The Lucas & Kanade Algorithm

Instructor - Simon Lucey

16-720B - Computer Vision

Carnegie Mellon
THE ROBOTICS INSTITUTE

Today

- Registration, Registration, Registration.
- Linearizing Registration.
- Lucas & Kanade Algorithm.

3 Biggest Problems in Computer Vision?

3 Biggest Problems in Computer Vision?

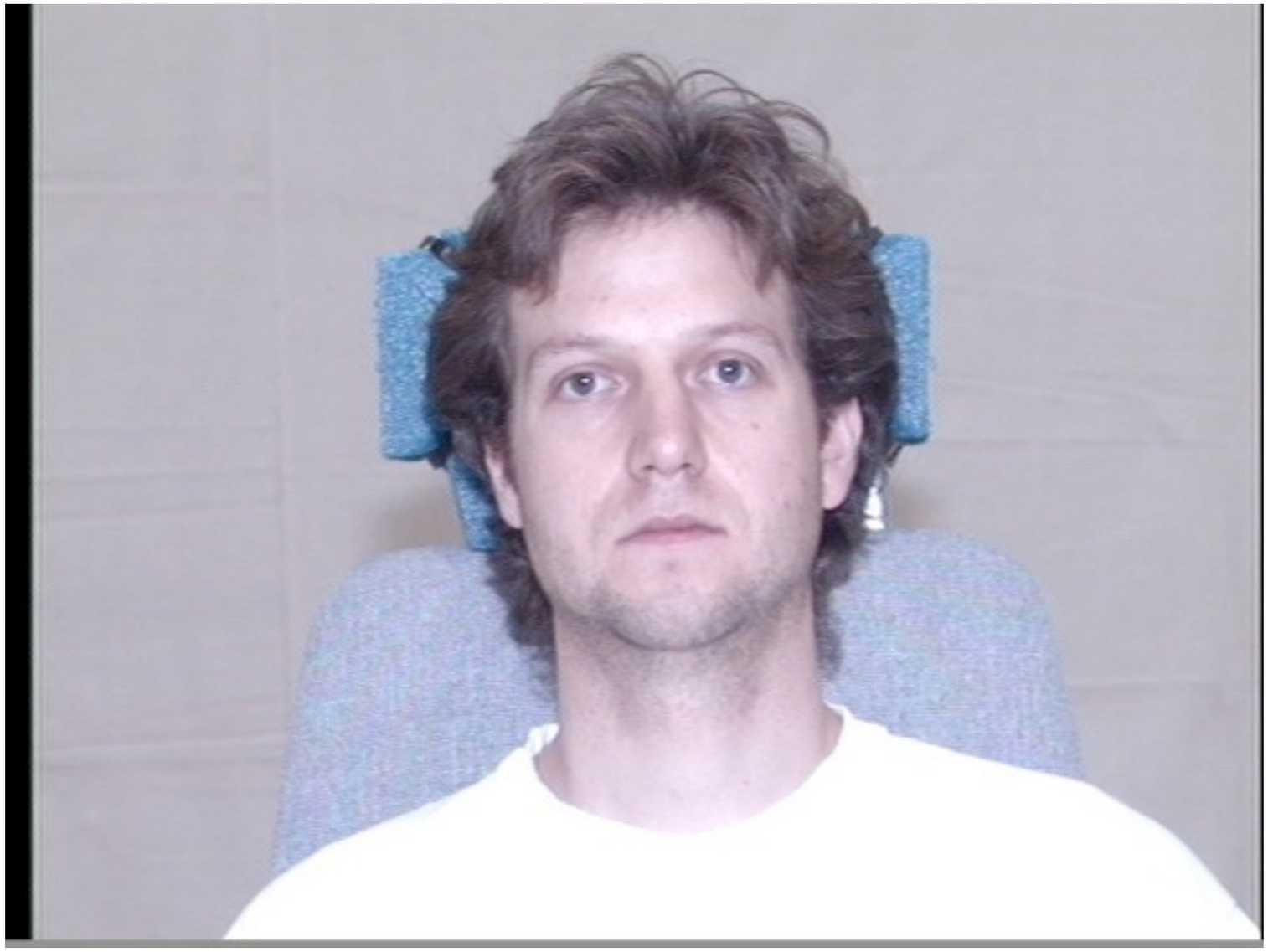
REGISTRATION

REGISTRATION

REGISTRATION

(Prof. Takeo Kanade - Robotics Institute - CMU.)

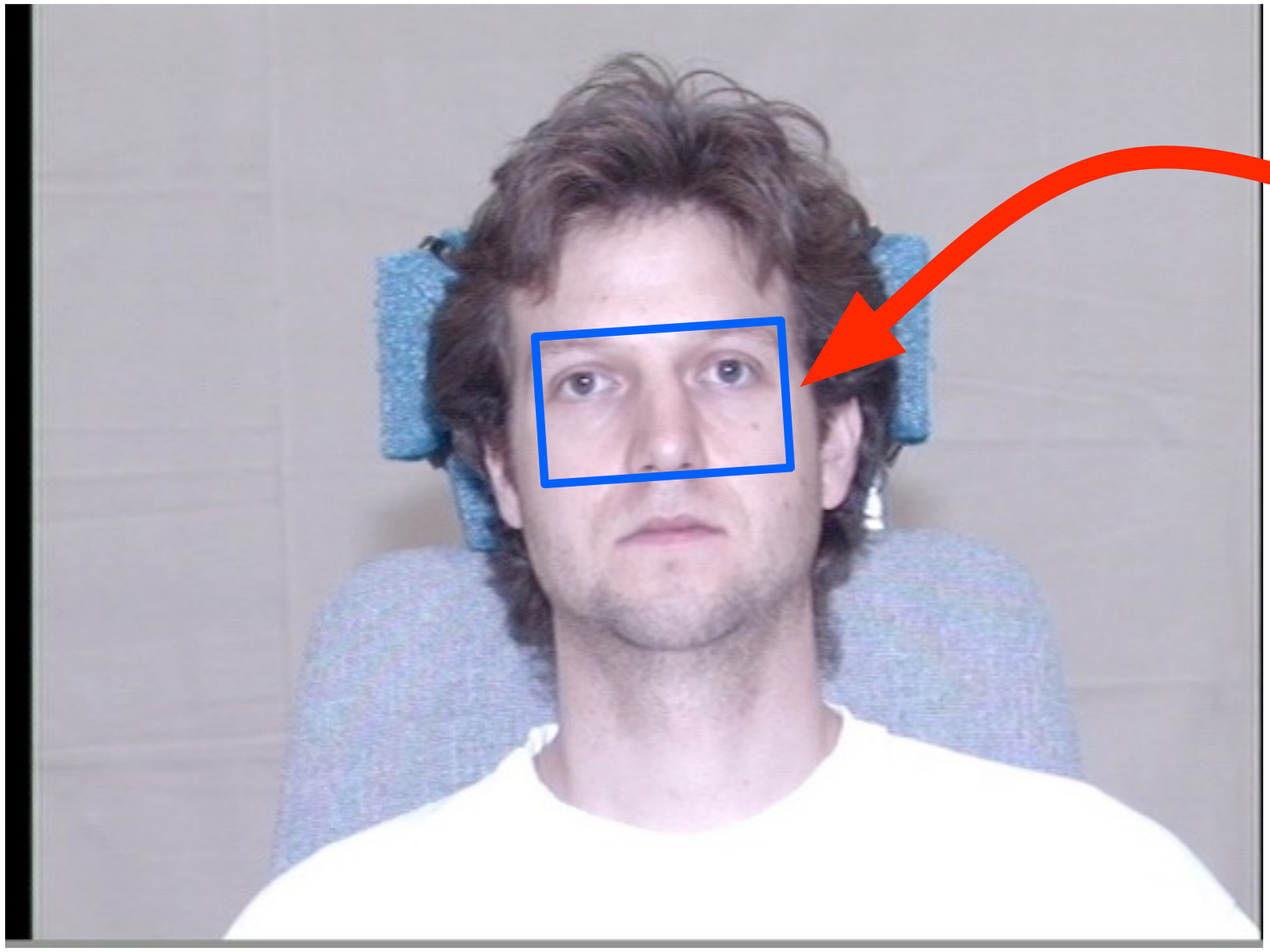
What is Registration?



“Source Image”

“Template”

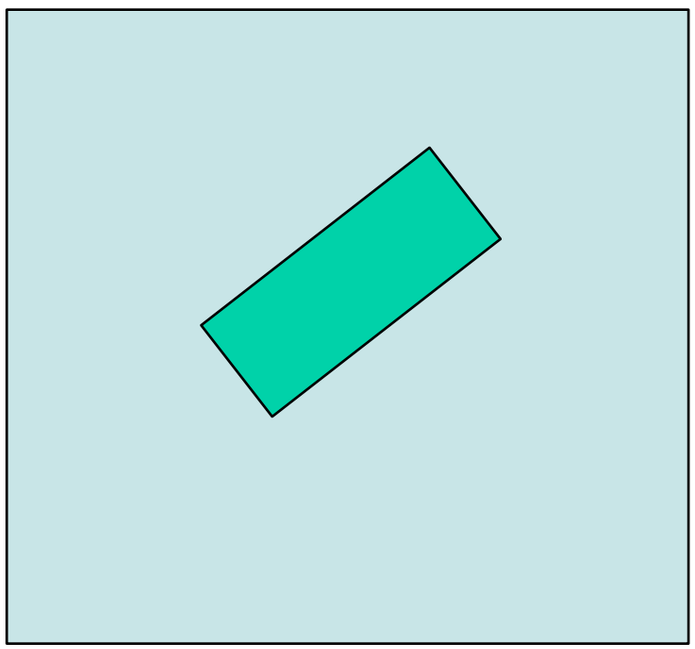
What is Registration?



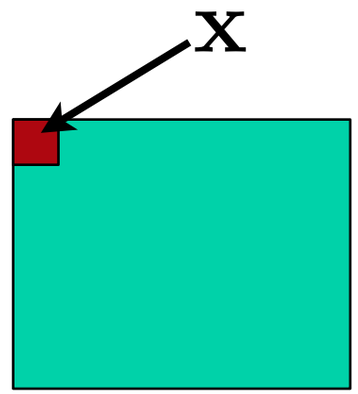
“Source Image”

“Template”

What is Registration?

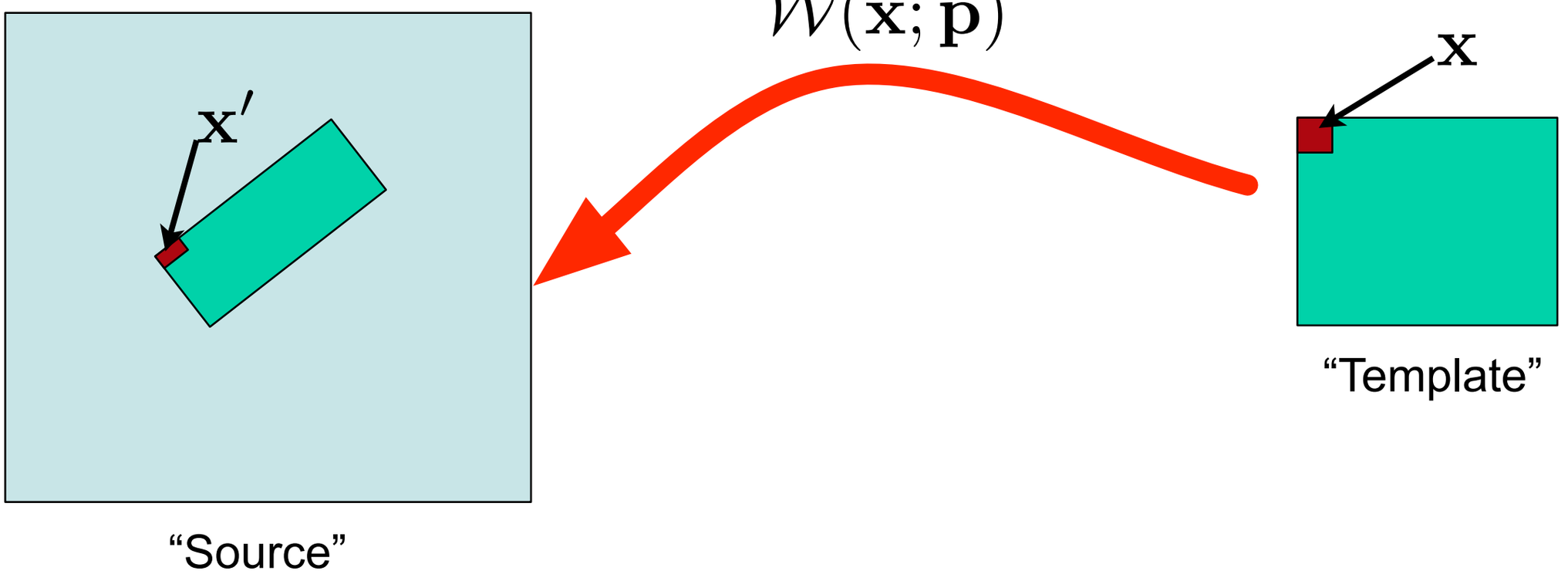


“Source”



“Template”

What is Registration?



Our goal is to find the warp parameter vector $\mathbf{p}$!

$\mathbf{x}$ = coordinate in template $[x, y]^T$

$\mathbf{x}'$ = corresponding coordinate in source $[x', y']^T$

$\mathcal{W}(\mathbf{x};\mathbf{p})$ = warping function such that $\mathbf{x}' = \mathcal{W}(\mathbf{x};\mathbf{p})$

$\mathbf{p}$ = parameter vector describing warp

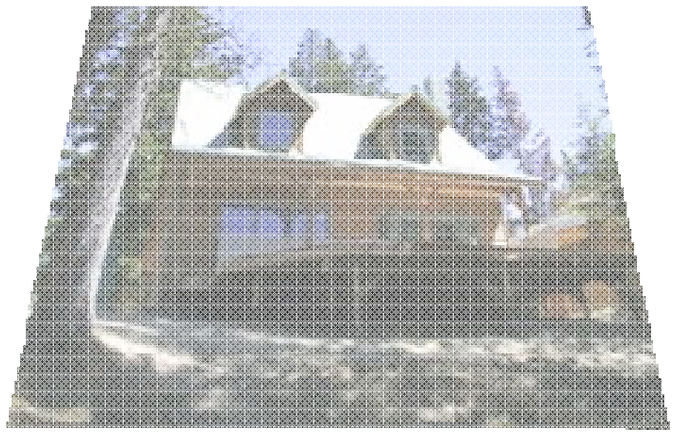
Different Warp Functions

translation

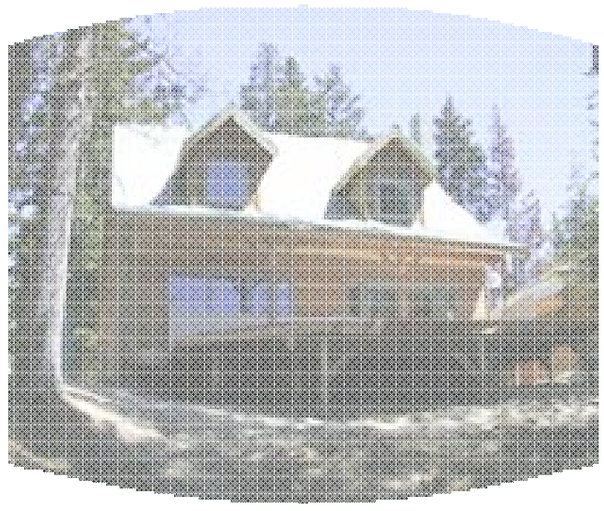
rotation

aspect

affine



perspective



cylindrical

(Szeliski and Fleet)

Defining Warp Functions

$$\mathcal{W}(\mathbf{x};\mathbf{p}) = \mathbf{x} + \mathbf{p}$$

$$\mathbf{p} = [p_1 \; p_2]^T$$

translation

$$\mathcal{W}(\mathbf{x};\mathbf{p}) = \mathbf{M}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix}1-p_1 & p_2 & p_3\\ p_4 & 1-p_5 & p_6\end{bmatrix}$$

$$\mathbf{p} = [p_1 \ldots p_6]^T$$

affine

rotation

aspect

translation

Defining Warp Functions

$$\mathcal{W}(\mathbf{x};\mathbf{p}) = \mathbf{x} + \mathbf{p}$$

$$\mathbf{p} = [p_1 \; p_2]^T$$

translation

$$\mathcal{W}(\mathbf{x};\mathbf{p}) = \mathbf{M}\begin{bmatrix}\mathbf{x}\\1\end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix}1+p_1 & p_2 & p_3\\ p_4 & 1+p_5 & p_6\end{bmatrix}$$

