Lecture 23: Motion Vectors

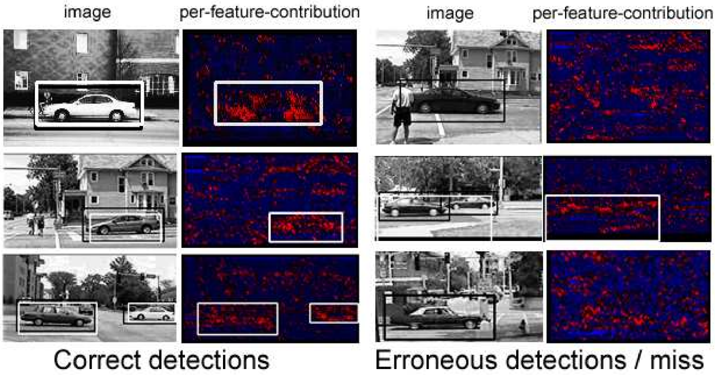
ECE 417: Multimedia Signal Processing
Mark Hasegawa-Johnson

University of Illinois

11/13/2017

Outline

Object Detection



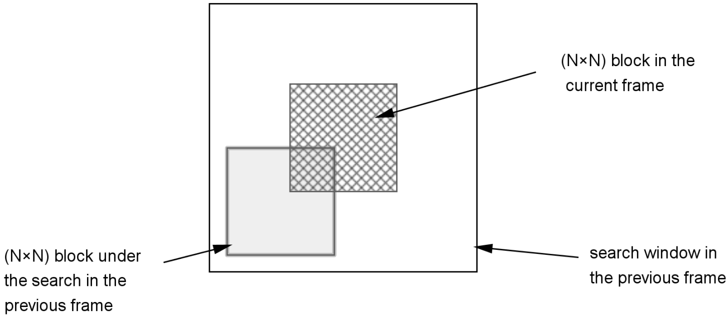
Zhuang, Zhou, Hasegawa-Johnson & Huang, "Efficient Object Localization with Gaussianized Vector Representation," IMCE 2009

Object Detection vs. Motion Vectors

- You already know that an "object detector" looks for rectangles in the image that match some pre-defined appearance classifier.
- A "motion vector" is calculated by finding a correspondence between rectangles at time t, and rectangles at time $t-1$, where t is the frame index in a video signal.

$$\vec{v} = \arg\min \| I(x, y, t) - I(x - v_1, y - v_2, t - 1) \|$$

Motion Vectors



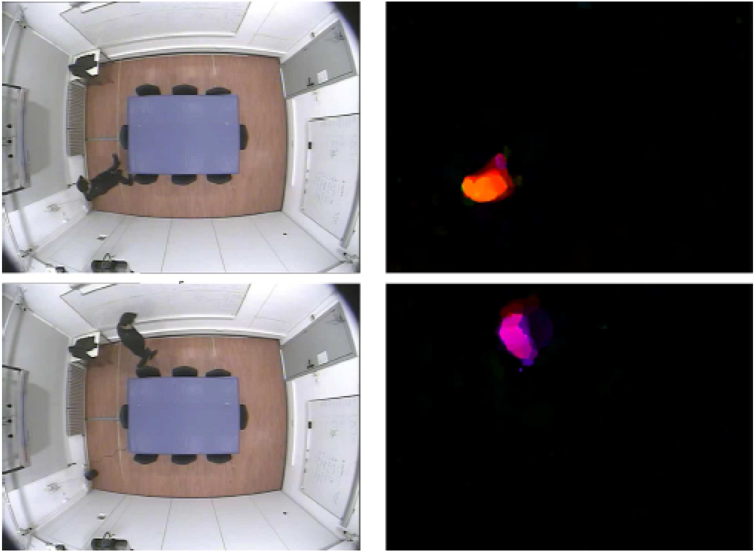
Applications of Motion Vectors: Video Coding

- Motion vectors were originally invented for video coding.
- For example, suppose that the pixels in a video are coded with 8 bits normally, $0 \leq I(x, y, t) \leq 255$.
- Suppose you can find $\vec{v}$ so that $-8 \leq \Delta I(x, y, t) \leq 7$, where

$$\Delta I(x, y, t) = I(x, y, t) - I(x - v_1, y - v_2, t - 1)$$

- Then you can code $\Delta I(x, y, t)$ using just 4 bits/pixel, instead of 8 bits/pixel.

