CMPUT 428: Optical Flow and Motion Detection; Image Registration

Roderick Lan

Contents

1 Lecture 1 - Jan 11

(Lec04 slides; Ch4 - 3DV; Ch8 - Szeliski)

1.1 Image Motion

Precursor to understand intensity exchange and what's geometrically happening in img.
Find relationship b/w pixel value changes and motion.

Example 1.1.1: Image Motion Video Example (slides)

Arrows point to general direction of motion.
Thresholds used (only compute for pixel changes that meet a certain threshold; account for shadows, etc.)

Overview 1.1.1: Image Motion

Somehow quantify frame-to-frame differences in image sequences.

1. Image intensity difference
2. Optic flow
3. 3-6 dim img motion computation

Relate two adjacent frames (small differences) by:

$$\text{Im}(x + \delta x, y + \delta y, t + \delta t) = \text{Im}(x, y, t)$$

(find local shift to match old frame with new frame; might not be exact match for different intensity values)

Motion is used for:

1. **Attention** - Detect and direct using eye/head motions
2. **Control** - Locomotion, manipulation, tools, etc.
3. **Vision** - Segment, depth, trajectory

1.1.1 Camera Reorientation

Subtract images to see changes in pixels.

1.2 Optic Flow Field

Vector field over img $[u, v] = f(x, y)$ where $u, v =$ vel vector, $x, y =$ Img position

Expln 1.2.1

FOE, FOC = Focus of Expansion, Contraction

$\mathrm{Im}(x+\delta x, y+\delta y, t+\delta t) = \mathrm{Im}(x,y,t)$

We get motion when camera is moved, but if we look strictly at the object, we cant tell if camera is moving or obj is moving.

We want to map points from diff perspectives

1.2.1 How to Compute Flow Vectors

Relate points. Go from image based measure to 3d point measure.
Possible Assumptions:

1. **color constancy -** point in H looks the same in I
2. **small motion -** points do not move very far

   This is the **optical flow** problem

Assume:

1. Image intensifies from obj points remain constant over times
2. Image displacement.motion small

Use median of pixel values.
If constant and changes small, can use Taylor expansion of intensity variation:

$\mathrm{Im}(x+\delta x, y+\delta y, t+\delta t) = \mathrm{Im}(x,y,t) + \frac{\partial\,\mathrm{Im}}{\partial x}\delta x + \frac{\partial\,\mathrm{Im}}{\partial y}\delta y \frac{\partial\,\mathrm{Im}}{\partial t}\delta t$

Using constancy assumption,

$0 = \frac{\partial\,\mathrm{Im}}{\partial x}\delta x + \frac{\partial\,\mathrm{Im}}{\partial y}\delta y \frac{\partial\,\mathrm{Im}}{\partial t}\delta t$

1.2.2 Solve for Optic Flow

Expln 1.2.2

$$\frac{\partial I}{\partial x} \approx \frac{I(x,y) - I(x-1,y)}{1}$$

$$\frac{\partial I}{\partial t} \approx I(x,y,t) - I(x,y,t-1)$$

Rewrite as dot product

$$-\frac{\partial \mathrm{Im}}{\partial t}\delta t = \left(\frac{\partial \mathrm{Im}}{\partial x}, \frac{\partial \mathrm{Im}}{\partial y}\right) \cdot \begin{bmatrix}\delta x \\ \delta y\end{bmatrix} = \nabla \mathrm{Im} \cdot \begin{bmatrix}\delta x \\ \delta y\end{bmatrix}$$

Flatten 2D images into vector → column vector where each column is stacked (top = first col, bottom = last col).

Use simultaneous equations.
Typically solve for motion in 2x2, 4x4, 8x8, or larger.
Dot product from Expln 1.2.2 is an over determined equation system;

$$d\text{Im} = \begin{bmatrix} \vdots \\ -\frac{\partial \text{Im}}{\partial t} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots \\ \frac{\partial \text{Im}}{\partial x} & \frac{\partial \text{Im}}{\partial y} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = M \cdot u$$

can be solved via:

LSS - $u = M \backslash d\text{Im}$

QR factorization - $(Q^\top M)u = Q^\top d\text{Im}$ $[Q, R] = qr(M), Q^\top M = R$

Expln 1.2.3: Conditions for Solvability

$A^\top A$ is invertible
$A^\top A$ entries not too small (noise)
$A^\top A$ well-conditioned
Study eigenvals; λ_1/λ_2 not too large ($\lambda_1 > \lambda_2$)

1.2.3 Aperture Problem

Gradients very large or very small; small λ_1, small λ_2
We want distinct edges, so large eigenvalues and gradients that are different w/ large magnitudes.
Looking at gradients more computationally stable

1.2.4 L1 Review

Use lots of tiles/small patches for 3d motion (small patches easy to track (?))

