CMPUT 428: Tracking

Roderick Lan

Contents

1 Lecture 3 - Jan 18

lec05TrackIntro
Turn temporal intensity change into spatial intensity change. Similar to optic flow. Aka Registration Tracking.

Expln 1.0.1: Why Visual Tracking?

Computers became fast enough. Went from text based to windows based interfaces. Processors evolved, became faster and related technology became cheap enough to be used commercially by all. Applications for visual tracking became commercially viable.
Many applications for it, ie. motion control, RCI, measurement.

1.1 Applications - Watching a moving target

Camera and computer tracks motion of user and interprets this in an online interaction
Uses:

Security - ie. monitoring people moving in a subway station or store

Measurement - ie. speed, alert on colliding trajectory

1.2 Applications - Human-Computer Interfaces

Camera and computer can determine how things move in a scene over time
Can interact w/ menus, buttons and interpret physical interactions

1.3 Applications - Interpreting tasks and functinality

Use tracking to interpret

I. Physics of a mechanism

II. How a car is repaired

III. How a human cooks in a kitchen

1.4 Applications - Human-Machine Interfaces

Camera and computer tracks otions of human user, interprets this and machine/robot carries out a task.
Uses:

Remote manipulation

Service robotics (ie. for the handicapped and elderly)

1.5 Applications - Human-Human Interfaces

Camera and computer tracks motions and expressions of human user, interprets, codes, transmits to read at remote location. Uses:

> **Ultra low bandwidth video communication** (ie. reanimate video comm)
>
> **Handheld Video cellphone**

1.5.1 Bandwidth and Processing Req.

For TV camera:

> Binary - 1 bit * 640x480 * 30 = 9.2 Mbits/sec
>
> Grey - 1 byte * 640x480 * 30 = 9.2 MB/sec
>
> Color - 3 byte * 640x480 * 30 = 27.6 MB/sec

1.6 Video Processing

Almost all pixels change
Constancy - physical scene stays the same
Fundamental types:

I. Visual motion detection
   A. relate 2 adjacent frames; optical flow
   B. small differences

II. Visual Tracking
   A. globally relating all frames - large differences
   B. similar to motion detection

1.7 Types of tracking

Point Tracking - extract point (pixel loc) of a particular image feature in each img
Segmentation Based Tracking - define identifying region property (ie. color, texture, stat); repeatedly extract this region in each img
Stabilization Based Tracking - formulate img geometry + appearance equations and use these to acquire img transform properties for each img.

> relate later instances and morph back to original coordinate system

1.8 Point Tracking

Simplest technique; already commercialized in motion capture systems.
Features commonly LED's (visible or IR), special markers (reflective, pattern)

easily differentiable from rest of image

Detection - (ie. pick brightest pixel(s), cross correlate image to find the best match for a known pattern)
Not exactly real time (has some delay), delay is problem[1]

1.9 Tracking by Segmentation

Doesn't need special markers, uses properties/features in video (ie. color).
Features should be differentiable from rest of the image/frame.
Can represent statistically (after casting to RGB space/"cube") with ellipsoid; find mean and covariance in each axis (RGB). Color outliers black, inliers white; gets mask.

1.9.1 Gray Level Thresholding

Get histogram of intensities. If histogram bimodal, put threshold between the models (if you can clearly separate).
Leads to binary image (?), imperfectly represents image.

1.9.2 Compare: Natural Image Slide

Graph is white noise statistics

1.10 Region Tracking (Segmentation Based)

Select a "cure" $\gamma(I(x,y))$

foreground enhancement

background subtraction

Segment

threshold

"clean up"

can serve as preprocessing

Compute Region geometry (moment images)

centroid (first moment; central gravity)

orientation (second moment)

scale (second moment)

[1]might not be a problem for all point tracking systems

1.10.1 Regions

$c_p = P \to \mathbb{R}^3$ (color space) - intrisic coloration

homogenous region - c_p roughly constant

texturized region - c_p significant intensity gradients horizontally and veticall

contour - c_p local rapid change in contrast (edges; can determine where obj ends)[2]

Expln 1.10.1: Homogenous Color Region - Photometry

Look at img in rgb space, use color statistics. Find something in rgb space that is distinguished from everything else; easy to track in just rgb.
middle vals (diag) is grayscale; all images have

1.10.2 Color Representation

$c_R = R \to \mathbb{R}^3$ - irradiance (reflected light) of region R
Color representation

DRM (Klinker et al): if P is Lambertian has matte line and highlight line

user selects matte pixels in R

PCA fits ellipsoid $(S, R^\top, T)$ to matte cluster

color similarity $\gamma(I(x, y))$ defined by Mahalanobis dist.

