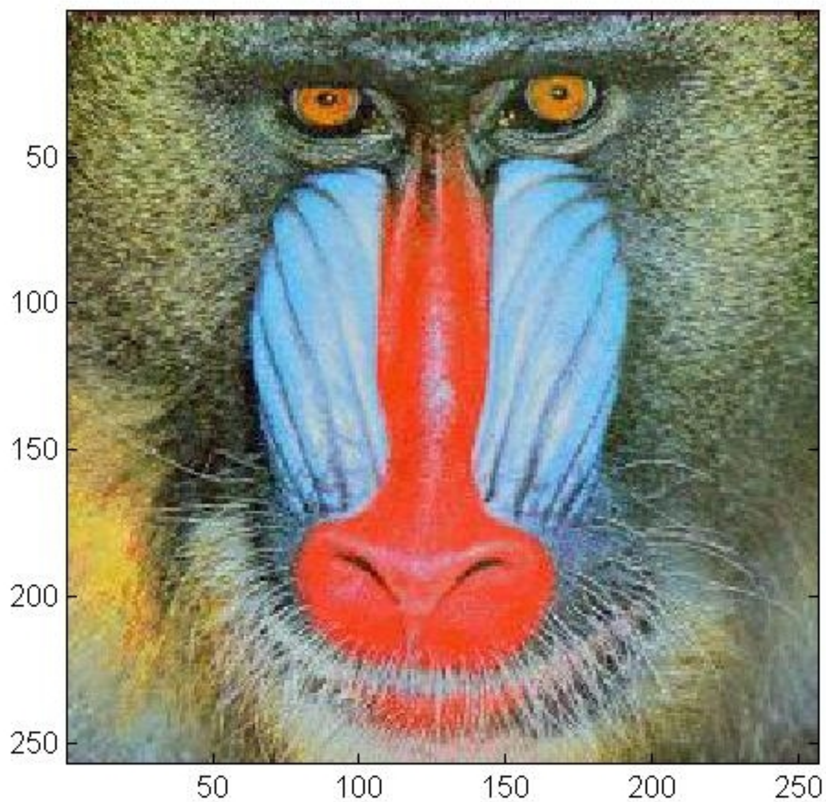
```
Dest = zeros(DH, DW,3);  % rows x columns x RGB
SW = length(Source(1,:,1));  % source width
SH = length(Source(:,1,1));   % source height
for y= 1:DH
   for x = 1:DW
     u = ([x,y]-DP)*(DQ-DP)'/((DQ-DP)*(DQ-DP)');
     v = ([x,y]-DP)*perp(DQ-DP)'/norm(DQ-DP);
     SourcePoint = SP+u*(SQ-SP) + v*perp(SQ-SP)/norm(SQ-SP);
     SourcePoint = max([1,1],min([SW,SH], SourcePoint));

     Dest(y,x,:)=Source(round(SourcePoint(2)),round(SourcePoint(1)),:);
   end;
end;
```

