# Image Warping

with examples in Matlab™

Marco Zuliani

marco.zuliani@gmail.com

vision.ece.ucsb.edu/~zuliani

October 11, 2008

# Presentation Overview

# What is Image Warping?

## Qualitatively...

Image warping is a transformation that is applied to the domain of an image, which modifies the geometrical properties of the image itself. Ideally, the intensity of the warped image is the same as the intensity of the original image at corresponding points.

Note that the filtering operations you have seen do far act on the range of an image.

*The Persistence of Memory* by Salvador Dalí blurred (filtering operation) and swirled (warping operation).

# The Goals of this Lecture

- After this class you will master...
  - the notion of forward and backward mapping
  - some fundamental mapping functions
  - some methods for image resampling
  - the basic programming techniques to implement image warping
- Disclaimer: some math is needed, but it'll be worthwhile!

# Presentation Overview

# The Formal Definition of an Image

- A (digital) image $\mathbf{I}$ is a function:

$$\begin{aligned} \mathbf{I}(\mathbf{x}) : \mathcal{D} \subseteq \mathbb{R}^n &\rightarrow \mathcal{C} \subseteq \mathbb{R}^m \\ \mathbf{x} &\mapsto \mathbf{I}(\mathbf{x}) \end{aligned}$$

- $m$ is the number of channels of the image (e.g. 1 if the image is gray-level, 3 if the image is RGB or an arbitrary number for multi-spectral images...).
- $n$ is the number of spatial dimensions (e.g. 2 if the images are traditional 2D picture, 3 for Computer Aided Tomography images...).

# Mapping Functions

- Also called warping functions.
- A 2D mapping $\mathbf{T}_{\boldsymbol{\theta}}$ is a function:

$$\begin{aligned} \mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^2 &\rightarrow \mathbb{R}^2 \\ (\mathbf{x}; \boldsymbol{\theta}) &\mapsto \mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}) \end{aligned}$$

- $\boldsymbol{\theta}$ is the vector of parameters of the transformation
- $\mathbf{x}$ is the point to be mapped.
- We will be concerned with invertible mappings, for which $\mathbf{T}_{\boldsymbol{\theta}}^{-1}$ is well defined.
    - Technically speaking we are considering bijective functions to perform the mapping.

# Formal Definition of Image Warping

**Our qualitative definition...**

Image warping is a transformation that is applied to the domain of an image, which modifies the geometrical properties of the image itself. Ideally, the intensity of the warped image is the same as the intensity of the original image at corresponding points.

**The analytical definition...**

Consider an image $\mathbf{I}$. Image warping produces a new image $\mathbf{I}'$ such that:

$$\mathbf{I}'(\mathbf{T}_\theta(\mathbf{x})) = \mathbf{I}(\mathbf{x})$$

for each $\mathbf{x} \in \mathcal{D}$.

- Note that $\mathbf{I} \neq \mathbf{I}'$. In general:
  - $\mathbf{I}$ and $\mathbf{I}'$ have different domain and range
  - The function return different values at the same location (if defined)

# Formal Definition of Image Warping

## Our qualitative definition. . .

Image warping is a transformation that is applied to the domain of an image, which modifies the geometrical properties of the image itself. Ideally, the intensity of the warped image is the same as the intensity of the original image at corresponding points.

## The analytical definition. . .

Consider an image $\mathbf{I}$. Image warping produces a new image $\mathbf{I}'$ such that:

$$\mathbf{I}'(\mathbf{T}_\theta(\mathbf{x})) = \mathbf{I}(\mathbf{x})$$

for each $\mathbf{x} \in \mathcal{D}$.

- Note that $\mathbf{I} \neq \mathbf{I}'$. In general:
  - $\mathbf{I}$ and $\mathbf{I}'$ have different domain and range
  - The function return different values at the same location (if defined)

# Formal Definition of Image Warping

**Our qualitative definition...**

Image warping is a transformation that is applied to the domain of an image, which modifies the geometrical properties of the image itself. Ideally, the intensity of the warped image is the same as the intensity of the original image at corresponding points.

**The analytical definition...**

Consider an image $\mathbf{I}$. Image warping produces a new image $\mathbf{I}'$ such that:

$$\mathbf{I}'(\mathbf{T}_\theta(\mathbf{x})) = \mathbf{I}(\mathbf{x})$$

for each $\mathbf{x} \in \mathcal{D}$.

