Motion Analysis

Motion analysis of digital images is based on a temporal sequence of image frames of a coherent scene.

"sparse sequence"	**=>**	**few frames, temporally spaced apart, considerable differences between frames**
"dense sequence"	**=>**	**many frames, incremental time steps, incremental differences between frames**
video	**=>**	**50 half frames per sec, interleaving, line-by-line sampling**

Motion detection
Register locations in an image sequence which have change due to motion

Moving object detection and tracking
Detect individual moving objects, determine and predict object trajectories, track objects with a moving camera

Derivation of 3D object properties
Determine 3D object shape from multiple views ("shape from motion")

1

Case Distinctions for Motion Analysis

stationary observer
moving observer

single moving object
multiple moving objects

rigid objects
jointed objects
deformable objects

perspective projection
weakly perspective projection
orthographic projection

rotation only
translation only
unrestricted motion

2 image analysis
multiple image analysis

incremental motion
large-scale motion

B/W images
colour images

xray images
IR images
natural images

noisy data
ideal data

monocular images
stereo images

dense flow
sparse flow
no flow
paralaxis

quatitative motion
qualitative motion

small objects
extended objects

polyeder
smooth objects
arbitrary objects

matte surfaces
specular surfaces
textured surfaces
arbitrary surfaces

without occlusion
with occlusion

uncalibrated camera
calibrated camera

data-driven
expectation-driven

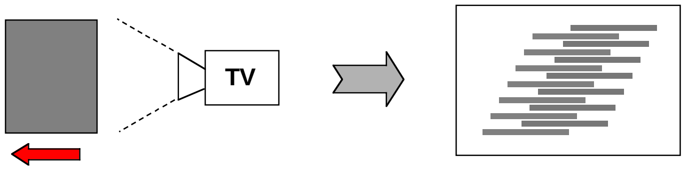
real-time
no real-time

parallel computation
sequential computation

Many motion analysis methods are only applicable in restricted cases!

2

Motion in Video Images



moving object

TV halfframes

TV-rate sampling affects images of moving objects:

- contours show saw-tooth pattern
- deformed angles
- limited resolution

Example:

- 512 pixels per row
- length of dark car is ca. 3.5 m ≈ 130 pixel
- speed is ca. 50 km/h ≈ 14 m/s
- displacement between halfframes is ca. 10 pixels

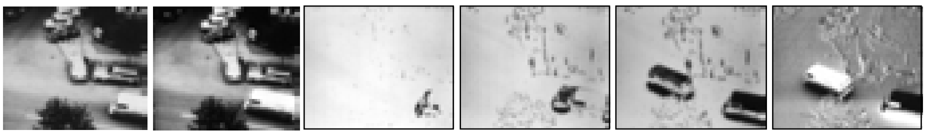
3

Difference Images

An obvious technique for motion detection is based on difference images:

- take the pixelwise difference of images of a sequence
- threshold the magnitude of the differences
- regions above threshold may be due to motion

Examples:



frame1 | frame12 | difference frame2 - frame1 threshold 30 | difference frame12 - frame1 threshold 30 | difference frame34 - frame1 threshold 30 | difference frame34 - frame1 without threshold

Note the effects prohibit reliable motion detection:

- phase jitter between frames (pixels do not correspond exactly)
- spurious motion of branches, pedestrians, dogs, etc.
- motion of uniform brightness regions does not show

4

Counting Differences

If the goal is to isolate the images of moving objects, it may be useful to

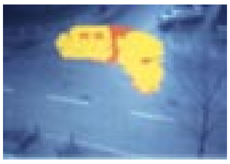
- count how often a pixel differs from its initial value (first-order difference picture FODP)
- count how often a pixel of a FODP region differs from its previous value (second-order difference picture SODP)

(R. Jain 76)

frame1

difference
frame4 - frame1
FODP (yellow)
SODP (red)

difference
frame10 - frame1
FODP (yellow)
SODP (red)



difference
frame30 - frame1
FODP (yellow)
SODP (red)

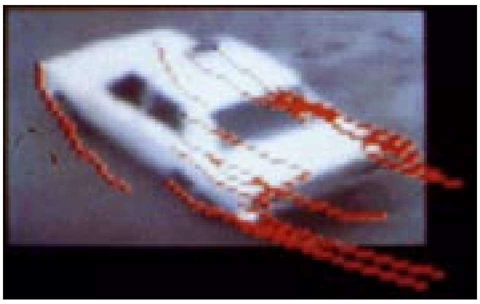
The problem of uniform brightness regions is not really overcome.

