

TSBB15

Computer Vision

Lecture 4

Motion estimation and optical flow

Motion

In many applications it is the case that

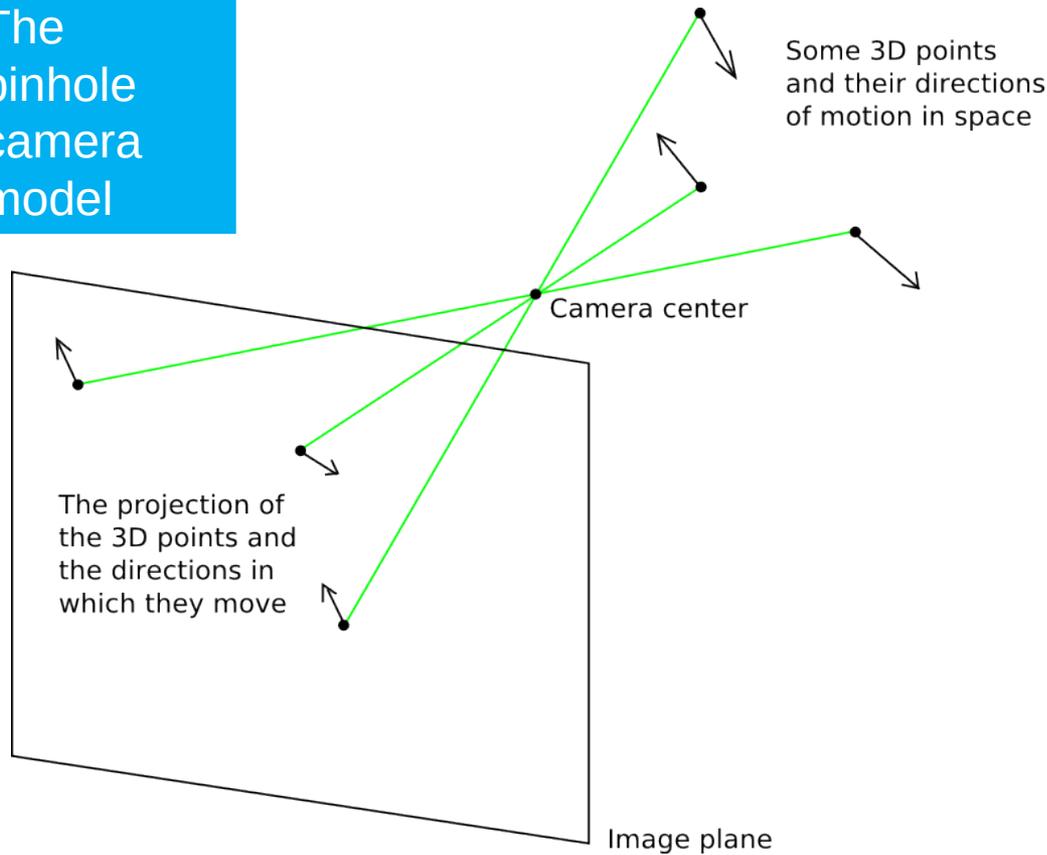
- the scene depicted in the image is dynamic
 - moving objects
 - deformable objects
- or the camera is moving relative to the scene
- in general: both cases

Motion

- From the camera's (viewer's) perspective these two cases are indistinguishable
 - Unless a high-level interpretation of the scene is available
- However, we can describe how points in the scene move relative to some reference frame, e.g., as defined by the camera