Applications of Motion Vectors: Object Tracking

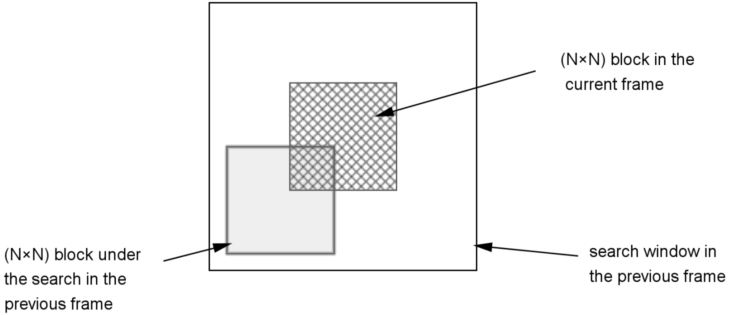


Huang, Zhuang & Hasegawa-Johnson, 2011, Fig. 1

Applications of Motion Vectors: Person Detection, Surveillance

Outline

The Block-Match Algorithm



The Block-Match Algorithm

For each position, $\vec{r}$, in the frame at time t, we find a position $\vec{r}-\vec{v}$ at time $t-1$ that best matches it. Here "best" is defined as minimum distance, e.g.,

$$\vec{v}(\vec{r}) = \arg\min \frac{1}{N^2} \sum_{\vec{m}=(1,1)}^{(N,N)} |I(\vec{r}+\vec{m},t) - I(\vec{r}+\vec{m}-\vec{v}(\vec{r}),t-1)|$$

or mean-squared error (MSE):

$$\vec{v}(\vec{r}) = \arg\min \frac{1}{N^2} \sum_{\vec{m}=(1,1)}^{(N,N)} |I(\vec{r}+\vec{m},t) - I(\vec{r}+\vec{m}-\vec{v}(\vec{r}),t-1)|^2$$

where $\vec{r}=[r_1,r_2]$ is the location of a size $N \times N$ block, and $\vec{v}(\vec{r})$ is the motion vector.

Computational Complexity

- The block-match algorithm is kind of computationally expensive.
- You have to check N^2 different possible velocity vectors...
- ...each of which requires N^2 additions...
- ...for every block in the image.
- Various sub-optimal search algorithms try to get close to finding the perfect velocity vector, without actually evaluating all N^2 of the possible velocity vectors.

Three-Step Search

1. Examine the 17 velocity vectors of the form $\vec{v} \in [\{-S, 0, S\}, \{-S, 0, S\}]$, where $S \in \{0, 1, 4\}$. Find the best $\vec{v}$ in this set.
2. Move the origin to the value of $\vec{v}$ with best match.
3. Repeat this process twice more.

Three-Step Search

	A	B	C	D	E	F	G	H	I
1	1				1				1
2									
3									
4				1	1	1			
5	1			1	0	1			1
6				1	1	1			
7									
8									
9	1				1				1
10									
11									

Outline

Optical Flow

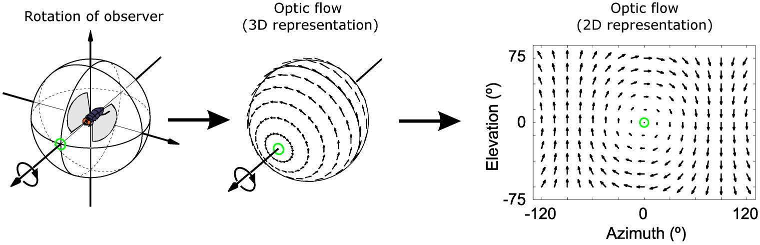
The basic idea of "optical flow" is that we treat the motion vector as a function of **continuous** position, rather than **discrete** position.