5

Corresponding Interest Points

Detection of moving objects by
- finding "interest points" in all frames of a sequence
- determining the correspondence of interest points in different frames
- chaining correspondences over time
- grouping interest points into object candidates

<u>Example</u>: Tracking interest points of a taxi turning off Schlüterstraße (Dreschler and Nagel 82)

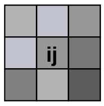


6

Moravec Interest Operator

Interest points (feature points) are image locations where an interest operator computes a high value. Interest operators measure properties of a local pixel neighbourhood.

Moravec interest operator: $M(i,j) = \frac{1}{8} \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} |g(m,n) - g(i,j)|$



This simple operator measures the distinctness of a point w.r.t. its surround.

Refinement of Moravec operator:

Determine locations with strong brightness variations along two orthogonal directions (e.g. based on variances in horizontal, vertical and diagonal direction).

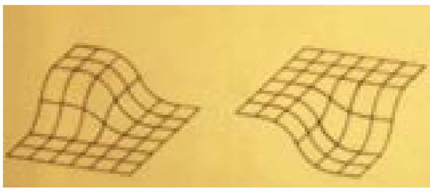
Interest points in different frames may not correspond to identical physical object parts due to their small neighbourhood and noise.

7

Corner Models

Interest points may be based on models of interesting facets of the image function, e.g. corners.

"corner" = location with extremal Gaussian curvatures (Dreschler and Nagel 81)



Zuniga-Haralick operator:

- **fit a cubic polynomial**

  $f(i,j) = c_1 + c_2x + c_3y + c_4x^2 + c_5xy + c_6y^2 + c_7x^3 + c_8x^2y + c_9xy^2 + c_{10}y^3$

  For a 5x5 neighbourhood the coefficients of the best-fitting polynomial can be directly determined from the 25 greyvalues

- **compute interest value from polynomial coefficients**

  $ZH(i,j) = \frac{-2(c_2^2 c_6 - c_2 c_3 c_5 - c_3^2 c_4)}{\left(c_2^2 + c_3^2\right)^{\frac{3}{2}}}$ **measure of "cornerness" of the polynomial**

8

Correspondence Problem

The correspondence problem is to determine which interest points in different frames of a sequence mark the same physical part of a scene.

Difficulties:

- scene may not offer enough structure to uniquely locate points
- scene may offer too much structure to uniquely locate points
- geometric features may differ strongly between frames
- photometric features differ strongly between frames
- there may be no corresponding point because of occlusion

Note that these difficulties apply to single-camera motion analysis as well as multiple-camera 3D analysis (e.g. binocular stereo).

9

Correspondence by Iterative Relaxation

Basic scheme (Thompson and Barnard 81) modified by Dreschler and Nagel:

- initialize correspondence confidences between all pairs of interest points in 2 frames based on
  - similarity of greyvalue neighbourhoods
  - plausibility of distance (velocity)
- modify confidences iteratively based on
  - similarity of displacement vectors in the neighbourhood
  - confidence of competing displacement vectors


interest points of 2 frames (red and blue)

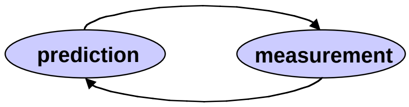

initialized confidences


confidences after 10 iterations

10

Kalman Filters (1)

A Kalman filter provides an iterative scheme for (i) predicting an event and (ii) incorporating new measurements.



Assume a linear system with observations depending linearly on the system state, and white Gaussian noise disturbing the system evolution and the observations:

$$\underline{x}_{k+1} = A_k \underline{x}_k + \underline{w}_k$$
$$\underline{z}_k = H_k \underline{x}_k + \underline{v}_k$$

What is the best estimate of $\underline{x}_k$ based on the previous estimate $\underline{x}_{k-1}$ and the observation $\underline{z}_k$?

