Introduction

- So far we have considered estimating structure and motion from a set of $n$ images $\{I_1, \ldots, I_n\}$.
- The images were not taken in any particular order or time.
- Here we analyse video data, intended as a sequence of images $I(t), t \geq 0$ captured by a moving camera.

In particular the goals of this lecture are to:

1. Study the properties of the projected image motion
2. Compute the optical flow as an approximation of the image motion
3. Design trackers to track objects in videos
Projected image motion

Projection from \( \mathbf{X} = (X, Y, Z)^T \)

\[
x = f Z X
\]  

(1)

Differentiate w.r.t. time to get projected motion

\[
\dot{x} = f \frac{Z}{Z^2} \dot{X}
\]

But the scene motion consists of translation \( \mathbf{V} \) and rotation \( \Omega \) relative to the camera. So:

\[
\dot{X} = (\dot{X}, \dot{Y}, \dot{Z})^T = \mathbf{V} + \Omega \times \mathbf{X}
\]

Hence writing \( \dot{Z} = \mathbf{X} \cdot \hat{z} \) and the using (1):

\[
\dot{x} = f \frac{Z}{Z^2} \left( \mathbf{V} + \Omega \times \mathbf{X} \right) - f \frac{Z}{Z^2} \mathbf{X}
\]

Note that:

Rotation

The depth \( Z \) does not appear with the rotation \( \Omega \):

The rotational component of motion is independent of the scene depth.

Translation

The depth \( Z \) appears with the translation \( \mathbf{V} \) as \( \mathbf{V}/Z \):

There is a depth/speed scaling ambiguity.

This means that one cannot tell whether something is large, far away and moving quickly or small, near-to and moving slowly.

Example: a simple motion alarm

In this example, the camera is rotating around a fixed position – hence \( \Omega \neq 0 \) but \( \mathbf{V} = 0 \). There is also an independently moving object in the scene.

The odometry from the camera axes provide \( \Omega \). Subtracting the motion due to the rotation (independent of the scene depth \( Z \)) reveals the moving object:

\[
\dot{x}_{\text{rotational}} = \Omega \times \mathbf{X} - f \frac{Z}{Z^2} \mathbf{X}
\]

Approaching and receding motion

In this example, the camera is approaching or receding but not rotating – hence \( V_x = V_y = 0 \), \( V_z \neq 0 \) and \( \Omega = 0 \).

The projected motion is a divergent/converged vector field, with a stationary point in the middle of the image \( x = 0, \dot{x} = 0 \).
Ego-motion and the focus of expansion

Generalising the previous example, if \( \mathbf{V} \neq \mathbf{0} \) but \( \mathbf{\Omega} = \mathbf{0} \), the motion field diverges/converges from a point, called focus of expansion.

To find the focus of expansion \( \mathbf{x} \), set \( \dot{\mathbf{x}} = \mathbf{0} \):

\[
0 = \dot{\mathbf{x}} = \frac{f}{Z} \mathbf{V} - \frac{V_z}{Z} \mathbf{x} \quad \Rightarrow \quad \mathbf{x} = \frac{f}{V_z} \mathbf{V}.
\]

Note:

▶ This equation says that you can treat the translational velocity \( \mathbf{V} \) as a ray, and find where it hits the image plane.
▶ This point in the image is called the focus of expansion (c.f. epipole)

Example: segmentation and motion alarm

Outline

Projected image motion

Optical flow and the aperture problem

Lucas-Kanade tracking
Computing the visual motion

- We have considered some uses of projected motion, but not yet discussed how to obtain motion field from imagery
- Three ways:
  1. Token based
     - compute corners in successive frames
     - match corners using some proximity and/or similarity scores
  2. Gradient based
     - directly from brightness values
     - great if corner detector or matcher unreliable
  3. Phase based
     - We will look at (2) in some detail
     - For (3) see work of (eg) Heeger, Fleet and others

Observable visual motion and projected motion

A word of warning:
- Compare a rotating snooker ball and a stationary light source, with a stationary ball and a moving light source:
  - moves, so visual motion is non-zero.
  - Scene still, but highlight moves.
  - Scene moves, but visual motion is zero.