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + \text{H.O.T}$$

In practice, we can measure the derivatives using some kind of discrete approximations, like

$$\begin{aligned}
\frac{\partial I}{\partial x} &= I(x,y,t) - I(x-1,y,t) \\
\frac{\partial I}{\partial y} &= I(x,y,t) - I(x,y-1,t) \\
\frac{\partial I}{\partial t} &= I(x,y,t) - I(x,y,t-1)
\end{aligned}$$

Optical Flow



Adapted from PloS Biology (CC licensed) article: Huston SJ, Krapp HG, 2008 Visuomotor Transformation in the Fly Gaze Stabilization System. PLoS Biol 6(7): e173. doi:10.1371/journal.pbio.0060173.

Optical Flow

Suppose we require that

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + \text{H.O.T}$$

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t)$$

That gives

$$0 \approx \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t$$

Optical Flow

$$0 = \frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t}$$

$$= \frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t}$$

Re-arranging, we get **the optical flow equation:**

$$\nabla I^T \vec{v} = -b$$

where we define $b = \frac{\partial I}{\partial t}$, and

$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]^T$$

Optical Flow

$$\textbf{OpticalFlow}: \ \nabla I^T \vec{v} = -b$$

Optical flow algorithms try to minimize b^2, by making different assumptions about $\vec{v}$.

- Lucas-Kanade assumes that $\vec{v}$ and $\partial I/\partial t$ are constant within a block.
- Horn-Schunck minimizes $\|\nabla \vec{v}\|$, assuming that $\partial I/\partial t = 0$.

Lucas-Kanade Algorithm

Suppose we have a block of N pixels, in which we can measure $\nabla I(\vec{r}_i)$, for $1 \leq i \leq N$.

- Assume that $\vec{v}(\vec{r}_i) = \vec{v}$, a constant.
- Assume that we can measure, at every pixel $\vec{r}_i$, the measurements $\frac{\partial I(\vec{r}_i)}{\partial x}$, $\frac{\partial I(\vec{r}_i)}{\partial y}$, and $\frac{\partial I(\vec{r}_i)}{\partial t}$.
- Then $\nabla I^T \vec{v} = -\frac{\partial I}{\partial t}$ gives

$$\left[\begin{array}{cc} \frac{\partial I(\vec{r}_1)}{\partial x} & \frac{\partial I(\vec{r}_1)}{\partial y} \\ \vdots & \vdots \\ \frac{\partial I(\vec{r}_N)}{\partial x} & \frac{\partial I(\vec{r}_N)}{\partial y} \end{array}\right] \vec{v} = - \left[\begin{array}{c} \frac{\partial I(\vec{r}_1}{\partial t} \\ \vdots \\ \frac{\partial I(\vec{r}_N)}{\partial t} \end{array}\right] + \vec{e}$$

- We can write that as

$$A\vec{v} = -\vec{b}$$

Lucas-Kanade Algorithm

Lucas-Kanade gives us

$$A\vec{v} \approx -\vec{b}$$

To minimize the error of the approximation, we want

$$\vec{v} = \arg\min E$$

where

$$E = \|A\vec{v} + \vec{b}\|^2$$

The solution is given by the pseudo-inverse:

$$\vec{v} = -(A^T A)^{-1} A^T \vec{b}$$

Outline

Conclusions

- Motion vectors can be used for video coding, object tracking, and person detection.
- Motion vectors can be computed using the block-matching algorithm, but computational complexity is kind of high, because we have to search for the best possible velocity vector out of an $N \times N$ set of possible velocity vectors.
- Optical flow computes a velocity vector at every pixel (actually, it treats the velocity vector as a continuously varying function of position)
- The optical flow equation, $\nabla I^T \vec{v} = -\frac{\partial I}{\partial t}$, is under-determined: both $\vec{v}$ and $\frac{\partial I}{\partial t}$ are unknown.
- The Lucas-Kanade algorithm assumes that both unknowns are constant over a local block, and solves using the pseudo-inverse.