- Note that $\mathbf{I} \neq \mathbf{I}'$. In general:
  - $\mathbf{I}$ and $\mathbf{I}'$ have different domain and range
  - The function return different values at the same location (if defined)

# Rotation Scale and Translation Mapping

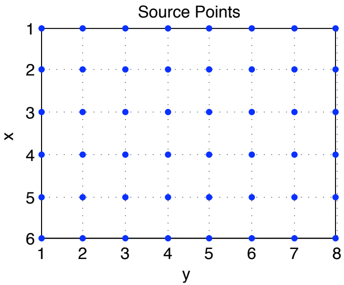
- Models a rotation, a scaling and a translation
- Forward mapping (belongs to the class of affine transformations):

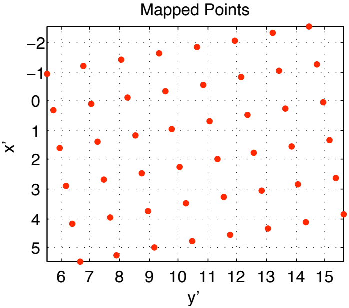
$$\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}) = \underbrace{s \begin{bmatrix} \cos\phi & -sin\phi \\ sin\phi & \cos\phi \end{bmatrix}}_{\text{scaling and rotation}} \mathbf{x} + \underbrace{\begin{bmatrix} t_x \\ t_y \end{bmatrix}}_{\text{translation}}$$

- $\boldsymbol{\theta} = \begin{bmatrix} t_x & t_y & s & \phi \end{bmatrix}^T \in \mathbb{R}^4$ encodes the rotation, the scaling and the translation
- The transformation has 4 degrees of freedom
- Backward mapping:

$$\mathbf{T}_{\boldsymbol{\theta}}^{-1}(\mathbf{y}) = \frac{1}{s} \begin{bmatrix} \cos\phi & sin\phi \\ -sin\phi & \cos\phi \end{bmatrix} \left( \mathbf{y} - \begin{bmatrix} t_x \\ t_y \end{bmatrix} \right)$$

# Rotation Scale and Translation Mapping - Example




- Remarks:
  - Axis orientation is à la Fortan
  - Origin is at $(1, 1)$, à la Matlab
  - The mapped points do not have integer coordinates!

# Swirling Mapping

- Cool effect (see also GIMP) to swirl an image around a point
- Forward mapping (easier in polar coordinates):

$$\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{bmatrix} \rho \cos(\phi + k\rho) \\ \rho \sin(\phi + k\rho) \end{bmatrix} + \mathbf{x}_c$$

- $\rho = \|\mathbf{x} - \mathbf{x}_c\|$, $\phi = \angle(\mathbf{x} - \mathbf{x}_c)$
- The transformation has 3 degrees of freedom
- Backward mapping: left as exercise

# Homographic Mapping

- Models the transformation of a planar surface seen from two pin-hole cameras from different points of view
- Forward mapping (it is a rational function, nonlinear in Euclidean space):

$$\mathbf{T}_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{bmatrix} \frac{\theta_1 x_1 + \theta_4 x_2 + \theta_7}{\theta_3 x_1 + \theta_6 x_2 + \theta_9} \\ \\ \frac{\theta_2 x_1 + \theta_5 x_2 + \theta_8}{\theta_3 x_1 + \theta_6 x_2 + \theta_9} \end{bmatrix}$$

  - It is linear in the projective space $\mathbb{P}^2$...
- The transformation has 8 degrees of freedom
  - Scale is immaterial
- Backward mapping: left as exercise

# Presentation Overview

# Motivation

- The backward and forward mapping in general yield points that have non integer coordinates.
- Images are usually defined over a discrete lattice defined at integer locations.
- Goal: Estimate (resample) the intensity value at a non integer location.
- Common resampling methods:
  - Nearest neighbour
  - Bilinear
  - Cubic
  - Lanczos
  - ...

# Motivation

- The backward and forward mapping in general yield points that have non integer coordinates.
- Images are usually defined over a discrete lattice defined at integer locations.
- Goal: Estimate (resample) the intensity value at a non integer location.
- Common resampling methods:
  - Nearest neighbour
  - Bilinear
  - Cubic
  - Lanczos
  - ...

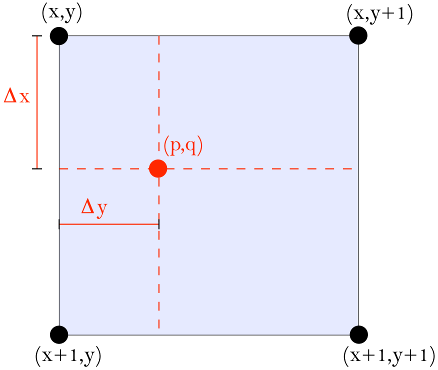
# Nearest Neighbor Interpolation

## The question

Let $x$ and $y$ be the integer coordinates of the lattice. What is the value of $f$ at $\begin{bmatrix} p & q \end{bmatrix}^T$?

