Image alignment
Homework 6 has been posted and is due on April 20\textsuperscript{th}.
- Any questions about the homework?
- How many of you have looked at/started/finished homework 6?

This week Yannis’ office hours will be on Wednesday 3-6 pm.
- Added an extra hour to make up for change.
Overview of today’s lecture

- Leftover from last time: Horn-Schunck flow.
- Motion magnification using optical flow.
- Image alignment.
- Lucas-Kanade alignment.
- Baker-Matthews alignment.
- Inverse alignment.
Most of these slides were adapted from:

Motion magnification using optical flow
How would you achieve this effect?

- Compute optical flow from frame to frame.
- Magnify optical flow velocities.
- Appropriately warp image intensities.
How would you achieve this effect?

- Compute optical flow from frame to frame.
- Magnify optical flow velocities.
- Appropriately warp image intensities.

In practice, many additional steps are required for a good result.
Some more examples
Some more examples
Image alignment
http://www.humansensing.cs.cmu.edu/intraface/
How can I find in the image?
Idea #1: Template Matching

Slow, combinatory, global solution
Idea #2: Pyramid Template Matching

Faster, combinatory, locally optimal
Idea #3: Model refinement

(when you have a good initial solution)

Fastest, locally optimal
Some notation before we get into the math…

2D image transformation

$$W(x; p)$$

2D image coordinate

$$x = \begin{bmatrix} x \\ y \end{bmatrix}$$

Parameters of the transformation

$$p = \{p_1, \ldots, p_N\}$$

Warped image

$$I(x') = I(W(x; p))$$

Pixel value at a coordinate
Some notation before we get into the math...

2D image transformation
\[ \mathbf{W}(x; p) \]

2D image coordinate
\[ x = \begin{bmatrix} x \\ y \end{bmatrix} \]

Parameters of the transformation
\[ p = \{ p_1, \ldots, p_N \} \]

Warped image
\[ I(x') = I(\mathbf{W}(x; p)) \]

**Translation**
\[
\mathbf{W}(x; p) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & p_1 \\ 0 & 1 & p_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}
\]

**Affine**
Some notation before we get into the math...

2D image transformation

\[ W(x; p) \]

2D image coordinate

\[ x = \begin{bmatrix} x \\ y \end{bmatrix} \]

Parameters of the transformation

\[ p = \{p_1, \ldots, p_N\} \]

Warped image

\[ I(x') = I(W(x; p)) \]

Pixel value at a coordinate

**Translation**

\[
W(x; p) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix}
= \begin{bmatrix} 1 & 0 & p_1 \\ 0 & 1 & p_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}
\]

**Affine**

\[
W(x; p) = \begin{bmatrix} p_1 x + p_2 y + p_3 \\ p_4 x + p_5 y + p_6 \end{bmatrix}
= \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}
\]

Warped image can be written in matrix form when linear affine warp matrix can also be 3x3 when last row is [0 0 1]
\[ W(x; p) \] takes a ______ as input and returns a ______

\[ W(x; p) \] is a function of ____ variables

\[ W(x; p) \] returns a ______ of dimension ___ x ___

\[ p = \{ p_1, \ldots, p_N \} \] where N is _____ for an affine model

\[ I(x') = I(W(x; p)) \] this warp changes pixel values?
Image alignment
(problem definition)

\[ \min_p \sum_x [I(W(x; p)) - T(x)]^2 \]

Find the warp parameters \( p \) such that the SSD is minimized
Find the warp parameters $p$ such that the SSD is minimized.
Image alignment
(problem definition)

\[
\min_p \sum_x \left[ I(\mathbf{W}(x; p)) - T(x) \right]^2
\]

Find the warp parameters \( p \) such that the SSD is minimized

How could you find a solution to this problem?
This is a non-linear (quadratic) function of a non-parametric function!

(Function $I$ is non-parametric)

\[ \min_p \sum_x \left[ I(W(x; p)) - T(x) \right]^2 \]

Hard to optimize

*What can you do to make it easier to solve?*
Hard to optimize

What can you do to make it easier to solve?

assume good initialization,
linearized objective and update incrementally

This is a non-linear (quadratic) function of a non-parametric function!

