Visual Tracking (briefly)

Computer Vision I

CSE252A

Lecture 18

Announcements

- HW 4 has been assigned – due next Thursday.
- Final exam: Friday, March 24, 3:00-6:00, here

Lukas Kanade: Integrate over a window

Assume a single velocity (u,v) for all pixels within an image patch, we can write a cost function indicating for a given (u,v), the extent that the optical flow equation is violated over a window Ω

$$E(u,v)=\sum_{x,y\in\Omega}\left(I_x(x,y)u+I_y(x,y)v+I_t\right)^2$$

$$\frac{dE(u,v)}{du}=\sum 2I_x\left(I_xu+I_yv+I_t\right)=0$$

$$\frac{dE(u,v)}{dv}=\sum 2I_y\left(I_xu+I_yv+I_t\right)=0$$

Condition for a local minimum

Solve with:

$$\begin{bmatrix}\sum I_x^2 & \sum I_xI_y\\ \sum I_xI_y & \sum I_y^2\end{bmatrix}\begin{pmatrix}u\\ v\end{pmatrix}=-\begin{pmatrix}\sum I_xI_t\\ \sum I_yI_t\end{pmatrix}$$

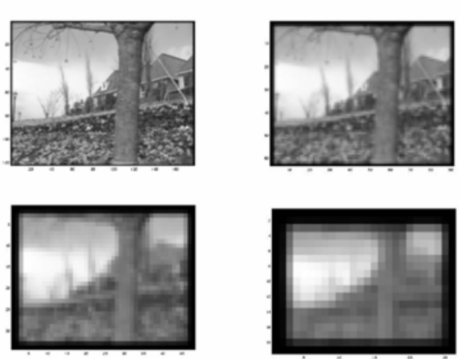
On the LHS: sum of the 2x2 outer product tensor of the gradient vector

$$\left(\sum \nabla I\nabla I^T\right)\vec{U}=-\sum \nabla II_t$$

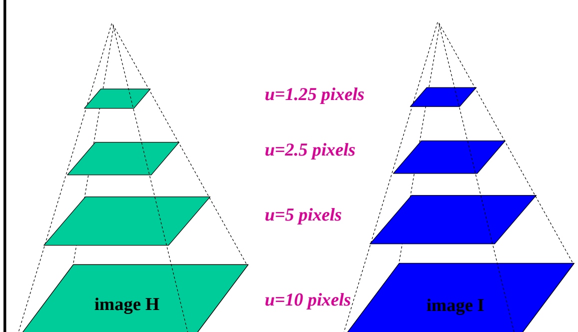
Revisiting the small motion assumption

- Is this motion small enough?
  - Probably not—it's much larger than one pixel (2nd order terms dominate)

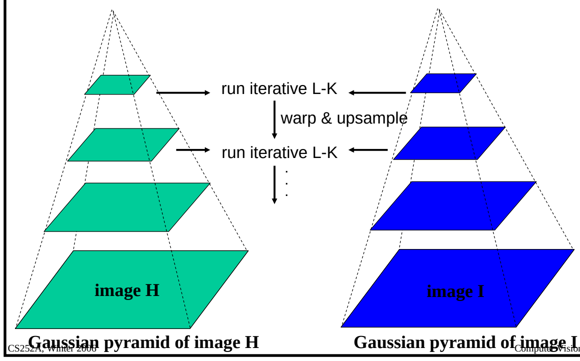
Pyramid / "Coarse-to-fine"



Coarse-to-fine optical flow estimation



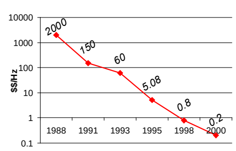
Coarse-to-fine optical flow estimation



Multi-resolution Lucas Kanade Algorithm

- Compute 'simple' LK at highest level
- At level i
  - Take flow u_{i-1}, v_{i-1} from level i-1
  - bilinear interpolate it to create u_i^*, v_i^* matrices of twice resolution for level i
  - multiply u_i^*, v_i^* by 2
  - compute f_t from a block displaced by $u_i^*(x,y)$, $v_i^*(x,y)$
  - Apply LK to get $u_i'(x, y)$, $v_i'(x, y)$ (the correction in flow)
  - Add corrections u_i' v_i', *i.e.* $u_i = u_i^* + u_i'$, $v_i = v_i^* + v_i'$.