## Nearest Neighbour Answer

$\hat{f}(p, q) = f(\text{round}(p), \text{round}(q))$



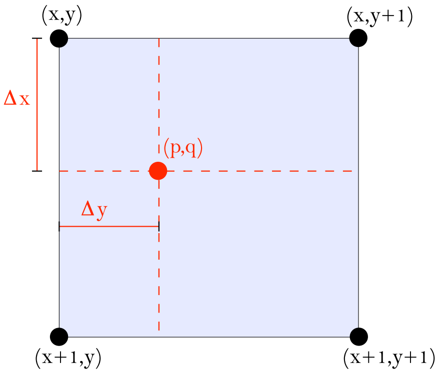
# Nearest Neighbor Interpolation

## The question

Let $x$ and $y$ be the integer coordinates of the lattice. What is the value of $f$ at $\begin{bmatrix} p & q \end{bmatrix}^T$?

## Nearest Neighbour Answer

$\hat{f}(p,q) = f(\text{round}(p), \text{round}(q))$



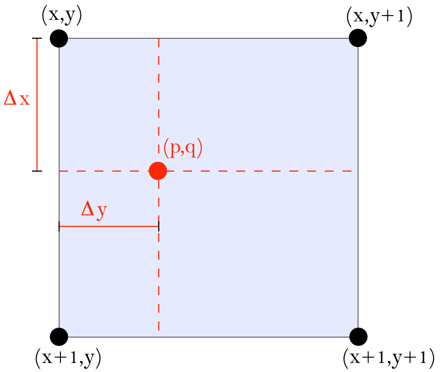
# Nearest Neighbor Interpolation

## The question

Let $x$ and $y$ be the integer coordinates of the lattice. What is the value of $f$ at $\begin{bmatrix} p & q \end{bmatrix}^T$?

## Nearest Neighbour Answer

$$\hat{f}(p, q) = f(\text{round}(p), \text{round}(q))$$

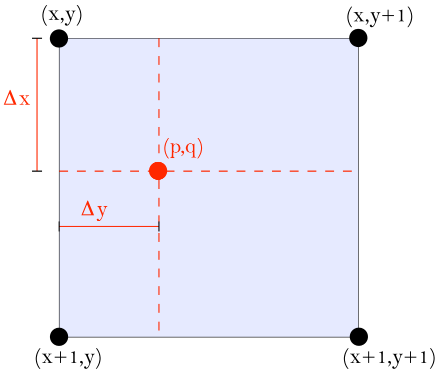


# Bilinear Interpolation: Notation

## The question

Let $x$ and $y$ be the integer coordinates of the lattice. What is the value of $f$ at $\begin{bmatrix} p & q \end{bmatrix}^T$?

- $F_{0,0} \stackrel{\text{def}}{=} f(x, y)$
- $F_{1,0} \stackrel{\text{def}}{=} f(x+1, y)$
- $F_{0,1} \stackrel{\text{def}}{=} f(x, y+1)$
- $F_{1,1} \stackrel{\text{def}}{=} f(x+1, y+1)$
- $\Delta x \stackrel{\text{def}}{=} p - x$ and $\Delta y \stackrel{\text{def}}{=} q - y$

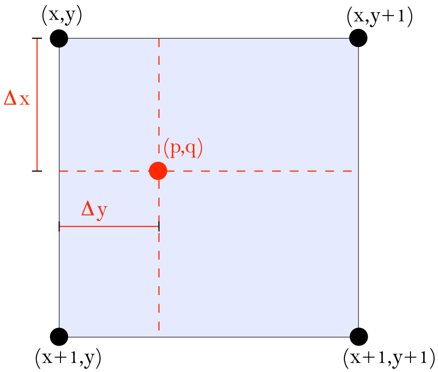


# Bilinear Interpolation: Notation

## The question

Let $x$ and $y$ be the integer coordinates of the lattice. What is the value of $f$ at $\begin{bmatrix} p & q \end{bmatrix}^T$?