The motion field

The
pinhole
camera
model



The *motion field* is the projection of the 3D motion onto the image plane

It can be represented as a vector valued function of the image coordinate

$\mathbf{m}(\mathbf{x})$

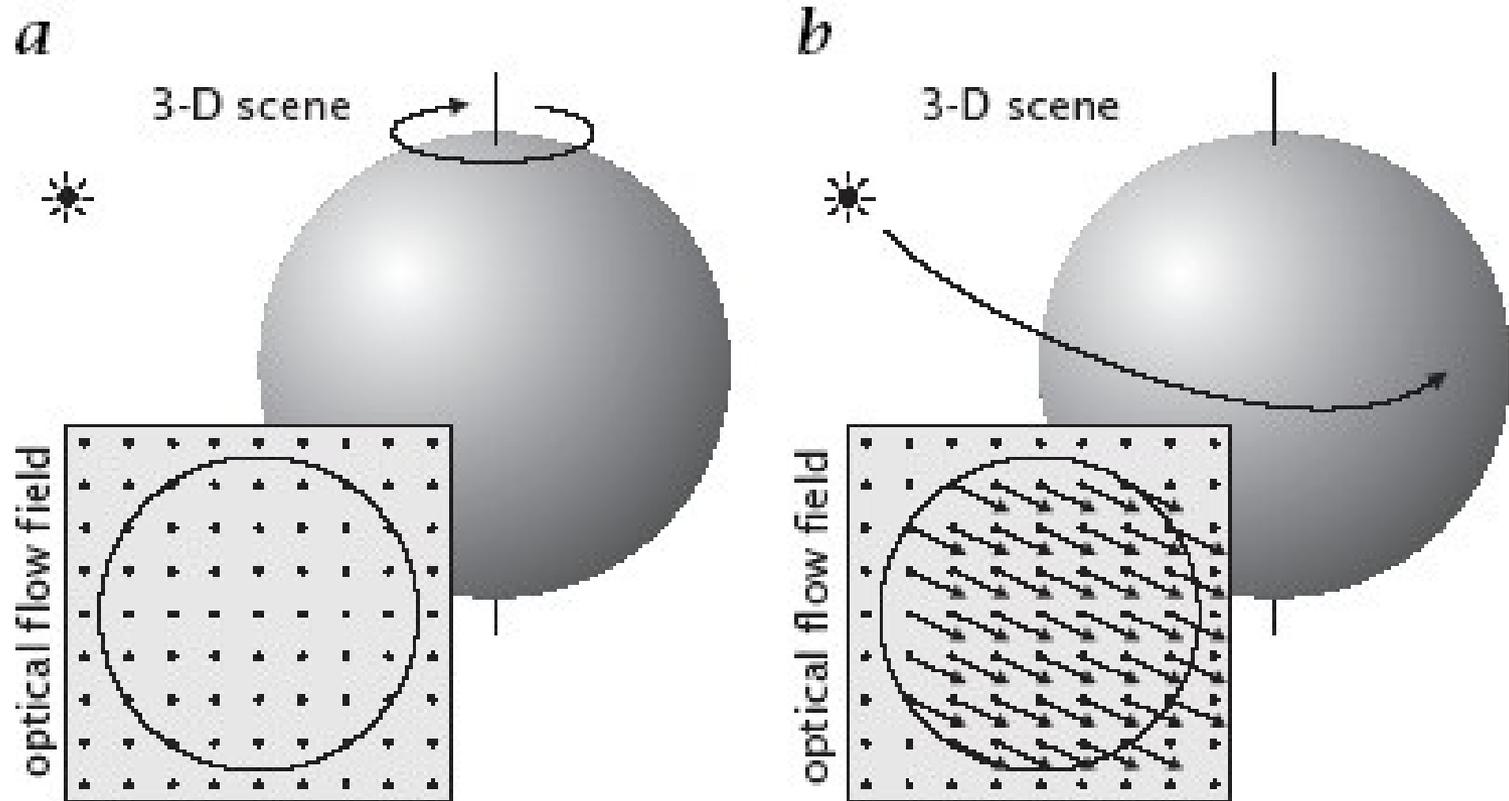
The motion field

- If we can measure the motion field $\mathbf{m}(\mathbf{x})$ it is possible to infer
 - how points and objects are moving relative the camera, or
 - how the camera is moving relative to the scene (*ego-motion* estimation)

The motion field

- In practice, we cannot measure $\mathbf{m}(\mathbf{x})$ directly
- However, we can measure how the image intensity moves/varies over time
 - Optical flow Will be formally defined shortly
- But there is no direct relation between the optical flow and the motion field
 - 3D motion may not always generate temporal variations in the image
 - 3D points that move along the projection lines have constant positions in the image
 - Temporal variations in the image may not always correspond to 3D motion