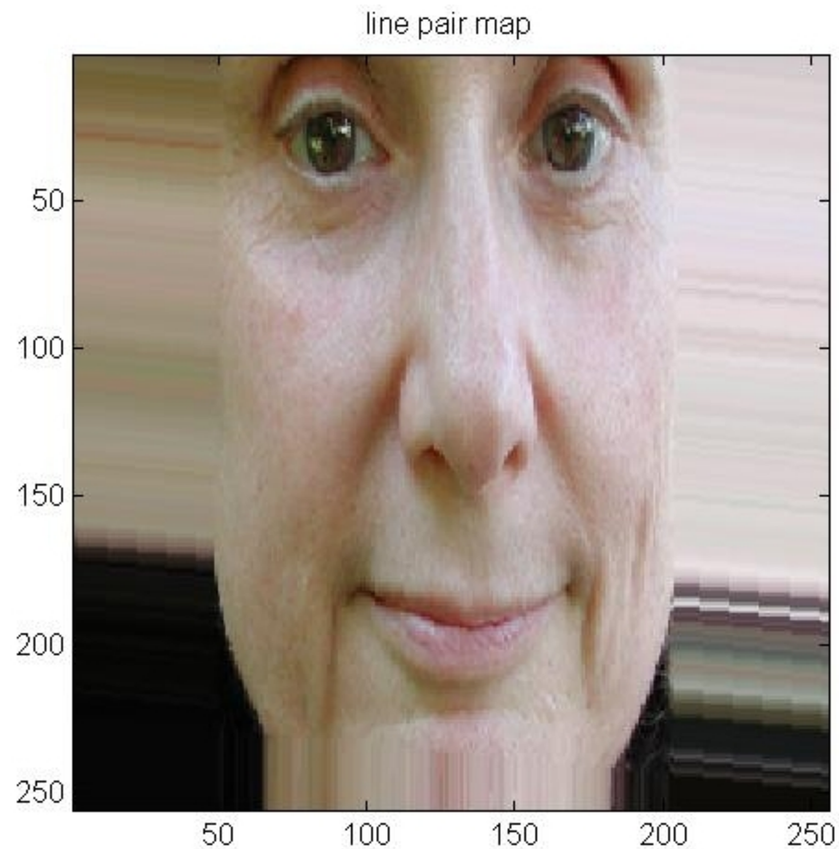


linePairMap.m extras

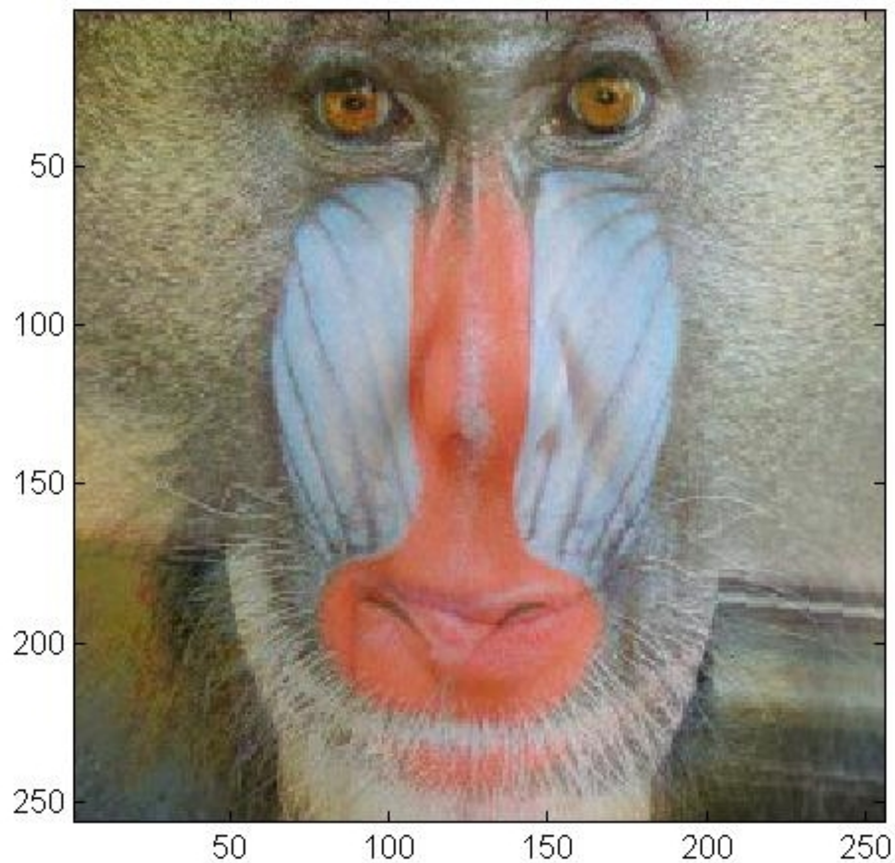
```
% display the image
figure, image(Dest/255,'CDataMapping','scaled');
axis equal;
title('line pair map');
xlim([1,DW]); ylim([1,DH]);


function Vperp = perp(V)
Vperp = [V(2), - V(1)];
```

