Catchy Title

Author:
Espen Johansen Velsvik

Supervisor:
Markus Grasmair

NTNU

Abstract

This is the abstract...

Contents

Preface

This is the preface.

1 Introduction

When an observer of an object moves relative to the object, there is an apparent relative motion in the image plane of the observer. The problem of determining this relative motion from a sequence of images is called the Optical Flow problem. The analysis is not so much dependent on prior knowledge of the scene, but on the image sequence itself. This independency makes it applicable in many different fields.

Let $\Omega \in \mathbb{R}$ be the image plane and $f(\xi, t) \in \mathbb{R}$ the brightness pattern of some vieo sequence at $\xi \in \Omega$ and some time $t \in [0, T]$ in the video. More concisely, one wants to find flow vector components $u, v \in \mathbb{R}$ by looking at the change in the brightness $f(\xi, t)$ from one frame to another in the video sequence. This is problematic because one can not independently determine a vector of 2 components using one constraint coming from the change in image brightness at a point $\xi \in \Omega$. Thus one needs to impose other constraints to make the problem solvable.

Is there always a relationship between the change in the brightness and the movement of objects in the image? It is not hard to see that the answer is no. For instance, imagine rotating a uniform sphere exhibiting a nonuniform brightness pattern over its surface. This rotation is not observable in the image plane, and would result in an apparent zero optical flow. Also, if the illumination of the image scene changes rapidly, brightness changes in the image plane may not be due to moving objects. These examples illustrate that optical flow does not always correspond to the relative movement of an object. Nonetheless, in the following model for optical flow the image scene is assumed to be simple so that brightness changes can be directly related to object motion.

2 The Optical Flow Equation

To solve the problem of recovering the optical flow from an image sequence one need to find an equation for the flow that admits a unique solution. To find this equation a common approach is to use two assumptions, which validity can be argued. These are called the brightness constancy assumption and the spatial coherence assumption.

2.1 The Brightness constancy assumption

The optical flow equation is derived from what is called brightness constancy assumption of Horn and Schunck [3]. Let $\mathbf{r}(t) = (x(t), y(t))$ be the parametrization of some line in Ω and let

$$\frac{d\mathbf{r}}{dt} = \mathbf{w}(\mathbf{r}(t), t) = [u(\mathbf{r}(t), t), v(\mathbf{r}(t), t)]^T$$

be the flow vector along this line. The brightness constancy assumption says that a point moving with velocity $\mathbf{w}(\xi(t), t)$ along the trajectory $\mathbf{r}(t)$ over time t does not change its appearance. This means that if the scene has the same lighting, then movement of an object along a trajectory does not change its brightness. In mathematical notation this is (under perfect conditions) equivalent to the following:

$$\frac{d}{dt} f(\mathbf{r}(t), t) = 0.$$

By using the chain rule for differentiation one gets

$$\frac{\partial f}{\partial x} u + \frac{\partial f}{\partial y} v + \frac{\partial f}{\partial t} = 0,$$

or equivalently

$$\nabla f^T \mathbf{w} + f_t = 0. \tag{2.1}$$

Unfortunately, since the flow consists of 2 components, this equation is not enough to determine the flow, but only the component of the flow in the direction of the gradient, or what is known as the normal flow. This is called the Aperture problem and it is illustrated in Figure (2.1). One way to solve this problem is to introduce another constraint.

2.2 The Smoothness Constraint

As noted in the previous section, the system does not admit a unique solution with the constraints given so far. A common approach, an idea introduced by Horn

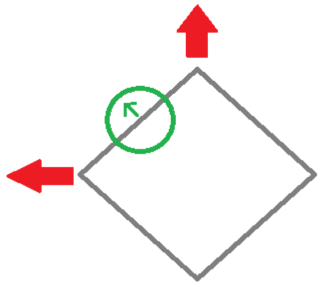

FIGURE 2.1: Illustration of the Aperture problem. Moving the gray rectangle in the direction of any of the two red arrows both give the movement shown by the green arrow if seen through the green circled aperture.

and Schunck [3], is to incorporate a smoothness constraint in the model. This smoothness constraint, also called the spatial coherence assumption [1], says that points can not move independently in the brightness pattern. There has to be some smoothness in the flow vector for points belonging to the same object. In other words, points on the same object moves with the same velocity. A natural way of obtaining a smoother solution would be to minimize some term depending on the sizes of the gradients ∇u and ∇v. As noted by Horn and Schunck, this leads to problems where one would expect discontinuous flow patterns. This is the case in images where occlusions are present, for instance an image of an object moving in a snow storm.

2.3 Variational Formulation

The aim now is to be able to combine these two constraint into one functional to be minimized. Now define

$$\rho_0(\mathbf{w}) = \nabla f \cdot \mathbf{w} + f_t.$$

If the constraint (2.1) coming from the brightness constancy assumption is to be satisfied, one would want to minimize a function that penalizes high values of ρ. For now this term is called the data term $M(u,v)$. At the same time one would want to satisfy the smoothness constraint, that is, one would want to define a smoothness function that depends on the values of the gradients ∇u and ∇v. Let this smoothness function be denoted as $V(\nabla u, \nabla v)$. To minimize both $M(u,v)$ and $V(\nabla u, \nabla v)$ one forms the following global energy function:

$$E(u,v) = \frac{1}{2}\int_{\Omega} \left(M(u,v) + \frac{1}{\sigma^2} V(\nabla u, \nabla v)\right) dx\, dy, \tag{2.2}$$

where $\sigma > 0$ is a regularization parameter. The problem is now to find the minimum of the energy functional $E(u,v)$. From calculus of variations we have that if $\mathbf{w}$

minimizes a functional

$$J(\mathbf{w}) = \iint_{\Omega} F(x, y, \mathbf{w}, \mathbf{w}_x, \mathbf{w}_y)\, dx\, dy,$$

then the first variation must be zero,

$$\delta J(\mathbf{w}; \boldsymbol{\eta}) = \frac{d}{d\epsilon}\left[J(\mathbf{w} + \epsilon\boldsymbol{\eta})\right] = 0,$$

