Lucas-Kanade Image Alignment Algorithms

Objective: Image Alignment

http://www.codeproject.com/KB/recipes/ImgAlign.aspx (C++)

http://www.ri.cmu.edu/research_project_detail.html?project_id=515&menu_id=261 (Matlab)
Image Alignment

• Align a template \( T(\mathbf{x}) \) to a target image \( I(\mathbf{x}) \), where \( \mathbf{x}=(x, y)^T \) is a vector containing the pixel coordinates.

• Let \( W(\mathbf{x}; \mathbf{p}) \) be a mathematical model which maps the pixels coordinates from one image to the other, parameterized by a set of warping parameters \( \mathbf{p} \).
Image Alignment

The goal of the Lucas-Kanade algorithm is to minimize the sum of squared error between two images, the template $T$ and the image $I$ warped back onto the coordinate frame of the template:

$$\sum_x \left[ W(x; p) - T(x) \right]^2$$  \hspace{1cm} (1)

To optimize the expression in (1), the Lucas-Kanade algorithm assumes that a current estimate of $p$ is known and then iteratively solves for increments to the parameters $\Delta p$; i.e. the following expression is (approximately) minimized:

$$\sum_x \left[ W(x; p + \Delta p) - T(x) \right]^2$$  \hspace{1cm} (2)
Image Alignment

Using the Taylor’s series expansion,

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

Therefore, (2) becomes

\[ \sum_x \left[ I(W(x; p)) \nabla I \frac{\partial W}{\partial p} \Delta p - T \right]^2 \]  \quad (3)

We need to solve for \( \Delta p \). The partial derivative of (3) with respect to \( \Delta p \) is

\[ 2 \sum_x \left[ \nabla I \frac{\partial W}{\partial p} \right]^T \left[ I(W(x; p)) \nabla I \frac{\partial W}{\partial p} \Delta p - T \right] \]  \quad (4)
Image Alignment

Setting (4) equal to zero and solving for $\Delta p$,

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

(5)

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

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

The Lucas-Kanade Algorithm

Iterate:

1. Warp \( I \) with \( W(x; p) \) to compute \( I(W(x; p)) \)
2. Compute the error image \( T(x) - I(W(x; p)) \)
3. Warp the gradient \( \nabla I \) with \( W(x; p) \)
4. Evaluate the Jacobian \( \frac{\partial W}{\partial p} \) at \( (x; p) \)
5. Compute the steepest descent images \( \nabla I \frac{\partial W}{\partial p} \)
6. Compute the Hessian matrix using Equation (11)
7. Compute \( \sum_x [\nabla I \frac{\partial W}{\partial p}]^T [T(x) - I(W(x; p))] \)
8. Compute \( \Delta p \) using Equation (10)
9. Update the parameters \( p \leftarrow p + \Delta p \)

until \( \| \Delta p \| \leq \epsilon \)

Image Alignment

Step 1:\nImage

Step 2:\nTemplate
\( T(x) \)
Warped
\( T(W(x, p)) \)
Warp Parameters
\( p \)
Parameter Updates
\( \Delta p \)
Inverse Hessian
\( H^{-1} \)
Hessian
\( H \)
Error
\( T(x) - T(W(x, p)) \)
SD Parameter Updates

Step 3:\nImage Gradient X

Step 4:\nImage Gradient Y

Step 5:\nWarped Gradients
\( \nabla I_x \)
\( \nabla I_y \)
Jacobian
\( \nabla^2 \)
Steepest Descent Images
\( \nabla I / \nabla^2 \)
Next step:

Efficient alternative of the Lucas-Kanade Image Alignment