Physical vs visual motion



From Jähne & Haussecker

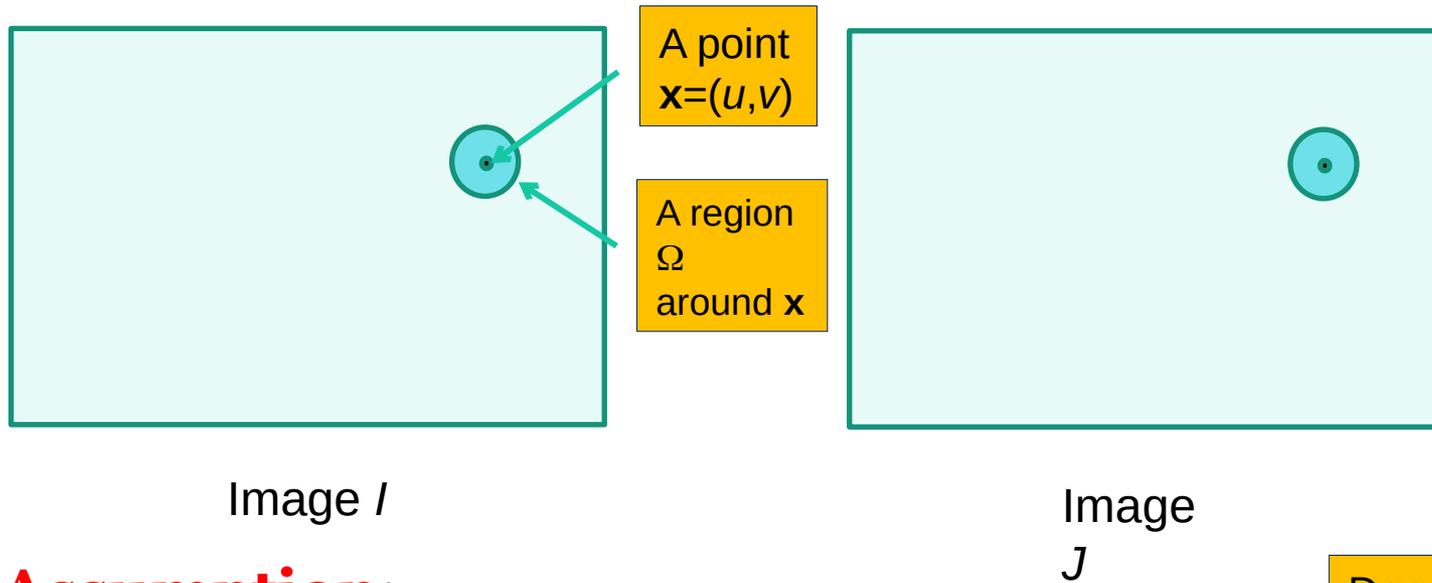
Displacement estimation

- One approach to motion estimation considers **two images** of the same scene, e.g.
 - Taken at two different time points, same camera position
 - Images from a video sequence, e.g., two consecutive images. Displacement is an estimate of the motion field $\mathbf{m}(\mathbf{x})$
 - Taken from two different positions, possibly at the same time point
 - Stereo images. Displacement is an estimate of depth in the scene (assuming a stationary scene)

Example (from *Middlebury*)



Mathematical model



- **Assumption:**

$$J(\mathbf{x}) = I(\mathbf{x} + \mathbf{d}) \quad \text{for all } \mathbf{x} \in \Omega$$

- Pixel values are constant, but displaced by \mathbf{d}
- How can we determine \mathbf{d} for each point \mathbf{x} ?

Depends on position of \mathbf{x}

Estimation of \mathbf{d}

- \mathbf{d} , at point \mathbf{x} , can be estimated by forming a cost function, based on the constancy of the pixel values:

$$\epsilon = \int_{\Omega_0} w(\mathbf{y}) (I(\mathbf{x} + \mathbf{y} + \mathbf{d}) - J(\mathbf{x} + \mathbf{y}))^2 d\mathbf{y}$$

A region of the origin, same size as Ω

A weighting function, e.g., a Gaussian, of same size as Ω

- The minimizer of ϵ is an estimate of \mathbf{d} at \mathbf{x} , which we then use as an estimate of $\mathbf{m}(\mathbf{x})$

Estimation of \mathbf{d}

- As an estimate of $\mathbf{m}(\mathbf{x})$, $\mathbf{d}(\mathbf{x})$ is referred to as ***optical flow*** (or optic flow)
- Finding the minimizer of ε is a non-linear estimation problem
 - Computationally complex problem
- It can be simplified by a linearization of I

Linearization of I

- At each point $\mathbf{x} + \mathbf{y}$, the dependency on \mathbf{d} in the intensity function I can be expressed as a Taylor expansion:

$$\nabla I(\mathbf{x} + \mathbf{y}) = \begin{pmatrix} \frac{\partial I}{\partial u} \\ \frac{\partial I}{\partial v} \end{pmatrix} = \text{Image gradient at } \mathbf{x} + \mathbf{y}$$

$$I(\mathbf{x} + \mathbf{y} + \mathbf{d}) = I(\mathbf{x} + \mathbf{y}) + \nabla I(\mathbf{x} + \mathbf{y}) \cdot \mathbf{d}$$