(Function $I$ is non-parametric)

$$\min_p \sum_x [I(W(x; p)) - T(x)]^2$$
Lucas-Kanade alignment
If you have a good initial guess $p$...

\[
\sum_x [I(W(x; p)) - T(x)]^2
\]

(can be written as ...)

\[
\sum_x [I(W(x; p + \Delta p)) - T(x)]^2
\]

(a small incremental adjustment)

(this is what we are solving for now)
This is **still** a non-linear (quadratic) function of a non-parametric function!

(Function $I$ is non-parametric)

$$\sum_x [I(W(x; p + \Delta p)) - T(x)]^2$$

*How can we linearize the function $I$ for a really small perturbation of $p$?*
This is **still** a non-linear (quadratic) function of a non-parametric function!

(Function $I$ is non-parametric)

\[
\sum_x [I(W(x; p + \Delta p)) - T(x)]^2
\]

*How can we linearize the function $I$ for a really small perturbation of $p$?*

Taylor series approximation!
Multivariable Taylor Series Expansion
(First order approximation)

\[ f(x, y) \approx f(a, b) + f_x(a, b)(x - a) + f_y(a, b)(y - b) \]
Multivariable Taylor Series Expansion
(First order approximation)

\[ f(x, y) \approx f(a, b) + f_x(a, b)(x - a) + f_y(a, b)(y - b) \]

Recall: \[ x' = W(x; p) \]

\[
I(W(x; p + \Delta p)) \approx I(W(x; p)) + \frac{\partial I(W(x; p))}{\partial p} \Delta p
\]

\[
= I(W(x; p)) + \frac{\partial I(W(x; p))}{\partial x'} \frac{\partial W(x; p)}{\partial p} \Delta p
\]

chain rule

\[
= I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p
\]

short-hand
Multivariable Taylor Series Expansion
(First order approximation)

\[ f(x, y) \approx f(a, b) + f_x(a, b)(x - a) + f_y(a, b)(y - b) \]

Linear approximation

\[ \sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2 \]

*What are the unknowns here?*
\[
\sum_x \left[ I(W(x; p + \Delta p)) - T(x) \right]^2
\]

Multivariable Taylor Series Expansion
(First order approximation)

\[f(x, y) \approx f(a, b) + f_x(a, b)(x - a) + f_y(a, b)(y - b)\]

Linear approximation

\[
\sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2
\]

Now, the function is a linear function of the unknowns
\sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2

\mathbf{x} \quad \text{is a \underline{________} of dimension ___ x ___}

\text{output of} \quad \mathbf{W} \quad \text{is a \underline{________} of dimension ___ x ___}

\mathbf{p} \quad \text{is a \underline{________} of dimension ___ x ___}

I(\cdot) \quad \text{is a function of _____ variables}
The Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$

(A matrix of partial derivatives)

\[ \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \]

\[ \mathbf{W} = \begin{bmatrix} W_x(x, y) \\ W_y(x, y) \end{bmatrix} \]

\[ \frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \cdots & \frac{\partial W_x}{\partial p_N} \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \cdots & \frac{\partial W_y}{\partial p_N} \end{bmatrix} \]

Affine transform

\[ \mathbf{W}(\mathbf{x}; \mathbf{p}) = \begin{bmatrix} p_1 x + p_3 y + p_5 \\ p_2 x + p_4 y + p_6 \end{bmatrix} \]

\[ \frac{\partial W_x}{\partial p_1} = x \quad \frac{\partial W_x}{\partial p_2} = 0 \quad \cdots \]

\[ \frac{\partial W_y}{\partial p_1} = 0 \quad \cdots \]

Rate of change of the warp

\[ \frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{bmatrix} \]
$$\sum_{x} \left[ I(W(x;p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2$$

\(\nabla I\) is a _________ of dimension ___ \(\times\) ___

\(\frac{\partial W}{\partial p}\) is a _________ of dimension ___ \(\times\) ___

\(\Delta p\) is a _________ of dimension ___ \(\times\) ___
\[ \sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2 \]
\[ \sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2 \]
\[
\sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2
\]
\[ \sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2 \]
\[
\sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2
\]
\[
\sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2
\]
\[
\sum_x \left[ I(\mathbf{W}(x; p)) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} - T(x) \right]^2
\]
\[ \sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2 \]