In general, observable motion $\neq$ projected motion
- But often its the best we can do (and a reasonable approximation...)

Gradient-based visual motion

Regard the image as a sampling of a continuous irradiance function $I = I(x, y, t)$.

Follow a particular image patch over time.

In this case $x = x(t)$ and $y = y(t)$, and total $dl/dt$ must exists.

Using the chain rule:

$$\frac{dl}{dt} = \frac{\partial l}{\partial t} + \frac{\partial l}{\partial x} \frac{dx}{dt} + \frac{\partial l}{\partial y} \frac{dy}{dt}$$

But if $I$ is constant in the image patch, $dl/dt = 0$, so that

$$0 = \frac{\partial l}{\partial t} + \frac{\partial l}{\partial x} \frac{dx}{dt} + \frac{\partial l}{\partial y} \frac{dy}{dt}$$

Hence if we write

$$\nabla I = \begin{bmatrix} \frac{\partial l}{\partial x} \\ \frac{\partial l}{\partial y} \end{bmatrix}, \quad \mu = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix}$$

we arrive at the motion or brightness constraint equation

$$\nabla I \cdot \mu = -\frac{\partial l}{\partial t}$$

where $\mu$ is called the optical flow.
The aperture problem

- We have not recovered \( \mu \) ...
- ... but only the component of \( \mu \) along the
direction of \( \nabla I \).
- This edge-normal flow is

\[
v = \left( \frac{\mu \cdot \nabla I}{|\nabla I|} \right) \left( \frac{\nabla I}{|\nabla I|} \right) = - \left( \frac{\partial I}{\partial t} \right) \frac{\nabla I}{|\nabla I|^2}
\]

This failure to recover both components is called the **Aperture Problem**. Taking gradients means the information about the intensity is local, as if seen through an aperture.

Overcoming the aperture problem

Intuition tells us that there should be no aperture problem if there is a brightness gradient in “two
directions”; i.e. at **corners**.

Drawing inspiration from the Plessey corner detector, proceed as follows:

- Consider “tracking” a small image patch assumed to have translational image motion.
- Let

\[
E(u, v) = \sum_{i,j} \left[ \frac{\partial l(x + i, y + j)}{\partial x} u + \frac{\partial l(x + i, y + j)}{\partial y} v + \frac{\partial l(x + i, y + j)}{\partial t} \right]^2
\]

Overcoming the aperture problem /ctd

\[
E(u, v) \approx \sum_{i,j} \left[ \frac{\partial l(x + i, y + j)}{\partial x} u + \frac{\partial l(x + i, y + j)}{\partial y} v + \frac{\partial l(x + i, y + j)}{\partial t} \right]^2
\]

- Differentiating w.r.t. \( u, v \) yields

\[
\frac{\partial E}{\partial u} = \sum_{i,j} \left[ \frac{\partial l}{\partial x} u + \frac{\partial l}{\partial y} v + \frac{\partial l}{\partial t} \right] = 0
\]
\[
\frac{\partial E}{\partial v} = \sum_{i,j} \left[ \frac{\partial l}{\partial x} u + \frac{\partial l}{\partial y} v + \frac{\partial l}{\partial t} \right] = 0
\]

- Rewriting as

\[
\begin{bmatrix}
W \otimes (\frac{\partial l}{\partial x})^2
& W \otimes (\frac{\partial l}{\partial y})^2

W \otimes (\frac{\partial l}{\partial x} \frac{\partial l}{\partial y})
& W \otimes (\frac{\partial l}{\partial x} \frac{\partial l}{\partial t})
\end{bmatrix}
\begin{bmatrix}
u
v
\end{bmatrix}
= - \begin{bmatrix}
W \otimes (\frac{\partial l}{\partial x} \frac{\partial l}{\partial y})

W \otimes (\frac{\partial l}{\partial x} \frac{\partial l}{\partial t})
\end{bmatrix}
\]

\( H \) is the matrix describing the local autocorrelation. Thus:

- In a region of uniform brightness, \( H \) has rank 0 and we can’t solve for optic flow
- On an edge, \( H \) has rank 1, and we can find the **normal component** of optic flow
- At a corner, \( H \) has full rank (i.e. 2), and we can solve the equations to find both components of flow.