$|S^{-1}R^\top(I(x, y) - T)| <$ pixels inside threshold/tracked obj

Color extension

$H_i =$ num of pixels in class i

Histograms are vectors of bin counts

Can use Fourier transform if going beyond color

[2]homog and texturized for inliers, contours for edge

2 Lecture 4 - Jan 23

lec05bRegTrack2
Registration Tracking (Lucas-Kanade)
optical flow $\rightarrow$ tracking: relate pixels from t to $t+1$ $\rightarrow$ relate from t to $t+n$
Geometry in 3d is 8DOF (not 3DOF)
Goal of tracking is to be able to compare template img w/ region in img (warp region to fit to template)

Simple Approach:

> Minimize brightness difference $E(u,v) = \sum_{x,y}(I(x+u,y+v) - I(x,y))^2$
> This only works when there is a distinct difference

Simple SSD bad, no subpixel accuracy (important in robotics) and not efficient.

2.1 Lucas-Kanade Algorithm

$$\begin{aligned} E(u,v) &= \sum_{x,y}(I(x+u,y+v) - I(x,y))^2 \\ &= \sum_{x,y}(I(x,y) - T(x,y) + uI_x + uI_y)^2 \end{aligned}$$

(LK OF - solve for translational motion of a patch (same as regular OF))

Reference img serves as template of obj
Tracking Cycle:

I. Predict - prior states predict new appearance

II. Image warping - generate normalized view; align to reference
Able to compare w/ reference

III. Model inverse - compute error from nominal

IV. State integration - apply correction to state

aka. Stabilization based tracking / Registration based tracking

2.1.1 Optic Flow to Tracking

same as optic flow but update "tracking state" $\vec{p}$

$$\vec{p} = \vec{p} + \vec{u}$$

$\vec{p}$ represents total motion; global displacement
For $t = 1, 2, \ldots$ $\text{Im}_t = \text{Im}(t, x+p) - \text{Im}(0, x)$ (temporal derivative is b/w current and first frame)
Round p to integer to avoid indexing issues

2.1.2 Tracking LK

init $\vec{p} = \vec{0}$, template T
For $t = 1 \ldots$

```
    Receive $I(t+1)$
    Compute $\text{dIm} = I(t+1, x+p) - T$
    Solve $-\text{Im}_t = Mu$ via $u = M \backslash \text{Im}_t$
    Update $p = p + u$
```

Doesnt account for resizing (zoom in, zoom out or obj go closer/away from camera)

2.2 Registration: Translation to Warp

Translation $\mathbf{u} \rightarrow$ Warp $\mathbf{w}(x,p)$

$$\begin{aligned} \text{dIm} &= I(t+1, x + [p_u \; p_v]^\top) - T \\ &= I(t+1, w(x,p)) - T \text{ where } w = x + p \end{aligned}$$

$$\begin{aligned} w(x,p) &= R(p_3) + \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \\ w(x,p) &= \begin{bmatrix} \cos p_3 & \sin p_3 \\ \sin p_3 & \cos p_3 \end{bmatrix} + \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \text{ Transl + Rot } (R) \text{ SE2, 3DOF} \\ w(x,p) &= p_4 R(p_3) + \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \text{ Transl + Rot + Scaling } (p_y) \text{ 4DOF; } R \text{ is } 2 \times 2 \\ w(x,p) &= Ax + \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} p_3 & p_4 \\ p_5 & p_6 \end{bmatrix} x + \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \text{ Affine 6DOF} \end{aligned}$$

Let $= \begin{bmatrix} x \\ y \\ \mathbb{1} \end{bmatrix} \sim \begin{bmatrix} x \\ y \\ w \end{bmatrix}$ (3 dimensional vector)

$$w(x,p) = Hx = \begin{bmatrix} p_1 & \cdots & \cdot \\ \vdots & \ddots & \vdots \\ \cdot & \cdots & p_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} \text{ } H \text{ is a } 3 \times 3 \text{ matrix}$$

2.2.1 Intensity based tracking w/ LK

Find image warping function parameters s.t.

$$\mathcal{I}(\mathbf{w}(\mathbf{x}; \mathbf{p}_i)) = \mathcal{I}_T(\mathbf{p}_i)$$