- **Pixel coordinate**: \((2 \times 1)\)
- **Image intensity**: \((1 \times 2)\)
- **Warp function**: \((2 \times 1)\)
- **Warp parameters**: \((6 \text{ for affine})\)
- **Image gradient**: \((1 \times 2)\)
- **Partial derivatives of warp function**: \((2 \times 6)\)
- **Incremental warp**: \((6 \times 1)\)
When you implement this, you will compute everything in parallel and store as matrix … don’t loop over x!
Summary
(of Lucas-Kanade Image Alignment)

Problem:

\[ \min_p \sum_x [I(W(x; p)) - T(x)]^2 \]

warped image template image

Difficult non-linear optimization problem

Strategy:

\[ \sum_x [I(W(x; p + \Delta p)) - T(x)]^2 \]

Assume known approximate solution
Solve for increment

\[ \sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2 \]

Taylor series approximation
Linearize
then solve for \( \Delta p \)
OK, so how do we solve this?

\[
\min_{\Delta p} \sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2
\]
Another way to look at it...

$$\min_{\Delta p} \sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2$$

(moving terms around)

$$\min_{\Delta p} \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \Delta p - \{ T(x) - I(W(x; p)) \} \right]^2$$

Have you seen this form of optimization problem before?
Another way to look at it…

\[
\min_{\Delta p} \sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2
\]

\[
\min_{\Delta p} \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \Delta p - \{T(x) - I(W(x; p))\} \right]^2
\]

Looks like \[
Ax - b
\]

*How do you solve this?*
Least squares approximation

\[ \hat{x} = \arg \min_x \|Ax - b\|^2 \]

is solved by

\[ x = (A^T A)^{-1} A^T b \]

Applied to our tasks:

\[ \min_{\Delta p} \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \Delta p - \{T(x) - I(W(x; p))\} \right]^2 \]

is optimized when

\[ \Delta p = H^{-1} \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^\top [T(x) - I(W(x; p))] \]

where

\[ H = \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^\top \left[ \nabla I \frac{\partial W}{\partial p} \right] \]

\[ x = (A^T A)^{-1} A^T b \]
Solve:

\[
\min_p \sum_x \left[ I(W(x; p)) - T(x) \right]^2
\]

warped image  template image

Difficult non-linear optimization problem

Strategy:

\[
\sum_x \left[ I(W(x; p + \Delta p)) - T(x) \right]^2
\]

Assume known approximate solution
Solve for increment

\[
\sum_x \left[ I(W(x; p)) + \nabla I \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2
\]

Taylor series approximation
Linearize

Solution:

\[
\Delta p = H^{-1} \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^T \left[ T(x) - I(W(x; p)) \right]
\]

Solution to least squares approximation

\[
H = \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^T \left[ \nabla I \frac{\partial W}{\partial p} \right]
\]

Hessian
This is called…

Gauss-Newton gradient decent non-linear optimization!
Lucas Kanade (Additive alignment)

1. Warp image \( I(\mathbf{W}(\mathbf{x}; \mathbf{p})) \)

2. Compute error image \([T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))]\)

3. Compute gradient \( \nabla I(\mathbf{x}') \)

4. Evaluate Jacobian \( \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \)

5. Compute Hessian \( H = \sum_{\mathbf{x}} \left[ \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^\top \left[ \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right] \)

6. Compute \( \Delta \mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[ \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^\top [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))] \)

7. Update parameters \( \mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p} \)

Just 8 lines of code!
Baker-Matthews alignment
Image Alignment

(start with an initial solution, match the image and template)
Image Alignment Objective Function

\[ \sum_{x} [I(\mathbf{W}(x; p)) - T(x)]^2 \]

Given an initial solution...several possible formulations

Additive Alignment

\[ \sum_{x} [I(\mathbf{W}(x; p + \Delta p)) - T(x)]^2 \]

incremental perturbation of parameters
Image Alignment Objective Function

$$\sum_x [I(W(x; p)) - T(x)]^2$$

Given an initial solution...several possible formulations

**Additive Alignment**

$$\sum_x [I(W(x; p + \Delta p)) - T(x)]^2$$

incremental perturbation of parameters

**Compositional Alignment**

$$\sum_x [I(W(W(x; \Delta p); p) - T(x)]^2$$

incremental warps of image
Additive strategy

- First shot
- Second shot
- Go back, adjust and try again
Compositional strategy

start from here

first shot

second shot

start from here
Additive

$I(x)$

$W(x;p)$
Additive

\[ W(x; p + \Delta p) \]

\[ W(x; p) \]
W(x; p + Δp) = W(x; p)
\[ W(x; 0 + \Delta p) = W(x; \Delta p) \]