- Assumption:** higher order terms in \mathbf{d} can be neglected

Linear estimation of \mathbf{d}

With this linearization of I at hand:

$$\epsilon = \int_{\Omega_0} w(\mathbf{y}) \left(I(\mathbf{x} + \mathbf{y}) - J(\mathbf{x} + \mathbf{y}) + \underbrace{\nabla I(\mathbf{x} + \mathbf{y}) \cdot \mathbf{d}} \right)^2 d\mathbf{y}$$

Equation
(A)

$$\frac{\partial I}{\partial u} v_1 + \frac{\partial I}{\partial v} v_2$$

- We want to find the minimum of ϵ with respect to the elements of $\mathbf{d} = (v_1, v_2)$
- Find \mathbf{d} where $\begin{pmatrix} \frac{\partial \epsilon}{\partial v_1} \\ \frac{\partial \epsilon}{\partial v_2} \end{pmatrix} = \mathbf{0}$

Determining \mathbf{d}

$$\begin{pmatrix} \frac{\partial \epsilon}{\partial v_1} \\ \frac{\partial \epsilon}{\partial v_2} \end{pmatrix} = \begin{pmatrix} 2 \int_{\Omega_0} w(\mathbf{y}) (I(\mathbf{x} + \mathbf{y}) - J(\mathbf{x} + \mathbf{y}) + \nabla I(\mathbf{x} + \mathbf{y}) \cdot \mathbf{d}) \frac{\partial I}{\partial u} d\mathbf{y} \\ 2 \int_{\Omega_0} w(\mathbf{y}) (I(\mathbf{x} + \mathbf{y}) - J(\mathbf{x} + \mathbf{y}) + \nabla I(\mathbf{x} + \mathbf{y}) \cdot \mathbf{d}) \frac{\partial I}{\partial v} d\mathbf{y} \end{pmatrix}$$

⇓

$$\int_{\Omega_0} w(\mathbf{y}) \begin{pmatrix} \frac{\partial I}{\partial u} \\ \frac{\partial I}{\partial v} \end{pmatrix} (I(\mathbf{x} + \mathbf{y}) - J(\mathbf{x} + \mathbf{y}) + \nabla^T I(\mathbf{x} + \mathbf{y}) \mathbf{d}) d\mathbf{y} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

The Lucas-Kanade equation

Assumption: \mathbf{d} is constant within Ω , i.e., \mathbf{d} is independent of \mathbf{y}

$$\int_{\Omega_0} w(\mathbf{y}) \nabla I(\mathbf{x} + \mathbf{y}) \nabla^T I(\mathbf{x} + \mathbf{y}) d\mathbf{y} \mathbf{d} = \int_{\Omega_0} w(\mathbf{y}) \nabla I(\mathbf{x} + \mathbf{y}) (J(\mathbf{x} + \mathbf{y}) - I(\mathbf{x} + \mathbf{y})) d\mathbf{y}$$

$\underbrace{\hspace{15em}}_{=\mathbf{T}(\mathbf{x})} \qquad \Downarrow \qquad \underbrace{\hspace{15em}}_{=\mathbf{s}(\mathbf{x})}$

The structure
tensor

$$\mathbf{T} \mathbf{d} = \mathbf{s}$$

This is *the Lucas-Kanade equation (LK-equation)*.
One equation per pixel in the image (gives one \mathbf{d} per pixel)

Determining \mathbf{d}

- In principle, \mathbf{d} can be determined from the LK-equation as

$$\mathbf{d} = \mathbf{T}^{-1} \mathbf{s}$$

- Only works if \mathbf{T} is not singular, i.e., I in Ω **must not be i1D**
- Lucas & Kanade: *An Iterative Image Registration Technique with an Application to Stereo Vision*, IUW, 1981

Alternative derivation of LK

- The LK-equation derived here is based on finding the local displacement between two images
- An alternative derivation is provided by the brightness constancy principle

Brightness constancy

- Think of the intensity function I as explicitly depending on the 3 variables, $I = I(u, v, t)$
- Basic assumption:
 - If we observe intensity I at (u, v, t) , this intensity **remains constant over time**, but it may change position as a function of time
- This is referred to as: ***brightness constancy***

Mathematical formulation

Means: the total derivative of I w.r.t. t is $= 0$

$$\frac{dI}{dt} = 0$$

Expand in partial derivatives of I :

$$\frac{\partial I}{\partial t} \frac{dt}{dt} + \frac{\partial I}{\partial u} \frac{du}{dt} + \frac{\partial I}{\partial v} \frac{dv}{dt} = 0$$

Mathematical formulation

Cont.