at $\epsilon = 0$ for any arbitrary function $\boldsymbol{\eta}(x, y)$. We get

$$\begin{aligned}
\delta J(\mathbf{w}; \boldsymbol{\eta}) &= \iint_{\Omega} \frac{d}{d\epsilon}\Big|_{\epsilon=0} F(x, y, \mathbf{w} + \epsilon\boldsymbol{\eta}, \mathbf{w}_x + \epsilon\boldsymbol{\eta}_x, \mathbf{w}_y + \epsilon\boldsymbol{\eta}_y)\, dx\, dy \\
&= \iint_{\Omega} \boldsymbol{\eta} F_{\mathbf{w}} + \boldsymbol{\eta}_x F_{\mathbf{w}_x} + \boldsymbol{\eta}_y F_{\mathbf{w}_y}\, dx\, dy \\
&= \iint_{\Omega} \boldsymbol{\eta} F_{\mathbf{w}} + \frac{d}{dx}(\boldsymbol{\eta} F_{\mathbf{w}_x}) + \frac{d}{dy}(\boldsymbol{\eta} F_{\mathbf{w}_y}) - \boldsymbol{\eta}\left(\frac{d}{dx} F_{\mathbf{w}_x} + \frac{d}{dy} F_{\mathbf{w}_y}\right) dx\, dy.
\end{aligned}$$

Now let Γ_E, Γ_W, Γ_N and Γ_S be the east, west, north and south boundary of our domain respectively. Then using Gauss' Theorem gives

$$\begin{aligned}
&\iint_{\Omega} \frac{d}{dx}(\boldsymbol{\eta} F_{\mathbf{w}_x}) + \frac{d}{dy}(\boldsymbol{\eta} F_{\mathbf{w}_y})\, dx\, dy \\
&= \int_{\Gamma_E} \boldsymbol{\eta} F_{\mathbf{w}_x}\, dx - \int_{\Gamma_W} \boldsymbol{\eta} F_{\mathbf{w}_x}\, dx + \int_{\Gamma_N} \boldsymbol{\eta} F_{\mathbf{w}_y}\, dy - \int_{\Gamma_S} \boldsymbol{\eta} F_{\mathbf{w}_y}\, dy
\end{aligned}$$

Using this result, we get

$$\begin{aligned}
\delta J(\mathbf{w}; \boldsymbol{\eta}) = &\iint_{\Omega} \boldsymbol{\eta}\left(F_{\mathbf{w}} - \frac{d}{dx} F_{\mathbf{w}_x} - \frac{d}{dy} F_{\mathbf{w}_y}\right) dx\, dy \\
&+ \left(\int_{\Gamma_E} \boldsymbol{\eta} F_{\mathbf{w}_x}\, dx - \int_{\Gamma_W} \boldsymbol{\eta} F_{\mathbf{w}_x}\, dx + \int_{\Gamma_N} \boldsymbol{\eta} F_{\mathbf{w}_y}\, dy - \int_{\Gamma_S} \boldsymbol{\eta} F_{\mathbf{w}_y}\, dy\right) = 0.
\end{aligned}$$

Since this must hold for any arbitrary function $\boldsymbol{\eta}(x, y)$ it follows that

$$\begin{aligned}
F_{\mathbf{w}} - \frac{d}{dx} F_{\mathbf{w}_x} - \frac{d}{dy} F_{\mathbf{w}_y} &= 0 \quad \text{in } \Omega \\
F_{\mathbf{w}_x} &= 0 \quad \text{on } \Gamma_e \text{ and } \Gamma_w \\
F_{\mathbf{w}_y} &= 0 \quad \text{on } \Gamma_n \text{ and } \Gamma_s
\end{aligned}$$

This is called the Euler-Lagrange equation of variational calculus. From this result it is easy to see that the following must hold for (2.2):

$$\begin{aligned} \partial_{\mathbf{w}} M - \frac{1}{\sigma^2}\left(\frac{d}{dx}\partial_{\mathbf{w}_x} V + \frac{d}{dy}\partial_{\mathbf{w}_y} V\right) &= 0 \quad \text{in } \Omega, \\ \partial_{\mathbf{w}_x} V &= 0 \quad \text{on } \Gamma_E \text{ and } \Gamma_W, \\ \partial_{\mathbf{w}_y} V &= 0 \quad \text{on } \Gamma_N \text{ and } \Gamma_S. \end{aligned} \tag{2.3}$$

2.4 Penalizing The Data Term

When the brightness constancy assumption was first introduced, Horn and Schunck used a quadratic penalization of ρ,which is equivalent to least-squares minimization, but other penalization functions have been proposed. Least-squares minimization is very sensitive to outliers. In the case of the data term, these outliers would be noise in the image (a pixel that jumps from low intensity to high intensity without corresponding to the motion of an object). Black and Anandan [1] proposed several subquadratic penalizer functions, arguing that a subquadratic penalizer would improve the robustness in the presence of outliers. In general the data term $M(u,v)$ can be written as

$$M(u,v) = \Psi_M(\rho_0^2), \tag{2.4}$$

where $\Psi_M(s^2)$ is some penalizing function aiming to minimize ρ_0^2.

2.5 Penalizing The Smoothness Term

The smoothness terms considered here will be quadratically dependent on the gradient in each direction, thus it is convenient to write the smoothness term in the form

$$V(\nabla u, \nabla v) = \Psi_V(\nabla u^T \Theta_u \nabla u) + \Psi_V(\nabla v^T \Theta_v \nabla v), \tag{2.5}$$

where $\Psi_V(s^2)$ is a penalizing function that can be either quadratic or subquadratic, and $\Theta_u = \Theta_u(x,y,u,v)$ and $\Theta_v = \Theta_v(x,y,u,v)$ are matrices steering the diffusion process of the Euler-Lagrange system (2.3). Their eigenvectors and corresponding eigenvalues gives the direction and magnitude of smoothing respectively.

2.6 The Euler-Lagrange System

After defining the data term and the smoothness term using penalizing functions one can see that equation (2.3) takes the form

$$\begin{aligned} \Psi_M'(\rho_0^2)(f_x u + f_y v + f_t) f_x - \frac{1}{\sigma^2}\operatorname{div}\left(\Psi_V'(\nabla u^T \Theta_u \nabla u)\Theta_u \nabla u\right) &= 0 \\ \Psi_M'(\rho_0^2)(f_x u + f_y v + f_t) f_y - \frac{1}{\sigma^2}\operatorname{div}\left(\Psi_V'(\nabla v^T \Theta_v \nabla v)\Theta_v \nabla v\right) &= 0 \end{aligned} \tag{2.6}$$

for $(x,y) \in \Omega$ with Neumann boundary conditions. The theoretical framework presented up to this point is the same for all the methods considered here. The main distinction for each method will be across which boundaries the flow field is smoothed, that is, the choice of diffusion matrix. We start with the simplest choice; the uniform smoothness approach by Horn and Schunck.