$$\mathbf{p} = [p_1 \ldots p_6]^T$$

affine

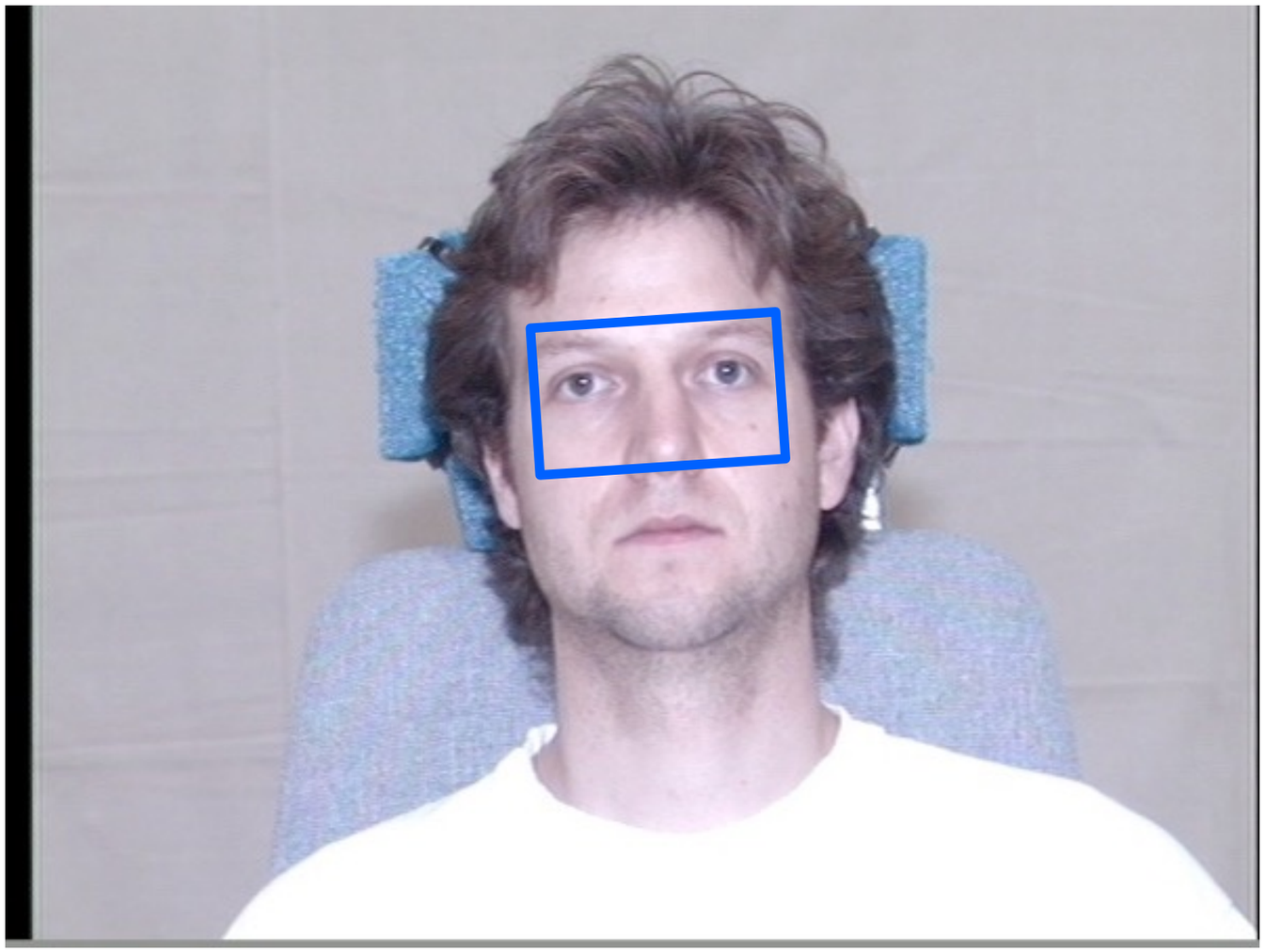
rotation

aspect

translation

Naive Approach to Registration

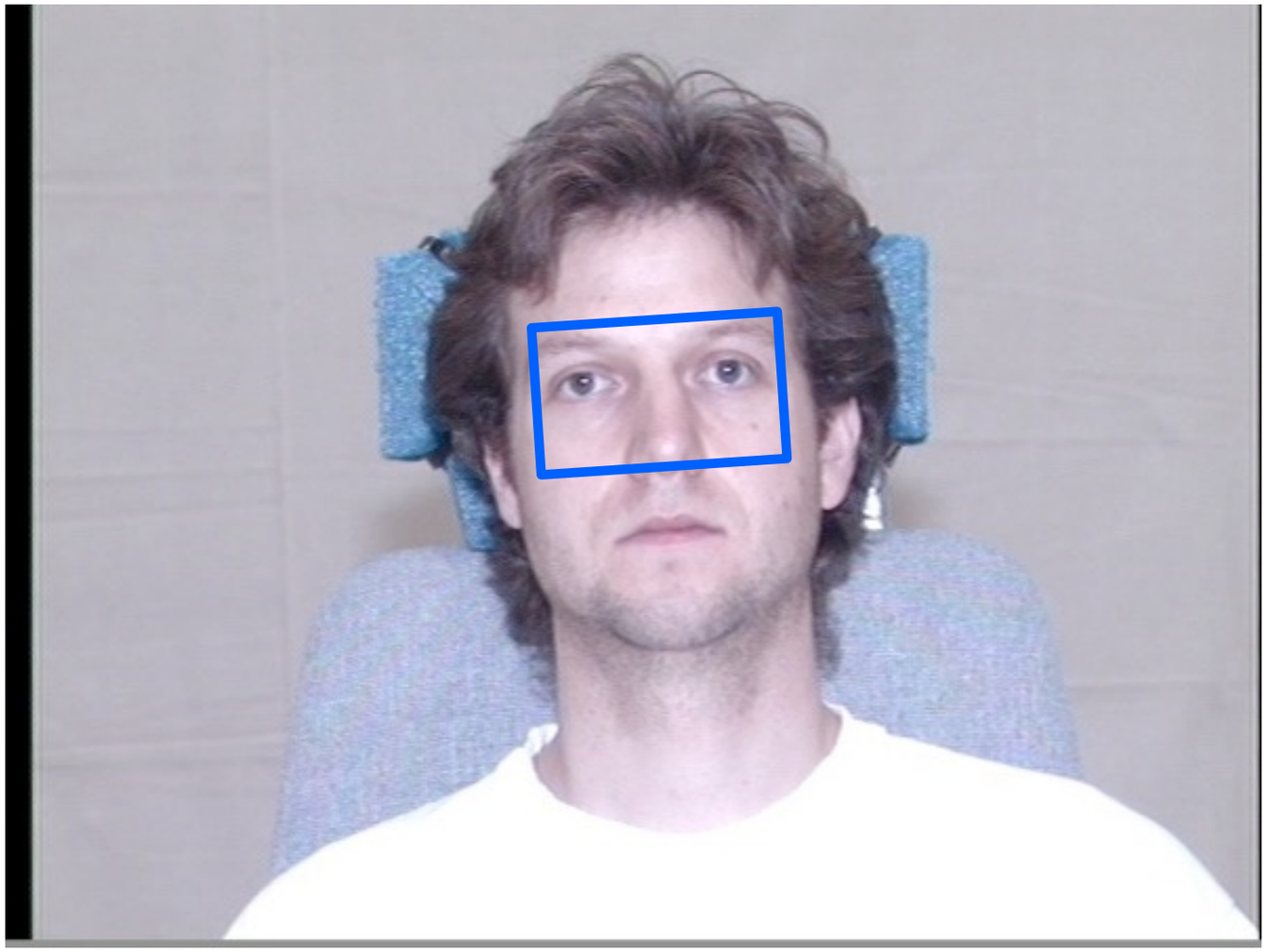
- If you were a person coming straight from machine learning you might suggest,



“Images of Object at various warps”

Naive Approach to Registration

- If you were a person coming straight from machine learning you might suggest,



"Images of Object at various warps"

[255,134,45,.....,34,12,124,67]

[123,244,12,.....,134,122,24,02]

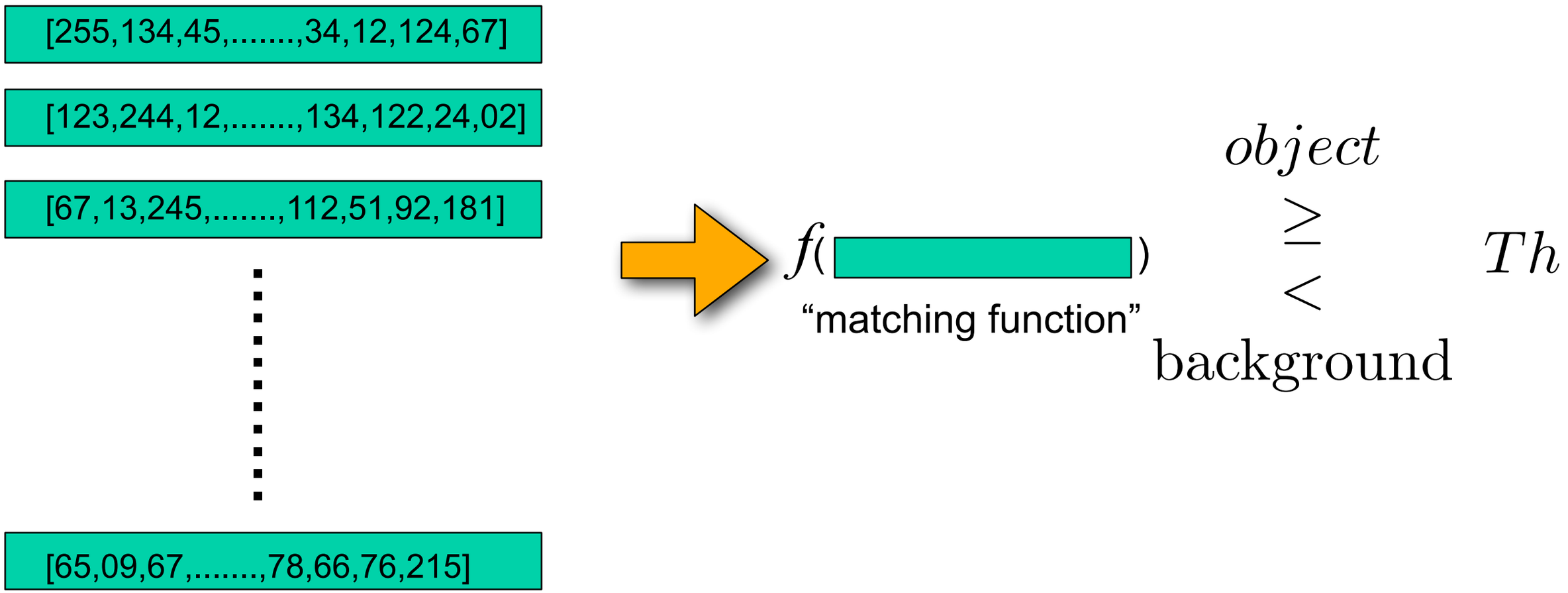
[67,13,245,.....,112,51,92,181]

[65,09,67,.....,78,66,76,215]

"Vectors of pixel values at each warp position"

Naive Approach to Registration

- If you were a person coming straight from machine learning you might suggest,



"Vectors of pixel values at each warp position"

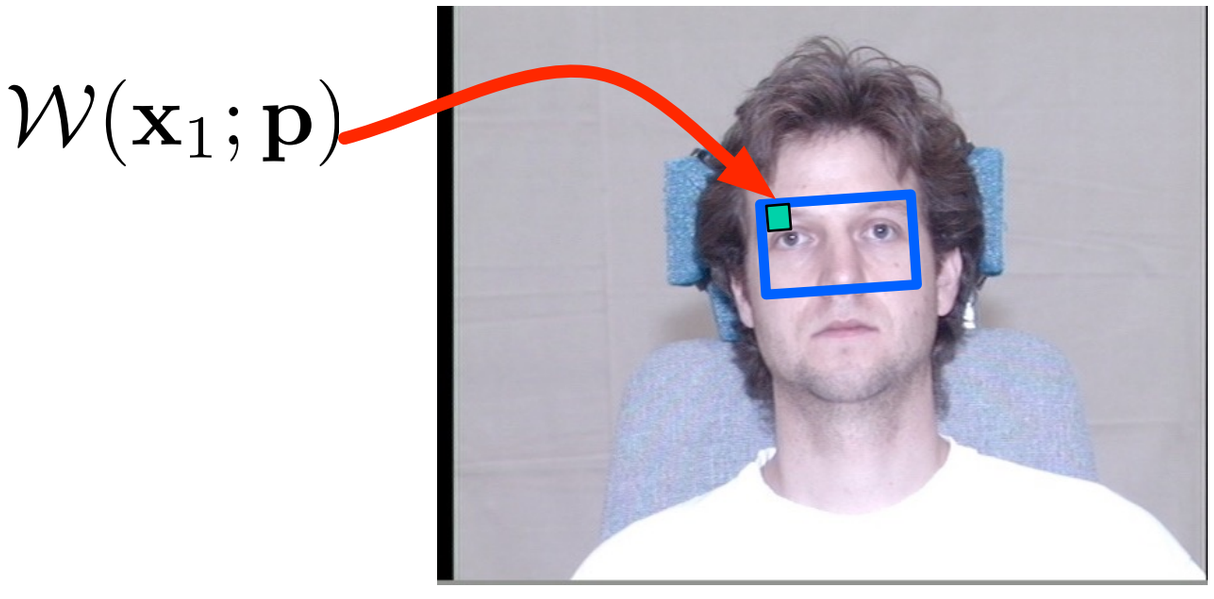
Naive Approach to Registration

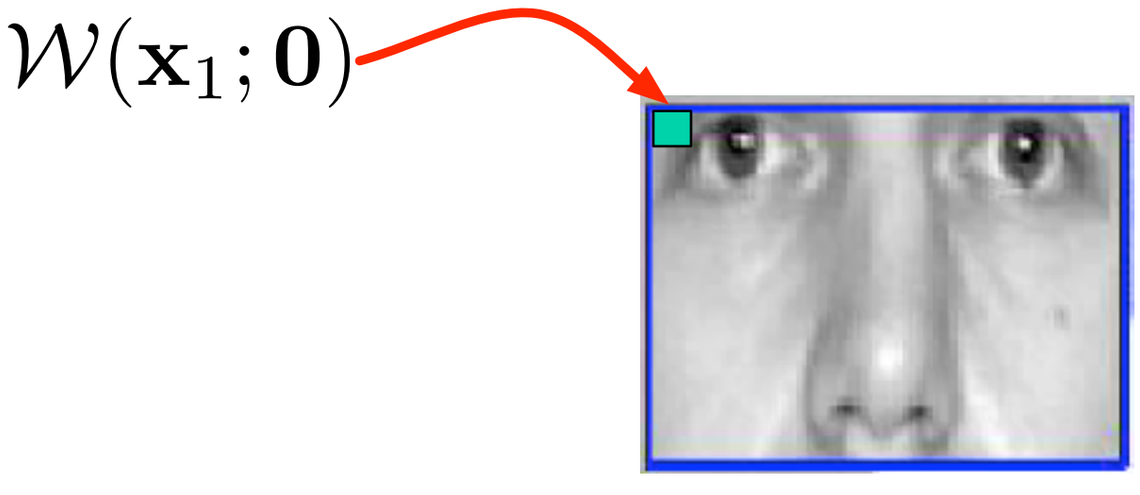
- Problem,
  - Do we sample every warp parameter value?
  - Plausible if we are only doing translation?
  - If the image is high resolution, do we need to sample every pixel?
  - What happens if warps are more complicated (e.g., scale)?
  - Becomes prohibitively expensive computationally as warp dimensionality expands.

Measures of Image Similarity

- Although not always perfect, a common measure of image similarity is:
  - Sum of squared differences (SSD)

$$\mathrm{SSD}(\mathbf{p}) = \sum_{i=1}^{N} ||\mathcal{I}\{\mathcal{W}(\mathbf{x}_i;\mathbf{p})\} - \mathcal{T}(\mathbf{x}_i)||_2^2$$

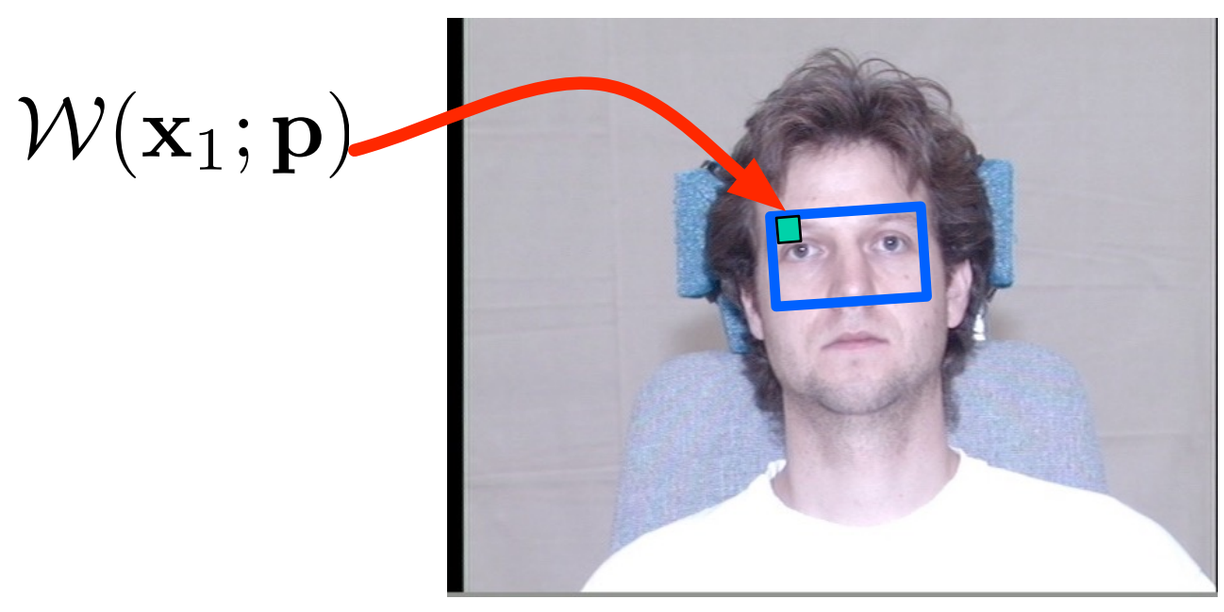


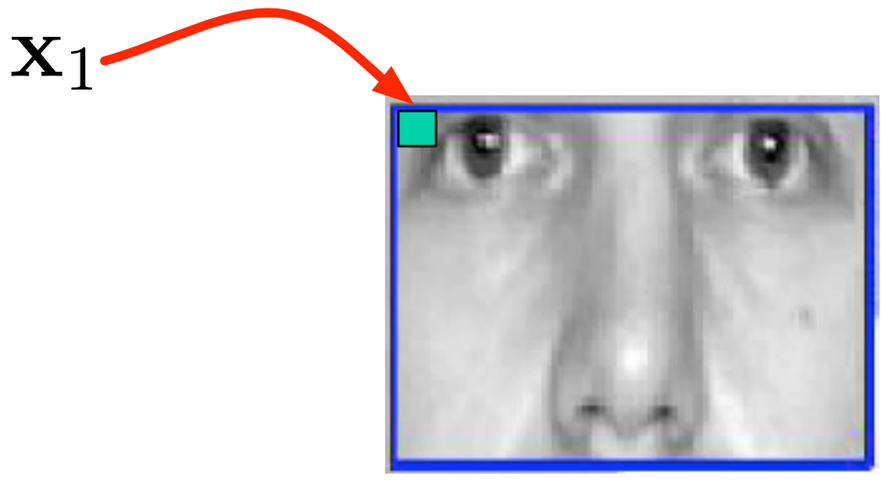


Measures of Image Similarity

- Although not always perfect, a common measure of image similarity is:
  - Sum of squared differences (SSD)

$$\mathrm{SSD}(\mathbf{p}) = \sum_{i=1}^{N} ||\mathcal{I}\{\mathcal{W}(\mathbf{x}_i;\mathbf{p})\} - \mathcal{T}(\mathbf{x}_i)||_2^2$$





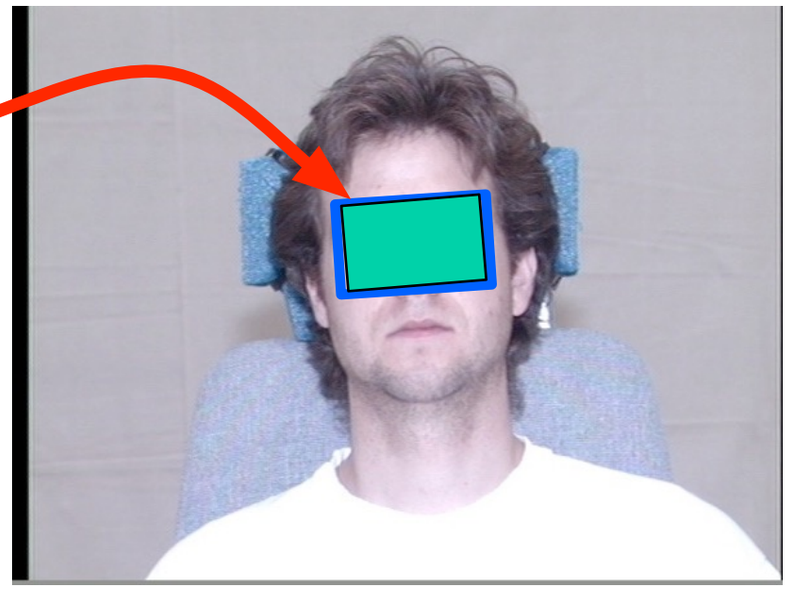
Measures of Image Similarity

- Although not always perfect, a common measure of image similarity is:
  - Sum of squared differences (SSD)

$$\mathrm{SSD}(\mathbf{p}) = ||\mathcal{I}(\mathbf{p}) - \mathcal{T}(\mathbf{0})||_2^2$$

"Vector Form"

$$\begin{bmatrix} \mathcal{W}(\mathbf{x}_1;\mathbf{p}) \\ \vdots \\ \mathcal{W}(\mathbf{x}_N;\mathbf{p}) \end{bmatrix}$$



$\mathcal{I}$

"Source Image"

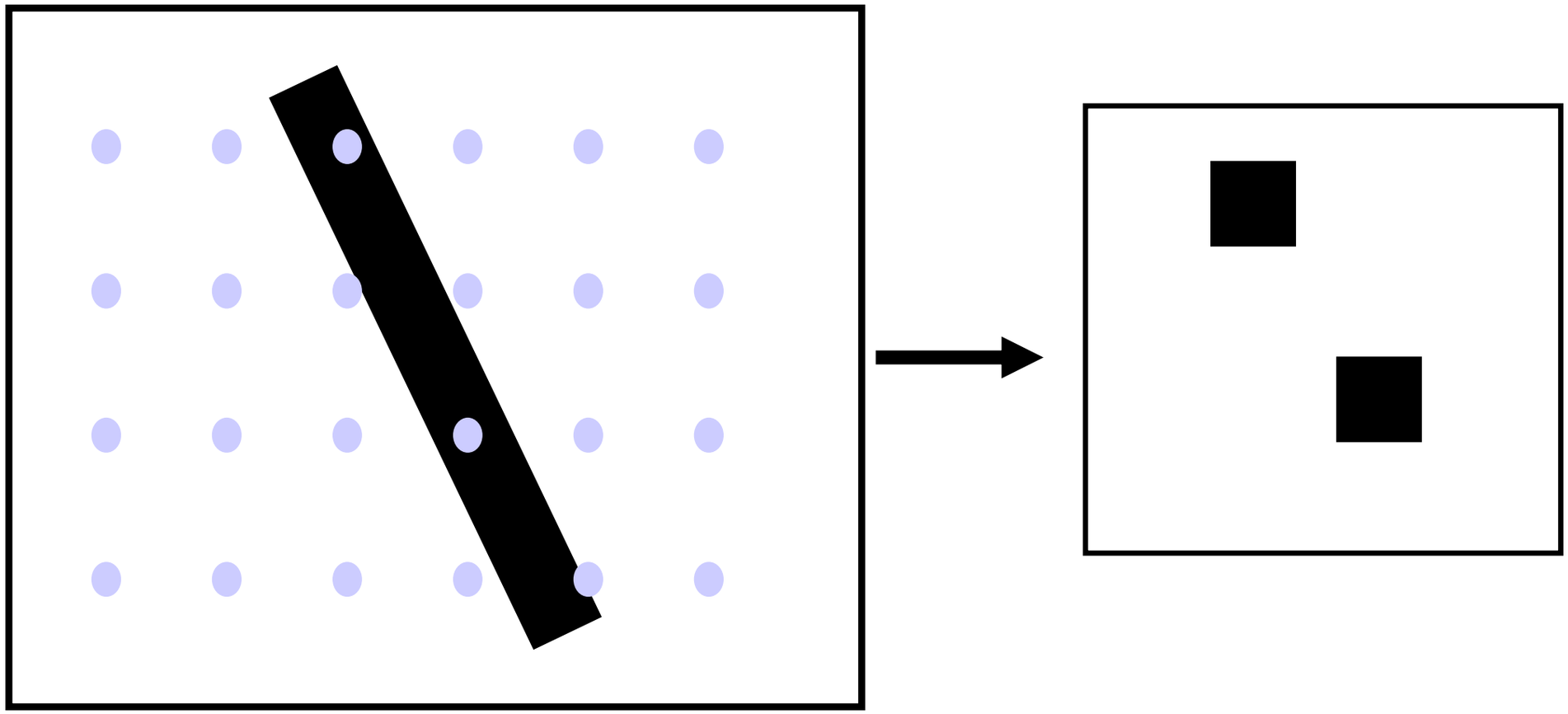
$$\begin{bmatrix} \mathcal{W}(\mathbf{x}_1;\mathbf{0}) \\ \vdots \\ \mathcal{W}(\mathbf{x}_N;\mathbf{0}) \end{bmatrix}$$

$\mathcal{T}$

"Template"

Fractional Coordinates

- The warp function gives nearly always fractional output.
- But we can only deal in integer pixels.
- What happens if we need to subsample an image?

