1 Theory Questions

Q1.1 Calculating the Jacobian

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

$$\frac{\partial L}{\partial P} = \sum_x 2(T(x) - I(W(x;p)) \frac{\partial I(w(x;p))}{\partial w(x;p)} (-1) \frac{\partial W(x;p)}{\partial P}$$

$$= -2 \sum_x (T(x) - I(W(x;p)) \nabla(I(x)) \frac{\partial \begin{bmatrix} u + p_1 u + v p_3 + p_5 \\ p_2 u + v + v p_4 + p_5 \end{bmatrix}}{\partial \begin{bmatrix} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 \end{bmatrix}}$$

$$= -2 \sum_x (T(x) - I(W(x;p)) \nabla(I(x)) \begin{bmatrix} u & 0 & v & 0 & 1 & 0 \\ 0 & u & 0 & v & 0 & 1 \end{bmatrix}$$

Q1.2 What is the computational complexity of inverse compositional method?

n(pix in T) m(pix in I) p(# parameters of W)

J
$gradient of template -> Separable convolution -> O(2 * 3 * n)$
$Jacobian -> O(n * p)$ (size)
$H -> j^T * j -> O(p * n * p)$ (matrix multiplication)([p *n] X [n * p])
H^{-1} ->O(p^{2.3~3})(matrix inverse, depends on implement)

for each runtime iteration:

image warping template: O(n)

calculate error: O(n)

caculate delta p: O(p*n* 2) ([p *n] X [n * 2])

->time complexity per iteration: $O(np)$

How does this compare to the run time of the regular Lucas-Kanade method?

forward Lucas-Kanade should take $O(np^2)$ to compute Hessian which is worse then $O(np)$

2. Lucas-Kanade Tracker

Q2.4 Test your trackers on the short car video sequence (data/car1/) by running the wrapper scripts lk demo.m and mb demo.m. What sort of templates work well for tracking? At what point does the tracker break down? Why does this happen?

(My ubuntu machine cannot render mp4 file as it lacks the required profile. submitted avi instead)
templates that contain edges and corner should have more change in gradient easier to track.
The black car seems lacking feature so it is kind of hard to track. at about 0:03, the tracker seems to have more intrest on the tree on on top left corner, thus losing track.

In your write-up: Submit your best video of the car being tracked. Save it as results/car.mp4.
saved as results/car_lk.avi, results/car_mb.avi

Q2.5 Try running your tracker on the landing video (data/landing/). This video was taken during a runway approach. With a model of the markings (the lengths of the line segments etc) the output of the tracker can be used to estimate the camera position with respect to the runway and can be used in an automated landing system.

(My ubuntu machine cannot render mp4 file as it lacks the required profile. submitted avi instead)
In your write-up: Submit your best video of the runway markings being tracked. Save it as results/landing.mp4.
saved as results/landing_lk.avi, results/landing_mb.avi

Q3.1x Add illumination robustness

saved as ec/car_Robust.avi, ec/landing_Robust.avi

Q3.2x LK Tracking on an image pyramid.

saved as ec/car_Pyramid.avi, ec/landing_Pyramid.avi