- $\underline{x}_k$ quantity of interest ("state") at time k
- A_k model for evolution of $\underline{x}_k$
- $\underline{w}_k$ zero mean Gaussian noise with covariance Q_k
- $\underline{z}_k$ observations at time k
- H_k relation of observations to state
- $\underline{v}_k$ zero mean Gaussian noise with covariance R_k

Often, A_k, Q_k, H_k and R_k are constant.

11

Kalman Filters (2)

The best a priori estimate of $\underline{x}_k$ before observing $\underline{z}_k$ is

$$\underline{x}_k' = A_{k-1}\underline{x}_{k-1}$$

After observing $\underline{z}_k$, the a priori estimate is updated by

$$\underline{x}_k'' = \underline{x}_k' + K_k(\underline{z}_k - H_k\underline{x}_k')$$

K_k is Kalman gain matrix. K_k is determined to minimize the a posteriori variance P_k'' of the error $x_k - \underline{x}_k''$. The minimizing K_k is

$$K_k = P_k' H_k^T (H_k P_k' H_k^T + R_k)^{-1}$$

with $P_k' = A_k P_{k-1}'' A_k^T + Q_{k-1}$ and $P_k'' = (I - K_k H_k) P_k'$

P_k' is covariance of error $x_k - \underline{x}_k'$ before observation of $\underline{z}_k$.

Iterative order of computations:

$\underline{x}_1' \rightarrow$ (1) $\underline{x}_k' = A_{k-1}\underline{x}_{k-1}''$

$P_1' \rightarrow$ (2) $P_k' = A_k P_{k-1}'' A_k^T + Q_{k-1}$

initialization

(1) $K_k = P_k' H_k^T (H_k P_k' H_k^T + R_k)^{-1}$

(2) $\underline{x}_k'' = \underline{x}_k' + K_k(\underline{z}_k - H_k\underline{x}_k')$

(3) $P_k'' = (I - K_k H_k) P_k'$

12

Kalman Filter Example

Track positions p_k and velocities v_k of an object moving along a straight line. Assume unknown accelerations a_k with probability density $N(0, q^2)$ and measurements of positions p_k corrupted by white noise b_k with probability density $N(0, r^2)$.

$$\underline{x}_{k+1} = A_k \underline{x}_k + \underline{w}_k \quad \Rightarrow \quad \begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} a_k$$

T is time increment

$$\underline{z}_k = H_k \underline{x}_k + \underline{v}_k \quad \Rightarrow \quad \begin{bmatrix} z_k \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \begin{bmatrix} b_k \\ 0 \end{bmatrix}$$

$z_k = p_k + b_k$

$$\underline{x}_0' = \begin{bmatrix} p_0 \\ v_0 \end{bmatrix} \quad P_0' = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

initialization (here: position and velocity values are known with certainty)

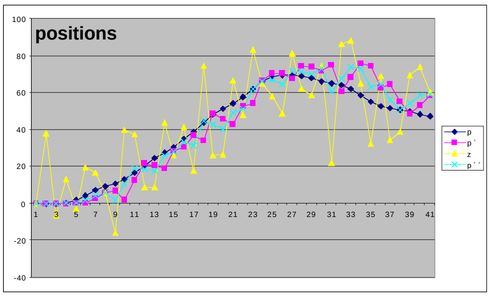
$$K_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \underline{x}_0'' = \begin{bmatrix} p_0 \\ v_0 \end{bmatrix} \quad P_0'' = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

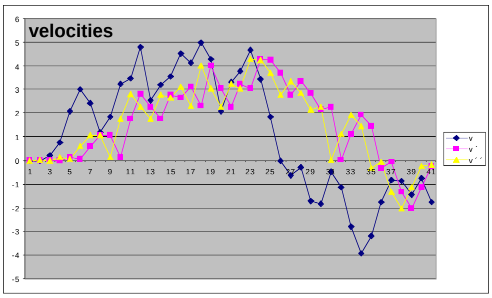
$$\underline{x}_1' = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} p_0 + v_0 T \\ v_0 \end{bmatrix} \quad P_1' = q^2 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$K_1 = \frac{q^2}{q^2 + r^2} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \underline{x}_1'' = \begin{bmatrix} p_0 + v_0 T \\ v_0 \end{bmatrix} + \frac{q^2}{q^2 + r^2} \begin{bmatrix} z_1 - (p_0 + v_0 T) \\ 0 \end{bmatrix} \quad P_1'' = \frac{q^2}{q^2 + 1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