3 The Data Term

The goal of the penalization of the data term is to make

$$\rho_0(\mathbf{w}) = (\nabla f^T \mathbf{w} + f_t),$$

as close to zero as possible. This is commonly done by penalizing high values of

$$\rho_0^2 = (\nabla f^T \mathbf{w} + f_t)^2.$$

3.1 Normalization

[7] reported that normalizing the data term can be beneficial. This can be seen from rewriting the squared term above as

$$\begin{aligned}
(\nabla f^T \mathbf{w} + f_t)^2 &= \left[|\nabla f| \left((\frac{\nabla f^T \mathbf{w}}{|\nabla f|} + \frac{f_t}{|\nabla f|} \right) \right]^2 \\
&= |\nabla f|^2 \left[\frac{\nabla f^T}{|\nabla f|} \left(\mathbf{w} + \frac{f_t \nabla f}{|\nabla f|^2} \right) \right]^2 \\
&= |\nabla f|^2 \left[\frac{\nabla f^T}{|\nabla f|} (\mathbf{w} - \mathbf{w}_n) \right]^2,
\end{aligned}$$

where

$$\mathbf{w}_n = -\frac{f_t \nabla f}{|\nabla f|^2} \tag{3.1}$$

is called the normal flow, which was briefly mentioned in section (2.1). Now define

$$d = \frac{\nabla f^T}{|\nabla f|} (\mathbf{w} - \mathbf{w}_n), \tag{3.2}$$

so that

$$\rho_0(u, v)^2 = |\nabla f|^2 d^2. \tag{3.3}$$

Let the flow vector $\mathbf{w}$ define a point in the uv-plane. The image gradient ∇f is then orthogonal to the line defined by $\rho_0(u, v) = 0$. Indeed, for two points $\mathbf{w}_1$ and $\mathbf{w}_2$, both satisfying $\rho_0(u, v) = 0$,

$$\nabla f^T (\mathbf{w}_1 - \mathbf{w}_2) = f_t - f_t = 0.$$

From this one can conclude that d is the distance from the line defined by $\rho_0(u, v) = 0$ and the point $\mathbf{w}$ in the uv-plane. A point with $d = 0$ would satisfy the constraint.

From this geometric interpretation Zimmer et al. [7] suggested that one should ideally use d^2 to penalize the data term, which is weighted by the square of the image gradient in the expression for ρ_0^2, as seen in (3.3). Thus the normalized data constraint is

$$\bar{\rho}_0 = \theta_0(\nabla f^T \mathbf{w} + f_t), \tag{3.4}$$

where the normalisation factor θ_0 is defined as

$$\theta_0 = \frac{1}{|\nabla f|^2 + \zeta^2}.$$

The regularization parameter $\zeta > 0$ avoids division by zero and simultaneously reduces the effect of small gradients.

3.2 The Gradient Constancy Assumption

In the model so far one has assumed that the illumination is the same for the whole scene, but this assumption is very likely to be violated. Thus, to make the model more robust against additive illumination changes in the image scene [references comes here] proposed to include a constraint regarding the gradients of the brightness. The assumption is called the gradient constancy assumption, and it says that gradients remain constant under their displacement, that is

$$\frac{d}{dt}\nabla f(\mathbf{r}(t), t)) = 0, \tag{3.5}$$

which gives

$$\nabla f_x \mathbf{w} + f_{xt} = 0 \qquad \text{and} \qquad \nabla f_y \mathbf{w} + f_{yt} = 0. \tag{3.6}$$

To combine the two constancy assumption into one constraint, define the normalized penalization terms coming from the gradient constancy assumption as

$$\bar{\rho}_x(\mathbf{w}) = \theta_x(\nabla f_x \mathbf{w} + f_{xt}) \qquad \text{and} \qquad \bar{\rho}_y(\mathbf{w}) = \theta_y(\nabla f_y \mathbf{w} + f_{yt}), \tag{3.7}$$

where

$$\theta_x = \frac{1}{|\nabla f_x|^2 + \zeta^2}. \qquad \text{and} \qquad \theta_y = \frac{1}{|\nabla f_y|^2 + \zeta^2},$$

and let the normalized joint penalization term $\bar{\rho}$ be defined by the following equation

$$\begin{aligned}\bar{\rho}(\mathbf{w})^2 &= \bar{\rho}_0(\mathbf{w})^2 + \gamma(\bar{\rho}_x(\mathbf{w})^2 + \bar{\rho}_y(\mathbf{w})^2) \\ &= \bar{\rho}_0(\mathbf{w})^2 + \gamma\bar{\rho}_{xy}(\mathbf{w})^2.\end{aligned}$$

3.3 Robustness Enhancements

As noted in (2.4) the data term can be penalized in different ways. Horn and Schunck [3] used a quadratic penalizer,

$$M(u,v) = \rho(u,v)^2 = (\nabla f^T \mathbf{w} + f_t)^2. \tag{3.8}$$

but as previously asserted this penalizer has relatively poor robustness against outliers. [Brox et al., 2004 Reference coming soon] used the subquadratic penaliser

$$\Psi_M(s^2) = \sqrt{s^2 + \epsilon^2},$$

with a small regularization parameter $\epsilon > 0$. A separate penalization of the constraints coming from the two constancy assumptions was proposed by [Bruhn and Weickert, 2005], arguing that this would be constructive if one of the assumptions produces an outlier. Using a separate penalization gives the data term

$$M(u,v) = \Psi_M(\bar{\rho}_0^2) + \gamma \Psi_M(\bar{\rho}_{xy}^2). \tag{3.9}$$