**Additive**

\[ W(x; p + \Delta p) \]

**Compositional**

\[ W(x; p) \circ W(x; \Delta p) \]
Compositional Alignment

Original objective function (SSD)

\[
\min_p \sum_x [I(W(x; p)) - T(x)]^2
\]

Assuming an initial solution \(p\) and a compositional warp increment

\[
\sum_x [I(W(W(x; \Delta p); p)) - T(x)]^2
\]
Compositional Alignment

Original objective function (SSD)

$$\min_{p} \sum_{x} [I(W(x; p)) - T(x)]^2$$

Assuming an initial solution $p$ and a compositional warp increment

$$\sum_{x} [I(W(W(x; \Delta p); p) - T(x)]^2$$

Another way to write the composition

$W(x; p) \circ W(x; \Delta p) \equiv W(W(x; \Delta p); p)$

Identity warp

$W(x; 0)$
Compositional Alignment

Original objective function (SSD)

$$\min_{p} \sum_{x} \left[ I(\mathbf{W}(x; p)) - T(x) \right]^2$$

Assuming an initial solution $p$ and a compositional warp increment

$$\sum_{x} \left[ I(\mathbf{W}(\mathbf{W}(x; \Delta p); p)) - T(x) \right]^2$$

Another way to write the composition

$$\mathbf{W}(x; p) \circ \mathbf{W}(x; \Delta p) \equiv \mathbf{W}(\mathbf{W}(x; \Delta p); p)$$

Identity warp

$$\mathbf{W}(x; 0)$$

Skipping over the derivation…the new update rule is

$$\mathbf{W}(x; p) \leftarrow \mathbf{W}(x; p) \circ \mathbf{W}(x; \Delta p)$$
So what’s so great about this compositional form?
Additive Alignment

\[ \sum_x [I(W(x; p + \Delta p)) - T(x)]^2 \]

linearized form

\[ \sum_x \left[ I(W(x; p)) + \nabla I(x') \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2 \]

Compositional Alignment

\[ \sum_x [I(W(x; \Delta p); p) - T(x)]^2 \]

linearized form

\[ \sum_x \left[ I(W(x; p)) + \nabla I(x') \frac{\partial W(x; 0)}{\partial p} \Delta p - T(x) \right]^2 \]
The Jacobian is constant. Jacobian can be precomputed!
Compositional Image Alignment

Minimize

$$\sum_x [I(W(W(x; \Delta p); p)) - T(x)]^2 \approx \sum_x \left[ I(W(x; p)) + \nabla I(W) \frac{\partial W}{\partial p} \Delta p - T(x) \right]^2$$

Jacobian is simple and can be precomputed.
Lucas Kanade (Additive alignment)

1. Warp image \( I(W(x; p)) \)

2. Compute error image \( [T(x) - I(W(x; p))]^2 \)

3. Compute gradient \( \nabla I(x') \)

4. Evaluate Jacobian \( \frac{\partial W}{\partial p} \)

5. Compute Hessian \( H \)

6. Compute \( \Delta p \)

7. Update parameters \( p \leftarrow p + \Delta p \)
Shum-Szeliski (Compositional alignment)

1. Warp image \(I(W(x; p))\)

2. Compute error image \([T(x) - I(W(x; p))]^2\)

3. Compute gradient \(\nabla I(x')\)

4. Evaluate Jacobian \(\frac{\partial W(x; 0)}{\partial p}\)

5. Compute Hessian \(H\)

6. Compute \(\Delta p\)

7. Update parameters \(W(x; p) \rightarrow W(x; p) \circ W(x; \Delta p)\)
Any other speed up techniques?
Inverse alignment
Why not compute warp updates on the template?

Additive Alignment

$$\sum_x [I(W(x; p + \Delta p)) - T(x)]^2$$

Compositional Alignment

$$\sum_x [I(W(W(x; \Delta p); p) - T(x)]^2$$
Why not compute warp updates on the template?

**Additive Alignment**

$$\sum_x [I(W(x; p + \Delta p)) - T(x)]^2$$

**Compositional Alignment**

$$\sum_x [I(W(W(x; \Delta p); p) - T(x)]^2$$

What happens if you let the template be warped too?