Visual Tracking



Visual Tracking

Main Challenges

1. 3-D Pose Variation
2. Occlusion of the target
3. Illumination variation
4. Camera jitter
5. Expression variation
   etc.

[Ho, Lee, Kriegman]

Main tracking notions

- State: usually a finite number of parameters (a vector) that characterizes the "state" (e.g., location, size, pose, deformation of thing being tracked.
- Dynamics: How does the state change over time? How is that changed constrained?
- Representation: How do you represent the thing being tracked
- Prediction: Given the state at time *t-1*, what is an estimate of the state at time *t*?
- Correction: Given the predicted state at time *t*, and a measurement at time *t*, update the state.
- Initialization – what is the state at time t=0?

What is state?

- 2-D image location, $\Phi=(u,v)$
- Image location + scale $\Phi=(u,v,s)$
- Image location + scale + orientation $\Phi=(u,v,s,\theta)$
- Affine transformation
- 3-D pose
- 3-D pose plus internal shape parameters (some may be discrete).
  - e.g., for a face, 3-D pose +facial expression using FACS + eye state (open/closed).
- Collections of control points specifying a spline
- Above, but for multiple objects (e.g. tracking a formation of airplanes).
- Augment above with temporal derivatives $(\phi, \dot{\phi})$

State Examples:

- object is ball, state is 3D position+velocity, measurements are stereo pairs
- object is person, state is body configuration, measurements are frames
- What is state here?

Example: Blob Tracker

- From input image *I(u,v)* (color?) at time *t*, create a binary image by applying a function f(I(u,v)).
- Clean up binary image using morphological operators
- Perform connected component exploration to find "blobs." – connected regions.
- Compute their moments (mean and covariance of coordinates of region), and use as state
- Using state estimate from *t-1* and perform "data association" to identify state in from t.

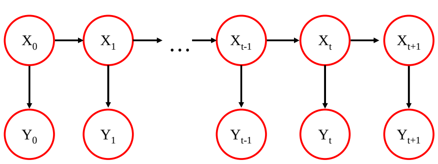
Blob Tracking in IR Images

- Threshold about body temperature
- Connected component analysis
- Position, scale, orientation of regions
- Temporal coherence

Tracking: Probabilistic framework

- Very general model:
  - We assume there are moving objects, which have an underlying state X
  - There are measurements Y, some of which are functions of this state
  - There is a clock
    - at each tick, the state changes
    - at each tick, we get a new observation

Tracking State



- Instead of "knowing state" at each instant, we treat the state as random variables X_t characterized by a pdf $P(X_t)$ or perhaps conditioned on other Random Variables e.g., $P(X_t \mid X_{t-1})$, etc.
- The observation (measurement) *Yt* is a random variable conditioned on the state $P(Y_t \mid X_t)$
- Generally, we don't observe the state – it's hidden.

Three main steps

- **Prediction:** we have seen $\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{i-1}$ — what state does this set of measurements predict for the i'th frame? to solve this problem, we need to obtain a representation of $P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1})$.
- **Data association:** Some of the measurements obtained from the i-th frame may tell us about the object's state. Typically, we use $P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_{i-1} = \boldsymbol{y}_{i-1})$ to identify these measurements.
- **Correction:** now that we have $\boldsymbol{y}_i$ — the relevant measurements — we need to compute a representation of $P(\boldsymbol{X}_i | \boldsymbol{Y}_0 = \boldsymbol{y}_0, \ldots, \boldsymbol{Y}_i = \boldsymbol{y}_i)$.

Tracking Modalities

- Color
  - Histogram [Birchfield 1998; Bradski 1998]
  - Volume [Wren *et al*., 1995; Bregler, 1997; Darrell, 1998]
- Shape
  - Deformable curve [Kass *et al*.1988]
  - Template[Blake *etal*. 1993; Birchfield 1998]
  - Example-based [Cootes *et al*., 1993; Baumberg & Hogg, 1994]
- Appearance
  - Correlation [Lucas & Kanade, 1981; Shi & Tomasi, 1994]
  - Photometric variation [Hager & Belhumeur, 1998]
  - Outliers [Black *et al*., 1998; Hager & Belhumeur, 1998]
  - Nonrigidity [Black *et al*., 1998; Sclaroff & Isidoro, 1998]
- Motion
  - Background model [Wren *et al*., 1995; Rosales & Sclaroff, 1999; Stauffer & Grimson, 1999]
  - Optical flow [Cutler & Turk]
  - Egomotion [Sawhney & Ayer, 1996; Irani & Anandan, 1998]
- Stereo
  - Blob correlation [Azarbayejani & Pentland, 1996]
  - Disparity map [Kanade *et al*., 1996; Konolige, 1997; Darrell *et al*., 1998]