4 The Approach of Horn and Schunck

The research of Horn and Shunck has formed the basis of further research in the field of optical flow. They proposed the following quadratic penalized data term

$$M(\mathbf{w}) = (\nabla f^T \mathbf{w} + f_t)^2, \tag{4.1}$$

which is equivalent to choosing

$$\Psi_M(\rho^2) = \rho^2$$

in the framework of (2.4). The contribution to (2.3) from the data term is thus

$$\partial_{\mathbf{w}} M = 2\nabla f(\nabla f^T \mathbf{w} + f_t) \tag{4.2}$$

The smoothness term used by Horn and Schunck is

$$V(\nabla u, \nabla v) = |\nabla u|^2 + |\nabla v|^2.$$

This is a homogeneous regularizer which means that it applies an equal amount of diffusion in all directions. In the framework of (2.6), this is equivalent to the diffusion matrices Θ_u and Θ_v being the identity matrix. Using this function as a flow regularizer gives

$$\partial_{\mathbf{w}_{x_i}} V = 2\mathbf{w}_{x_i},$$

for $x_i = x, y$. Dividing by 2 in all terms results in (2.3) taking the form

$$\begin{aligned}
(f_x u + f_y v + f_t) f_x - \frac{1}{\sigma^2}\left(\frac{d}{dx} u_x + \frac{d}{dy} u_y\right) &= 0 \quad \text{in } \Omega, \\
(f_x u + f_y v + f_t) f_y - \frac{1}{\sigma^2}\left(\frac{d}{dx} v_x + \frac{d}{dy} v_y\right) &= 0 \quad \text{in } \Omega \\
\mathbf{w}_x &= 0 \quad \text{on } \Gamma_e \text{ and } \Gamma_w, \\
\mathbf{w}_y &= 0 \quad \text{on } \Gamma_n \text{ and } \Gamma_s,
\end{aligned} \tag{4.3}$$

which can be seen as a system of coupled Poisson equations with Neumann boundary conditions:

$$\begin{aligned}
-\frac{1}{\sigma^2}\Delta u + f_x^2 u &= -(F(v) + f_t f_x) \\
-\frac{1}{\sigma^2}\Delta v + f_y^2 v &= -(F(u) + f_t f_y),
\end{aligned}$$

where $F(q) = f_x f_y q$.

4.1 Discretizing the Horn and Schunck method

Let now our image be of size m-by-n, and let f^1 and f^2 be the image at $t = 1$ an $t = 2$ respectively. Also, we flatten the regular 2-dimensional grid in Ω and consider now $(\xi^i)_{i\in[mn]}$ so that $(\xi^i) = (x, y) = (\lfloor i/m \rfloor, i - \lfloor i/m \rfloor)$, when assuming the distance between vertical and horizontal grid points in Ω are 1. The corresponding vector representation of the image f is denoted as $\mathbf{f}(\xi^i) \in \mathbb{R}^{mn}$. Continuing this notation, the discrete flow values $\mathbf{w}(\xi^i)$ is represented as the following vector in $\mathbb{R}^{2mn}$:

$$\mathbf{w}(\xi^i) = \begin{bmatrix} u(\xi^i)_{i\in[mn]} \\ v(\xi^i)_{i\in[mn]} \end{bmatrix}.$$

For the discretization of the image gradients in (4.3), the forward difference was used on f^1, producing the two vectors $\mathbf{d}_x(\xi^i)$ and $\mathbf{d}_y(\xi^i)$ in $\mathbb{R}^{mn)}$, where the gradients on the boundary are assumed to be zero. The time derivative f_t is discretized using forward difference with time step $\Delta t = t_2 - t_1 = 1$ as shown below:

$$\mathbf{c}(\xi^i) = \mathbf{f}^2(\xi^i) - \mathbf{f}^1(\xi^i),$$

where $\mathbf{c}(\xi^i)$ is a vector in $\mathbb{R}^{mn}$. Normally when choosing derivative approximations one wants as high order as possible so that the truncation error goes to zero as one increases the number of grid points. In this case the number of grid points is fixed, so there is little to gain from choosing higher order approximations; it is best to choose the derivative approximation that results in the simplest discretization. Thus for the flow vector in (4.3), the first derivative was approximated using backward difference, and the second was approximated using forward difference. Let now L_x and L_y be the matrices performing a forward difference on the components of the vector $\mathbf{w}$ in x- and y-direction respectively. The gradient is then represented as

$$L\mathbf{w}(\xi^i) = \begin{bmatrix} \tilde{u}_x(\xi^i)_{i\in[mn]} \\ \tilde{u}_y(\xi^i)_{i\in[mn]} \\ \tilde{v}_x(\xi^i)_{i\in[mn]} \\ \tilde{v}_y(\xi^i)_{i\in[mn]} \end{bmatrix}$$

where $\tilde{u}_x, \tilde{v}_x, \tilde{u}_y, \tilde{v}_y \in \mathbb{R}^{mn}$ are vector approximations to the derivatives. This means that L is the following block matrix:

$$L = \begin{bmatrix} L_x & 0 \\ L_y & 0 \\ 0 & L_x \\ 0 & L_y \end{bmatrix}.$$

Since (4.3) gives one set of equations for the interior nodes and one for the boundary nodes, these have to separated into two coupled systems. The interior system can be written as

$$(D^T D + \frac{1}{\sigma^2} L^T L)\mathbf{w}(\xi^i) = -D^T \mathbf{c}(\xi^i), \tag{4.4}$$

for $(\xi^i) \in \Omega$ where D is the block matrix

$$D = \left[\begin{array}{c|c} D_x & D_y \end{array}\right].$$

D_x and D_y are diagonal matrices with the elements of $\mathbf{d}_x(\xi^i)$ and $\mathbf{d}_y(\xi^i)$ for $(\xi^i) \in \Omega$ along its diagonals respectively. When using a forward difference approximation of the derivative, the derivative approximations in the points next to Γ_E and Γ_S on the grid will be dependent on points on these boundaries respectively, and the derivative on these boundaries are set to zero. Let $\alpha(\xi^i) = (\mathbf{d}_x(\xi^i)u(\xi^i) + \mathbf{d}_y(\xi^i)v(\xi^i) + \mathbf{c}(\xi^i))$. For $(\xi^{i+m}) \in \Gamma_E$ one gets

$$\alpha(\xi^i)\mathbf{d}_x(\xi^i) - \frac{1}{\sigma^2}\left[L_x^T\left(u(\xi^i) - u(\xi^{(i+m)})\right) + L_y^T\left(u(\xi^i) - u(\xi^{(i+1)})\right)\right] = 0$$
$$\alpha(\xi^i)\mathbf{d}_y(\xi^i) - \frac{1}{\sigma^2}\left[L_x^T\left(v(\xi^i) - v(\xi^{(i+m)})\right) + L_y^T\left(v(\xi^i) - v(\xi^{(i+1)})\right)\right] = 0,$$

but since the derivative on the boundary is zero, one must enforce that $-L_x^T q(\xi^{i+m}) = 0$ for $q = u, v$. This leads to

$$u(\xi^i) = u(\xi^{(i+m)}) \tag{4.5}$$
$$v(\xi^i) = v(\xi^{(i+m)}) \tag{4.6}$$

so

$$\alpha(\xi^i)\mathbf{d}_x(\xi^i) - \frac{1}{\sigma^2}L_y^T L_y u(\xi^i) = 0$$
$$\alpha(\xi^i)\mathbf{d}_y(\xi^i) - \frac{1}{\sigma^2}L_y^T L_y v(\xi^i) = 0.$$