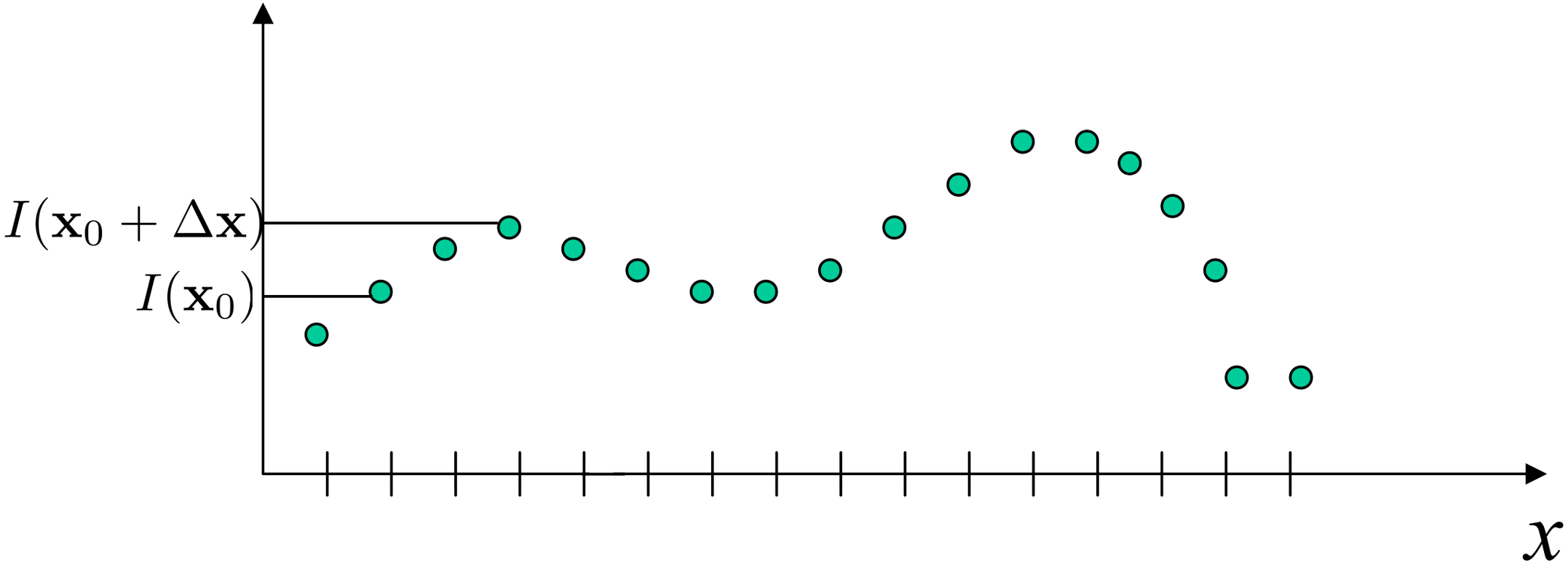


(Black)

Image Interpolation

- Simply take the Taylor series approximation.

$$I(\mathbf{x}_0 + \Delta\mathbf{x}) \approx I(\mathbf{x}_0) + \frac{\partial I(\mathbf{x}_0)}{\partial \mathbf{x}}^T \Delta\mathbf{x}$$

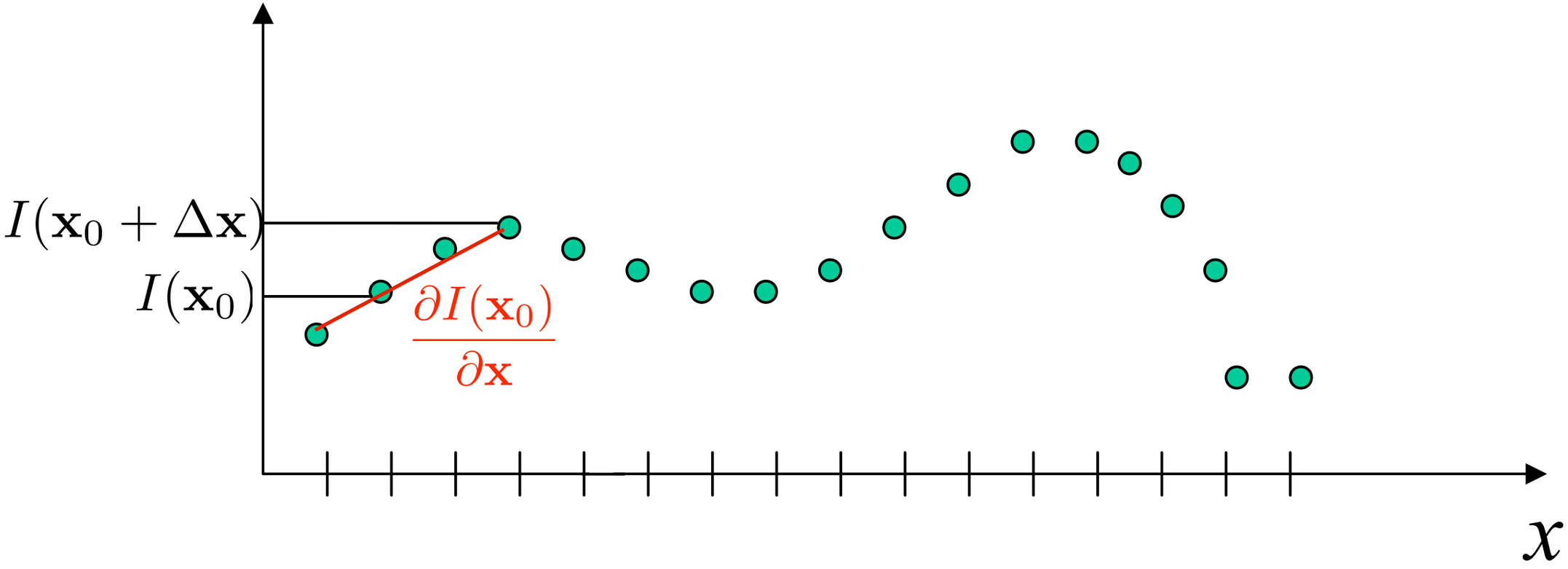


(Black)

Image Interpolation

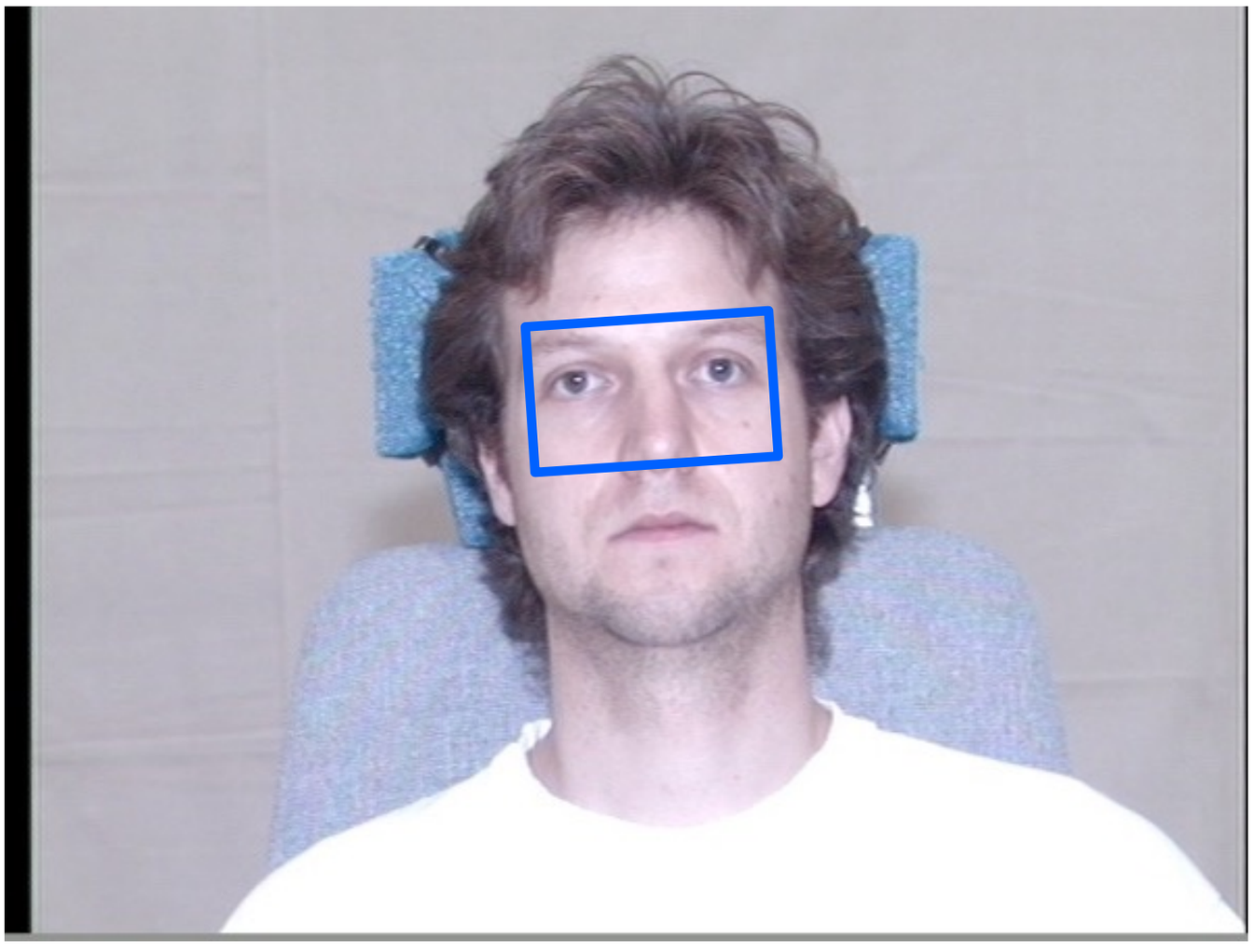
- Simply take the Taylor series approximation.

$$I(\mathbf{x}_0 + \Delta\mathbf{x}) \approx I(\mathbf{x}_0) + \frac{\partial I(\mathbf{x}_0)}{\partial \mathbf{x}}^T \Delta\mathbf{x}$$



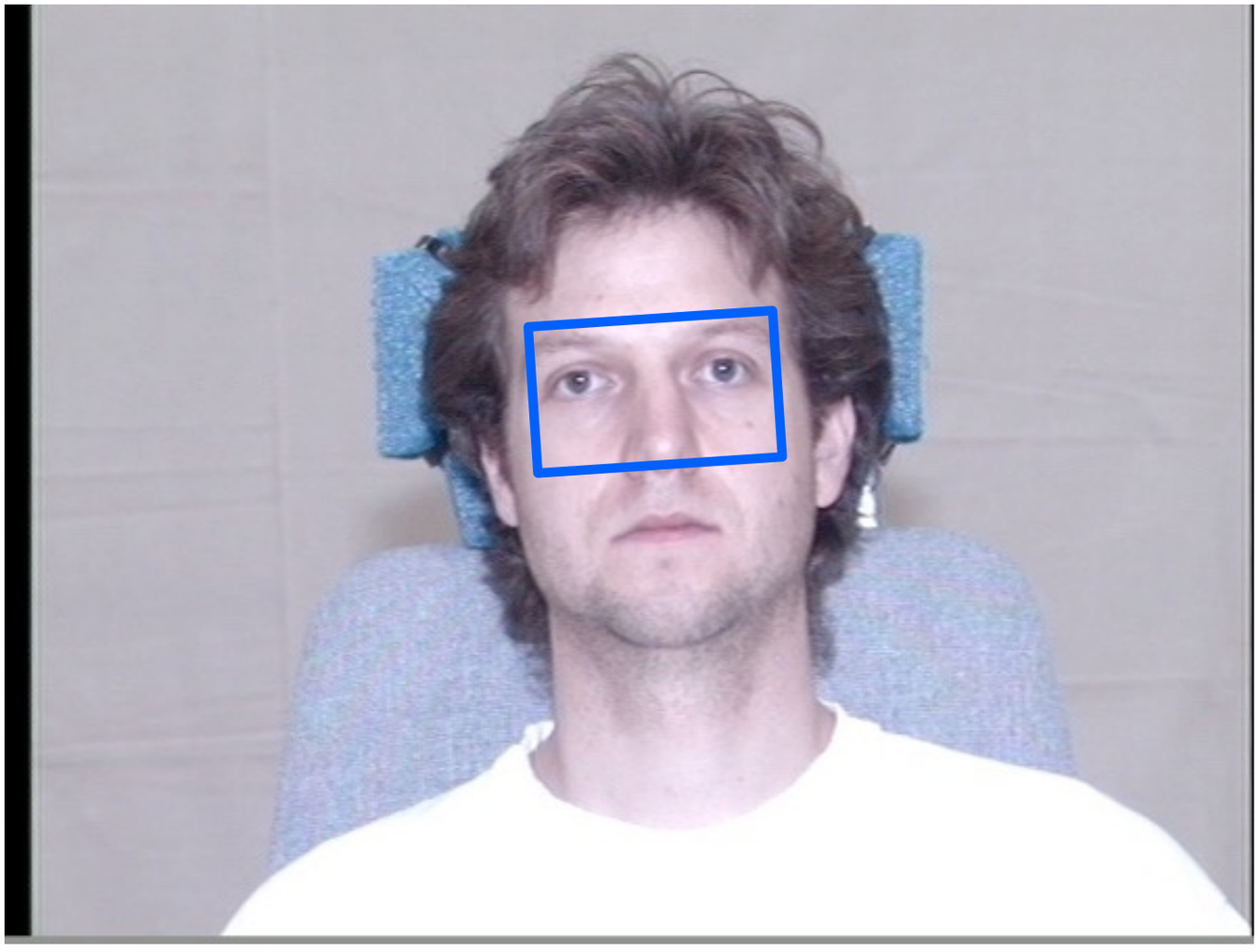
(Black)

Exhaustive Search



“Images at various warps”

Exhaustive Search



“Images at various warps”

[255,134,45,.....,34,12,124,67]

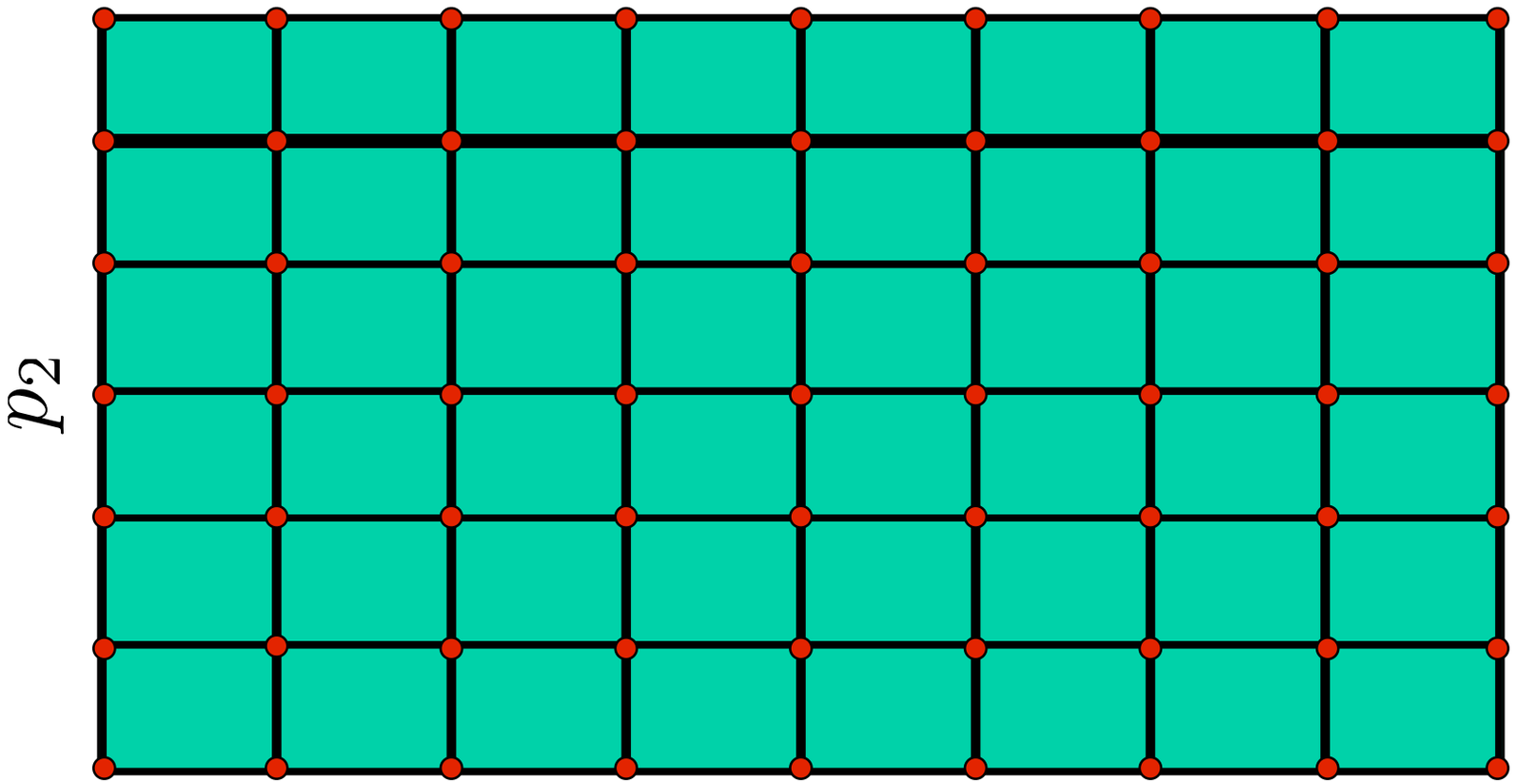
[123,244,12,.....,134,122,24,02]

[67,13,245,.....,112,51,92,181]

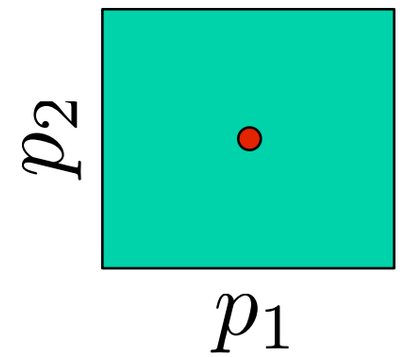
[65,09,67,.....,78,66,76,215]

“Vectors of pixel values at each warp position”

Exhaustive Search



"Possible Source Warps"