Color Blob tracking

- **Color-based tracker gets lost on white knight: Same Color**

Snakes: Active Contours

- Contour C: continuous curve on smooth surface in $\Re^3$
- Snake S: projection of C to image
- Curve types
  - Edge between regions on surface with contrasting properties
  - Line that contrasts with surface properties on both side
  - Silhouette of surface against contrasting background
- General Algorithm:
  - Perform edge detection
  - Fit parametric or non-parametric curve to data

Snakes: Basic Approach

- Parameterize a closed contour

$$\mathbf{Q} = (q_0^x .. q_n^x, q_0^y .. q_n^y)$$

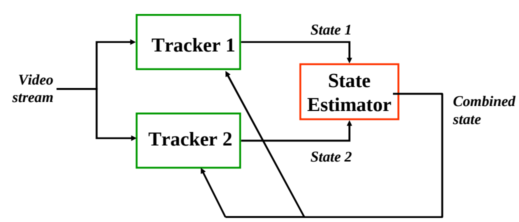
$$\mathbf{U}(s) = \begin{pmatrix} \mathbf{B}(s)^t & 0 \\ 0 & \mathbf{B}(s)^t \end{pmatrix}$$

- $\mathbf{r}(s) = \mathbf{q}^t\mathbf{B}(s)$ or $\mathbf{r}(s) = \mathbf{U}(s)\mathbf{Q}$
- Given a predicted state **q,** search radially for edges
- Solve a least squares problem for new state

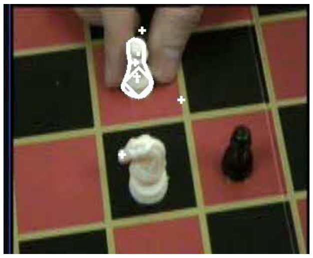
Tracker Composition: Only Shape (Snakes)

- **Geometry-based tracker gets lost on black pawn: Same shape**

Tracker Composition

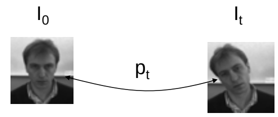


Tracker Composition: Color and Shape



- **Combining Trackers => Robustness**
- **Trackers in video, IR and range**

Visual Tracking using regions



Variability model: $I_t = g(I_0, p_t)$

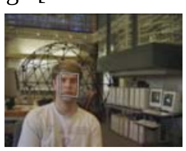
Incremental Estimation: From I_0, I_{t+1} and p_t compute Δp_{t+1}

$$|| I_0 - g(I_{t+1}, p_{t+1}) ||^2 \implies \min$$

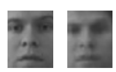
Tracking using Textured Regions

- Mean intensity difference between **I** and affine warp of template image [Shi & Tomasi, 1994]

Template I_R



Tracked state I_c



$\mathbf{I}_R$ $\mathbf{I}_C$

$$\psi_{tregion}(x,y) = \sum_{(x,y)\in W} (\mathbf{I}_R(x,y) - \mathbf{I}_C(x,y))^2$$

$|\mathbf{I}_R - \mathbf{I}_C|$

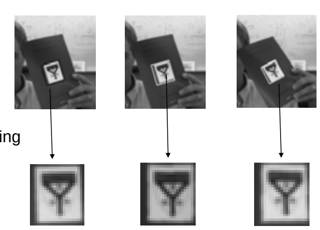
Image Warping

- Warping is a change of coordinates:
  J(u,v) = I(f(u,v,p),g(u,v,p))
- Always prefer to warp to destination to avoid gaps
- Two interpolation schemes
  - nearest neighbor
  - bilinear
- J(**u**) = I(A **u**)
- Note that we can "unroll" the loop to avoid the matrix multiply
- For much of tracking, nearest neighbor works well