Computing the optic flow

Rewrite as

\[
\begin{bmatrix}
W \otimes (\frac{\partial l}{\partial x})^2
& W \otimes (\frac{\partial l}{\partial y})^2

W \otimes (\frac{\partial l}{\partial x} \frac{\partial l}{\partial y})
& W \otimes (\frac{\partial l}{\partial x} \frac{\partial l}{\partial t})
\end{bmatrix}
\begin{bmatrix}
u
v
\end{bmatrix}
= - \begin{bmatrix}
W \otimes (\frac{\partial l}{\partial x} \frac{\partial l}{\partial y})

W \otimes (\frac{\partial l}{\partial x} \frac{\partial l}{\partial t})
\end{bmatrix}
\]

\( H \) \( \mu = -b \)

H is the matrix describing the local autocorrelation. Thus:

- In a region of uniform brightness, \( H \) has rank 0 and we can’t solve for optic flow
- On an edge, \( H \) has rank 1, and we can find the **normal component** of optic flow
- At a corner, \( H \) has full rank (i.e. 2), and we can solve the equations to find both components of flow.
Computing the optic flow

Notes

- The first order approximation used only holds for very small values of $\mu$, $|\mu| < 1$.
- To recover bigger flows, use idea of a Gaussian pyramid:
  - Successively smooth and subsample images

Outline

Projected image motion

Optical flow and the aperture problem

Lucas-Kanade tracking

Visual tracking

The aim of visual tracking is to locate the position/pose of a target in a image successively over an extended sequence of images.

Target: require some “appearance” model
- image patch, colour histogram
- deformable contours (i.e. edges)
- 3D CAD model

Position:
- image position
- image position plus deformation parameters
- 3D translation and rotation
- full body pose of articulated object...

Let's start simply by tracking a planar image patch...

Tracking as optimisation

$\mathbf{f}(\mathbf{p})$

We seek the image position
$\mathbf{p} = [t_x \ t_y]^T$ which maximises the similarity to the template
- Exhaustive search possible but undesirable
- Better to do gradient ascent/descent
Lucas-Kanade tracker
The case of pure translation

▶ Formulate search as an optimisation problem using brightness constraint as our objective function.
▶ Optimising non-convex function so need to start near solution.
▶ Suppose starting point is \([t_x \ t_y]^\top\).

Minimise:

\[
E(\delta t_x, \delta t_y) = \sum_{x, y \in W} [l(x + t_x + \delta t_x, y + t_y + \delta t_y) - T(x, y)]^2
\]

w.r.t. \(\delta t_x, \delta t_y\).

Lucas-Kanade tracker
Pure translation

▶ Expanding to first order we obtain:

\[
E(\delta t_x, \delta t_y) \approx \sum_{x, y \in W} [l(x + t_x, y + t_y) + \nabla l \left( \begin{array}{c} \delta t_x \\ \delta t_y \end{array} \right) - T(x, y)]^2
\]

Taking partials w.r.t. \(\delta t_x, \delta t_y\) yields:

\[
2 \sum_{x, y \in W} \nabla \nabla^\top l \left( \begin{array}{c} \delta t_x \\ \delta t_y \end{array} \right) = 0
\]

Hence

\[
\begin{bmatrix} \sum_{x, y \in W} \nabla \nabla^\top l \end{bmatrix} \begin{bmatrix} \delta t_x \\ \delta t_y \end{bmatrix} = - \sum_{x, y \in W} \nabla \nabla^\top l \left( [l(x + t_x, y + t_y) - T(x, y)] \right)
\]

Note the (unsurprising) similarity to the optic flow solution.

▶ Set \(t_x \leftarrow t_x + \delta t_x, t_y \leftarrow t_y + \delta t_y\) and iterate until change negligible.

Lucas-Kanade tracker
Example