**Inverse Compositional Alignment**

$$\sum_x [T(W(x; \Delta p) - I(W(x; p))]^2$$
Compositional

$$W(x;0 + \Delta p) = W(x;\Delta p)$$

Inverse compositional

$$W(x;p) o W(x;\Delta p)^{-1}$$
Compositional strategy

start from here

first shot

second shot

start from here
Inverse Compositional strategy

first shot

move the hole
So what’s so great about this inverse compositional form?
Inverse Compositional Alignment

Minimize

\[
\sum_x [T(W(x; \Delta p) - I(W(x; p))]^2 \approx \sum_x \left[T(W(x; 0)) + \nabla T \frac{\partial W}{\partial p} \Delta p - I(W(x; p)) \right]^2
\]

Solution

\[
H = \sum_x \left[\nabla T \frac{\partial W}{\partial p} \right]^\top \left[\nabla T \frac{\partial W}{\partial p} \right]
\]

\[
\Delta p = \sum_x H^{-1} \left[\nabla T \frac{\partial W}{\partial p} \right]^\top \left[T(x) - I(W(x; p)) \right]
\]

Update

\[
W(x; p) \leftarrow W(x; p) \circ W(x; \Delta p)^{-1}
\]
Properties of inverse compositional alignment

**Jacobian** can be precomputed
It is constant - evaluated at $W(x;0)$

**Gradient of template** can be precomputed
It is constant

**Hessian** can be precomputed

$$H = \sum_x \left[ \nabla_T \frac{\partial W}{\partial p} \right]^\top \left[ \nabla_T \frac{\partial W}{\partial p} \right]$$

$$\Delta p = \sum_x H^{-1} \left[ \nabla_T \frac{\partial W}{\partial p} \right]^\top \left[ T(x) - I(W(x; p)) \right]$$

(main term that needs to be computed)

**Warp** must be invertible
Lucas Kanade (Additive alignment)

1. Warp image \( I(W(x;p)) \)

2. Compute error image \( [T(x) - I(W(x;p))]^2 \)

3. Compute gradient \( \nabla I(W) \)

4. Evaluate Jacobian \( \frac{\partial W}{\partial p} \)

5. Compute Hessian \( H \)

6. Compute \( \Delta p \)

7. Update parameters \( p \leftarrow p + \Delta p \)
Shum-Szeliski (Compositional alignment)

1. Warp image \( I(W(x; p)) \)

2. Compute error image \([T(x) - I(W(x; p))]\)

3. Compute gradient \( \nabla I(x') \)

4. Evaluate Jacobian \( \frac{\partial W(x; 0)}{\partial p} \)

5. Compute Hessian \( H \)

6. Compute \( \Delta p \)

7. Update parameters \( W(x; p) \leftarrow W(x; p) \circ W(x; \Delta p) \)
Baker-Matthews (Inverse Compositional alignment)

1. Warp image $I(W(x; p))$

2. Compute error image $[T(x) - I(W(x; p))]$

3. Compute gradient $\nabla T(W)$

4. Evaluate Jacobian $\frac{\partial W}{\partial p}$

5. Compute Hessian $H$

\[ H = \sum_x \left[ \nabla T \frac{\partial W}{\partial p} \right] ^T \left[ \nabla T \frac{\partial W}{\partial p} \right] \]

6. Compute $\Delta p$

\[ \Delta p = \sum_x H^{-1} \left[ \nabla T \frac{\partial W}{\partial p} \right] ^T [T(x) - I(W(x; p))] \]

7. Update parameters $W(x; p) \leftarrow W(x; p) \circ W(x; \Delta p)^{-1}$
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Efficient</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwards Additive</td>
<td>No</td>
<td>Lucas, Kanade</td>
</tr>
<tr>
<td>Forwards compositional</td>
<td>No</td>
<td>Shum, Szeliski</td>
</tr>
<tr>
<td>Inverse Additive</td>
<td>Yes</td>
<td>Hager, Belhumeur</td>
</tr>
<tr>
<td>Inverse Compositional</td>
<td>Yes</td>
<td>Baker, Matthews</td>
</tr>
</tbody>
</table>
Basic reading:
• Szeliski, Section 8.1.