- $F_{0,0} \stackrel{\text{def}}{=} f(x, y)$
- $F_{1,0} \stackrel{\text{def}}{=} f(x+1, y)$
- $F_{0,1} \stackrel{\text{def}}{=} f(x, y+1)$
- $F_{1,1} \stackrel{\text{def}}{=} f(x+1, y+1)$
- $\Delta x \stackrel{\text{def}}{=} p - x$ and $\Delta y \stackrel{\text{def}}{=} q - y$

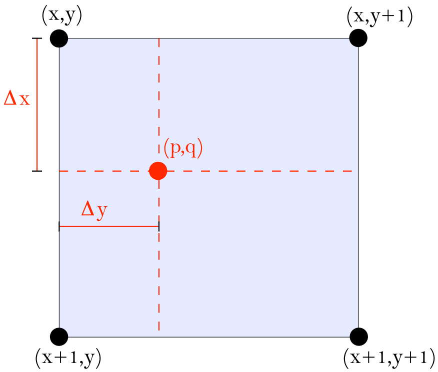


# Bilinear Interpolation: Notation

## The question

Let $x$ and $y$ be the integer coordinates of the lattice. What is the value of $f$ at $\begin{bmatrix} p & q \end{bmatrix}^T$?

- $F_{0,0} \stackrel{\text{def}}{=} f(x, y)$
- $F_{1,0} \stackrel{\text{def}}{=} f(x + 1, y)$
- $F_{0,1} \stackrel{\text{def}}{=} f(x, y + 1)$
- $F_{1,1} \stackrel{\text{def}}{=} f(x + 1, y + 1)$
- $\Delta x \stackrel{\text{def}}{=} p - x$ and $\Delta y \stackrel{\text{def}}{=} q - y$



# Bilinear Interpolation - I

## The question

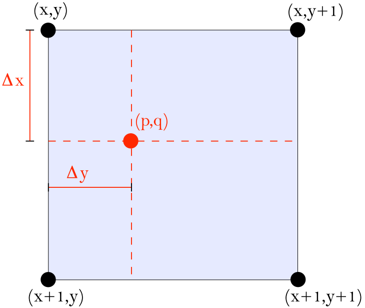
Let $x$ and $y$ be the integer coordinates of the lattice. What is the value of $f$ at $\begin{bmatrix} p & q \end{bmatrix}^T$?

- Linear interpolation in the $x$ direction:

$$\begin{aligned} f_y(\Delta x) &= (1-\Delta x)F_{0,0} + \Delta x F_{1,0} \\ f_{y+1}(\Delta x) &= (1-\Delta x)F_{0,1} + \Delta x F_{1,1} \end{aligned}$$

- Linear interpolation in the $y$ direction:

$$\hat{f}(p,q) = (1-\Delta y) f_y + \Delta y f_{y+1}$$



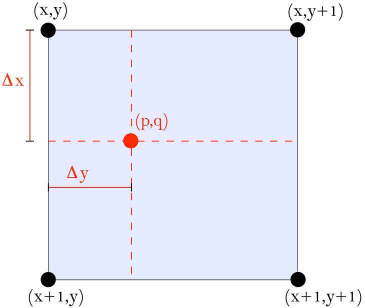
## The question

Let $x$ and $y$ be the integer coordinates of the lattice. What is the value of $f$ at $\begin{bmatrix} p & q \end{bmatrix}^T$?

## Bilinear Interpolation Answer

Note that $\hat{f}(p,q)$ "passes through" the samples.

$$\hat{f}(p,q) = (1-\Delta y)(1-\Delta x)F_{0,0} + (1-\Delta y)\Delta x F_{1,0} + \Delta y(1-\Delta x)F_{0,1} + \Delta y \Delta x F_{1,1}$$



# Bilinear Interpolation - II

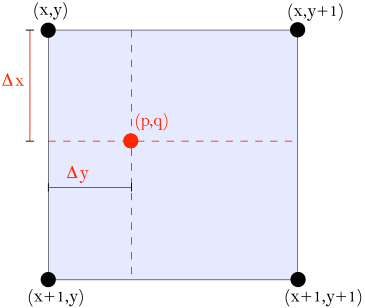
## The question

Let $x$ and $y$ be the integer coordinates of the lattice. What is the value of $f$ at $\begin{bmatrix} p & q \end{bmatrix}^T$?