$$\frac{\partial I}{\partial t} \underbrace{\frac{dt}{dt}}_{=1} + \frac{\partial I}{\partial u} \underbrace{\frac{du}{dt}}_{=v_1} + \frac{\partial I}{\partial v} \underbrace{\frac{dv}{dt}}_{=v_2} = 0$$

- $\mathbf{v} = (v_1, v_2)$ is the velocity vector of the intensity I at (u, v, t)
- \mathbf{v} is a function of (u, v, t) , $\mathbf{v} = \mathbf{v}(\mathbf{x})$
- Local estimate of the motion field $\mathbf{m}(\mathbf{x})$

BCCE / Optic flow equation

Cont.
$$\frac{\partial I}{\partial t} + \frac{\partial I}{\partial u} v_1 + \frac{\partial I}{\partial v} v_2 = 0$$

Alternative
formulation:

$$\frac{\partial I}{\partial t} + \nabla I \cdot \mathbf{v} = 0$$

- This is the ***Brightness Constancy Constraint Equation (BCCE)***
- A.k.a. the optical (optic) flow equation

BCCE

- Is a differential equation
- It assumes that we can determine/estimate the temporal derivative of I at (u, v, t)
 - In practice, it must be estimated in terms of finite differences
 - Compare to the two-image derivation of the LK-eq
- BCCE is one equation per pixel (and time)
 - But it has 2 unknowns: (v_1, v_2)
 - Cannot be solved at the pixel level

Determining \mathbf{v}

- At a pixel $\mathbf{x} = (u, v)$, at time t , we can formulate a cost function

$$\epsilon = \int_{\Omega_0} w(\mathbf{y}) \left(\frac{\partial I}{\partial t} + \nabla I(\mathbf{x} + \mathbf{y}) \cdot \mathbf{v} \right)^2 d\mathbf{y}$$

- Assumes that \mathbf{v} is constant within Ω
- This cost function is very similar to the one used for the 2-image case, [Equation \(A\)](#), slide 14

LK-equation, again...

- Minimizing ε , therefore, implies finding \mathbf{v} such that

$$\mathbf{T} \mathbf{v} = \mathbf{s}$$

Continuous time LK-
eq

- Where

$$\mathbf{T}(\mathbf{x}) = \int_{\Omega_0} w(\mathbf{y}) \nabla I(\mathbf{x} + \mathbf{y}) \nabla^T I(\mathbf{x} + \mathbf{y}) d\mathbf{y}$$

$$\mathbf{s}(\mathbf{x}) = - \int_{\Omega_0} w(\mathbf{y}) \frac{\partial I}{\partial t} \nabla I(\mathbf{x} + \mathbf{y}) d\mathbf{y}$$

The aperture problem

- Regardless of how the LK-eq has been derived, it cannot be solved robustly for pixels where I in Ω is i1D
- Even approximately i1D may cause problems
- This is related to the so-called aperture problem:
 - ***In a i1D region we cannot determine the local displacement/velocity along a line/edge***

The aperture problem

- Is the pattern in the circle moving down, right, or right-down?
- Since the pattern is 1D, its velocity cannot be completely determined
- We can, however, determine a unique *normal velocity*
 - *How?*