2 Lecture 2 - Jan 16

lec05bRegTrack2
For temporal difference (ie. frame 1 - frame 0), doesn't make sense if camera is moving.
Vector fields characterize vertical and horizontal motion (**u**)
image constancy - amount of light being reflected is constant; frame to frame problem

$$d\text{Im} = \begin{bmatrix} \vdots \\ -\frac{\partial \text{Im}}{\partial t} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots \\ \frac{\partial \text{Im}}{\partial x} & \frac{\partial \text{Im}}{\partial y} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = M \cdot u$$

Overdetermined equation,

Expln 2.0.1: Avoid Normal Equations

Ill conditioned; doubles the sensitivity

2.1 Image Registration: Related Areas

Optic flow is directly responsible for things like video compression (mpeg)
Parametric motion

- medical image-to-anatomy atlas registration
- aligining separate photos to form a panormaa img

Video Tracking

- (Lab 2)

2.1.1 Tracking

Three assumptions in Image Registration

1. Brightness consistency
   (a) img measurement (ie. brightness) in small region remains the same despite location change
2. Spatial coherence
   (a) neighboring points belong to same surface; have similar motion
3. Temporal persistence
   (a) img motion of surface patch changes gradually over time

2.2 Image Registration: 3 Applications

geometric space for 2d motion easy, 3d hard.
warped plane 8d instead of 2d (4 points + another 2d)
Image Alignment - $I(x), T(x)$ are two images
Tracking - $T(x)$ is a small (salient) patch around point p in first video img, $t = 0$. $I(x)$ is the img at $t + 1$
Optical Flow - $T(x), I(x)$ are patches of imgs at $t, t + 1$

2.3 Simple Approach (for translation; OF)

Minimize brightness difference

$E(u, v) = \sum_{x,y}(I(x + u, y + v) - T(x, y))^2$

$u, v = x, y$ shift

(Plot error landscape; can help understand issues ie. line $\rightarrow$ distinct one way, not the other)

higher res $\rightarrow$ harder to associate stuff, can use the natural variance of an object (ie. pattern in granite)

Problems:

not efficient

no subpixel accuracy

2.4 Lucas-Kanade Algorithm

$$E(u,v) = \sum_{x,y} (I(x+u, y+v) - T(x,y))^2$$

$I(x+u, y+v) \approx I(x,y) + uI_x + vI_y$

$$= \sum_{x,y} (I(x,y) - T(x,y) + uI_x + vI_y)^2$$

$$= \sum_{x,y} (d\text{Im}_{t(x,y)} + uI_x + vI_y)^2$$

Overdetermined system

$$-\text{Im}_t = \begin{bmatrix} \vdots \\ -\frac{\partial \text{Im}}{\partial t} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots \\ \frac{\partial \text{Im}}{\partial x} & \frac{\partial \text{Im}}{\partial y} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} = M \cdot \mathbf{u}$$

Draw optic flor vector $\mathbf{u}$

2.5 MPEG Compression

Image doesn't change much from frame to frame; can reliably predict next image via optic flow vectors for 2-10 frames. Space is 1/10th of what it would have been if new jpeg made for each frame.
Save initial compressed jpeg, use optic flow vectors to move patches around for next 10 frames instead of rendering new jpeg each time.
Movie compression about being able to do OF well; more computationally complex to make than to reproduce. (calculations slow, moving patches fast)

2.6 Pyramid Tracking

Look at OG img at high res, pixel shifts at higher resolutions correspond to smaller movements (opposite for lower resolution)

Take average or gaussian filter to create subsample img w/ fewer processing issues
Gaussian Pyramid and Laplace Pyramid
(pyramid formed via taking gaussian, etc.)

2.6.1 Coarse to fine OF

(improves performance for large motion)
Create pyramid; run up pyramid where full motion at pixel
run on coarse (low res); warp and upsample, run OF to refine motion (?)

2.7 Visual Tracking/Stabilization

Globally relating all frames - **large difference**

big challenge w/ tracking - worst case = task failed (rely on whole sequence going right)
compression worst case is just movie ugly for a few frames; recovers shortly after

2.8 Manually Selected Template in First Frame

Manually selected template $T = (I(t = 0))$
prediction - how far template moved a t-1; error from actual shift $(p + u)$ and predicted/template shift (p)
Optic flow - compute $\mathbf{u}$ b/w frames
Tracking - keep track of global motion $\mathbf{p}$
Similar to a pyramid, pyramid does this by artificially generating diff imgs

2.9 Image Registration

Transform a "source" img to match a "target" img Formulation
Similar to tracking/optic flow, but a bigger problem than tracking and motion.
Have to get global representation/projection (?).
Represent simiilarity wih a similarity score

depends on data

Different transformations: rigid, affine, projection
Non=rigid registration
can solve as OF problem w/ pyramid, more complex methods also available.