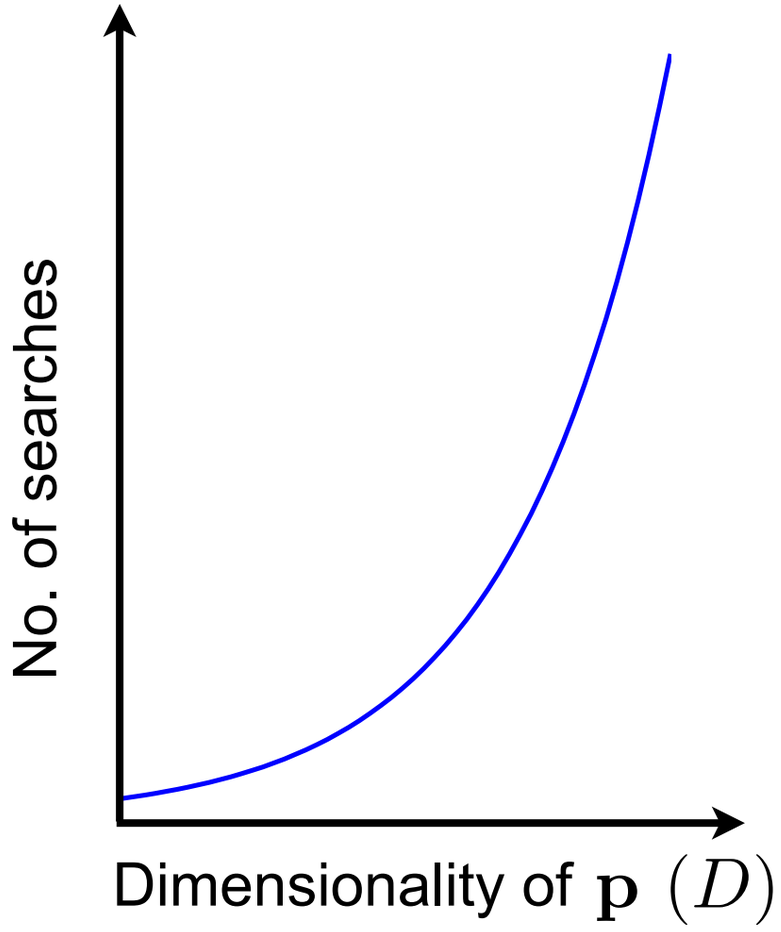
"Possible Template Warps"

$$\mathbf{p} = \{p_1, p_2\}$$

Exhaustive Search

- One can see that as the dimensionality D of $\mathbf{p}$ increases, and, assuming the same number of discrete samples n per dimension, the number of searches becomes,

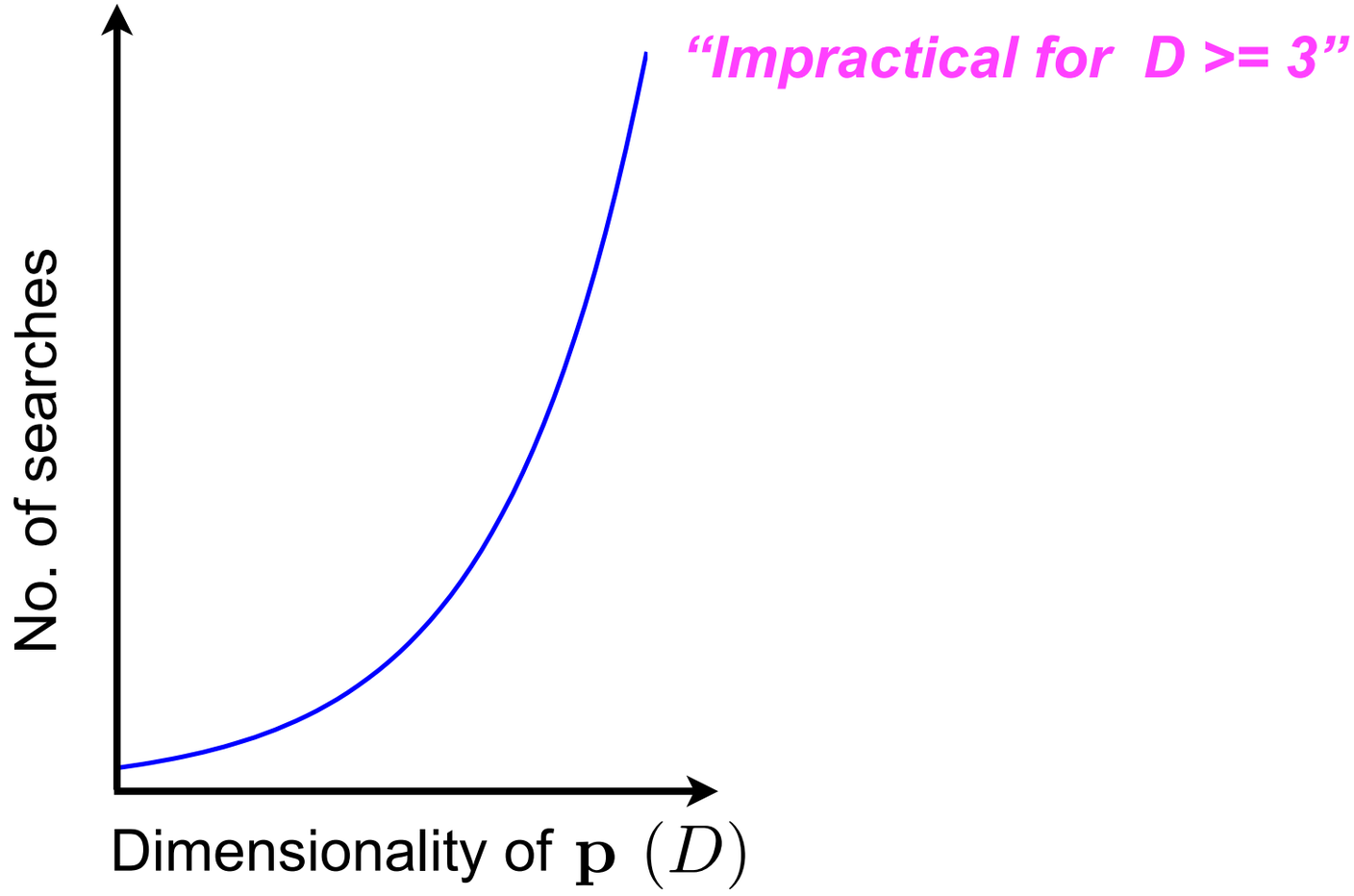
$$\text{number of searches} \rightarrow O(n^D)$$



Exhaustive Search

- One can see that as the dimensionality D of $\mathbf{p}$ increases, and, assuming the same number of discrete samples n per dimension, the number of searches becomes,

$$\text{number of searches} \rightarrow O(n^D)$$

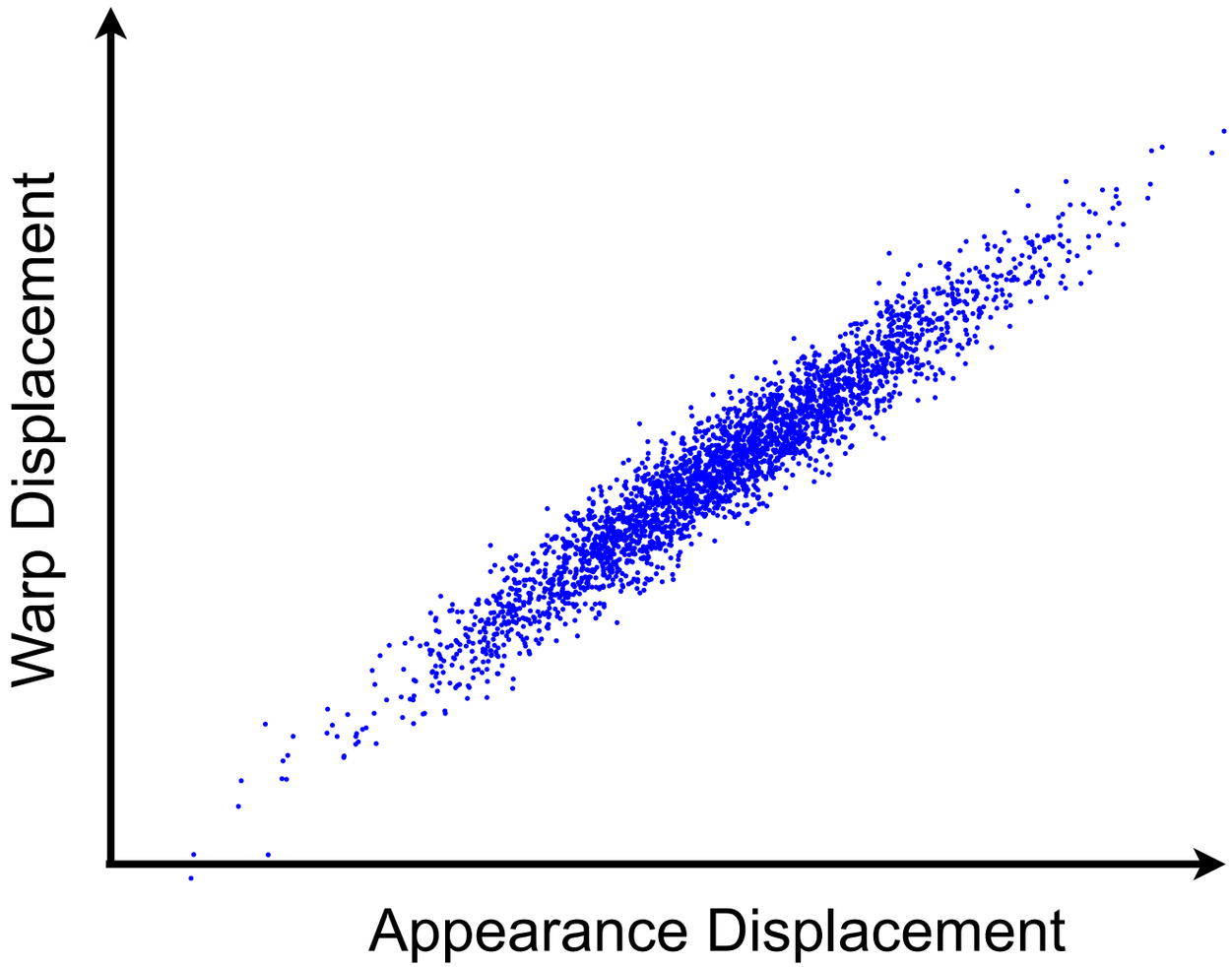


Today

- Registration, Registration, Registration.
- Linearizing Registration.
- Lucas & Kanade Algorithm.

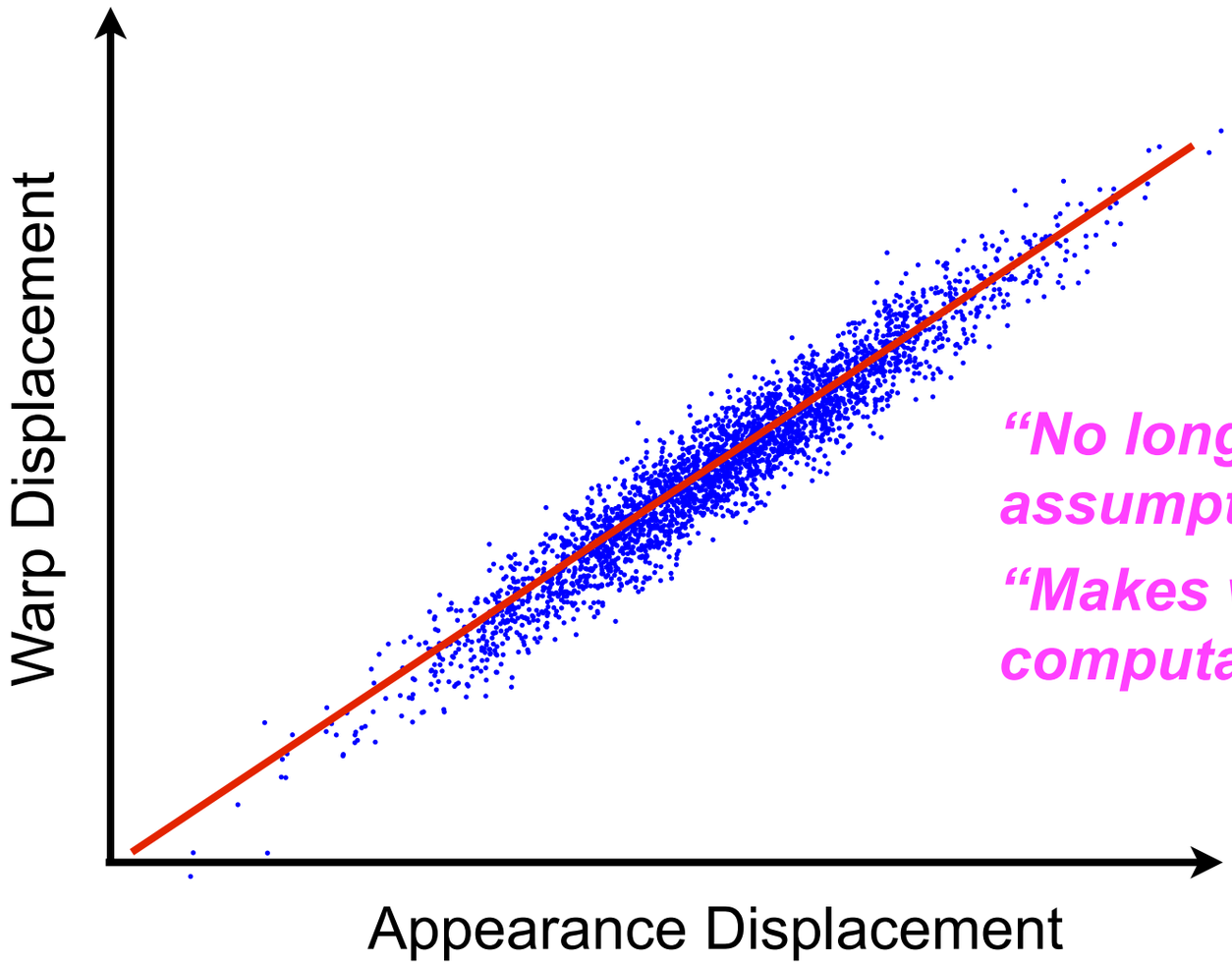
Slightly Less Naive Approach

- Instead of sampling through all possible warp positions to find the best match, let us instead learn a regression,



Slightly Less Naive Approach

- Instead of sampling through all possible warp positions to find the best match, let us instead learn a regression,



"No longer have to make discrete assumptions about warps."

"Makes warp estimation computationally feasible."

Problems?

- For us to learn this regression effectively we need to make a couple of assumptions.
  - What is the distribution $\Delta \mathbf{p}$ of warp displacements?
  - Is there a relationship between appearance displacement and warp displacement?
  - When does this relationship occur, when does it fail?

$\mathcal{T}(\mathbf{0})$

$\mathcal{T}(\Delta \mathbf{p})$

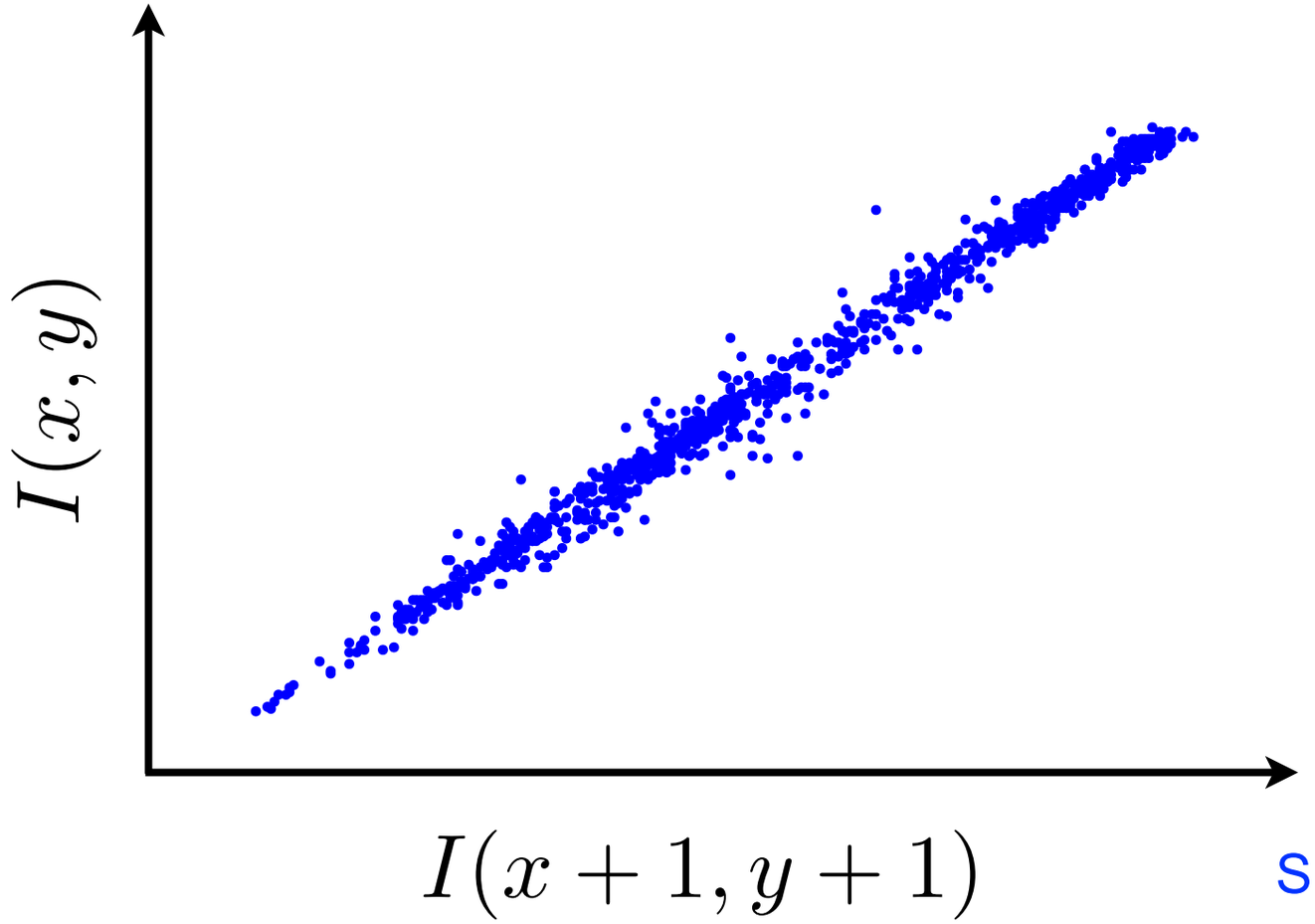
Problems?

- For us to learn this regression effectively we need to make a couple of assumptions.
  - What is the distribution $\Delta \mathbf{p}$ of warp displacements?
  - Is there a relationship between appearance displacement and warp displacement?
  - When does this relationship occur, when does it fail?