BCCE revisited

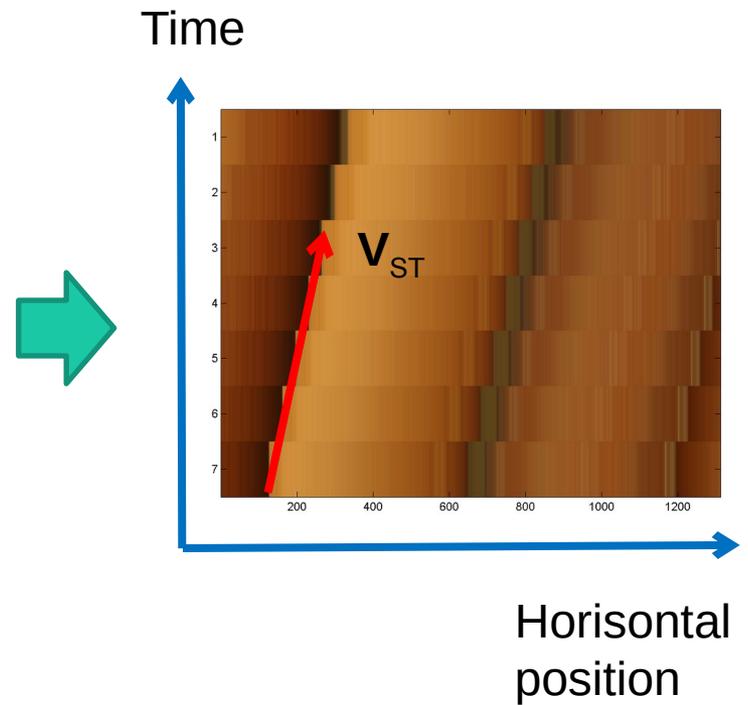
- **A consequence of BCCE:**

In the 3D spatio-temporal volume,
 I must be constant in a direction given by

$$\mathbf{v}_{ST} = (v_1, v_2, 1)$$

- This implies that $\nabla_{ST} I$, the 3D spatio-temporal gradient of I , is orthogonal to \mathbf{v}_{ST}

Example



A new cost function

- We define a new cost function ϵ_{ST} as

$$\epsilon_{\text{ST}} = \int_{\Omega_0} w(\mathbf{y}) \left(\hat{\mathbf{v}}_{\text{ST}}^{\text{T}} \nabla_{\text{ST}} I \right)^2 d\mathbf{y}$$

where

$$\hat{\mathbf{v}}_{\text{ST}} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}, \quad \|\hat{\mathbf{v}}_{\text{ST}}\| = 1, \quad \nabla_{\text{ST}} I = \begin{pmatrix} \frac{\partial I}{\partial x_1} \\ \frac{\partial I}{\partial x_2} \\ \frac{\partial I}{\partial x_3} \end{pmatrix}$$

Spatio-temporal motion vector

- $\hat{\mathbf{v}}_{ST}$ (and \mathbf{v}_{ST}) is called the *spatio-temporal motion vector* (it is 3-dimensional)
- $\nabla_{ST} I$ is the spatio-temporal gradient of I (also 3-dimensional)
- We will minimize ε_{ST} over $\hat{\mathbf{v}}_{ST}$ with the additional constraint

$$\|\hat{\mathbf{v}}_{ST}\| = 1$$

- This is a *total least squares* formulation of how to determine $\mathbf{v}(\mathbf{x})$

Finding the minimum of ε_{ST}

- The constraint can be expressed as

$$c = \|\hat{\mathbf{v}}_{ST}\|^2 = r_1^2 + r_2^2 + r_3^2 = 1$$

- The solution is given by $\hat{\mathbf{v}}_{ST} = (r_1, r_2, r_3)$ that satisfies

$$\frac{\partial}{\partial r_k} \varepsilon = \lambda \frac{\partial}{\partial r_k} c$$

Lagrange's method
for minimisation
with constraints

for $k = 1, 2, 3$ (**why?**)

The 3D structure tensor revisited

- These 3 equations can be rewritten as

$$\left[\int_{\Omega} w(\mathbf{x}) \nabla_{ST} I \nabla_{ST}^T I d\mathbf{x} \right] \hat{\mathbf{v}}_{ST} = \lambda \hat{\mathbf{v}}_{ST}$$

(why?)

- Note that the expression inside the bracket is a 3D structure tensor!

The 3D structure tensor revisited

- We rewrite this as

$$\mathbf{T}_{3D} \hat{\mathbf{v}}_{ST} = \lambda \hat{\mathbf{v}}_{ST}$$

- This means that the $\hat{\mathbf{v}}_{ST}$ which minimizes ε must be an eigenvector of \mathbf{T}_{3D}
- It should also be normalized: $\|\hat{\mathbf{v}}_{ST}\| = 1$
- The eigenvector that minimizes ε is the one of smallest eigenvalue (**why?**)

The 3D structure tensor revisited

- Once $\hat{\mathbf{v}}_{ST} = (r_1, r_2, r_3)$ has been determined we can find \mathbf{v}_{ST} that is
 - Parallel to $\hat{\mathbf{v}}_{ST}$
 - Has its last component = 1
- The first two components of \mathbf{v}_{ST} are the motion vector $\mathbf{v} = (v_1, v_2)$

$$v_1 = \frac{r_1}{r_3} \quad v_2 = \frac{r_2}{r_3}$$

Summary

- We now have 2 alternatives to local motion estimation based on BCCE:
 1. least squares minimization
(based on \mathbf{T}_{2D} and \mathbf{s})
 2. total least squares minimization
(based on \mathbf{T}_{3D})

Summary: Least squares minimization

- Minimize

$$\varepsilon_{ST} = \int_{\Omega} w(\mathbf{x}) [\mathbf{v}_{ST} \cdot \nabla_3 I]^2 d\mathbf{x}$$

where $\mathbf{v}_{ST} = (v_1, v_2, 1)$ over the motion components
 $\mathbf{v} = (v_1, v_2)$