Non-linear minimization problem;
Use Gauss-Newton to minimize, applies first order Taylor series approx

LK - Approx by linearization

Jacobian matrix of current and gradient imgs

2.2.2 Image Derivatives

Can get derivs from comparing shifts, warps/rotates
We want M to be linearly independent (numerical stability)

Reference 2.2.1: Related Chapter

Read ch5. in Heath textbook (NLLS)

2.2.3 Planar Texture Variability; Affine Variability

Affine Warp Func:

$$\begin{bmatrix} u_w \\ v_w \end{bmatrix} = \mathcal{W}_a(\mathbf{p}, \mathbf{a}) = \begin{bmatrix} \mathbf{a}_3 & \mathbf{a}_4 \\ \mathbf{a}_5 & \mathbf{a}_6 \end{bmatrix} \mathbf{p} + \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix}$$

Image Variability:

$$\sum_{i=1}^{6} \frac{\partial}{\partial a_i} \mathbf{I}_w \Delta a_i = \left[\frac{\partial I}{\partial u}, \frac{\partial I}{\partial v}\right] \begin{bmatrix} \frac{\partial u}{\partial a_1} & \cdots & \frac{\partial u}{\partial a_6} \\ \frac{\partial v}{\partial a_1} & \cdots & \frac{\partial v}{\partial a_6} \end{bmatrix} \begin{bmatrix} \Delta a_1 \\ \vdots \\ \Delta a_6 \end{bmatrix}$$

When we track, try to figure out ys

2.2.4 Planar Texture Variability; Projective Variability

Dont need more than 3 col when it is a plane
(finish)

2.3 LK Iterative Minimization

Minimize sum of squared differences

$$f(\Delta \mathbf{x}) = \frac{1}{2} \|\mathbf{y}(\mathbf{0}) + \mathbf{J}_{\mathbf{y}}(\mathbf{0}) \Delta \mathbf{x}\|^2$$

parameter inc. has closed form solution (Gauss-Newton)

$$\Delta \mathbf{x} = -\mathbf{H}^{-1} \mathbf{J}_{\mathbf{y}}(\mathbf{0})^\top \mathbf{y}(\mathbf{0})$$

$$\mathbf{H} = \mathbf{J}_{\mathbf{y}}(\mathbf{0})^\top \mathbf{J}_{\mathbf{y}}(\mathbf{0})$$

Update param approx

$$\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \Delta \mathbf{x}$$

(using multiplication instead of addition can help stability, was able to calculate for affine with this switch)
Slide 34-40

error akin to temporal difference

2.4 Registration based Tracking

$$\mathbf{p_t} = \arg\max_{\mathbf{p}} f(\mathbf{I_0}(\mathbf{x}), \mathbf{I_t}(\mathbf{w}(\mathbf{x}, \mathbf{p})))$$

Tracking not perfect; Estimates lag a bit, could iterate more on same image to improve
Need tracking for 3d, (pixel correspondence and relating b/w frames)

3 Lecture 5 - Jan 25

lec05cCRV16IROS17ModularShort22.pdf

Reference 3.0.1: Relevant paper

Modular Decomposition and Analysis of Registration Based Trackers (MTF)

Expln 3.0.1: Registration Based Tracking

Find the optimal warp or geometric transformation that registers each image in a sequence with the template

$$\mathbf{p_t} = \arg\max_{\mathbf{p}} f(\mathbf{I_0}(\mathbf{x}), \mathbf{I_t}(\mathbf{w}(\mathbf{x}, \mathbf{p})))$$

”align to find best overlap given some warp”
$\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ (vector of coordinates in patch) $\mathbf{x}_k = [x_k, y_k]^\top \in \mathbb{R}^2$
$\mathbf{I}(\mathbf{x}) = [I(x_1, y_1), \ldots, I(x_N, y_N)]^\top \in \mathbb{R}^N, I(x, y) : \mathbb{R}^2 \mapsto \mathbb{R}$
$\mathbf{p} = [p_1, \ldots, p_S]$ (parameters), S : DOF of image motion - ”state space model”
$\mathbf{w} : \mathbb{R}^2 \times \mathbb{R}^S \mapsto \mathbb{R}^2$
$f : \mathbb{R}^N \times \mathbb{R}^N \mapsto \mathbb{R}$

Points dont have to be on a grid, can be arbitrarily placed.