## Bilinear Interpolation Answer

Note that $\hat{f}(p,q)$ "passes through" the samples.

$$\begin{aligned}\hat{f}(p,q) = &(1-\Delta y)(1-\Delta x)F_{0,0} + \\ &(1-\Delta y)\Delta x F_{1,0} + \\ &\Delta y(1-\Delta x)F_{0,1} + \\ &\Delta y \Delta x F_{1,1}\end{aligned}$$



# A Word Regarding Cubic Interpolation

- Common misconception: "it is obtained fitting a parabola rather than a line and proceeding as for bilinear interpolation"
- No! A little bit more involved...
  - Not only the value of the function is matched at the vertices of the lattice, but also the derivatives
  - A neighborhood of $4 \times 4$ points is needed (derivatives must be estimated via finite differences)
  - For an appropriate choice of the coefficients $a_{i,j}$:

$$\hat{f}(p,q) = \sum_{i,j=-1}^{2} a_{i,j} \Delta x^{i+1} \Delta y^{j+1}$$

# Image Resampling Comparison

Clock detail from *The Persistence of Memory* rotated by 30 degrees and magnified of 20%. The left images is obtained via nearest neighbour interpolation, the middle figure via bilinear interpolation and the right one via bicubic interpolation.

# Presentation Overview

# The Warping Recipe

- The fundamental steps of a warping algorithm are the following:
  1. Computation of the bounding box of the warped image (forward mapping).
  2. Backward mapping of lattice points that sample the bounding box of the warped image (avoid "holes")
  3. Validation of the backward mapped points (must belong to the domain of the source image)
  4. Intensity transfer via resampling
- We will look into their implementation using Matlab™ in the next slides

# Step 1: Bounding Box

```
function bw = computeBoundingBox(bs, fun, varargin)

% use single precision to speed up the calculations
[xs ys] = ndgrid(bs(1):bs(2), bs(3):bs(4));
[xw yw] = feval(fun, ...
    single(xs), single(ys), ...
    true, varargin{:});

bw(1) = floor(min(xw));
bw(2) = ceil(max(xw));
bw(3) = floor(min(yw));
bw(4) = ceil(max(yw));

return
```

- The bounding box of the warped image is defined by signed integers

# Step 2: Backward Mapping

```
% sample inside the bounding box of the destination image
[xw yw] = ndgrid(options.bw(1):options.bw(2), ...
    options.bw(3):options.bw(4));
% apply the inverse mapping
[xs ys] = feval(fun, single(xw), single(yw), ...
    false, varargin{:});
```

- The bounding box of the warped image is sampled at the integer locations of the lattice to avoid holes
- Such samples are mapped backwards onto the original image

# Step 3: Validation

```
% define the margin according to the interpolation method
switch (options.interp_method)
    case 0
        margin = [0 0];
    case 1
        margin = [0 1];
    case 2
        margin = [1 2];
end;

% make sure the point in the source image stay withing the source
% bounding box
flag = ...
    (xs ≥ 1+margin(1)) & (xs ≤ heights-margin(2)) & ...
    (ys ≥ 1+margin(1)) & (ys ≤ widths-margin(2));

% filter away the bad points. Note the use of logical indexing,
% in general faster than using find
xs = xs(flag);
ys = ys(flag);
xw = xw(flag);
yw = yw(flag);
```

- Margins are imposed for the sampling
- We reject the backward mapped points that do not belong to the source image domain

# Step 4: Intensity Transfer Via Resampling

```
% compute the linear indices
indw = heightw*(yw-options.bw(3))+(xw-options.bw(1)+1);

% preallocate the warped image (single precision)
Iw = zeros(heightw, widthw, Nc, 'single');

% populate each bit plane of the image
offsetw = 0;
for nc = 1:Nc
    Iwi = imsample(Is(:,:,nc), xs, ys, options.interp_method);
    Iw(indw+offsetw) = Iwi;
    offsetw = offsetw + sizew;
end;
```

- The linear indices are computed the column major convention (Fortran and Matlab) and setting the starting index to 1
- For multichannel images we offset the linear indices
- `imsample` function performs the interpolation

# Resampling Code Snippet

```
Is = zeros(1, N, 'single');
x = single([xs; ys]);

for h = 1:N

    xx = floor(x(:, h));

    switch mode

        % nearest neighbour
        case 0

            Is(h) = single(I(xx(1), xx(2)));

        % bilinear
        case 1

            Δ = x(:, h) − xx;
            Δ_m = 1 − Δ;

            F_00 = single(I(xx(1), xx(2)));
            F_01 = single(I(xx(1), xx(2)+1));
            F_10 = single(I(xx(1)+1, xx(2)));
            F_11 = single(I(xx(1)+1, xx(2)+1));

            F_A = F_00*Δ_m(1)+F_10*Δ(1);
            F_B = F_01*Δ_m(1)+F_11*Δ(1);

            Is(h) = F_A*Δ_m(2)+F_B*Δ(2);
```

# Presentation Overview

# Anamorphosis

- Etymology: from Greek anamorphōsis, forming anew (ana-, again + morphoun, to form)
- A pictorial technique introduced by Renaissance painters to hide images in images by distorting (or, using our terminology, by warping) them.
- Hidden images can be seen looking at the painting from a specific point of view.
- Paintings are planar: the transformation must be represented by an homography!