Equivalently when $(\xi^{i+1}) \in \Gamma_S$,

$$u(\xi^i) = u(\xi^{(i+1)}) \tag{4.7}$$
$$v(\xi^i) = v(\xi^{(i+1)}) \tag{4.8}$$

which results in the following equations:

$$\alpha(\xi^i)\mathbf{d}_x(\xi^i) - \frac{1}{\sigma^2}L_x^T L_x u(\xi^i) = 0$$
$$\alpha(\xi^i)\mathbf{d}_y(\xi^i) - \frac{1}{\sigma^2}L_x^T L_x v(\xi^i) = 0.$$

On the two other boundaries Γ_W and Γ_N, enforcing the forward differences to be zero at $(\xi^i) \in \Gamma_W$ leads to

$$u(\xi^i) = u(\xi^{(i+m)}) \tag{4.9}$$
$$v(\xi^i) = v(\xi^{(i+m)}), \tag{4.10}$$

and likewise for $(\xi^i) \in \Gamma_N$,

$$u(\xi^i) = u(\xi^{(i+1)}) \tag{4.11}$$
$$v(\xi^i) = v(\xi^{(i+1)}). \tag{4.12}$$

FIGURE 4.1: First image in the Hamburg taxi sequence.

FIGURE 4.2: Second image in the Hamburg taxi sequence.

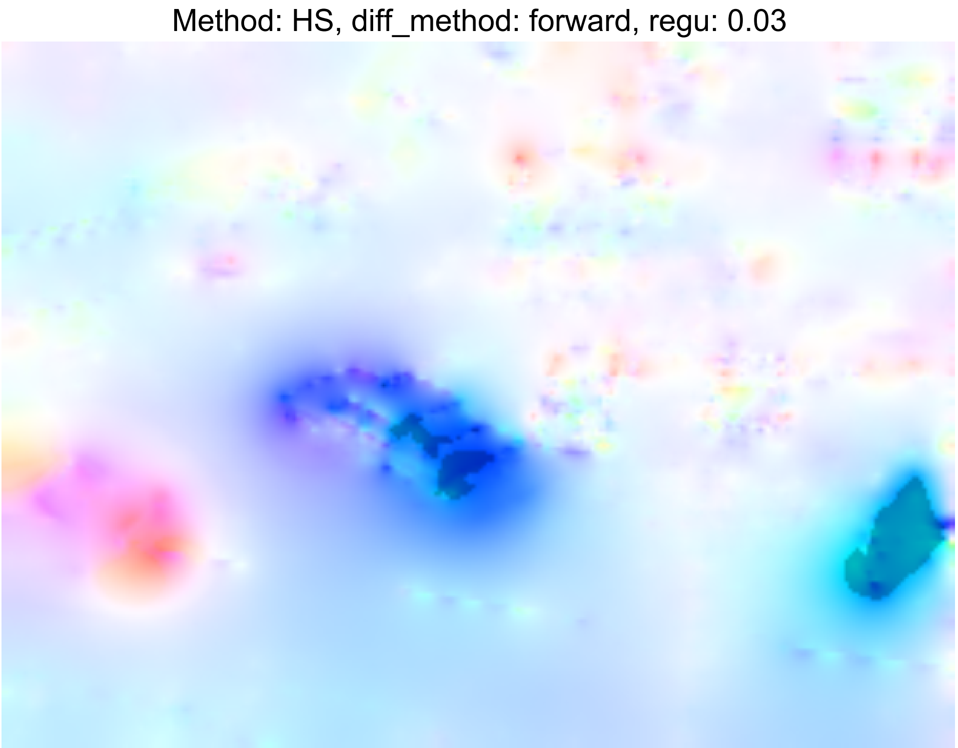

FIGURE 4.3: Result for Horn and Schunck with regularization 0.3 and quadratic data term penalization.

4.2 Results for the Horn and Schunck method

Results for the HS method.... quadratic and subquadratic penaliser

5 Anisotropic Image-Driven Regularization

The homogeneous regularizer of Horn and Schunck smoothes the flow field in all directions. This is a desirable property for internal pixels of an object, but not for pixels on the flow boundary. A homogeneous regularizer would smooth the flow field at pixels where one would expect the flow field to be discontinuous, which results in blurry flow edges.

5.1 Regularization according to image structure

As one of the assumptions to the optical flow model is that different object has different brightness patterns, one would expect that flow boundaries coincide with image edges. Thus an amendment to this issue is to make a smoothness term which takes the gradients of the image into account, and smooths the flow field along image edges instead of across them. Such methods are called image-driven regularization methods. The anisotropic reguralizer of Nagel and Enkelmann [4] performs smoothing along the image gradients and prevents smoothing across image edges. This is done by introducing a regularised projection matrix P, defined as

$$P(\nabla f) = \frac{1}{|\nabla f|^2 + 2\kappa^2}(\nabla^{\perp} f (\nabla^{\perp})^T + \kappa^2 I),$$

where $\nabla^{\perp} f = \left[-f_y, f_x\right]^T$, and $\kappa > 0$ is a regularization parameter. The smoothness term of Nagel and Enkelmann can now be written as the following

$$V_{AI}(\nabla u, \nabla v) = \nabla^T u P(\nabla f)\nabla u + \nabla^T v P(\nabla f)\nabla v,$$

or written out

$$V_{AI}(\nabla u, \nabla v) = \frac{\kappa^2}{|\nabla f|^2 + 2\kappa^2}\left(u_{\mathbf{s}_1}^2 + v_{\mathbf{s}_1}^2\right) + \frac{|\nabla f|^2 + \kappa^2}{|\nabla f|^2 + 2\kappa^2}\left(u_{\mathbf{s}_2}^2 + v_{\mathbf{s}_2}^2\right),$$

where

$$\mathbf{s}_1 = \frac{1}{|\nabla f|}\begin{bmatrix} f_x \\ f_y \end{bmatrix}, \qquad \mathbf{s}_2 = \frac{1}{|\nabla f|}\begin{bmatrix} -f_y \\ f_x \end{bmatrix}, \tag{5.1}$$

and $q_{\mathbf{s}_i} = \mathbf{s}_i^T \nabla u$ for $q = u, v$. That is, $q_{\mathbf{s}_i}$ is the directional derivative of q in the direction of the image gradient ($i = 1$) or the orthogonal direction ($i = 2$). This means that setting $\Theta = P$ in (2.6) steers the diffusion so that flow vectors are smoothed along image edges and not across them.