$\mathcal{T}(\mathbf{0})$

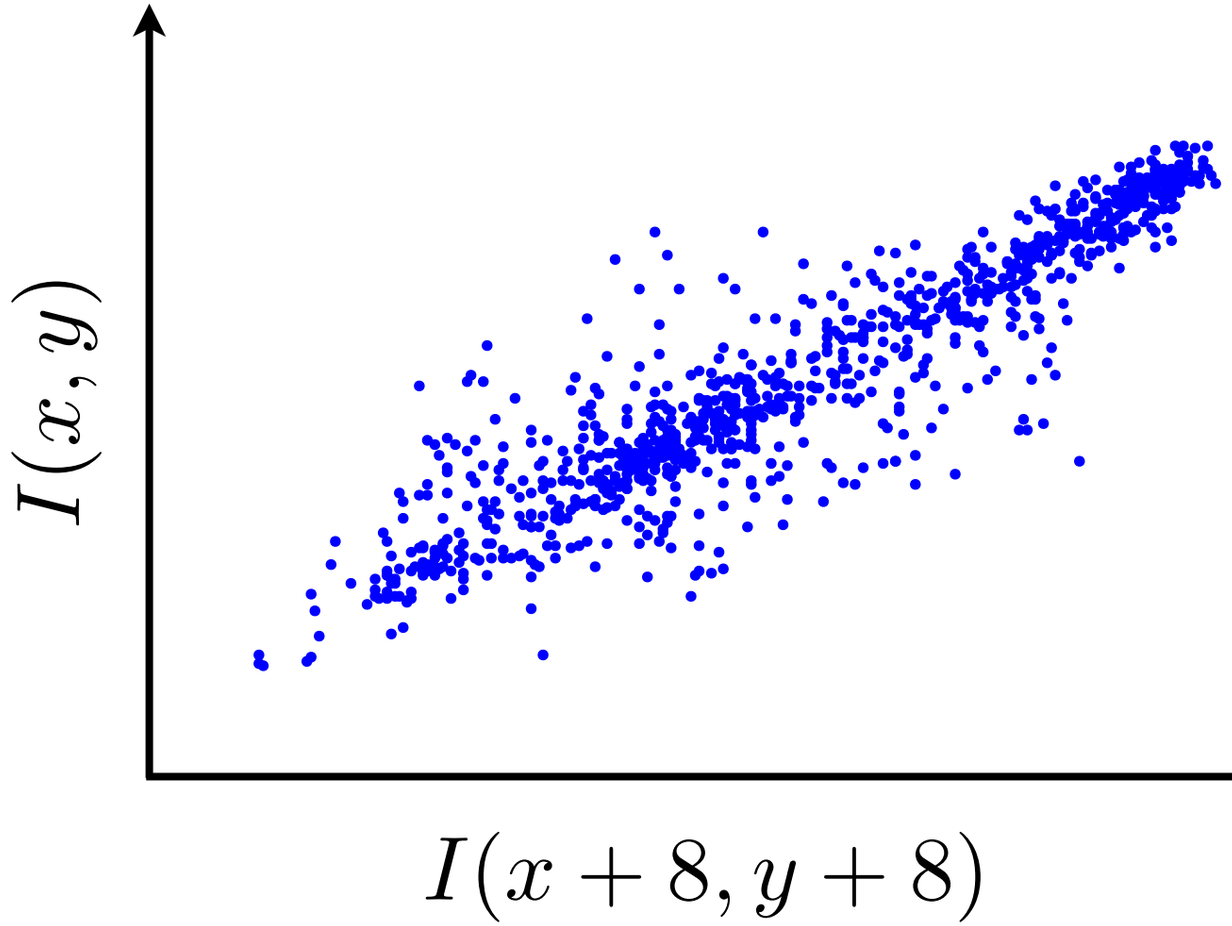
$\mathcal{T}(\Delta \mathbf{p})$

I



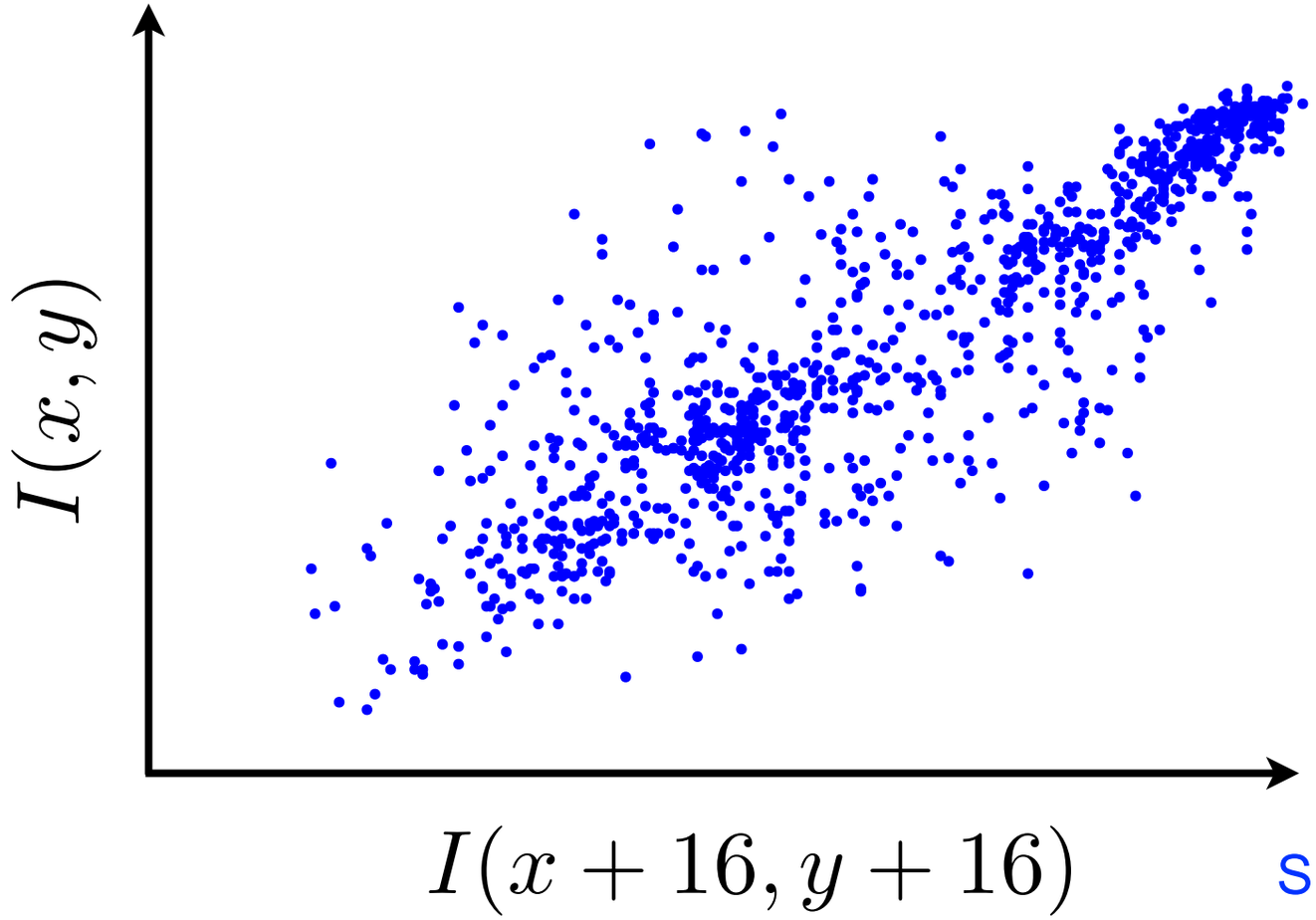
Simoncelli & Olshausen 2001

I



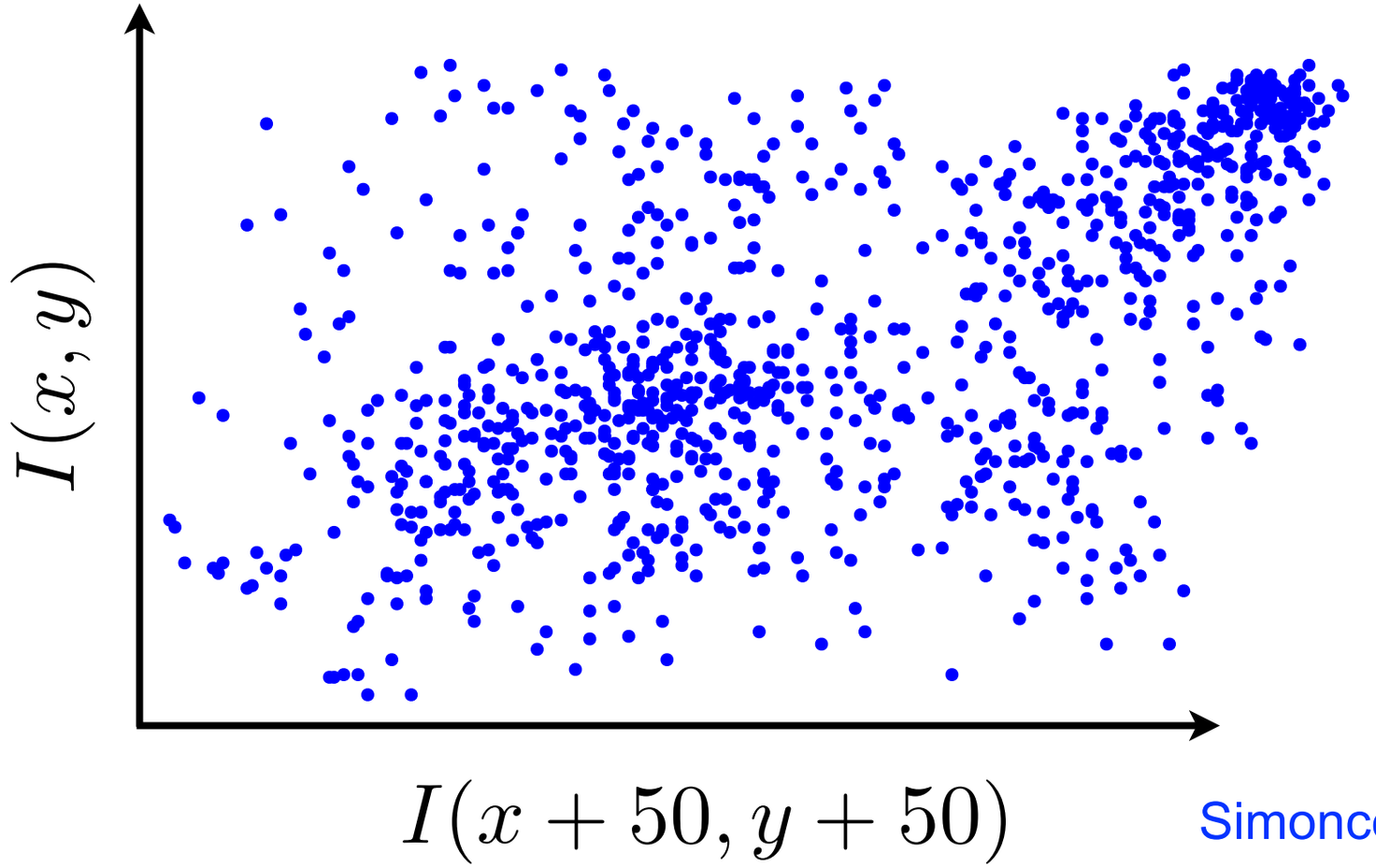
Simoncelli & Olshausen 2001

I



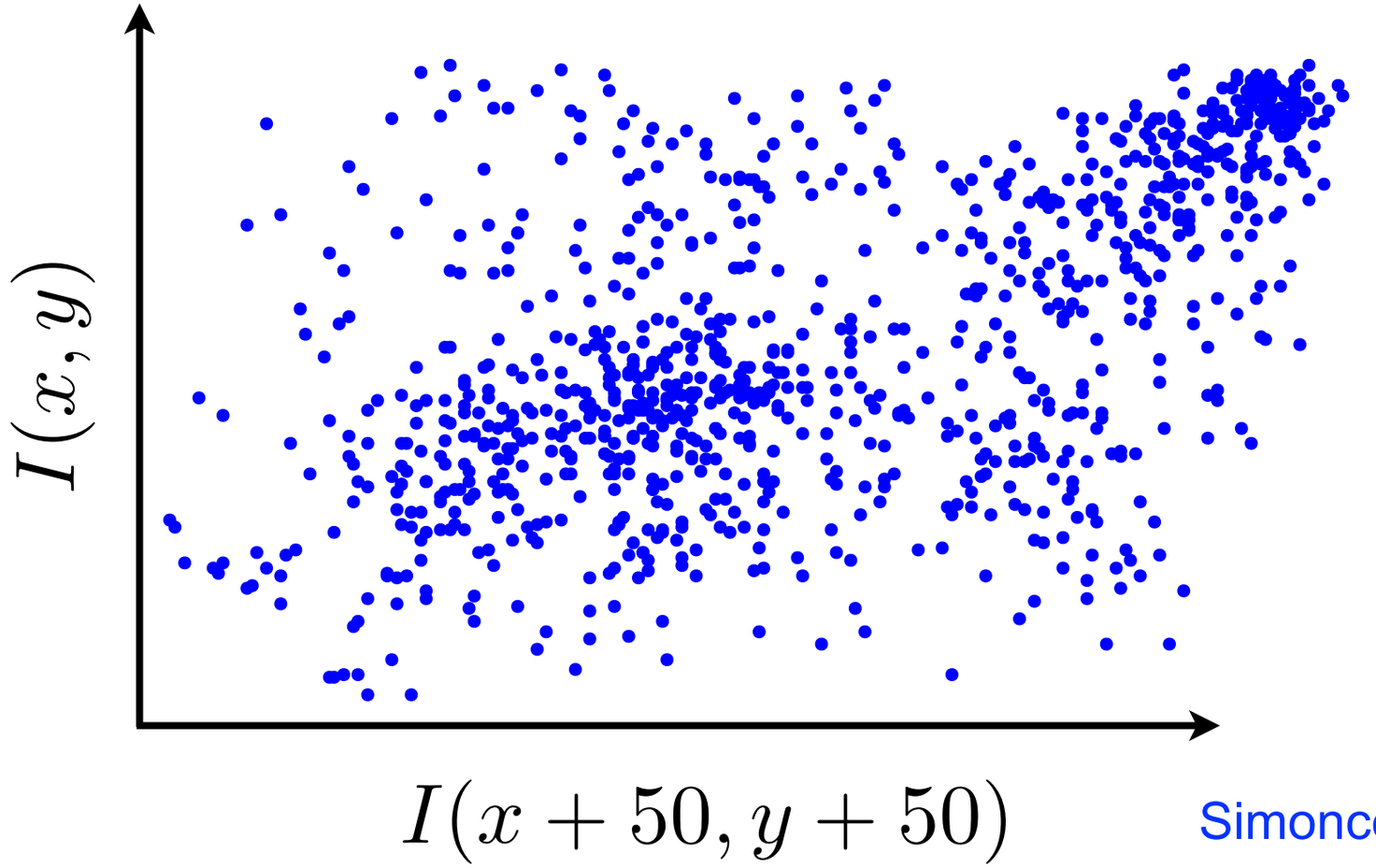
Simoncelli & Olshausen 2001

I



Simoncelli & Olshausen 2001

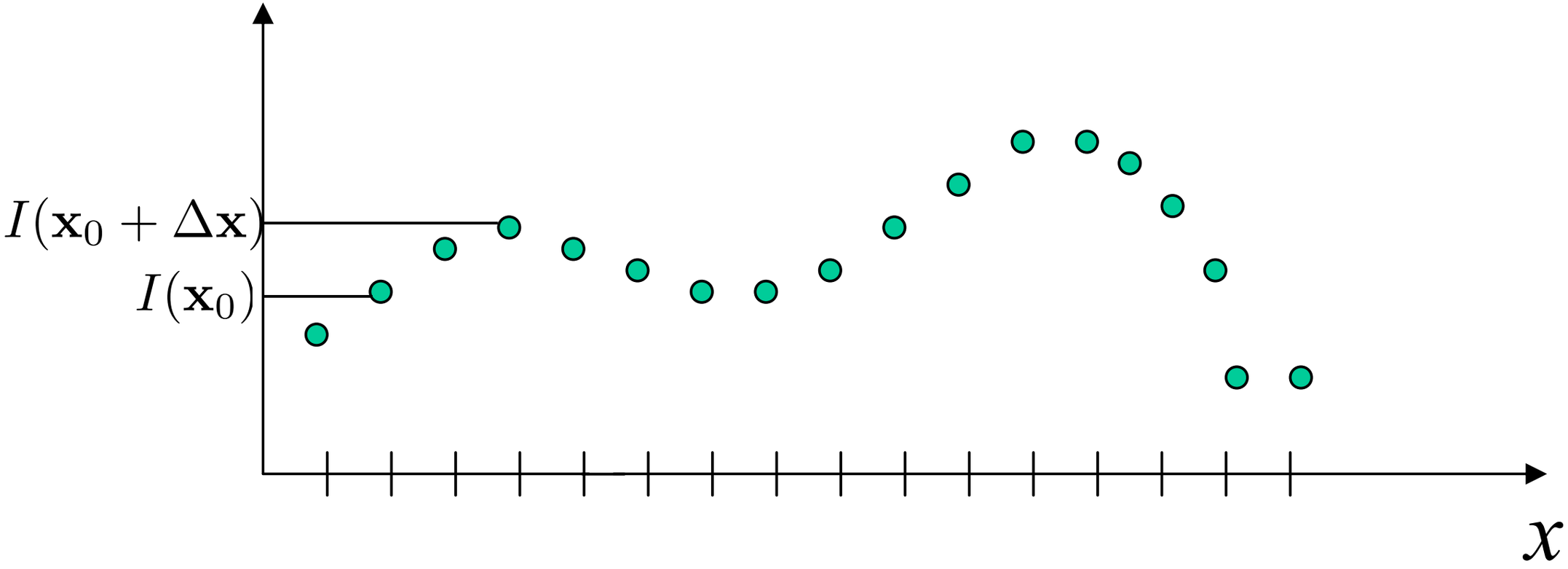
I



Simoncelli & Olshausen 2001

Linearizing Registration

- What if I want to know $\Delta \mathbf{x}$ given that I have only the appearance at $I(\mathbf{x}_0)$ and $I(\mathbf{x}_0 + \Delta \mathbf{x})$?

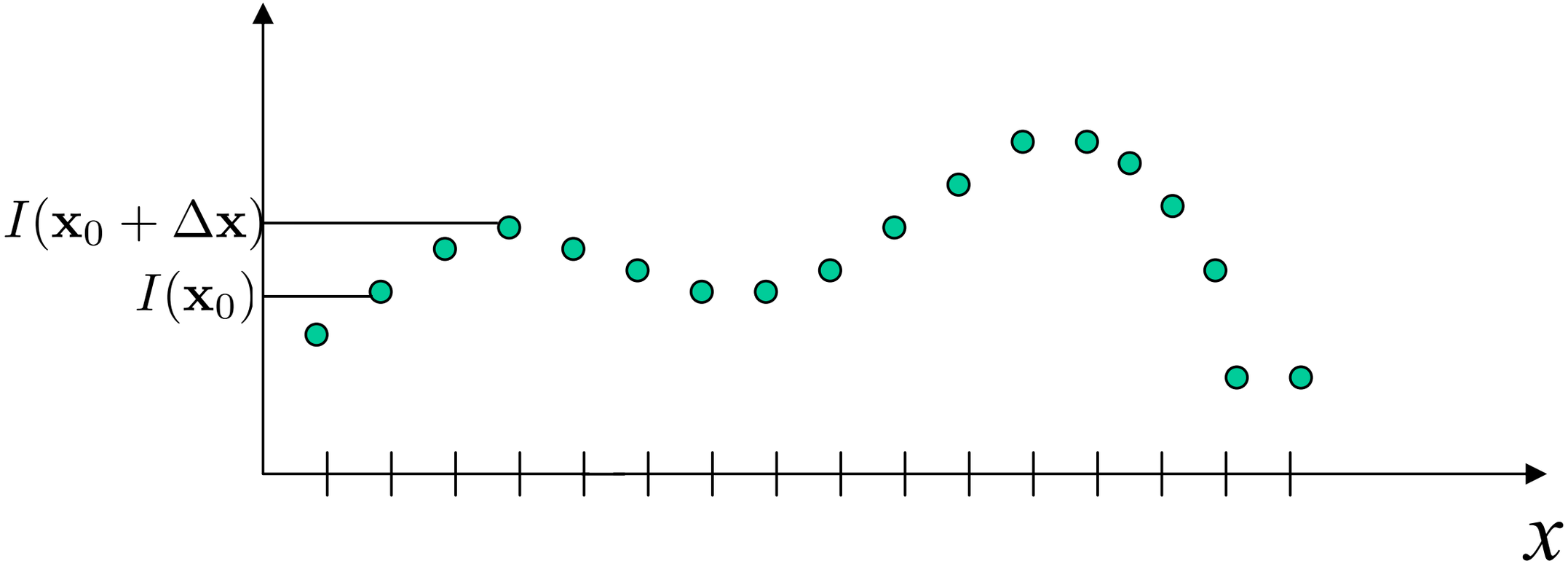


(Black)

Linearizing Registration

- Again, simply take the Taylor series approximation?

$$\mathcal{I}(\mathbf{x}_0 + \Delta\mathbf{x}) \approx \mathcal{I}(\mathbf{x}_0) + \frac{\partial \mathcal{I}(\mathbf{x}_0)}{\partial \mathbf{x}^T}\Delta\mathbf{x}$$

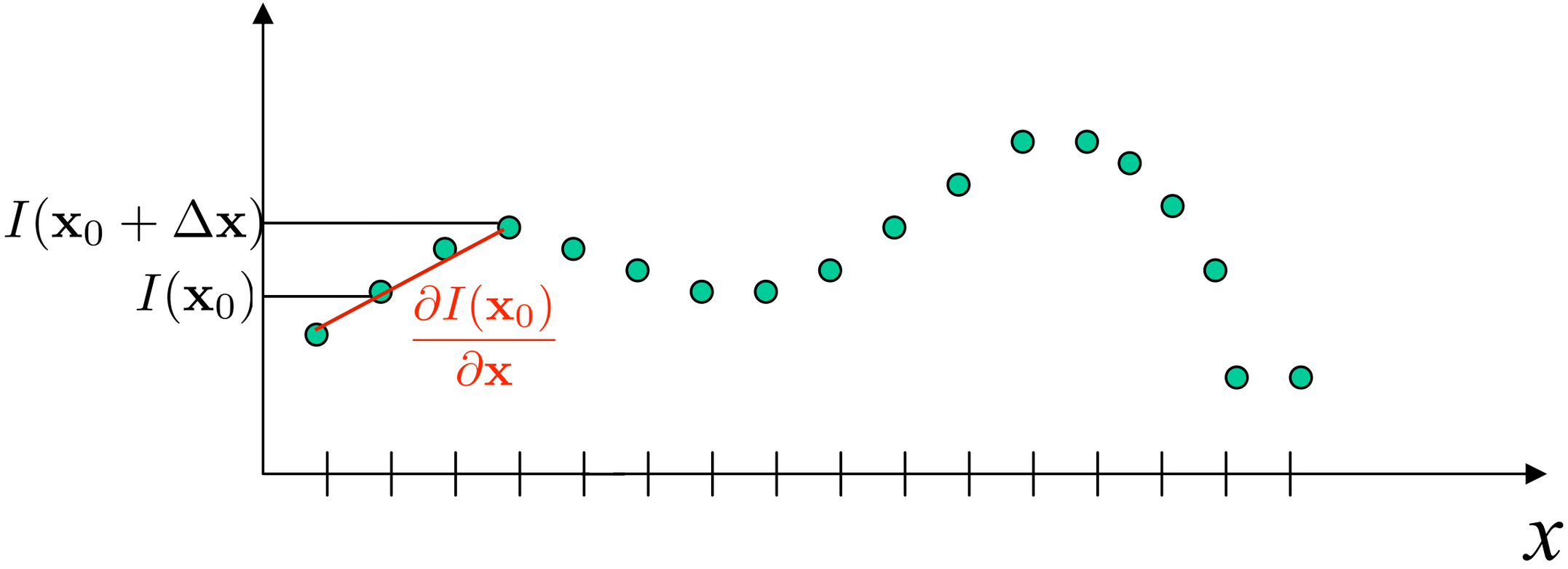


(Black)

Linearizing Registration

- Again, simply take the Taylor series approximation?

$$\mathcal{I}(\mathbf{x}_0 + \Delta\mathbf{x}) \approx \mathcal{I}(\mathbf{x}_0) + \frac{\partial \mathcal{I}(\mathbf{x}_0)}{\partial \mathbf{x}^T}\Delta\mathbf{x}$$



(Black)

Linearizing Registration

- Again, simply take the Taylor series approximation,

$$\mathcal{I}(\mathbf{x}_0 + \Delta\mathbf{x}) \approx \mathcal{I}(\mathbf{x}_0) + \frac{\partial \mathcal{I}(\mathbf{x}_0)}{\partial \mathbf{x}^T}\Delta\mathbf{x}$$

- Therefore,

$$\Delta\mathbf{x} \approx \left[\frac{\partial \mathcal{I}(\mathbf{x}_0)}{\partial \mathbf{x}}\frac{\partial \mathcal{I}(\mathbf{x}_0)}{\partial \mathbf{x}^T}\right]^{-1}\frac{\partial \mathcal{I}(\mathbf{x}_0)}{\partial \mathbf{x}}[\mathcal{I}(\mathbf{x}_0 + \Delta\mathbf{x}) - \mathcal{I}(\mathbf{x}_0)]$$

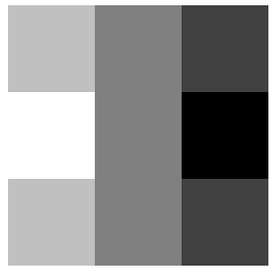
- We refer to $\frac{\partial \mathcal{I}(\mathbf{x}_0)}{\partial \mathbf{x}}$ as the gradient of the image at $\mathbf{x}_0$.