13

Diagrams for Kalman Filter Example (1)



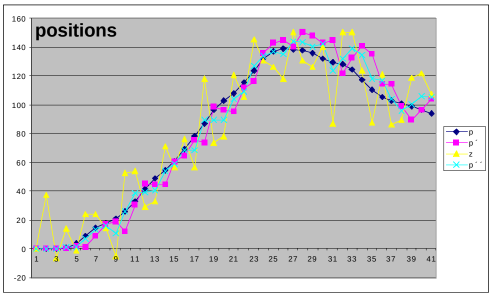


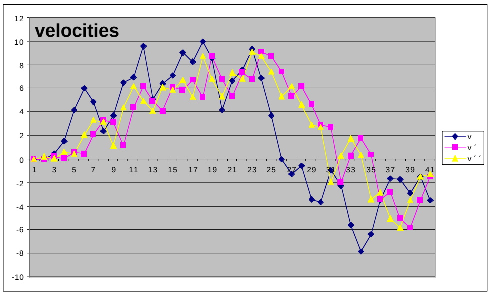
$T = 1$	time step
$q = 1$	standard deviation of acceleration bursts
$r = 20$	standard deviation of position sensor
$p_0 = 0$	initial position
$v_0 = 0$	initial velocity

The standard deviation of the estimated position p is around 12 before observing $\underline{z}$ and around 10 after observing $\underline{z}$.

14

Diagrams for Kalman Filter Example (2)





$T = 1$ time step

$q = 2$ standard deviation of acceleration bursts

$r = 20$ standard deviation of position sensor

$p_0 = 0$ initial position

$v_0 = 0$ initial velocity

The standard deviation of the estimated position p is around 15 before observing <u>z</u> and around 12 after observing <u>z</u>.

Optical Flow Constraint Equation

Optical flow is the displacement field of surface elements of a scene during an incremental time interval dt ("velocity field").

Assumptions:
- Observed brightness is constant over time (no illumination changes)
- Nearby image points move similarly (velocity smoothness constraint)

For a continuous image g(x, y, t) a linear Taylor series approximation gives

$$g(x+dx, y+dy, t+dt) \approx g(x, y, t) + g_x dx + g_y dy + g_t dt$$

For motion without illumination change we have

$$g(x+dx, y+dy, t+dt) = g(x, y, t)$$

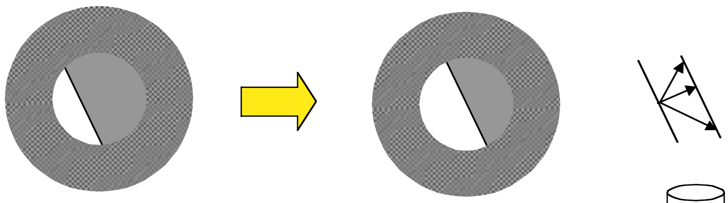
Hence $g_x dx/dt + g_y dy/dt = g_x u + g_y v = -g_t$ u, v velocity components

$\boxed{g_x u + g_y v = -g_t}$ **optical flow constraint equation**

$g_x \approx \Delta g/\Delta x$, $g_y \approx \Delta g/\Delta y$, $g_t \approx \Delta g/\Delta t$ may be estimated from the spatial and temporal surround of a location (x, y), hence the optical flow constraint equation provides <u>one</u> equation for the <u>two</u> unknowns u and v.

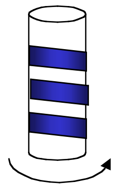
Aperture Effect

The optical flow constraint allows for ambiguous motion interpretations. This can be illustrated by the aperture effect.



In which direction has the edge moved?

Compare with the barber pole effect:



Due to the linear approximation of the image function, the velocity vector cannot be determined uniquely from a local neighbourhood.

