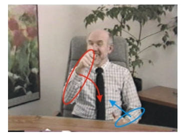
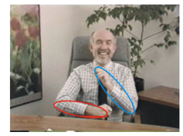
Image Processing:
4. Optical Flow

Aleix M. Martinez
aleix@ece.osuedu

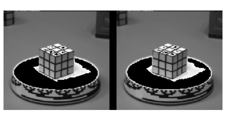
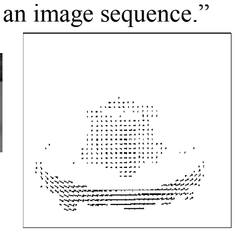
Motion estimation

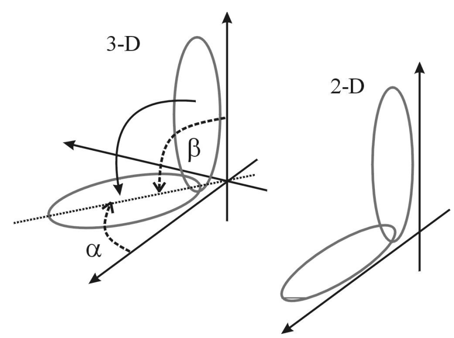
- Optical flow is used to compute the motion of the pixels of an image sequence. It provides a dense (point to point) pixel correspondance.
- *Correspondence problem*: determine where the pixels of an image at time *t* are in the image at time *t+1*.
- Large number of applications.

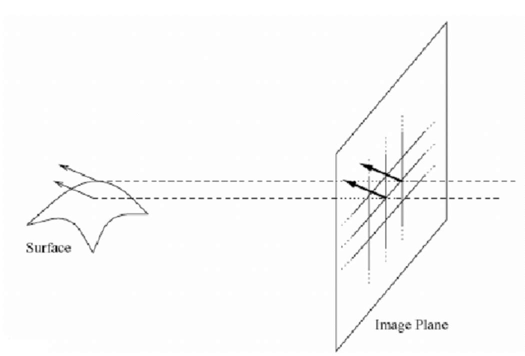



Two *important* definitions

- **Motion field:** "the 2-D projection of a 3-D motion onto the image plane."
- **Optical flow:** "the apparent motion of the brightness pattern in an image sequence."





The method of Horn and Schunck

• This is the most fundamental optical flow algorithm.

• As you will see, it has several *important* flaws that makes its use inappropriate in a large number of applications.

• Most of the other algorithms proposed to date are based on the formulation advanced by Horn and Schunck.

- If the brightness is assumed to be constant from frame to frame, then the motion associated to each pixel *(x,y)* of an image *I* can be modeled as:

$$I(x,y,z) = I(x+u\delta t, y+v\delta t, t+\delta t)$$

This is known as the *data conservation constraint*.

- The 1st-order Taylor expansion

$$I_x u + I_y v + I_t = 0$$
$$E_D = \iint_R (I_x u + I_y v + I_t)^2 \, dxdy$$

- Derivations of previous result:

$$I(x,y,z) = I(x+u\delta t, y+v\delta t, t+\delta t)$$

$$I(x,y,t) = I(x,y,t) + \partial x \frac{\partial I}{\partial x} + \partial y \frac{\partial I}{\partial y} + \partial t \frac{\partial I}{\partial t} + \varepsilon$$

$$\partial t \to 0 \;\Rightarrow\; \frac{dx}{dt}\frac{\partial I}{\partial x} + \frac{dy}{dt}\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \Rightarrow \frac{dI}{dt} = 0$$

$$u = \frac{dx}{dt}, \quad v = \frac{dy}{dt}, \quad I_x = \frac{\partial I}{\partial x}, \quad I_y = \frac{\partial I}{\partial y}, \quad I_t = \frac{\partial I}{\partial t}.$$

$$I_x u + I_y v + I_t = 0$$

- The method is underdetermined.
- We can add an additional constraint known as: *spatial coherence*

$$E_S = \iint_R \left\|\nabla(u,v)\right\|^2 dxdy$$

$$E_S = \iint_R ((u_x^2 + u_y^2) + (v_x^2 + v_y^2)) dxdy$$

- The solution can be obtained by minimizing the functional:

$$E_D + \lambda E_S$$

Minimization

$$\iint F(u,v,u_x,u_y,v_x,v_y)\,dxdy$$

To minimize the above integral, we can use calculus of variations. The *Euler* equations are:

$$F_u - \frac{\partial}{\partial x} F_{u_x} - \frac{\partial}{\partial y} F_{u_y} = 0$$
$$F_v - \frac{\partial}{\partial x} F_{v_x} - \frac{\partial}{\partial y} F_{v_y} = 0$$