Template tracking: Planar Case

Planar Object => Affine motion model: $u'_i = A u_i + d$

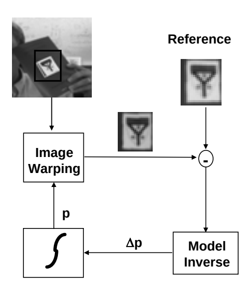
Warping



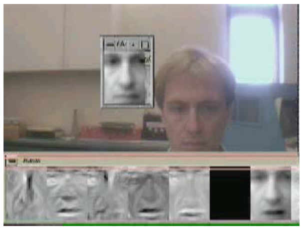
$I_t = g(p_t, I_0)$

Hager/Toyama: Tracking Cycle

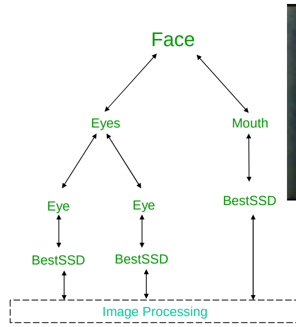
- Prediction
  - Prior states predict new appearance
- Image warping
  - Generate a "normalized view"
- Model inverse
  - Compute error from nominal
- State integration
  - Apply correction to state



SSD Tracking



XVision: A tracking System



Composition of Primitive Trackers

Recognition

Computer Vision I

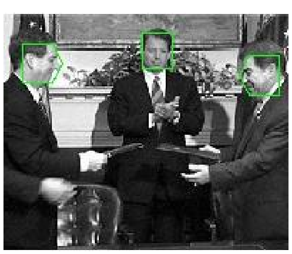
CSE252A

Lecture 18

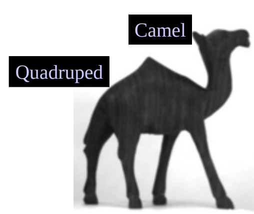
Recognition

Given a database of objects and an image determine what, if any of the objects are present in the image.

Recognition



Given a database of objects and an image determine what, if any of the objects are present in the image.

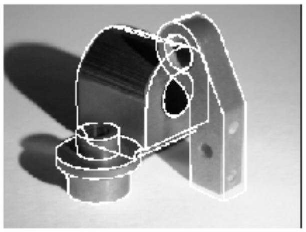


Problem:

Recognizing instances

Recognizing categories

Recognition

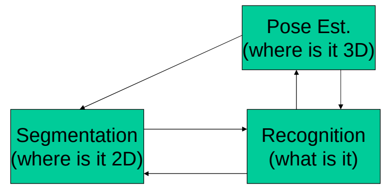


Given a database of objects and an image determine what, if any of the objects are present in the image.

Object Recognition: The Problem

Given: A database D of "known" objects and an image I:

1. Determine which (if any) objects in D appear in I
2. Determine the pose (rotation and translation) of the object



WHAT AND WHERE!!!

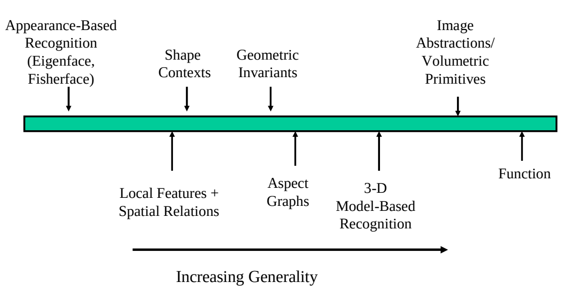
Recognition Challenges

- Within-class variability
  - Different objects within the class have different shapes or different material characteristics
  - Deformable
  - Articulated
  - Compositional
- Pose variability:
  - 2-D Image transformation (translation, rotation, scale)
  - 3-D Pose Variability (perspective, orthographic projection)
- Lighting
  - Direction (multiple sources & type)
  - Color
  - Shadows
- Occlusion – partial
- Clutter in background -> false positives

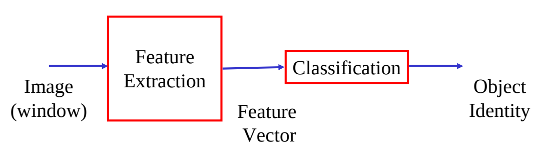
Object Recognition Issues:

- How general is the problem?
  - 2D vs. 3D
  - range of viewing conditions
  - available context
  - segmentation cues
- What sort of data is best suited to the problem?
  - Whole images
  - Local 2D features (color, texture,
  - 3D (range) data
- What information do we have in the database?
  - Collection of images?
  - 3-D models?
  - Learned representation?
  - Learned classifiers?
- How many objects are involved?
  - small: brute force search
  - large: ??

A Rough Recognition Spectrum



Sketch of a Pattern Recognition Architecture



Example: Face Detection

- Scan window over image.
- Classify window as either:
  - Face
  - Non-face