Optical Flow Smoothness Constraint

For dynamic scenes one can often assume that the velocity field changes smoothly in a spatial neighbourhood:

- **large objects**
- **translational motion**
- **observer motion, distant objects**

Hence, as an additional constraint, one can minimize a smoothness error:

$$e_s = \iint ((u_x^2 + u_y^2) + (v_x^2 + v_y^2)) \, dx \, dy$$

One also wants to minimize the error in the optical flow constraint equation:

$$e_c = \iint (g_x u + g_y v + g_t)^2 \, dx \, dy$$

Using a Lagrange multiplier λ, both constraints can be combined into an error functional, to be minimized by the calculus of variations:

$$e = \iint (g_x u + g_y v + g_t)^2 + \lambda (u_x^2 + u_y^2 + v_x^2 + v_y^2) \, dx \, dy$$

Optical Flow Algorithm

The solution for optical flow with smoothness constraint is given in terms of a pair of partial differential equations:

$$u = \bar{u} - g_x \frac{g_x \bar{u} + g_y \bar{v}}{\lambda^2 + g_x^2 + g_y^2} \qquad v = \bar{v} - g_y \frac{g_x \bar{u} + g_y \bar{v}}{\lambda^2 + g_x^2 + g_y^2}$$

$\bar{u}$ and $\bar{v}$ denote mean velocity values based on the local neighbourhood

The equations can be solved by a Gauss-Seidel iteration based on pairs of consecutive images (Horn & Schunck 81).

Basic optical flow algorithm (Sonka et al. 98, pp. 687):

1. Initialize velocity vectors $\underline{c}(i, j)$ for all (i, j) where $\underline{c}^T = [u\ v]$
2. Estimate g_x, g_y, g_t for all (i, j) from the pair of consecutive images
3. For the k-th iteration, compute

$$u^k(i,j) = \bar{u}^{k-1}(i,j) - g_x(i,j)Q^{k-1}(i,j)$$
$$v^k(i,j) = \bar{v}^{k-1}(i,j) - g_y(i,j)Q^{k-1}(i,j)$$

with $Q^{k-1}(i,j) = \dfrac{g_x(i,j)\bar{u}^{k-1}(i,j) + g_y(i,j)\bar{v}^{k-1}(i,j)}{\lambda^2 + g_x^2(i,j) + g_y^2(i,j)}$

4. Repeat step 3 until the error e is below a threshold

$$e^k = \sum_i \sum_j [g_x(i,j)u^k(i,j) + g_y(i,j)v^k(i,j) + g_t(i,j)]^2 + \lambda[u_x^{k^2}(i,j) + u_y^{k^2}(i,j) + v_x^{k^2}(i,j) + v_y^{k^2}(i,j) < \varepsilon$$

λ is a fixed value chosen to balance the constraints

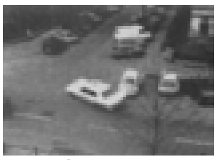
19

Optical Flow Improvements

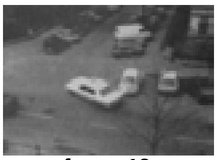
(from Nagel and Enkelmann 86)

Several improvements of the Horn & Schunck optical flow computation have been suggested. For example, Nagel (1983) introduced the "oriented smoothness constraint" which does not enforce smoothness across edges.

2 frames of the taxi sequence

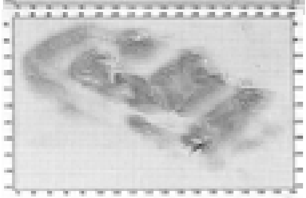


frame 11



frame 12

needle diagram of optical flow for taxi motion with isotropic smoothness constraint after 30 iterations



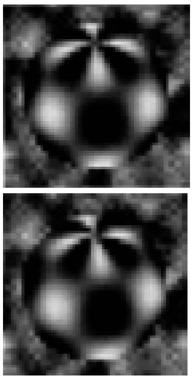
the same with oriented smoothness constraint

20

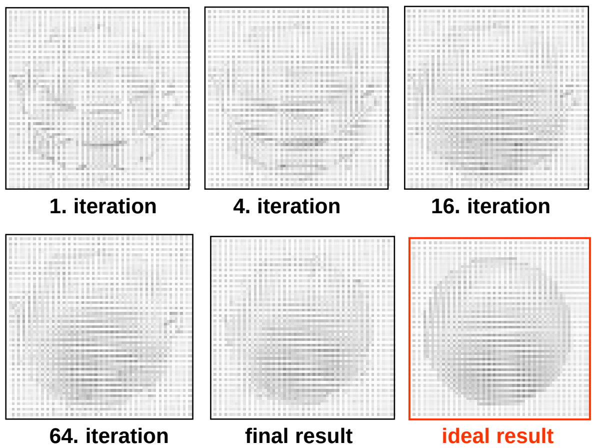
Optical Flow and Segmentation

The optical flow smoothness constraint is not valid at occluding boundaries ("silhouettes"). In order to inhibit the constraint, one may try to segment the image based on optical flow discontinuities while performing the iterations.