5.1.1 Discretizing the Nagel and Enkelmann smoothness term

Setting $\Theta = P$ in (2.6) leads to the following Euler-Lagrange system:

$$\frac{\partial M}{\partial u} - \frac{1}{\sigma^2}\mathrm{div}(P\nabla u) = 0$$
$$\frac{\partial M}{\partial v} - \frac{1}{\sigma^2}\mathrm{div}(P\nabla v) = 0.$$

Using the same discretizations for the derivatives as in 4.1 one gets

$$(D^T D + \frac{1}{\sigma^2}L^T P L)\mathbf{w} = -D^T\mathbf{c}$$

for the internal pixels and with Neumann boundary conditions given in equations (4.5) to (4.12). (Maybe some more details on discretization?)

5.1.2 Results for the anisotropic image-driven regularization

Experiments were run to find the best regularization parameter in the Nagel and Enkelmann smoothness term. The regularization parameter σ found for the Horn and Schunck method is used to regularize the whole smoothness term. The resulting flow field for different choices of the regularization parameter κ, while keeping σ constant, is shown in Figure (**??**). It is seen that choosing $\kappa = 0.8$ gives a fairly good segmentation of the object. Since the values for the gradient from the sobel derivative are relatively high, it is expected to also having to choose a relatively large value for κ for sufficient regularization. Figure (**??**) compares the anisotropic smoothness term of Nagel and Enkelmann with $\kappa = 0.8$ with the homogeneous smoothness term of Horn and Schunck, both with using regularization parameter $\sigma = 0.003$.

6 The Isotropic Flow-Driven Method

One drawback of the image-driven approach to regularization is that there is often a great deal of oversegmentation, since image boundaries are not necessarily flow boundaries. The solution is to decrease the smoothing on flow boundaries, a so called flow-driven approach. But an obvious problem is that the flow boundaries are not known a priori.

6.1 Subquadratic Penalization

Shulman and Herve [5] suggested using a subquadratic penalizer instead of a quadratic one, arguing that a quadratic penalizer assumes a Gaussian distribution of the flow gradients which would penalize large gradients, assumed to correspond to flow boundaries, too much. The smoothness term can be written as

$$\begin{aligned} V_{IF}(\nabla u, \nabla v) &= \psi_V\left(|\nabla u|^2 + |\nabla v|^2\right) \\ &= \psi_V\left(u_{\mathbf{s}_1}^2 + u_{\mathbf{s}_2}^2 + v_{\mathbf{s}_1}^2 + v_{\mathbf{s}_2}^2\right), \end{aligned}$$

where $\psi_V(s^2)$ is some subquadratic penalizer function performing a nonlinear isotropic diffusion. The contribution to (2.3) is $\nabla \cdot (\partial_{u_x} V, \partial_{u_y} V)$. Computing the individual components, we get

$$\begin{aligned} \partial_{u_x} V &= 2\psi_V'\left(u_{\mathbf{s}_1}^2 + u_{\mathbf{s}_2}^2 + v_{\mathbf{s}_1}^2 + v_{\mathbf{s}_2}^2\right) u_x \\ \partial_{u_y} V &= 2\psi_V'\left(u_{\mathbf{s}_1}^2 + u_{\mathbf{s}_2}^2 + v_{\mathbf{s}_1}^2 + v_{\mathbf{s}_2}^2\right) u_y, \end{aligned}$$

and equivalent for $(\partial_{v_x} V, \partial_{v_y} V)$. Thus the diffusion matrix of (2.6) is given as

$$\Theta_{IF} = 2\psi_V'\left(u_{\mathbf{s}_1}^2 + u_{\mathbf{s}_2}^2 + v_{\mathbf{s}_1}^2 + v_{\mathbf{s}_2}^2\right) I,$$

where I is the identity matrix, which is seen to be a function of the flow gradient. As a convex penaliser, Cohen (1993) suggested the following total variation regulariser:

$$\psi_V(s^2) = \sqrt{s^2 + \epsilon^2}, \tag{6.1}$$

which gives

$$\psi_V'(s^2) = \frac{1}{2\sqrt{s^2 + \epsilon^2}}.$$

This penaliser function results in the following Euler-Lagrange system

$$\begin{aligned} \partial_u M - \frac{1}{\sigma^2}\left(\frac{\partial}{\partial x}\left[\frac{u_x}{\sqrt{|\nabla u|^2 + |\nabla v|^2 + \epsilon^2}}\right] + \frac{\partial}{\partial y}\left[\frac{u_y}{\sqrt{|\nabla u|^2 + |\nabla v|^2 + \epsilon^2}}\right]\right) &= 0 \\ \partial_v M - \frac{1}{\sigma^2}\left(\frac{\partial}{\partial x}\left[\frac{v_x}{\sqrt{|\nabla u|^2 + |\nabla v|^2 + \epsilon^2}}\right] + \frac{\partial}{\partial y}\left[\frac{v_y}{\sqrt{|\nabla u|^2 + |\nabla v|^2 + \epsilon^2}}\right]\right) &= 0, \end{aligned} \tag{6.2}$$

which is clearly a non-linear system. (Something more about TV regularizers? Rudin et al.)