- Find \mathbf{v} by solving $\mathbf{T}_{2D} \mathbf{v} = \mathbf{s}$
- We can see \mathbf{v}_{ST} as a homogeneous representation of \mathbf{v}

Summary: Total least squares minimization

- Minimize

$$\varepsilon_{ST} = \int_{\Omega} w(\mathbf{x}) [\hat{\mathbf{v}}_{ST} \cdot \nabla_3 I]^2 d\mathbf{x}$$

over all components of $\hat{\mathbf{v}}_{ST} = (r_1, r_2, r_3)$ and with the constraint $\|\hat{\mathbf{v}}_{ST}\| = 1$

- Find $\hat{\mathbf{v}}_{ST}$ as the eigenvector of smallest eigenvalue with respect to \mathbf{T}_{3D}
- Find \mathbf{v} from $\hat{\mathbf{v}}_{ST}$ as $v_1 = \frac{r_1}{r_3} \quad v_2 = \frac{r_2}{r_3}$

The 3D tensor

- In the 3D case, we compute a structure tensor \mathbf{T}_{3D} , a symmetric 3×3 matrix, that can be decomposed as (the spectral theorem)

$$\mathbf{T}_{3D} = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \lambda_2 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \lambda_3 \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T$$

where $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ are the eigenvalues of \mathbf{T}_{3D} and $\hat{\mathbf{e}}_k$ are the corresponding eigenvectors (an orthonormal set)

The 3D structure tensor

- In general (*not only in the case of motion*) we can distinguish between three cases of the local 3D signal
 - The signal is constant on parallel planes (i1D)
 - The signal is constant on parallel lines (i2D)
 - The signal is isotropic
- Remember that \mathbf{T} is formed as

$$\mathbf{T}(\mathbf{x}) = \int_{\Omega_0} w(\mathbf{y}) \nabla I(\mathbf{x} + \mathbf{y}) \nabla^T I(\mathbf{x} + \mathbf{y}) d\mathbf{y}$$

The signal is constant on parallel planes (Lasagna)

- (Case 1) The 3D signal is i1D
 - The gradient $\nabla_3 I$ is always parallel to the normal vector of the planes

$$\mathbf{T} = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T$$

- \mathbf{T} has rank 1
- $\hat{\mathbf{e}}_1$ is a normal vector to the planes
- A moving 2D line generates a 3D signal that is i1D) \mathbf{T} has rank 1



The signal is constant on parallel planes

- In this case, the Fourier transform of I is concentrated along a line through the origin, in the direction of $\hat{\mathbf{e}}_1$

The signal is constant on parallel lines (Spaghetti)

- (Case 2) The 3D signal is intrinsic 2D (i2D)

- The gradient $\nabla_3 I$ is always perpendicular to the direction $\hat{\mathbf{e}}_3$ of the lines

$$\mathbf{T} = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \lambda_2 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T$$



- $\hat{\mathbf{e}}_3$ is an eigenvector of eigenvalue 0 relative to \mathbf{T}
- \mathbf{T} has rank 2
- A moving point generates a 3D signal that is i2D
 $\Rightarrow \mathbf{T}$ has rank 2

The signal is constant on parallel lines

- In this case, the Fourier transform of I is concentrated to a plane through the origin, that has $\hat{\mathbf{e}}_3$ as its normal vector
- In other words, the plane is spanned by $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$

The signal is isotropic (Dumpling)

- (Case 3) The signal varies uniformly in all directions
 - The gradient $\nabla_3 I$ is not restricted to some subspace



$$\mathbf{T} = \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \lambda_2 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \lambda_3 \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T$$

where λ_1 , λ_2 and λ_3 all are $\neq 0$.

- \mathbf{T} has rank 3
- Not consistent the BCCE

The signal is isotropic

- In the isotropic case, variations in all directions are uniformly distributed
- Implies that $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$
- We can write $\mathbf{T} = \lambda \mathbf{I}$ (\mathbf{I} is the identity tensor)
- The Fourier transform of the signal extends into all 3 dimensions