Checkered sphere rotating before randomly textured background



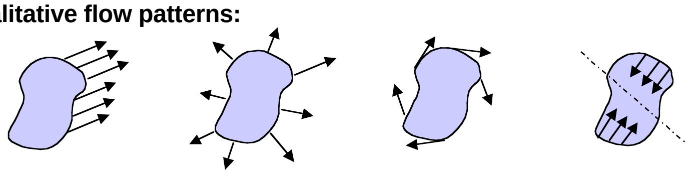
(From B.K.P. Horn, Robot Vision, 1986)



Optical Flow Patterns

Complex optical flow fields may be segmented into components which show a consistent qualitative pattern.

Qualitative flow patterns:



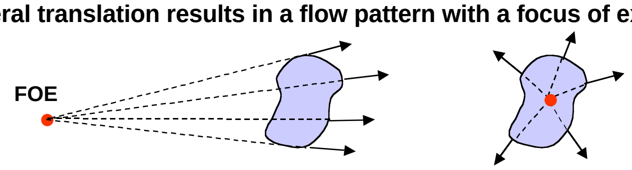
translation at constant distance

translation in depth

rotation at constant distance

rotation about axis parallel to image plane

General translation results in a flow pattern with a focus of expansion (FOE):



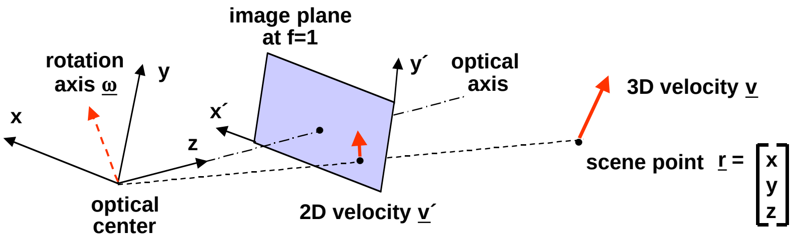
As the direction of motion changes, the FOE changes its location.

Optical Flow and 3D Motion (1)

In general, optical flow may be caused by an unknown 3D motion of an unkown surface.

How do the flow components u´, v´ depend on the 3D motion parameters?

Assume camera motion in a static scene, optical axis = z-axis, rotation about the origin.



3D velocity $\underline{v}$ of a point $\underline{r}$ is determined by rotational velocity $\underline{\omega}$ and translational velocity $\underline{t}$:

$$\underline{v} = -\underline{t} - \underline{\omega} \times \underline{r}$$

Optical Flow and 3D Motion (2)

By taking the component form of $\underline{v} = -\underline{t} - \underline{\omega} \times \underline{r}$ with $\underline{t}^T = [t_x, t_y, t_z]$, $\underline{\omega}^T = [a, b, c]$ and $\underline{r}^T = [x\ y\ z]$ and computing the perspective projection we get

$$u' = \frac{\dot{x}}{z} - \frac{x\dot{z}}{z^2} = \left(-\frac{t_x}{z} - b + cy'\right) - x'\left(-\frac{t_z}{z} - ay' + bx'\right)$$

$$v' = \frac{\dot{y}}{z} - \frac{y\dot{z}}{z^2} = \left(-\frac{t_y}{z} - cx' + a\right) - y'\left(-\frac{t_z}{z} - ay' + bx'\right)$$

Observation of u´and v´at location (x´, y´) gives 2 equations for 7 unknowns. Note that motion of a point at distance kz with translation k$\underline{t}$ and the same rotation ω will give the same optical flow, k any scale factor.

The translational and rotational parts may be separated:

$$u'_{translation} = -\frac{t_x + x't_z}{z} \qquad u'_{rotation} = ax'y' - b(x'^2 + 1) + cy'$$

$$v'_{translation} = -\frac{t_y + y't_z}{z} \qquad v'_{rotation} = a(y'^2 + 1) - bx'y' + cx'$$

For pure translation we have 2 equations for 3 unknows (z fixed arbitrarily).