Gradients through Filters

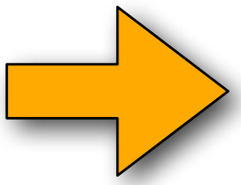
- Traditional method for calculating gradients in vision is through the use of edge filters. (e.g., Sobel, Prewitt).

I

$*$

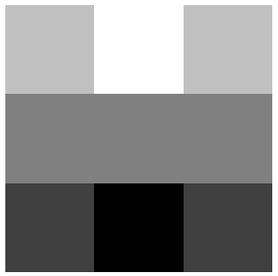


"Horizontal"

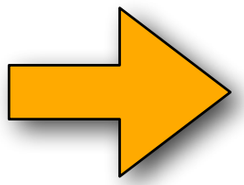


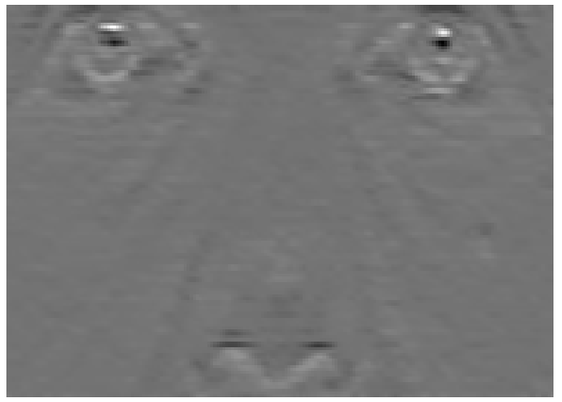
∇I_x

$*$



"Vertical"





∇I_y

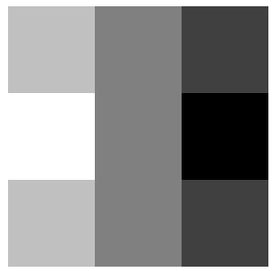
- where, $\dfrac{\partial I(\mathbf{x})}{\partial \mathbf{x}} = [\nabla I_x(\mathbf{x}), \nabla I_y(\mathbf{x})]^T$

Gradients through Filters

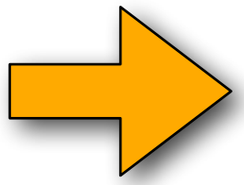
- Traditional method for calculating gradients in vision is through the use of edge filters. (e.g., Sobel, Prewitt).

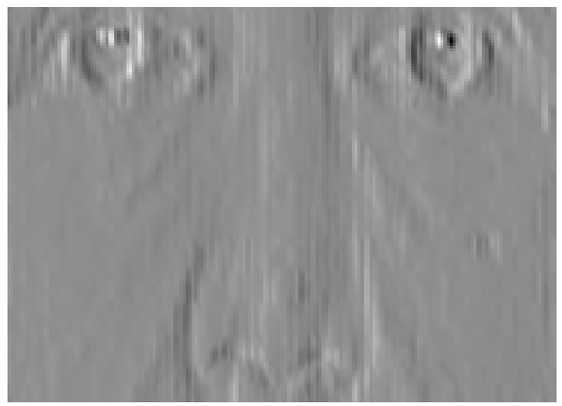
I

$*$



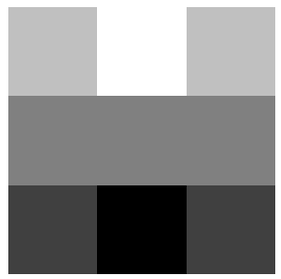
"Horizontal"



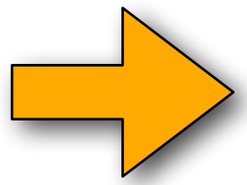


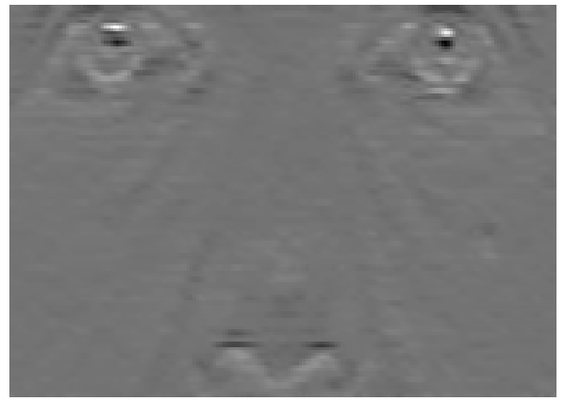
∇I_x

$*$



"Vertical"





∇I_y

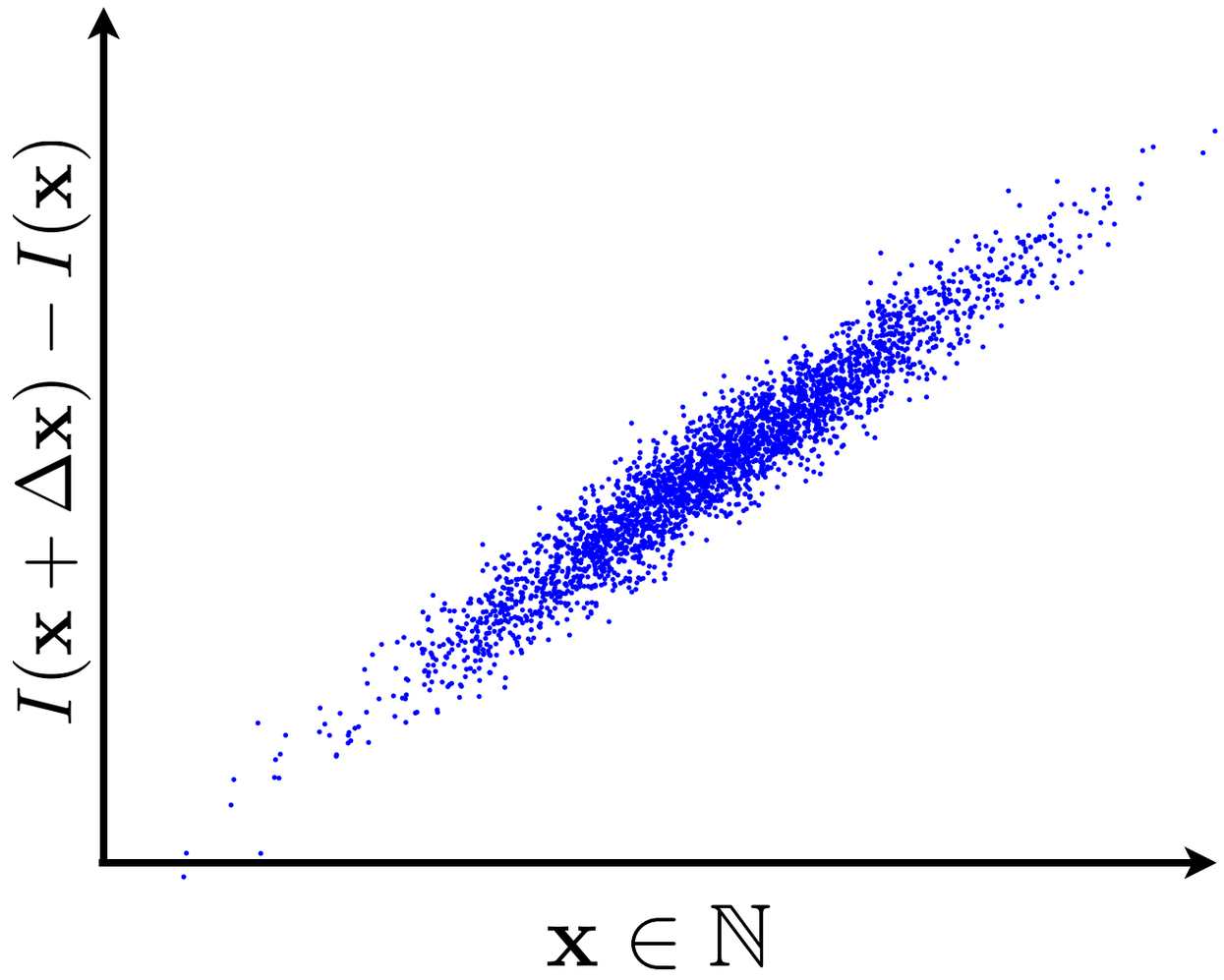
"Often have to apply a smoothing filter as well."

- where, $\dfrac{\partial I(\mathbf{x})}{\partial \mathbf{x}} = [\nabla I_x(\mathbf{x}), \nabla I_y(\mathbf{x})]^T$

Gradients through Regression

- Another strategy is to learn a least-squares regression for every pixel in the source image such that,

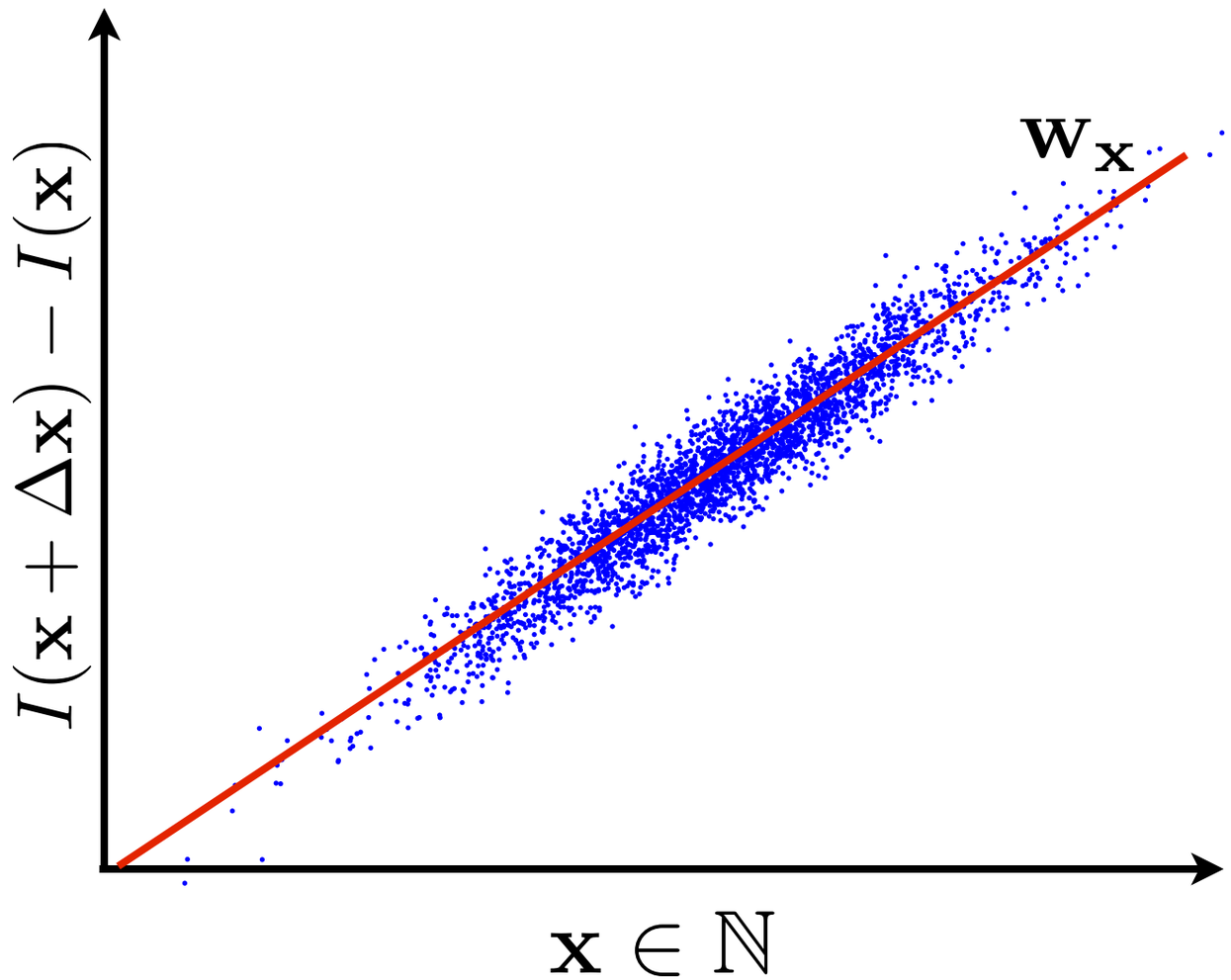
$$I(\mathbf{x} + \Delta\mathbf{x}) \approx I(\mathbf{x}) + \mathbf{w}_{\mathbf{x}}^{T} \Delta\mathbf{x} + b$$



Gradients through Regression

- Another strategy is to learn a least-squares regression for every pixel in the source image such that,

$$I(\mathbf{x} + \Delta\mathbf{x}) \approx I(\mathbf{x}) + \mathbf{w}_{\mathbf{x}}^{T} \Delta\mathbf{x} + b$$

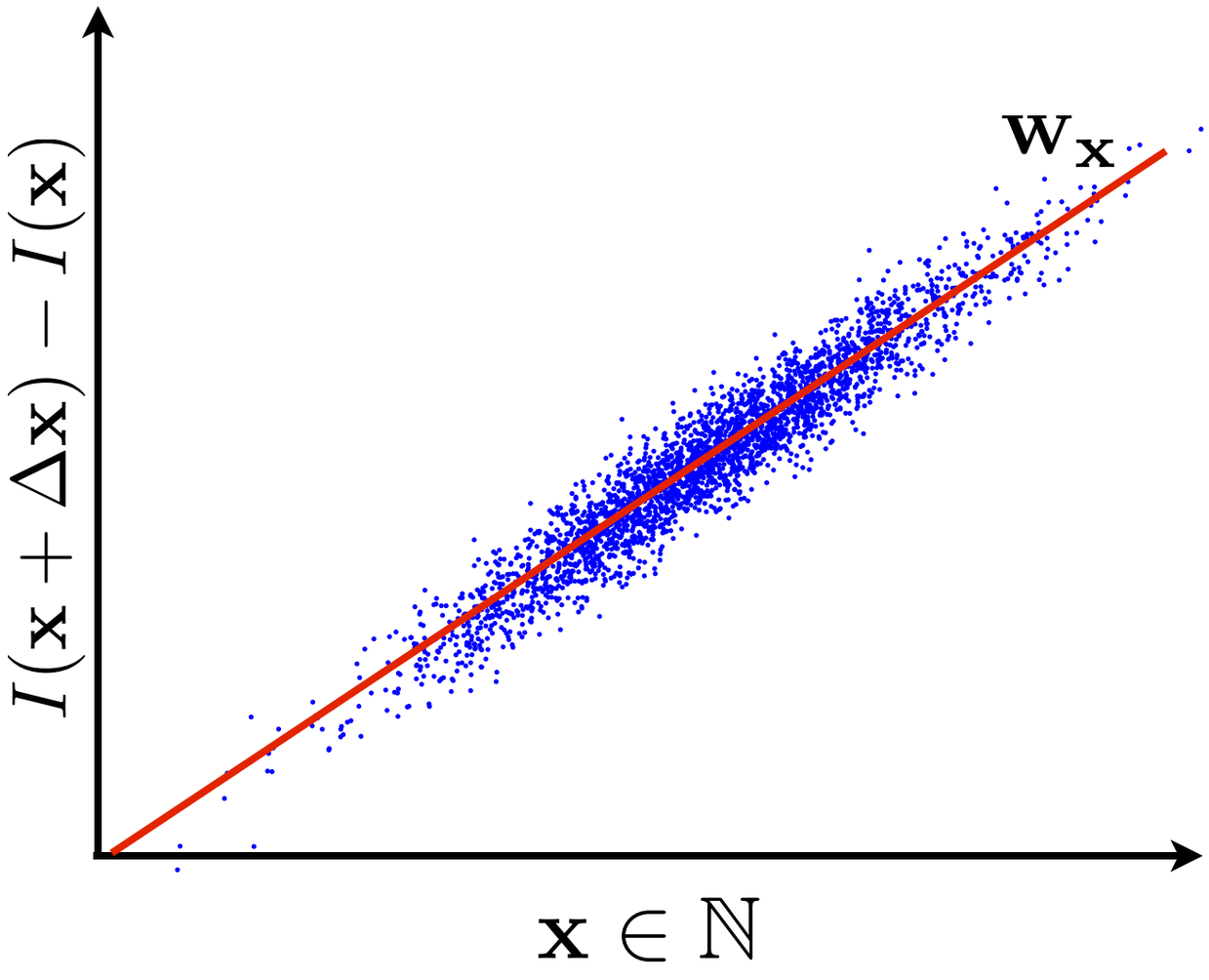


Gradients through Regression

- Another strategy is to learn a least-squares regression for every pixel in the source image such that,

$$I(\mathbf{x}+\Delta\mathbf{x}) \approx I(\mathbf{x}) + \mathbf{w}_{\mathbf{x}}^{T}\Delta\mathbf{x} + b$$

Why does b have to equal zero?



Gradients through Regression

- Can be written formally as,

$$\arg\min_{\mathbf{w_x}} \sum_{\Delta\mathbf{x} \in \mathbb{N}} ||I(\mathbf{x} + \Delta\mathbf{x}) - I(\mathbf{x}) - \mathbf{w}_\mathbf{x}^T \Delta\mathbf{x}||^2$$

- Solution is given as,

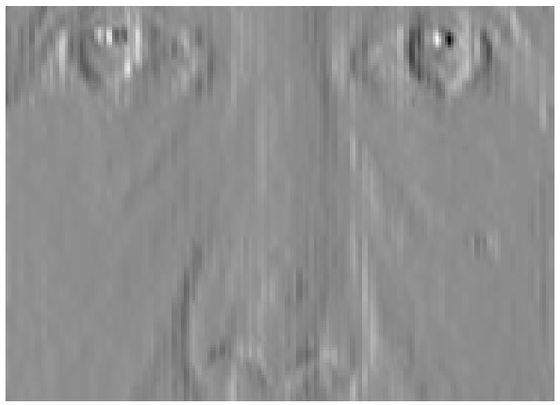
$$\mathbf{w_x} = \left(\sum_{\Delta\mathbf{x} \in \mathbb{N}} \Delta\mathbf{x}\Delta\mathbf{x}^T \right)^{-1} \left(\sum_{\Delta\mathbf{x} \in \mathbb{N}} \Delta\mathbf{x}[I(\mathbf{x} + \Delta\mathbf{x}) - I(\mathbf{x})] \right)$$

Gradients through Regression

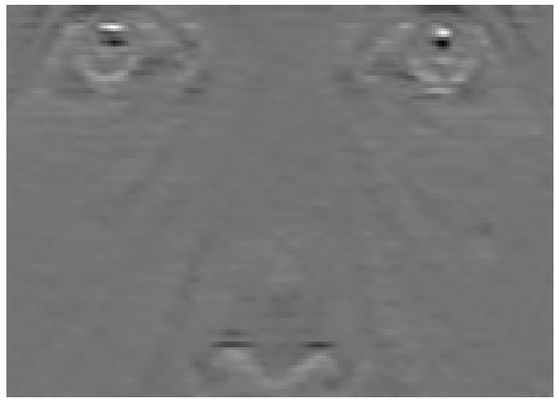
- Can solve for gradients using least-squares regression.
- Has nice properties: expand neighborhood, no heuristics.