And we want to minimize the expression:

$$F = \left(u_x^2 + u_y^2\right) + \left(v_x^2 + v_y^2\right) + \lambda\left(I_x u + I_y v + I_t\right)^2$$

$$\nabla^2 u = \lambda\left(I_x u + I_y v + I_t\right) I_x$$
$$\nabla^2 v = \lambda\left(I_x u + I_y v + I_t\right) I_y$$

Note that:

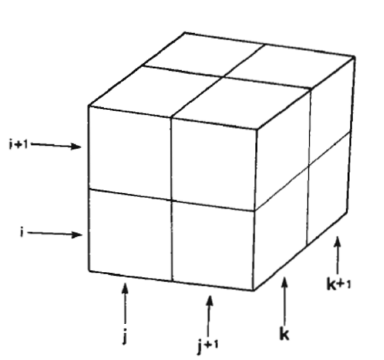
$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

(Details in: Horn, "Robot Vision", MIT Press, 1986.)

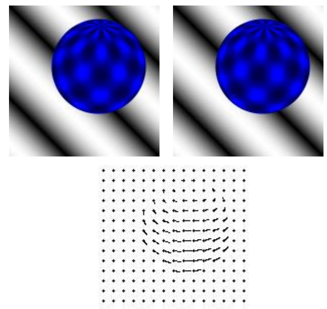
The discrete case

- We can estimate the derivatives, *Ix* and *Iy*, by using the following discreet approximation:

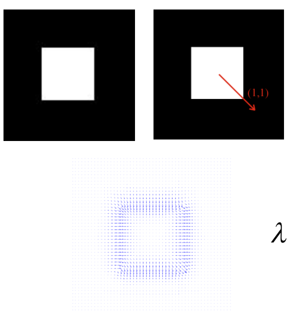
$$I_x \approx \frac{1}{4}\left(I_{i+1,j,k} + I_{i+1,j+1,k} + I_{i+1,j,k+1} + I_{i+1,j+1,k+1}\right)$$
$$-\frac{1}{4}\left(I_{i,j,k} + I_{i,j+1,k} + I_{i,j,k+1} + I_{i,j+1,k+1}\right)$$



Example



Example



$\lambda = 1$

Limitations

- The assumptions embedded in Horn & Schunck formulation are generally inappropriate.
- E.g. both constraints are violated at motion boundaries – known as motion discontinuities.
- The *aperture problem*: to gain accuracy R needs to be large, however, the larger R is the most probable that our assumptions become invalid.

Robust Statistics: Black & Anandan

- It is possible to regard motion discontinuities as *outliers*.
- We can discard outliers only if we can detect them.
- Outliers normally deviate largely from the mean motion.

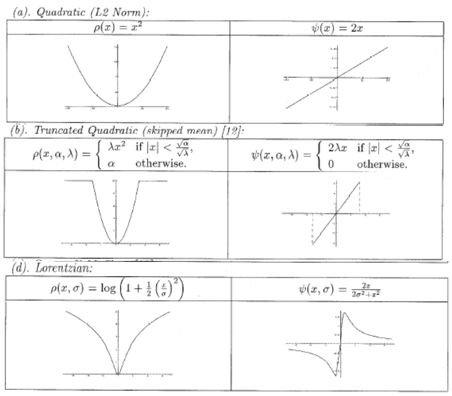
Same constraints as in Horn & Schunck

$$E_D = \iint_R \rho(I_x u + I_y v + I_t)\,dxdy$$

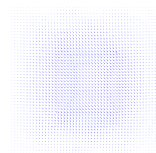
$$E_S = \iint_R \rho(\nabla(u,v))\,dxdy$$

New estimator, e.g.:

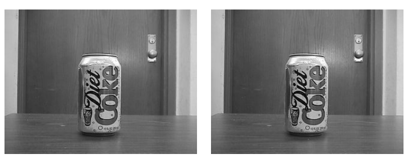
$$\rho(x,\sigma) = \log\left(1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right)$$

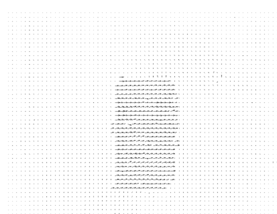


Examples



$\lambda = 1$

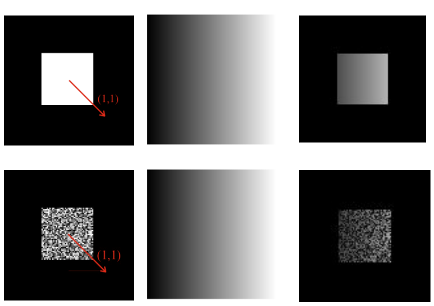




Illumination Changes: Negahdaripour

- The methods seen above assume no changes in the illumination of a scene from frame to frame.
- This is not realistic; even if the illumination source(s) is (are) not moving.
- It is possible to modify the constraint used by the two preceding method.

The illumination problem



Illumination Changes

- Illumination changes also violate the assumptions of E_D and E_S.
- B&A approach cannot handle large variations in lighting. Its formulation does not take this into account.
- We can easily incorporate this information in the form of a multiplier and an offset:

$$I(x+u\delta t, y+v\delta t, t+\delta t) = M(x,y,t)I(x,y,t) + C(x,y,t)$$

Gennert & Negahdaripour

$$E_D = \iint_R (I_x u + I_y v + I_t - I\ m_t - c_t)\,dxdy$$

$$E_S = \iint_R \|\nabla(u,v)\|_2^2\,dxdy$$

$$E_M = \iint_R \|\nabla m_t\|_2^2\,dxdy$$

$$E_C = \iint_R \|\nabla c_t\|_2^2\,dxdy$$

$$E = \lambda_D E_D + \lambda_S E_S + \lambda_M E_M + \lambda_C E_C$$

Closed solution

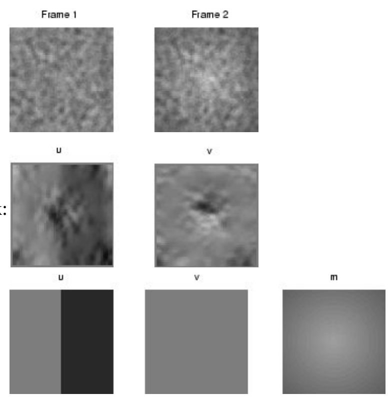
Negahdaripour has proposed the following closed-form solution:

$$\sum_W \begin{bmatrix} I_x^2 & I_x I_y & -I_x I & -I_x \\ I_x I_y & I_y^2 & -I_y I & -I_y \\ -I_x I & -I_y I & I^2 & I \\ -I_x & -I_y & I & 1 \end{bmatrix} \begin{bmatrix} \partial x \\ \partial y \\ \partial m \\ \partial c \end{bmatrix} = \sum_W \begin{bmatrix} -I_x I_t \\ -I_y I_t \\ I I_t \\ I_t \end{bmatrix}$$

Images:

Horn & Schunck:

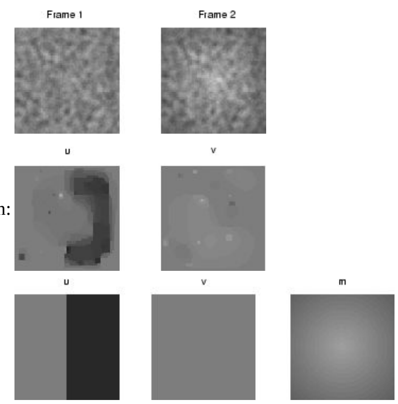
Ground-truth:



Images:

Black & Anandan:

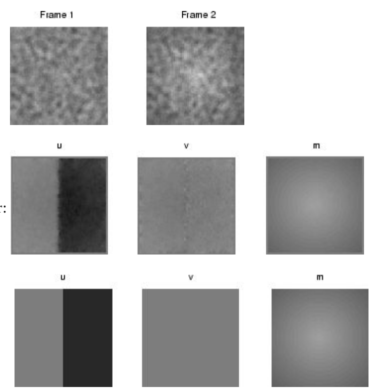
Ground-truth:



Images:

Gennert & Negahdaripour:

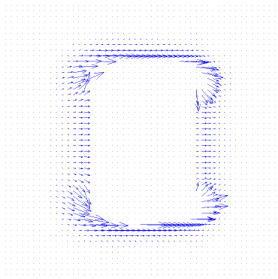
Ground-truth:



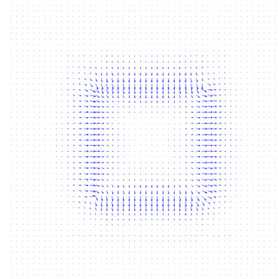
More Examples

Images:

Horn & Schunck

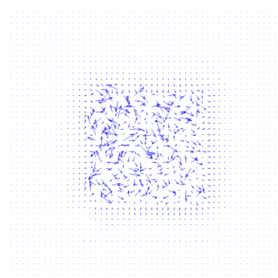


Gennert & Negahdaripour

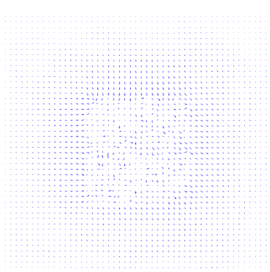


Images

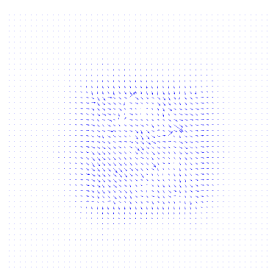
Horn & Schunck



Black & Anandan



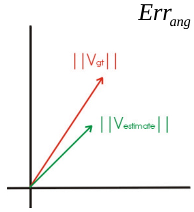
Gennert & Negahdaripour

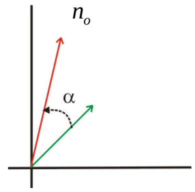


Error estimation

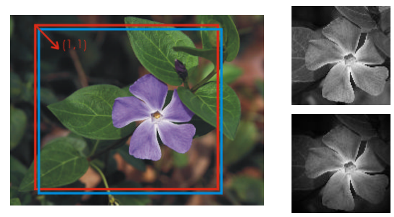
$$Err_{vel} = \frac{\sum_{i=1}^{n_v} \left| \left\| V_{estimate_i} \right\| - \left\| V_{gt_i} \right\| \right|}{n_v}$$

$$Err_{ang} = \frac{\sum_{i=1}^{n_o} \arccos\left(\frac{V_{estimate_i} \cdot V_{gt_i}}{\left\| V_{estimate_i} \right\| \left\| V_{gt_i} \right\|} \right)}{n_o}$$





The flower sequence

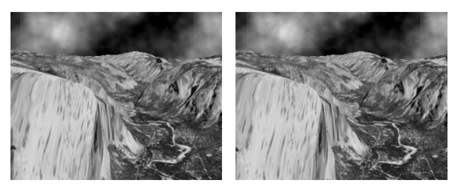


Gaussian

Flower sequence

	Vel. Err.	Stdv.	Density	Angl. Err.	Stdv.	Density
H&S	1.388	0.662	100%	78.144	65.407	39.96%
L&K	1.268	0.782	100.00%	73.061	64.264	62.43%
N&Y	0.562	0.313	100%	18.652	21.086	99.32%
B&A	0.531	0.386	100%	73.264	57.76	97.82%
G&N	0.821	0.283	100%	14.696	16.226	98.97%

The Yosemite sequence



Yosemite sequence

	Vel. Err.	Stdv.	Density	Angl. Err.	Stdv.	Density
H&S	0.935	1.35	100%	45.342	85.14	98.58%
L&K	0.746	1.159	99.88%	30.54	73.761	98.40%
N&Y	0.636	0.726	99.97%	33.877	76.764	99.01%
B&A	0.509	0.725	100%	18.775	57.809	99.13%
G&N	1.547	1.786	100%	41.999	77.642	98.59%

Some Examples


Diet
Coke