Confidence measures

- As confidence measures for the three cases we can use, *for example*:

$$c_1 = \frac{\lambda_1 - \lambda_2}{\lambda_1} \quad \text{Case 1}$$

$$c_2 = \frac{\lambda_2 - \lambda_3}{\lambda_1} \quad \text{Case 2}$$

$$c_3 = \frac{\lambda_3}{\lambda_1} \quad \text{Case 3}$$

Confidence measures

- They satisfy $c_1 + c_2 + c_3 = 1$.
- Furthermore
 - i1D-signal $\Rightarrow \mathbf{T}$ has rank 1 \Rightarrow
 $\lambda_1 > 0, \lambda_2 = \lambda_3 = 0 \Rightarrow c_1 = 1, c_2 = c_3 = 0$.
 - i2D-signal $\Rightarrow \mathbf{T}$ has rank 2 \Rightarrow
 $\lambda_1 \geq \lambda_2 > 0, \lambda_3 = 0 \Rightarrow c_2 \neq 0, c_3 = 0$.
 - Isotropic signal $\Rightarrow \mathbf{T}$ has rank 3 $\Rightarrow c_3 \neq 0$.

Decomposing \mathbf{T}

- Based on these confidence measures, \mathbf{T} can be decomposed as

$$\begin{aligned}
 \mathbf{T} &= \lambda_1 \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \lambda_2 \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \lambda_3 \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T \\
 &= (\lambda_1 - \lambda_2) \hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \\
 &\quad + (\lambda_2 - \lambda_3) (\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T) + \\
 &\quad + \lambda_3 (\hat{\mathbf{e}}_1 \hat{\mathbf{e}}_1^T + \hat{\mathbf{e}}_2 \hat{\mathbf{e}}_2^T + \hat{\mathbf{e}}_3 \hat{\mathbf{e}}_3^T) \\
 &= \lambda_1 [c_1 \mathbf{T}_{\text{rang1}} + c_2 \mathbf{T}_{\text{rang2}} + c_3 \mathbf{I}]
 \end{aligned}$$

Summary

- Given a local picture of the signal:
 - The directions along which the signal is constant correspond to the null space of \mathbf{T}
 - \mathbf{T} has a range that is orthogonal to this null space
 - In the Fourier domain: the energy is concentrated to the range of \mathbf{T}

Summary

- The rank of \mathbf{T} equals the dimension of its range
- The range represent the dimensions in the Fourier domain where there is energy
- We can define confidence measures (in various ways) that indicate which rank or case that \mathbf{T} represents
- In general, \mathbf{T} can be a combination of the different cases

Computation of the motion vector (rank 2)

- At each point (x_1, x_2, t) we can estimate the local 3D structure tensor \mathbf{T}
- If \mathbf{T} has rank 2 it corresponds to a non-1D signal in the 2D image
- Since \mathbf{T} has rank 2 we can "uniquely" determine an eigenvector of smallest eigenvalue:

$$\hat{\mathbf{v}}_{ST} = (r_1 \ r_2 \ r_3)$$

Computation of the motion vector (rank 2)

- From the previous derivations we know that

$$\hat{\mathbf{V}}_{ST} \sim \mathbf{v}_{ST} = (v_1 \ v_2 \ 1)$$

- Consequently, we can compute the motion components as

$$v_1 = \frac{r_1}{r_3} \quad v_2 = \frac{r_2}{r_3}$$

Computation of the motion vector (rank 1)

- If \mathbf{T} has rank 1 it means that the corresponding 2D-signal is 1D
 - A moving line or edge
- The null space of \mathbf{T} is 2-dimensional
- We cannot uniquely determine \mathbf{v}_{ST} , and therefore \mathbf{v} cannot be uniquely determined
- Related to the aperture problem

Computation of the motion vector (rank 1)

- However, in this case we can determine the *normal motion* of the 2D-signal
- Let $\mathbf{p}=(p_1, p_2, p_3)$ be an eigenvector of largest eigenvalue relative to \mathbf{T}

Computation of the motion vector (rank 1)

- The spatio-temporal normal motion vector \mathbf{v}_{ST} must satisfy

$$\mathbf{p}^T \mathbf{v}_{ST} = 0$$

$$p_1 v_1 + p_2 v_2 + p_3 = 0$$

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \kappa \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$$

– (why?)

1

2

Computation of the motion vector (rank 1)

- From these two relations, the normal motion is given as

$$\mathbf{v}_{\text{norm}} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = -\frac{p_3}{p_1^2 + p_2^2} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$$

Computation of the motion vector (rank 3)

- Finally, if \mathbf{T} has rank 3 this implies that the local signal does not satisfy the conditions expressed in BCCE. (**why?**)

A strategy for motion estimation

- Compute the 3D tensor \mathbf{T}_3
- Determine its eigenvalues
- Classify the tensor into each of the three cases, based on some confidence measures (**how?**)
- If rank 1: compute the normal motion
- If rank 2: compute the “true” motion
- If rank 3: no motion can be determined