I

Learn Regressor for each Pixel



∇I_x



∇I_y

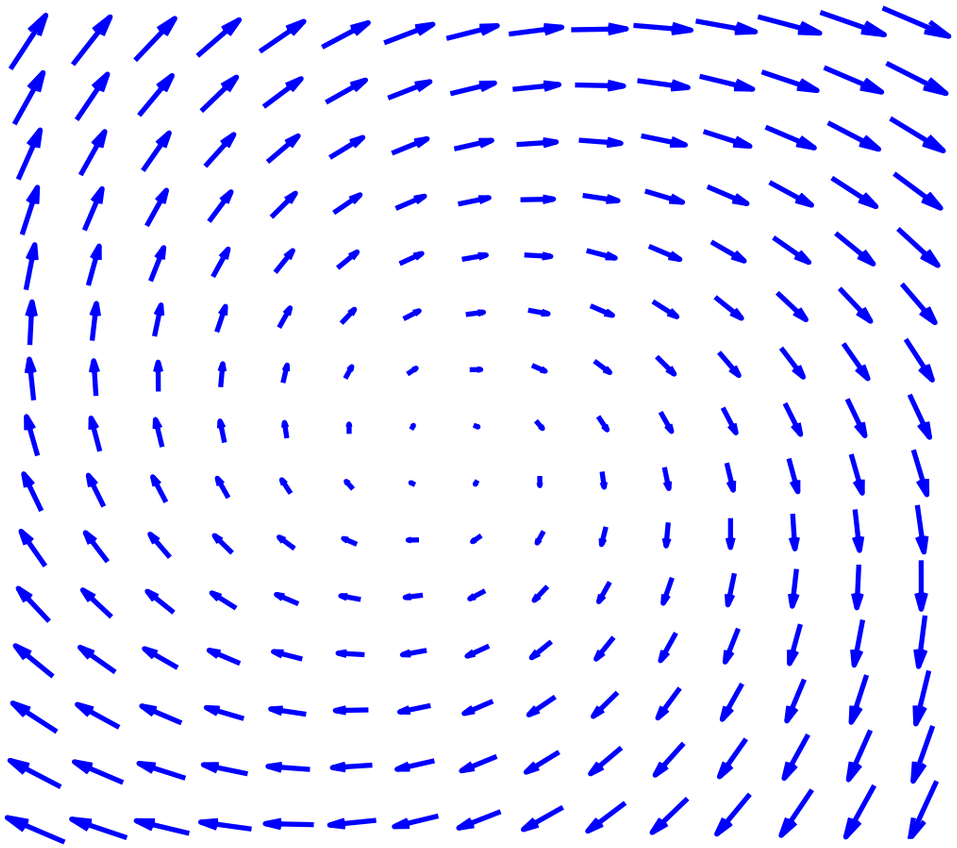
- where,

$$\frac{\partial I(\mathbf{x})}{\partial \mathbf{x}} = [\nabla I_x(\mathbf{x}), \nabla I_y(\mathbf{x})]^T = [w_x, w_y]^T$$

Possible Solution?

- Could we now just solve for individual pixel translation, and then estimate a complex warp from these motions?

$$\begin{bmatrix} \Delta\mathbf{x}_1 \\ \vdots \\ \Delta\mathbf{x}_N \end{bmatrix} \rightarrow$$

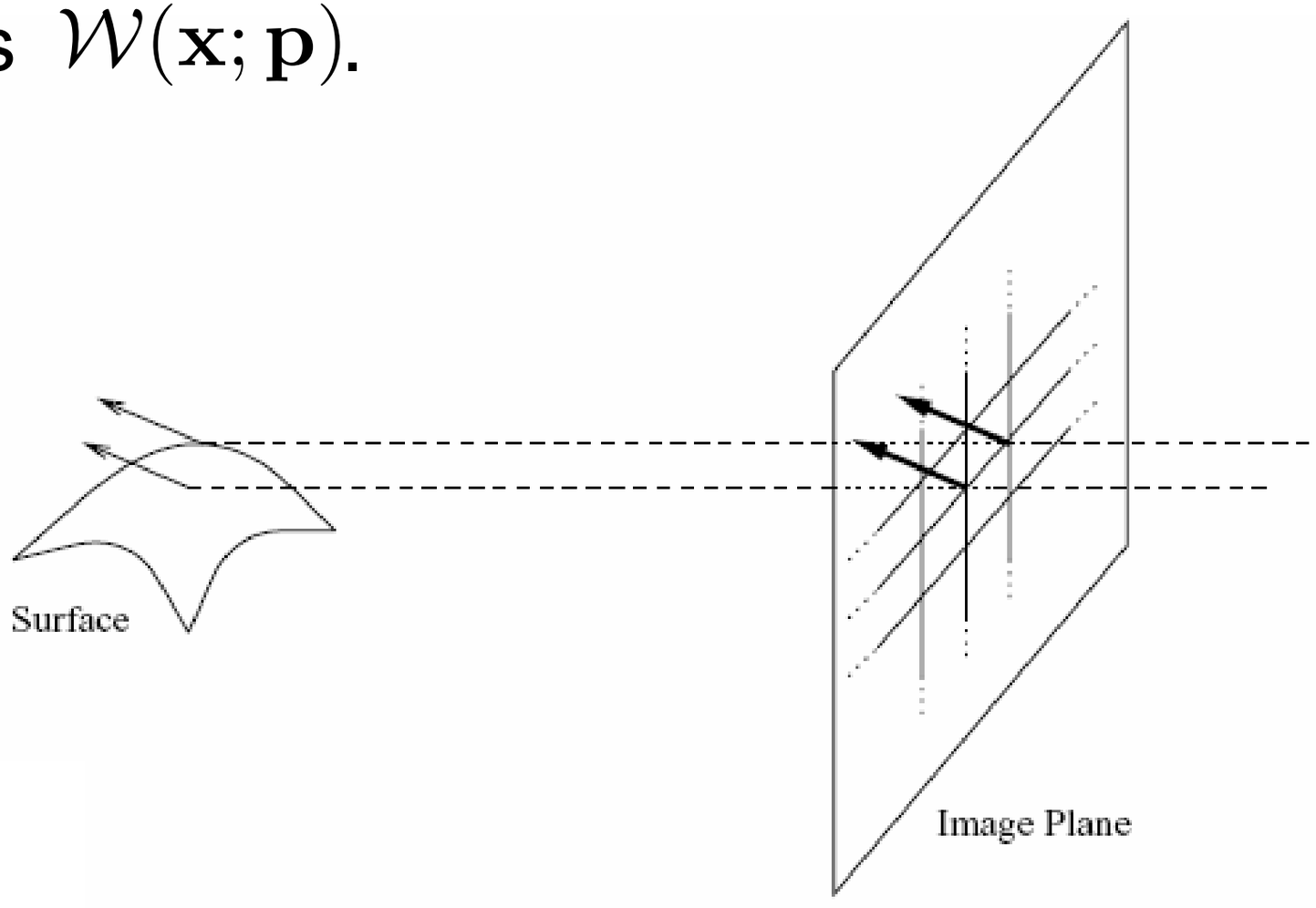


$$\rightarrow \Delta\mathbf{p}$$

“Motion Field for Rotation”

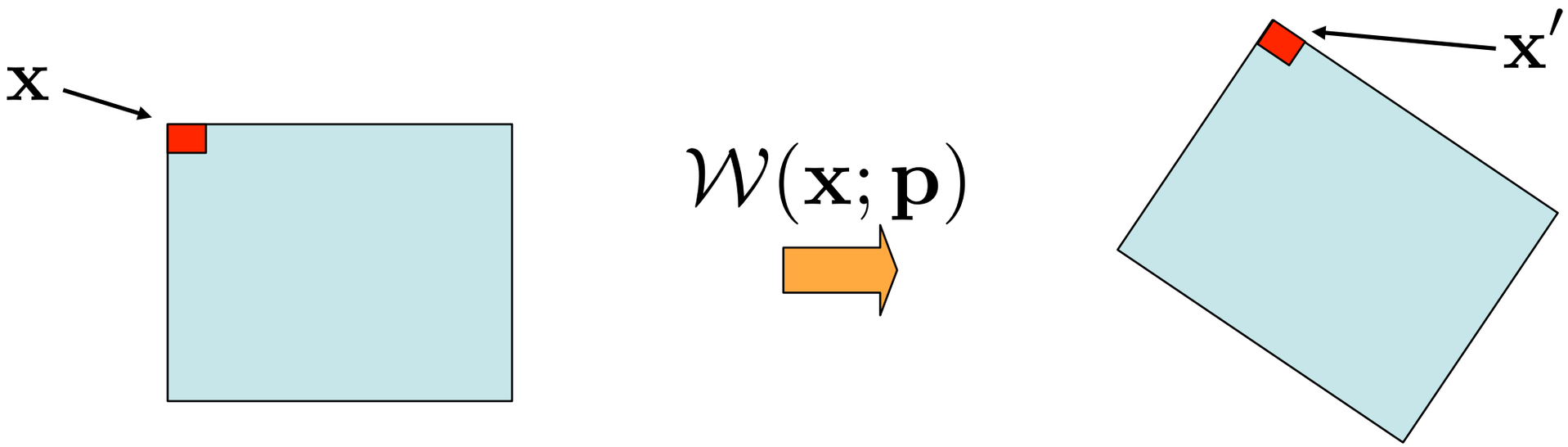
Spatial Coherence

- Problem is pixel noise. Individual pixels are too noisy to be reliable estimators of pixel movement (optical flow).
- Fortunately, neighboring points in the scene typically belong to the same surface and hence typically have similar motions $\mathcal{W}(\mathbf{x};\mathbf{p})$.



Reminder: Warp Functions

- To perform alignment we need a formal way of describing how the template relates to the source image.
- To do this we can employ what is known as a warp function:-



where,

$\mathbf{x}_i = i$-th 2D coordinate

$\mathbf{p} =$ parametric form of warp

Today

- Registration, Registration, Registration.
- Linearizing Registration.
- Lucas & Kanade Algorithm.

Lucas & Kanade Algorithm

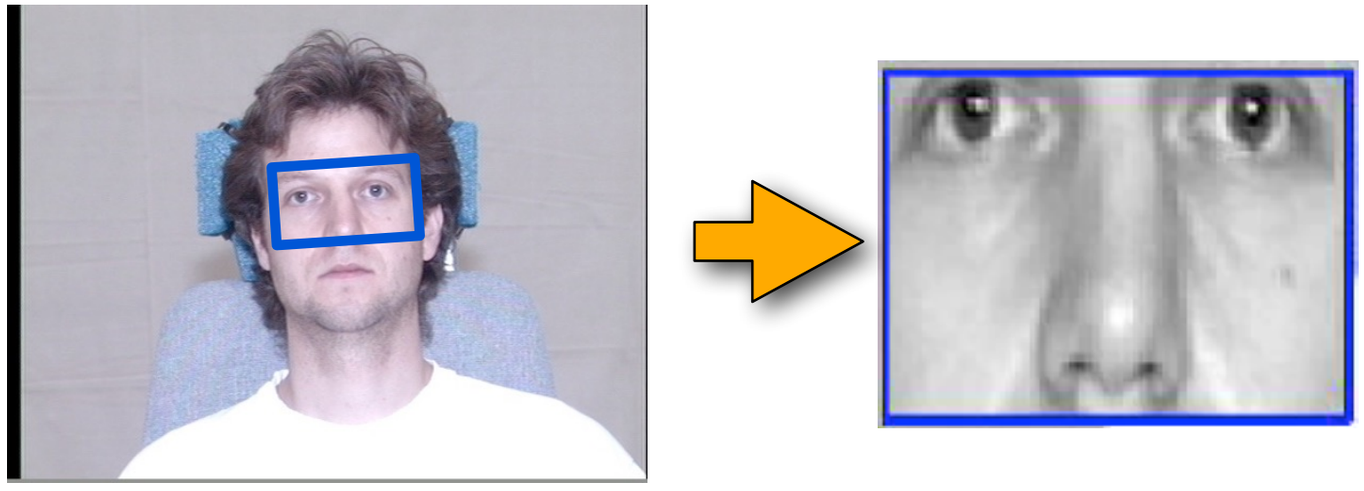
- Lucas & Kanade (1981) realized this and proposed a method for estimating warp displacement using the principles of ***gradients*** and ***spatial coherence***.
- Technique applies Taylor series approximation to any spatially coherent area governed by the warp $\mathcal{W}(\mathbf{x};\mathbf{p})$.

$$\mathcal{I}(\mathbf{p}+\Delta\mathbf{p}) \approx \mathcal{I}(\mathbf{p}) + \frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}^T}\Delta\mathbf{p}$$

Lucas & Kanade Algorithm

- Lucas & Kanade (1981) realized this and proposed a method for estimating warp displacement using the principles of ***gradients*** and ***spatial coherence***.
- Technique applies Taylor series approximation to any spatially coherent area governed by the warp $\mathcal{W}(\mathbf{x};\mathbf{p})$.

$$\mathcal{I}(\mathbf{p}+\Delta\mathbf{p}) \approx \mathcal{I}(\mathbf{p}) + \frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}^T}\Delta\mathbf{p}$$

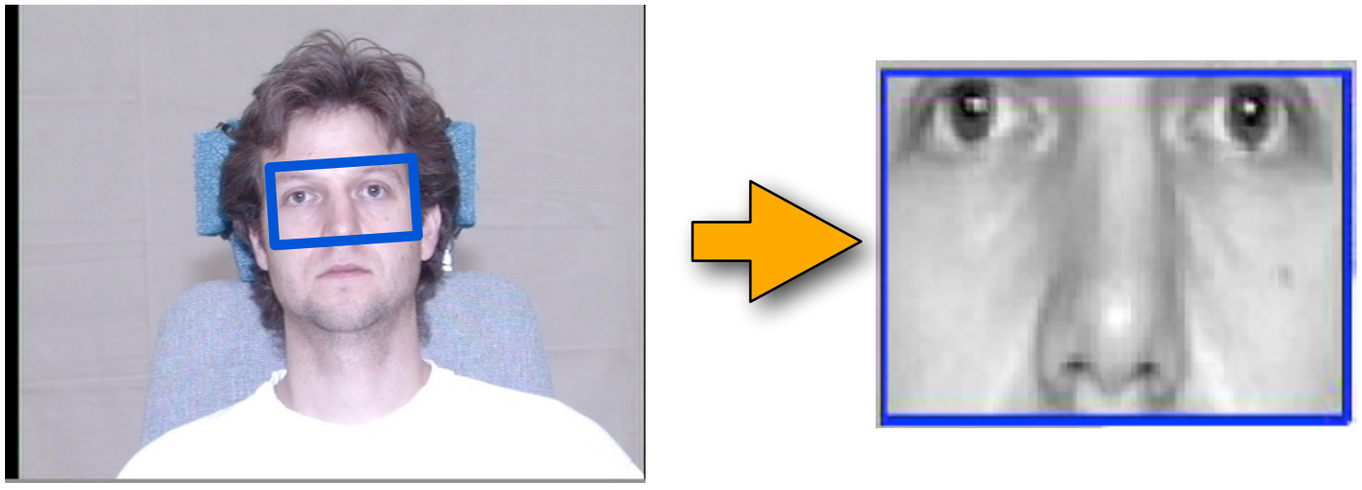


Lucas & Kanade Algorithm

- Lucas & Kanade (1981) realized this and proposed a method for estimating warp displacement using the principles of ***gradients*** and ***spatial coherence***.
- Technique applies Taylor series approximation to any spatially coherent area governed by the warp $\mathcal{W}(\mathbf{x};\mathbf{p})$.

$$\mathcal{I}(\mathbf{p}+\Delta\mathbf{p}) \approx \mathcal{I}(\mathbf{p}) + \frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}^T}\Delta\mathbf{p}$$

"We consider this image to always be static...."



Lucas & Kanade Algorithm

- Lucas & Kanade (1981) realized this and proposed a method for estimating warp displacement using the principles of ***gradients*** and ***spatial coherence***.
- Technique applies Taylor series approximation to any spatially coherent area governed by the warp $\mathcal{W}(\mathbf{x};\mathbf{p})$.

$$\mathcal{I}(\mathbf{p}+\Delta\mathbf{p}) \approx \mathcal{I}(\mathbf{p}) + \frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}^T}\Delta\mathbf{p}$$

Lucas & Kanade Algorithm

- Lucas & Kanade (1981) realized this and proposed a method for estimating warp displacement using the principles of ***gradients*** and ***spatial coherence***.
- Technique applies Taylor series approximation to any spatially coherent area governed by the warp $\mathcal{W}(\mathbf{x};\mathbf{p})$.

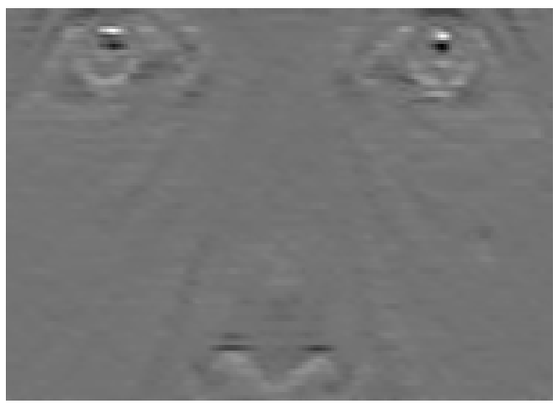
$$\mathcal{I}(\mathbf{p}+\Delta\mathbf{p}) \approx \mathcal{I}(\mathbf{p}) + \frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}^T}\Delta\mathbf{p}$$

$$\frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}^T} = \begin{bmatrix} \frac{\partial \mathcal{I}(\mathbf{x'}_1)}{\partial \mathbf{x'}_1^T} & \cdots & \mathbf{0}^T \\ \vdots & \ddots & \vdots \\ \mathbf{0}^T & \cdots & \frac{\partial \mathcal{I}(\mathbf{x'}_N)}{\partial \mathbf{x'}_N^T} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathcal{W}(\mathbf{x}_1;\mathbf{p})}{\partial \mathbf{p}^T} \\ \vdots \\ \frac{\partial \mathcal{W}(\mathbf{x}_N;\mathbf{p})}{\partial \mathbf{p}^T} \end{bmatrix}$$

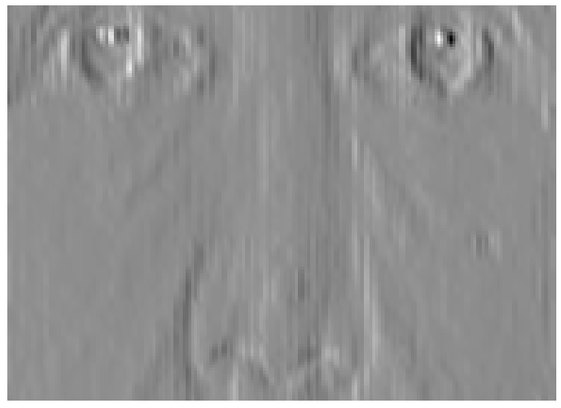
$$\mathbf{x}' = \mathcal{W}(\mathbf{x};\mathbf{p})$$

Lucas & Kanade Algorithm

$$\frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}^T} = \begin{bmatrix} \frac{\partial \mathcal{I}(\mathbf{x'}_1)}{\partial \mathbf{x'}_1^T} & \cdots & \mathbf{0}^T \\ \vdots & \ddots & \vdots \\ \mathbf{0}^T & \cdots & \frac{\partial \mathcal{I}(\mathbf{x'}_N)}{\partial \mathbf{x'}_N^T} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathcal{W}(\mathbf{x}_1;\mathbf{p})}{\partial \mathbf{p}^T} \\ \vdots \\ \frac{\partial \mathcal{W}(\mathbf{x}_N;\mathbf{p})}{\partial \mathbf{p}^T} \end{bmatrix}$$