3.1 Motivation of Reg Tracking

Process in registration based tracking has become fragmented since LK, intuitive way exists to relate these by decomposing the tracking task into three modules. (most contributinos confined to only 1-2 of these modules)
Modular Tracking Framework (**MTF**) to easily plug in new methods.

3.2 Modular Decomposition

Appearance model
Search Model - maximize
(finish)

3.3 State Space Model

$$\mathbf{p_t} = \arg\max_{\mathbf{p}} f(\mathbf{I_0}(\mathbf{x}), \mathbf{I_t}(\mathbf{w}(\mathbf{x}, \mathbf{p})))$$

Warping function or geometric transformation that represents set of allowable image motions of the object.
Embodies constraints place on the warp parameter space (search efficiency, alignment precision).
Includes DOF of allowed motion, actual paramterization of warping function (naive method of searching just an exhaustive search)

Expln 3.3.1: Registration: Trans → Warp

Find params of warping function s.t.

$$\mathcal{I}(\mathbf{w}(\mathbf{x}; \mathbf{p}_i)) = \mathcal{I}_T(\mathbf{p}_i)$$

for all template points

Overview 3.3.1: Homography

Planar Projective Warping

$$x_i' = Hx_i$$

Homography similar to affine; 8DOF

More DOF not necessarily better, more complex models (higher DOF), can be less stable.

3.4 Search Method

$$\mathbf{p}_t = \arg\max_{\mathbf{p}}$$

Optimization method that maximizes AM similarity functions 2 categories

I. **Gradient Descent -** Newton or Gauss-Newton method

II. **Stochastic Search -** Sampling based

3.4.1 Simple img reg w/ SSD error norm

Exhaustive Search (not really needed if near minimum, mathematical structure exists)
Solve w/ Gauss-Newton optimization

3.4.2 Homogenous Coordinates: Translate 2D point

old: $x' = x + dx$
new: $x' = M \cdot dx$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3.4.3 Search method examples slide

inverse additiive not good
inverse compositional much more efficient than otherse
multiplication better than addition

3.4.4 Jurie and Dhome Learning Phase

Apply set of random transformations/warps to initial template. If we do enough warps, then we can do exhaustive search and find/output nearest warp.
Warp with Δx warp parameters. Stack warps into matrix.

$$\mathbf{X} = \mathbf{AY}$$

or

$$\Delta\mathbf{X} = \mathbf{AY}$$

($\mathbf{A}$ is Jacobian in $\Delta\mathbf{X}$ case)
Find average linear predictor that speeds up Xs.
Jacobian works best near target
Learn linear ppredictor w/

$$\mathbf{A} = \mathbf{XY}^\top(\mathbf{YY}^\top)^{-1}$$

or

$$\mathbf{A}' = \mathbf{Y}' \backslash \mathbf{X}'$$

3.4.5 Example Search Methods

Nearest Neighbor Search (NN)

generate sample by warping $\mathbf{I_0}(\mathbf{x})$

(can create k-deep trees)

inverse method $\rightarrow$ tree precomputed

Particle Filter (PF)

generate samples, compute weight for each; estimate Δp as weighted avg

Expln 3.4.1: Issues with Affine

$$x' = Ax + b$$

A and b not similar

3.5 Appearance Model

$$\mathbf{I_0} \text{ and } \mathbf{I_t}$$

Similarity measure between two image patches:

candidate warped patch from current img

tempalte extracted from initial img

2 main categories

I. SSD like

II. Robust

3.5.1 Appearance Model - Examples

SCV and RSCV - statistical measures
ZNCC - correlation and convolution good at measuring similarity
NCC performs very well on regular images, not so much on irregular imgs. (?)
MI good for irregular imgs, ie. mapping map to sat img (?)
CCRE - more complex statistical measure
NCC generally the best

3.5.2 Texture

High texture region desired so we can find distinct points.

rocks are trackable (grain pattern)

3.6 System Design

$$\mathbf{p_t} = \arg\max_{\mathbf{p_s},\mathbf{p_a}} f(\mathbf{I_0}(\mathbf{x}), \mathbf{I_t}(\mathbf{w}(\mathbf{x}, \mathbf{p_s})), \mathbf{p_a})$$

Choosing search methods, appearance models, state space models
Choosing to combine trackers (ie. sample based stochastic tracker + newton based tracker; breadth + precision).
Adding illumination models.

more hz = more trackers
inverse methods faster than none inverse methods (due to getting rid of/changing + and Δ)
inverse methods let us precompute (?)