Line Pair Map

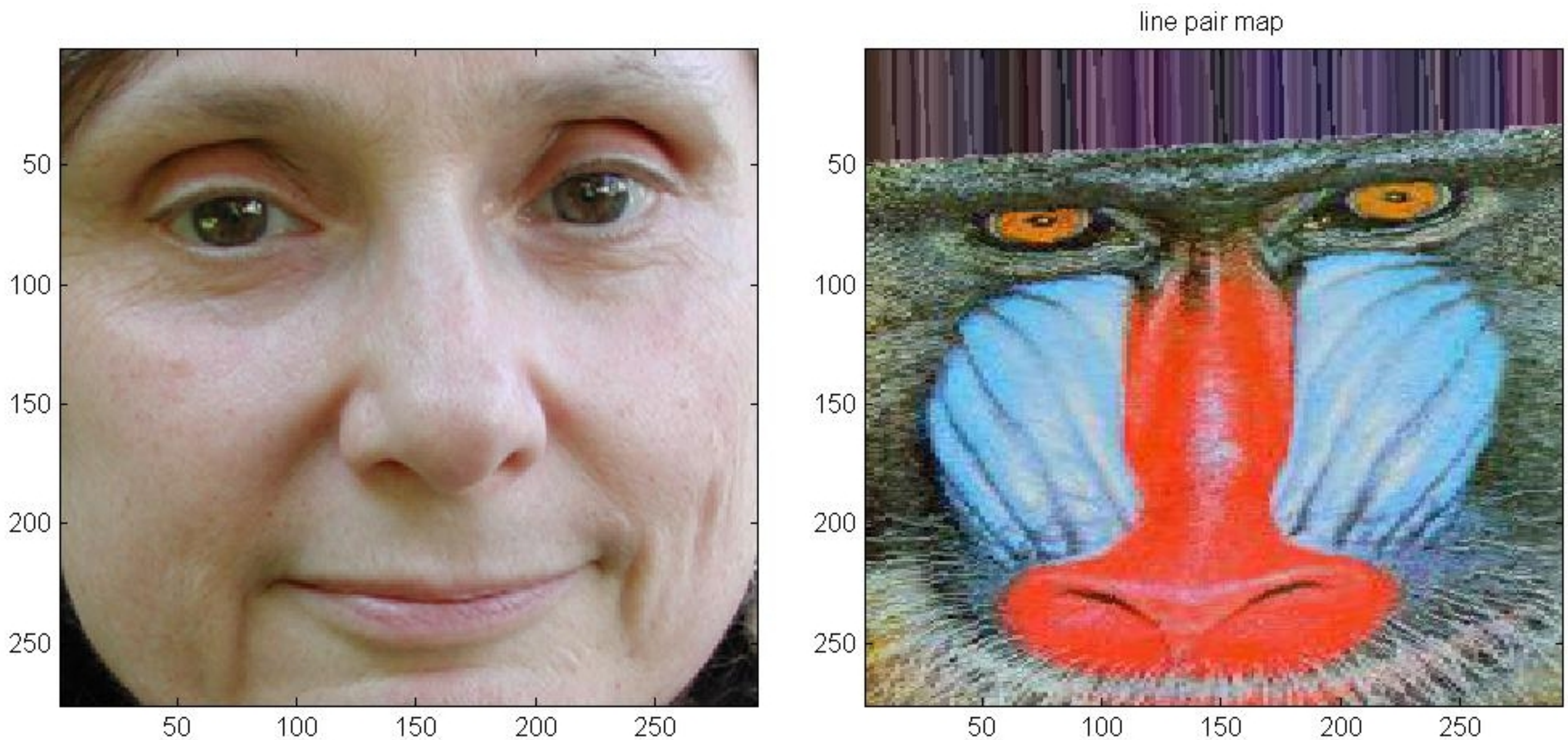




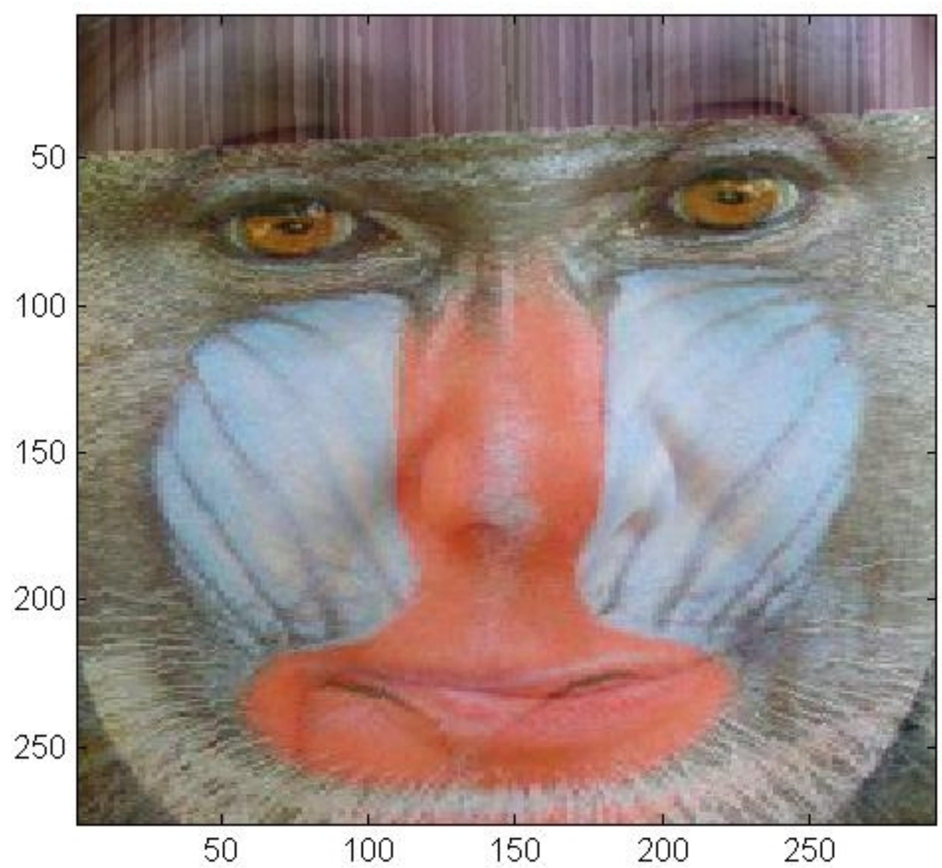
Line Pair Blend



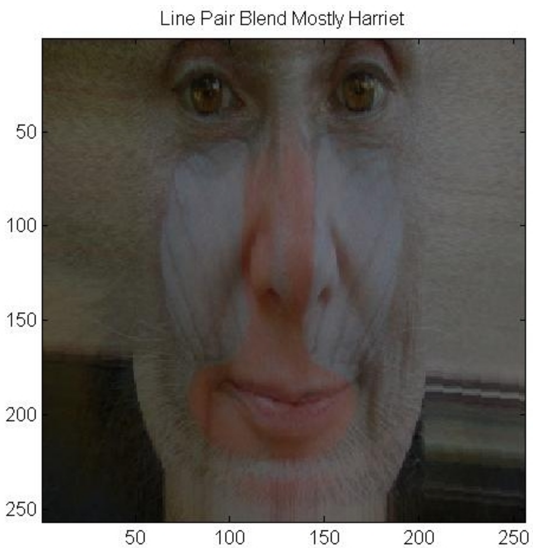
Line Pair Map 2

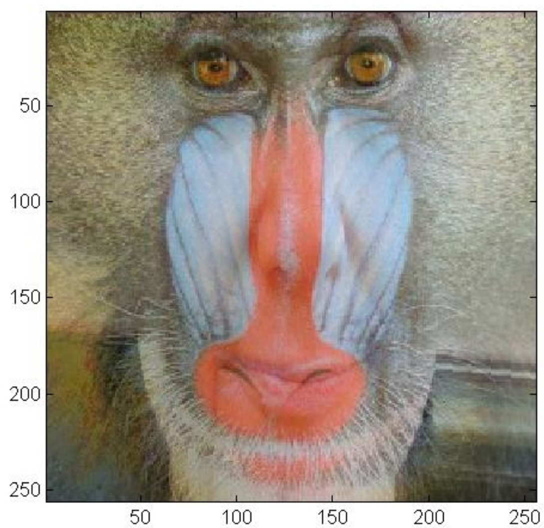


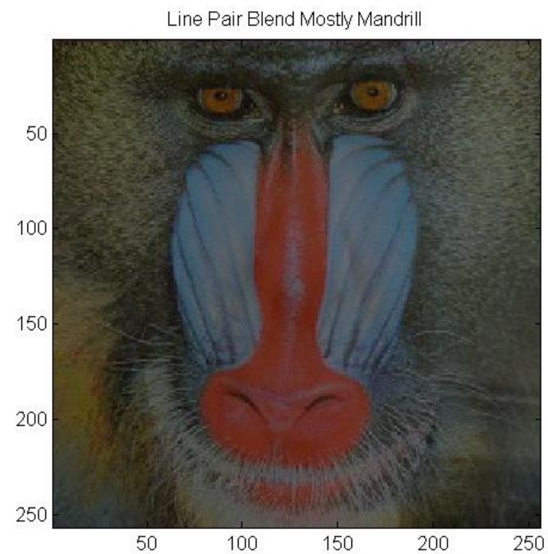
Line Pair Blend 2



Weighted Blends







Multiple Line Pairs

Find Xi' for the ith pair of lines.

Di = Xi' – X

Use a weighted average of the Di.

Weight is determined by the distance from X to the line.

$$weight = \left(\frac{length^p}{(a + dist)} \right)^b$$

length = length of the line

dist is the distance from the pixel to the line

a, b, and p are used to change the relative effect of the lines.

Add average displacement to X to determine X‘.

Let’s Morph

MorphX