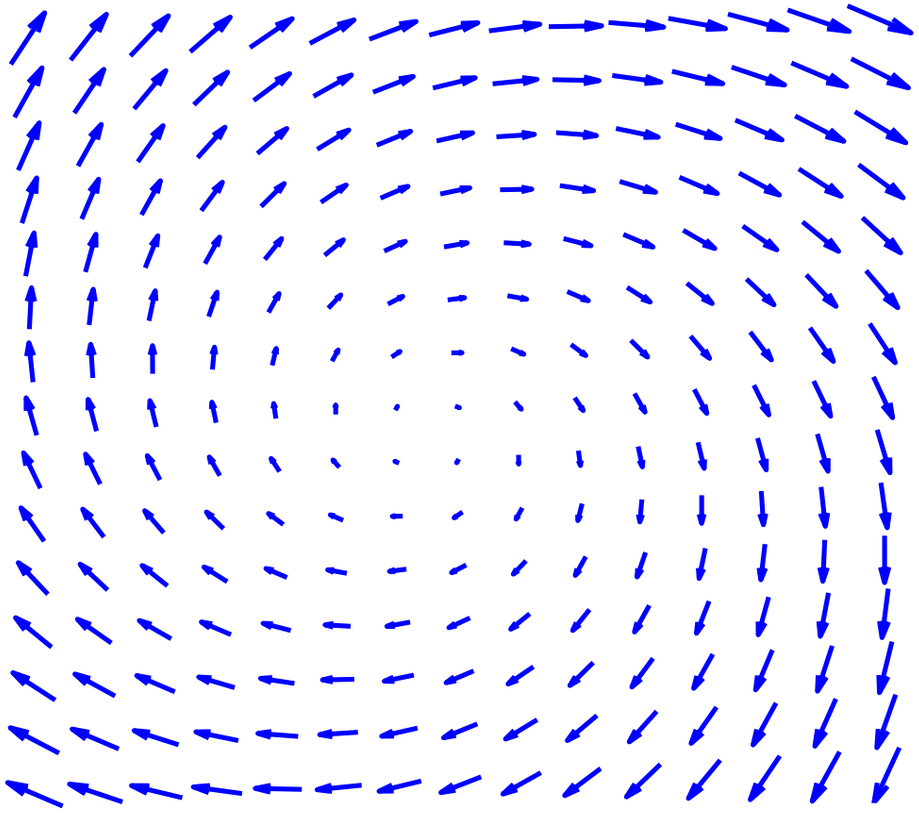
$\nabla_x \mathcal{I}$



$\nabla_y \mathcal{I}$

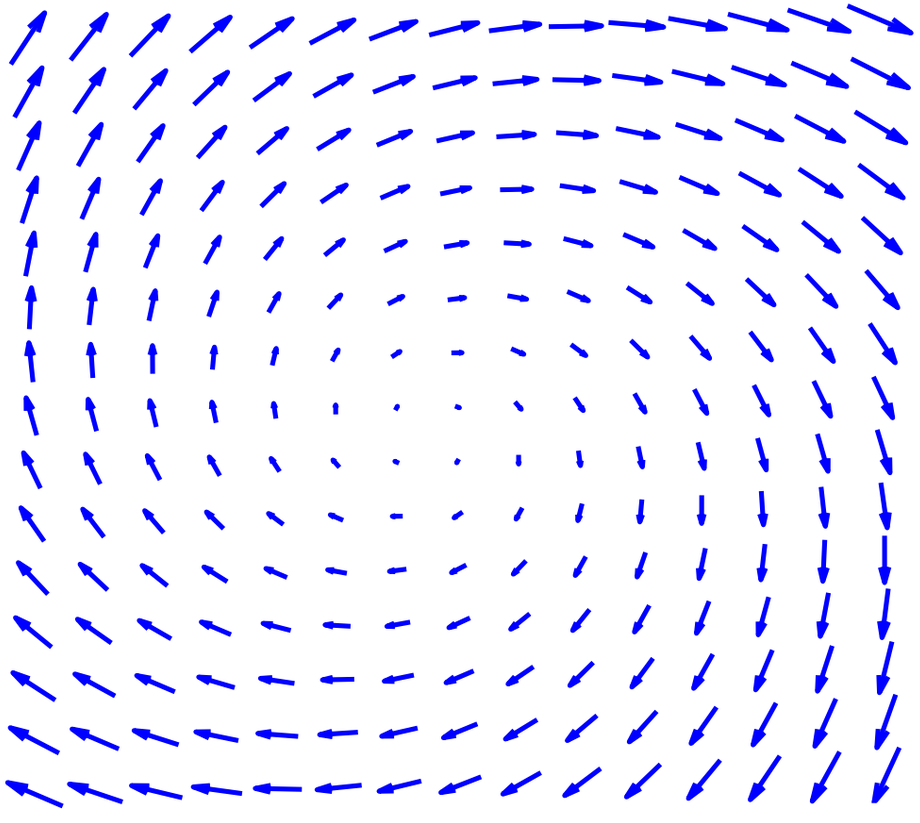
Lucas & Kanade Algorithm

$$\frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}^T} = \begin{bmatrix} \frac{\partial \mathcal{I}(\mathbf{x'}_1)}{\partial \mathbf{x'}_1^T} & \cdots & \mathbf{0}^T \\ \vdots & \ddots & \vdots \\ \mathbf{0}^T & \cdots & \frac{\partial \mathcal{I}(\mathbf{x'}_N)}{\partial \mathbf{x'}_N^T} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathcal{W}(\mathbf{x}_1;\mathbf{p})}{\partial \mathbf{p}^T} \\ \vdots \\ \frac{\partial \mathcal{W}(\mathbf{x}_N;\mathbf{p})}{\partial \mathbf{p}^T} \end{bmatrix}$$



Lucas & Kanade Algorithm

$$\frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}^T} = \begin{bmatrix} \frac{\partial \mathcal{I}(\mathbf{x'}_1)}{\partial \mathbf{x'}_1^T} & \cdots & \mathbf{0}^T \\ \vdots & \ddots & \vdots \\ \mathbf{0}^T & \cdots & \frac{\partial \mathcal{I}(\mathbf{x'}_N)}{\partial \mathbf{x'}_N^T} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathcal{W}(\mathbf{x}_1;\mathbf{p})}{\partial \mathbf{p}^T} \\ \vdots \\ \frac{\partial \mathcal{W}(\mathbf{x}_N;\mathbf{p})}{\partial \mathbf{p}^T} \end{bmatrix}$$



$$\mathcal{W}(\mathbf{x};\mathbf{p}) = \mathbf{M} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} 1+p_1 & p_2 & p_3 \\ p_4 & 1+p_5 & p_6 \end{bmatrix}$$

$$\mathbf{p} = [p_1 \ldots p_6]^T$$

Can you determine $\frac{\partial \mathcal{W}(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}}$ **?**

Lucas & Kanade Algorithm

- Lucas & Kanade (1981) realized this and proposed a method for estimating warp displacement using the principles of ***gradients*** and ***spatial coherence***.
- Technique applies Taylor series approximation to any spatially coherent area governed by the warp $\mathcal{W}(\mathbf{x};\mathbf{p})$.

$$\underbrace{\mathcal{I}(\mathbf{p}+\Delta\mathbf{p})}_{N\times 1} \approx \underbrace{\mathcal{I}(\mathbf{p})}_{N\times 1} + \underbrace{\frac{\partial \mathcal{I}(\mathbf{p})^T}{\partial \mathbf{p}}}_{N\times P}\Delta\mathbf{p}$$

N = number of pixels

P = number of warp parameters

Lucas & Kanade Algorithm

- Actual algorithm is just the application of the following steps,

Step 1:

$$\Delta \mathbf{p} = \arg\min_{\Delta \mathbf{p}} ||\mathcal{I}(\mathbf{p}) + \frac{\partial \mathcal{I}(\mathbf{p})}{\partial \mathbf{p}^T} \Delta \mathbf{p} - \mathcal{T}(\mathbf{0})||_2^2$$

Step 2:

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}$$

keep applying steps until $\Delta \mathbf{p}$ converges.

LK Algorithm

- Actual algorithm is just the application of the following steps,

Step 1:

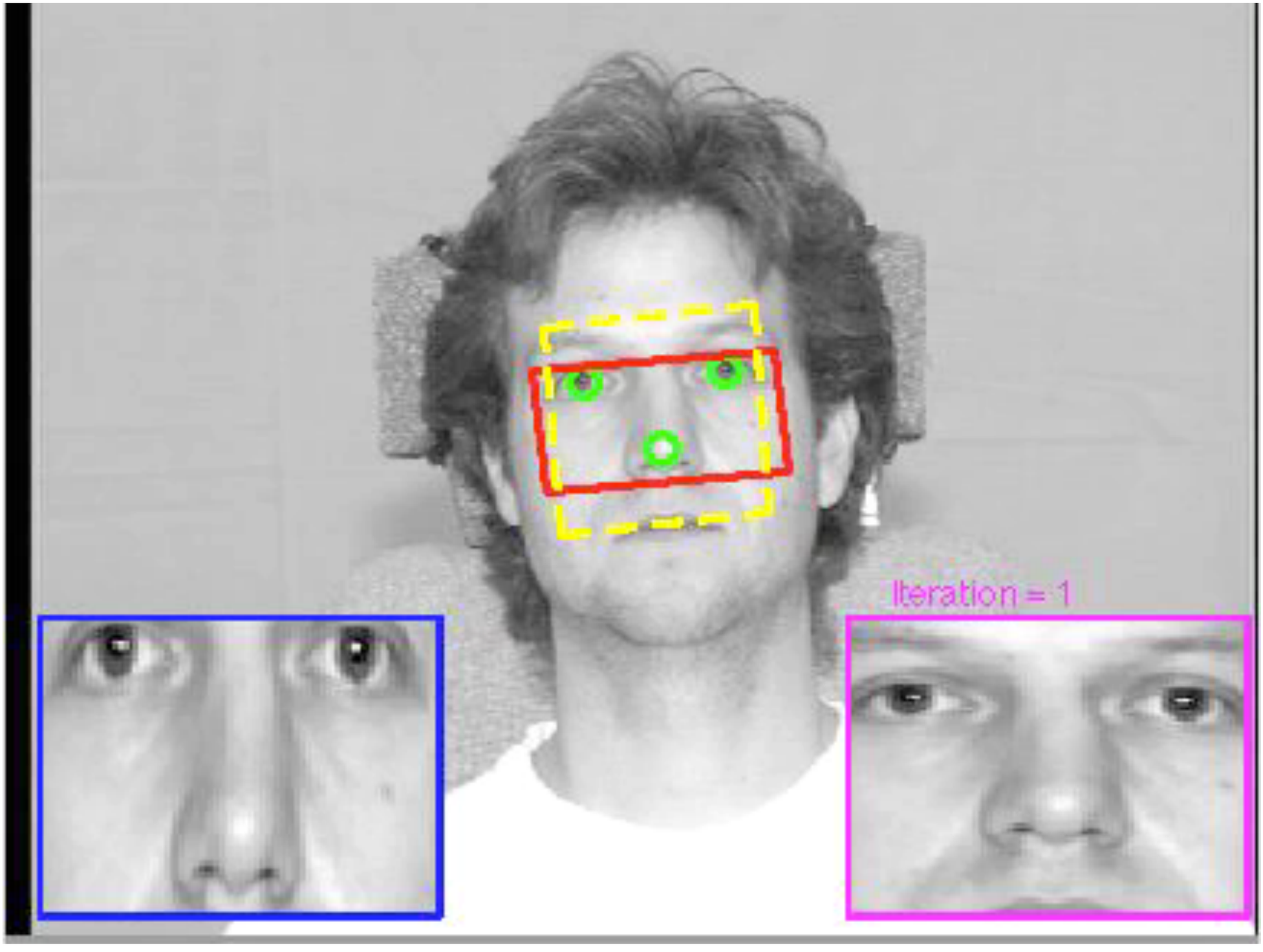
$$\Delta \mathbf{p} = \arg\min_{\Delta \mathbf{p}} ||\mathbf{A}\Delta \mathbf{p} - \mathbf{b}||_2^2$$

Step 2:

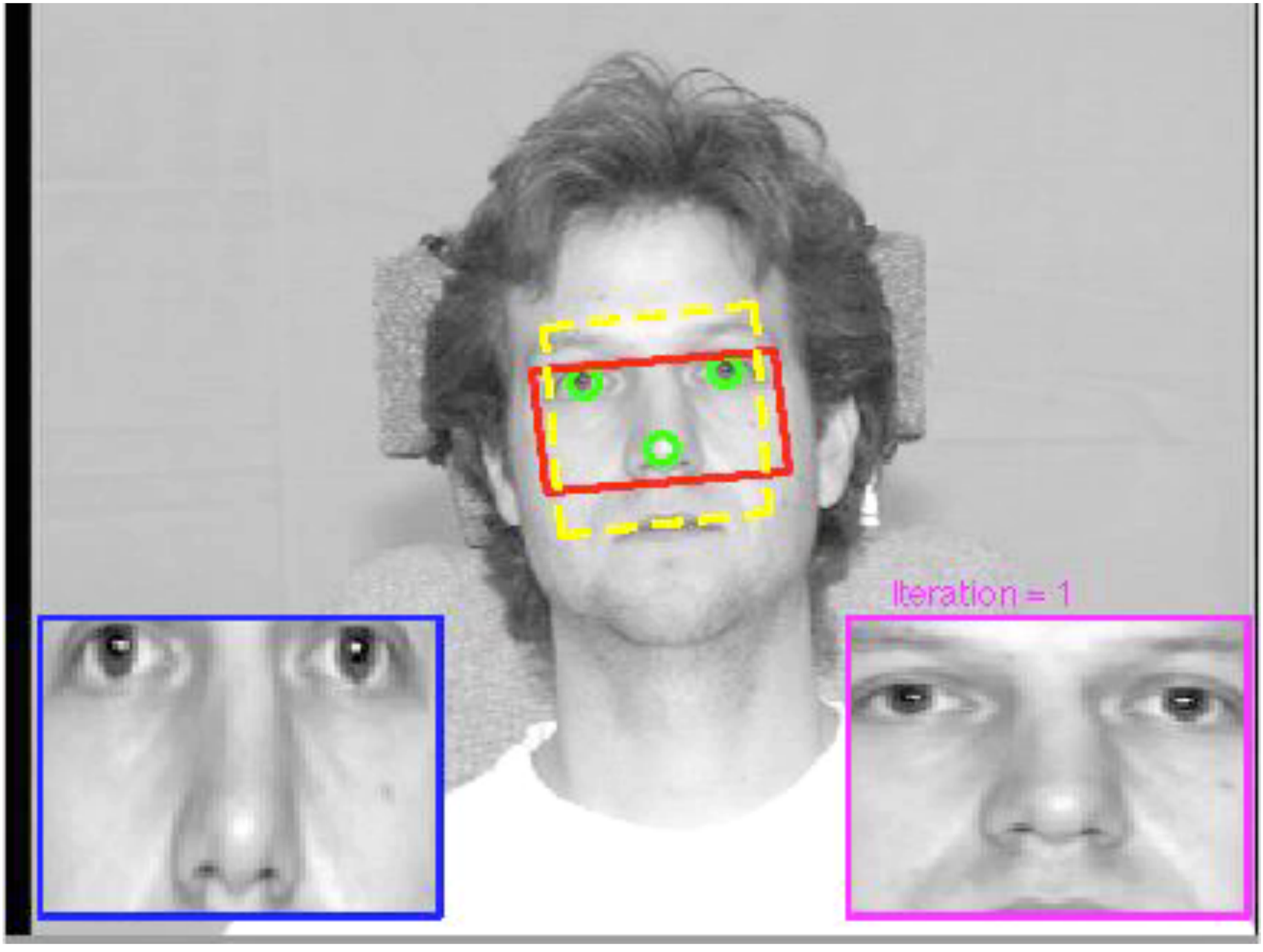
$$\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}$$

keep applying steps until $\Delta \mathbf{p}$ converges.
$(\mathbf{A}, \mathbf{b})$ change at each iteration.

Examples of LK Alignment



Examples of LK Alignment



More to read…

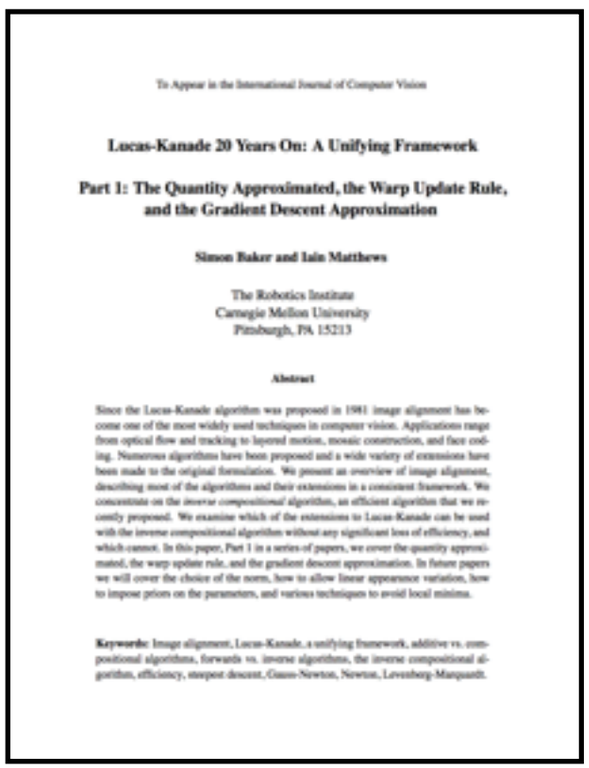
An Iterative Image Registration Technique with an Application to Stereo Vision

Abstract

1. Introduction

2. The registration problem

- Lucas & Kanade, “An Iterative Image Registration Technique with Application to Stereo Vision”, IJCAI 1981.
- Baker & Matthews, “Lucas-Kanade 20 Years On: A Unifying Framework”, IJCV 2004.
- Bristow & Lucey, “In Defense of Gradient-Based Alignment on Densely Sampled Sparse Features”, Springer Book on Dense Registration Methods 2015.



To Appear in the International Journal of Computer Vision

Lucas-Kanade 20 Years On: A Unifying Framework

Part 1: The Quantity Approximated, the Warp Update Rule, and the Gradient Descent Approximation

Simon Baker and Iain Matthews

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Since the Lucas-Kanade algorithm was proposed in 1981 image alignment has become one of the most widely used techniques in computer vision. Applications range from optical flow and tracking to layered motion, mosaic construction, and face coding. Numerous algorithms have been proposed and a wide variety of extensions have been made to the original formulation. We present an overview of image alignment, describing most of the algorithms and their extensions in a consistent framework. We concentrate on the *inverse compositional* algorithm, an efficient algorithm that we recently proposed. We examine which of the extensions to Lucas-Kanade can be used with the inverse compositional algorithm without any significant loss of efficiency, and which cannot. In this paper, Part 1 in a series of papers, we cover the quantity approximated, the warp update rule, and the gradient descent approximation. In future papers we will cover the choice of the norm, how to allow linear appearance variation, how to impose priors on the parameters, and various techniques to avoid local minima.

Keywords: Image alignment, Lucas-Kanade, a unifying framework, additive vs. compositional algorithms, forwards vs. inverse algorithms, the inverse compositional algorithm, efficiency, steepest descent, Gauss-Newton, Newton, Levenberg-Marquardt.

In Defense of Gradient-Based Alignment on Densely Sampled Sparse Features

Hilton Bristow and Simon Lucey

Abstract In this chapter, we explore the surprising result that gradient-based continuous optimization methods perform well for the alignment of image/object models when using densely sampled sparse features (HOG, dense SIFT, *etc.*). Gradient-based approaches for image/object alignment have many desirable properties – inference is typically fast and exact, and diverse constraints can be imposed on the motion of points. However, the presumption that gradients predicted on sparse features would be poor estimators of the true descent direction has meant that gradient-based optimization is often overlooked in favour of graph-based optimization. We show that this intuition is only partly true; sparse features are indeed poor predictors of the error surface, but this has no impact on the actual alignment performance. In fact, for general object categories that exhibit large geometric and appearance variation, sparse features are integral to achieving any convergence whatsoever. How the descent directions are predicted becomes an important consideration for these descriptors. We explore a number of strategies for estimating gradients, and show that estimating gradients via regression in a manner that explicitly handles outliers improves alignment performance substantially. To illustrate the general applicability of gradient-based methods to the alignment of challenging object categories, we perform unsupervised ensemble alignment on a series of non-rigid animal classes from ImageNet.

Hilton Bristow
Queensland University of Technology, Australia. e-mail: hilton.bristow@gmail.com

Simon Lucey
The Robotics Institute, Carnegie Mellon University, USA. e-mail: slucey@cs.cmu.edu