The Ambassadors by Hans Holbein the Younger, 1533

# Discovering The Skull via an Homographic Transformation

The original painting "The Ambassadors" (left) warped according to an homographic trasformation (center) that reveals the skull in the painting (right, detail). Bicubic interpolation was used.

# Warping for Image Registration & Mosaicking

- Image registration:
  - establish a mapping between two or more images possibly taken:
    - at different times,
    - from different viewpoints,
    - under different lighting conditions,
    - and/or by different sensors
  - align the images with respect to a common coordinate system coherently with the three dimensional structure of the scene
    - Warping happens using the techniques illustrated in these slides
- Image mosaicking: images are combined to provide a representation of the scene that is both geometrically and photometrically consistent.

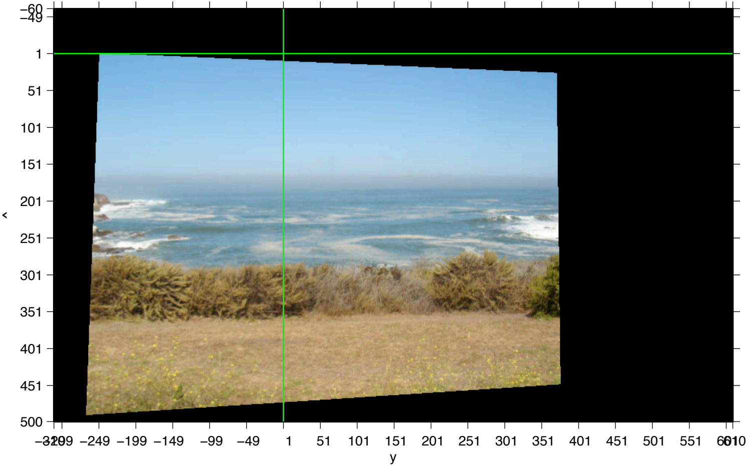
# Warping for Image Registration & Mosaicking

Overlapping views from Montaña De Oro, courtesy of C. Kumor

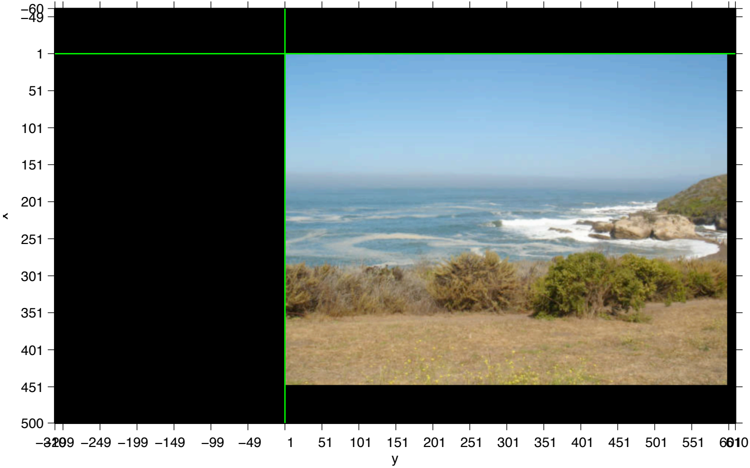
- The warping model can be approximated by an homography even if the scene:
  - is not planar
  - is not rigid

# Registration Warping Example - I




Left image warped onto the canvas (bicubic interpolation was used). Note the canvas bounds.

# Registration Warping Example - II




Right image warped onto the canvas (bicubic interpolation was used). Note the canvas bounds.

# Presentation Overview

# Wrapping Things Up

- In this tutorial you have been introduced to:
  - the notion of forward and backward mapping
  - some fundamental mapping functions
  - some methods for image resampling
  - the basic programming techniques to implement image warping
- Caveats:
  - Pay attention to the convention of the coordinates system
  - Be consistent according to the fact that the image origin is located at $(1, 1)$ or $(0, 0)$
  - Make sure the forward and backward mapping are correctly defined and implemented
  - Make sure you don't exceed the image bounds (border issues for the interpolation methods)