6.1.1 The Lagged Diffusivity Fixed Point Method

The non-linear system (6.2) is can be solved using the method of lagged diffusivity. Chan and Mulet [2] showed that the lagged diffusivity fixed point method used in TV restoration of images can be traced back to Weiszfeld's method of minimizing Euclidean lengths, and for which global and linear convergence can be shown. From (6.2) we can define the fixed point method as

$$\begin{aligned}
\partial_{u^{k+1}} M - \frac{1}{\sigma^2}\left(\frac{\partial}{\partial x}\left[\frac{u_x^{k+1}}{\sqrt{|\nabla u^k|^2 + |\nabla v^k|^2 + \epsilon^2}}\right] + \frac{\partial}{\partial y}\left[\frac{u_y^{k+1}}{\sqrt{|\nabla u^k|^2 + |\nabla v^k|^2 + \epsilon^2}}\right]\right) = 0 \\
\partial_{v^{k+1}} M - \frac{1}{\sigma^2}\left(\frac{\partial}{\partial x}\left[\frac{v_x^{k+1}}{\sqrt{|\nabla u^k|^2 + |\nabla v^k|^2 + \epsilon^2}}\right] + \frac{\partial}{\partial y}\left[\frac{v_y^{k+1}}{\sqrt{|\nabla u^k|^2 + |\nabla v^k|^2 + \epsilon^2}}\right]\right) = 0,
\end{aligned}$$

where we solve for u^{k+1} and v^{k+1} given u^k and v^k from the previous iteration. This is a linear system in each iteration, and it can be solved in the same manner as the previous methods. The system to be solved at iteration $k+1$ is

$$(D^T D + \frac{1}{\sigma^2} L^T \Theta^k L)\mathbf{w}(\xi)^{k+1} = -D^T \mathbf{c}(\xi),$$

for $\xi \in \Omega$, where Θ^k is the following 4mn-by-4mn diagonal block matrix

$$\Theta^k = \left[\begin{array}{c|c|c|c} \chi^k & 0 & 0 & 0 \\ \hline 0 & \chi^k & 0 & 0 \\ \hline 0 & 0 & \chi^k & 0 \\ \hline 0 & 0 & 0 & \chi^k \end{array}\right].$$

The submatrix χ^k is the mn-by-mn diagonal matrix with the values

$$\begin{aligned}
\chi_i^k &= 2\psi_V'\left(|\nabla \mathbf{w}(\xi^i)|^2\right) \\
&= \frac{1}{\sqrt{|\mathbf{w}(\xi^i)|^2 + \epsilon^2}}
\end{aligned}$$

along its diagonal.

6.1.2 Results for Isotropic flow driven

....

7 Image- and Flow-Driven Regularization

The image driven methods smooths the flow field along image boundaries, that is in the direction of $\nabla f^{\perp}$, while flow driven methods aims to decrease the smoothing at flow boundaries. Sun et al. [6] combined these two methods to construct an anisotropic image- and flow-driven regularizer in a statistical learning framework.

7.1 The Anisotropic regularizer

The continuous regularizer as presented by Zimmer et al. [7] is as follows:

$$V(\nabla u, \nabla v) = \psi_V(u_{\mathbf{s}_1}^2) + \psi_V(u_{\mathbf{s}_2}^2) + \psi_V(v_{\mathbf{s}_1}^2) + \psi_V(v_{\mathbf{s}_2}^2). \tag{7.1}$$

This regularizer smooths the flow field along image boundaries, but steers the smoothing strength according to the flow edges. Let $\mathbf{s}_i = [s_{i1}, s_{i2}]^T$. The contribution to the Euler-Lagrange system for $q \in u, v$ is

$$\begin{aligned}&\partial_x(\partial_{q_x} V) + \partial_y(\partial_{q_y} V)\\&= 2\partial_x\left(\psi_V'(q_{\mathbf{s}_1}^2)q_{\mathbf{s}_1}s_{11} + \psi_V'(q_{\mathbf{s}_2}^2)q_{\mathbf{s}_2}s_{21}\right) + 2\partial_y\left(\psi_V'(q_{\mathbf{s}_1}^2)q_{\mathbf{s}_1}s_{12} + \psi_V'(q_{\mathbf{s}_2}^2)q_{\mathbf{s}_2}s_{22}\right),\end{aligned}$$

which can be written as

$$\begin{aligned}&2\nabla^T\begin{bmatrix}\psi_V'(q_{\mathbf{s}_1}^2)\left(s_{11}^2 q_x + s_{11}s_{12}q_y\right) + \psi_V'(q_{\mathbf{s}_2}^2)\left(s_{21}^2 q_x + s_{21}s_{22}q_y\right)\\ \psi_V'(q_{\mathbf{s}_1}^2)\left(s_{11}s_{12}q_x + s_{12}^2 q_y\right) + \psi_V'(q_{\mathbf{s}_2}^2)\left(s_{21}s_{22}q_x + s_{22}^2 q_y\right)\end{bmatrix}\\&= 2\nabla^T\left(\psi_V'(q_{\mathbf{s}_1}^2)\mathbf{s}_1^T\mathbf{s}_1\nabla q + \psi_V'(q_{\mathbf{s}_2}^2)\mathbf{s}_2^T\mathbf{s}_2\nabla q\right)\\&= 2\nabla^T\left(\psi_V'(q_{\mathbf{s}_1}^2)\mathbf{s}_1^T\mathbf{s}_1 + \psi_V'(q_{\mathbf{s}_2}^2)\mathbf{s}_2^T\mathbf{s}_2\right)\nabla q.\end{aligned}$$

From the expression above one can conclude that the diffusion matrix of equation (2.6) is

$$\Theta_q = \psi_V'(q_{\mathbf{s}_1}^2)\mathbf{s}_1^T\mathbf{s}_1 + \psi_V'(q_{\mathbf{s}_2}^2)\mathbf{s}_2^T\mathbf{s}_2 \tag{7.2}$$

for $q \in u, v$ with the vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ given in (5.1).

7.2 TV functional as a Convex regularizer

Now, choosing.... (TV functional, discretization, lagged diffusivity...)

8 Rotational Invariant Constraint-Driven Regularization

Zimmer et. al [7] argued that the directional information coming from the eigenvectors $\mathbf{s}_1$ and $\mathbf{s}_2$ are inconsistent with the imposed constraints on the data term $M(u,v)$ if the $M(u,v)$ is designed to take into account the constraints (3.6) coming from the gradient constancy assumption. They opted for a regularization term that instead of steering the diffusion process along the image edges, steers the diffusion along constraint edges, defined by the eigenvectors $\mathbf{r}_1$ and $\mathbf{r}_2$ of the regularization matrix

$$R_\mu = K_\mu * \Big[\theta_0(\nabla f \nabla f^T) + \gamma\Big(\theta_x(\nabla f_x \nabla f_x^T) + \theta_y(\nabla f_y \nabla f_y^T)\Big)\Big]. \tag{8.1}$$

9 Testing content

9.1 Test section

This section contains a few examples to demonstrate the most important macros. Let's start with a quick list: $\|A\|$ is the norm, $|A|$ is the absolute value, A^* is the complex conjugate, $A^\dagger$ is the hermitian conjugate, A^T is the matrix transpose, $\langle\phi|$ is a bra vector, $|\psi\rangle$ is a ket vector, $\langle A\rangle = \langle\phi|A|\psi\rangle$ is a quantum expectation value, $R_\mu{}^\nu{}_\rho{}^\sigma$ is a tensor, $[\mathbf{A},\mathbf{B}]_- \equiv \mathbf{AB}-\mathbf{BA}$ is a commutator, and $[\mathbf{A},\mathbf{B}]_+ \equiv \mathbf{AB}+\mathbf{BA}$ is an anticommutator. If you need to typeset for example radioactive isotopes, just use the notationWe can also write out a simple three-dimensional integral:

$$\int_{\mathbb{R}^3} \mathrm{d}^3\mathbf{r}\, f(\mathbf{r}) = \int_0^{2\pi} \mathrm{d}\theta \int_0^{\pi} \mathrm{d}\phi \sin\phi \int_0^{\infty} r^2 f(r,\theta,\phi) \tag{9.1}$$

And how about a few differential equations:

$$\alpha\frac{\mathrm{d}^2 f}{\mathrm{d}x^2} + \beta\frac{\mathrm{d}f}{\mathrm{d}x} + \gamma f(x) = 0 \qquad \nabla^2\hat{\boldsymbol{\mathcal{A}}}(\mathbf{x}) = \sum_{ij} \frac{\partial\boldsymbol{\alpha}}{\partial x_i}\cdot\frac{\partial\boldsymbol{\beta}}{\partial x_j} \tag{9.2}$$

And maybe a few matrices and a determinant as well:

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \mathbf{B} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad \mathbf{C} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} \tag{9.3}$$

The rest of this chapter consists of the traditional *Lorem impsum* example text, with occasional examples of figures, tables, and code listings.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut faucibus, tortor porta imperdiet vestibulum, tortor lorem congue odio, a auctor velit dui a velit. Integer interdum metus ut neque semper, in aliquet nunc tristique. Phasellus ut mi elementum, semper felis vitae, varius urna. Nulla eget malesuada lorem. Nullam pharetra, elit in porta elementum, risus mauris ultrices massa, eget cursus lacus lectus molestie tortor. Vivamus consequat tempor mi vitae mattis. Cras vitae porttitor tellus. Duis tristique, diam nec hendrerit pulvinar, nisl ante consectetur diam, ut sagittis felis purus ut elit. Proin non tellus volutpat, consectetur est ut, mollis augue. Vivamus sollicitudin, nisl vel egestas bibendum, purus tortor rutrum enim, nec porttitor quam turpis eget velit. Mauris lacinia quis mi ut luctus. Sed ut eros metus.

Aenean vitae bibendum ligula, non cursus enim. Ut varius arcu nec nunc pulvinar, posuere mollis sem venenatis. Cras ullamcorper sollicitudin purus eu fermentum. Integer quis nulla eget leo viverra feugiat a eu turpis. Nam imperdiet nisl quis condimentum fermentum. Sed euismod diam egestas, varius magna et, aliquet metus. Nunc quis turpis mi. Maecenas feugiat facilisis ante vel porttitor. Aliquam eget metus sed lacus interdum vehicula at in nunc. Proin sit amet bibendum lorem. Nullam imperdiet at erat ultricies placerat. Sed ac dolor mollis, faucibus augue sed, lobortis tortor.

Figure 9.1: This is a test figure.

Sed id volutpat diam, at vestibulum dui. Mauris convallis justo nec neque luctus tincidunt. Donec vel ullamcorper nisi. Ut pulvinar quam non commodo lacinia. Nunc sollicitudin nibh eu malesuada egestas. Aenean euismod, sapien non molestie tincidunt, libero odio tempor massa, ac accumsan diam sapien sagittis magna. Donec non lectus turpis. Integer in consectetur ante. Nullam mollis suscipit ultricies. Ut auctor risus quis luctus euismod. Sed magna metus, mollis eget felis pretium, bibendum tempor sapien. Suspendisse sodales est sed tellus aliquam, quis bibendum orci laoreet. Sed consectetur justo metus, vitae tristique urna suscipit et. Curabitur id malesuada dui.

Table 9.1: This is a table caption.

Something	Something else	Something different
One	4.5	π
Two	4.7	e
Three	5.5	γ

Duis suscipit congue dolor, a tempus purus consequat sit amet. Sed a pulvinar eros. Phasellus ornare quam pulvinar, vestibulum justo vel, fringilla ipsum. Ut sagittis vehicula nunc, eu feugiat est vulputate vel. In vitae aliquam eros, et luctus dolor. Praesent varius ligula sit amet purus eleifend facilisis. Donec convallis, nisl vitae dignissim interdum, justo arcu convallis tortor, sit amet luctus ligula mi sit amet purus. Nullam congue neque in libero ultrices, a viverra turpis venenatis. Nullam at blandit ligula. Nunc non sodales nibh. Aliquam tempus arcu quis scelerisque convallis.

Listing 9.1: Here is the code caption...

```
#include <iostream>
using namespace std;

cout << "Hello␣world!" << endl;
```

Vestibulum rutrum placerat dapibus. Donec vitae leo mollis, convallis lorem vitae, tincidunt eros. Sed eleifend non sem a euismod. Fusce vel tincidunt diam. Praesent nec tristique lectus. Suspendisse a quam bibendum, mattis magna sed, feugiat tortor. Aenean ipsum est, fermentum a dui euismod, aliquam tristique nisl. Integer a faucibus elit. Donec pharetra justo ut lorem convallis, ut dictum ligula accumsan. Suspendisse vitae nisl vel massa placerat fringilla at a metus. Integer sed volutpat urna, in rutrum erat. Fusce eget arcu pulvinar dui ullamcorper tincidunt in ut augue. Aliquam erat volutpat. Aliquam tempus arcu non sapien tincidunt, ac ornare justo pharetra. Donec dignissim porttitor ornare. Integer eget sodales diam.

10 References and citations

10.1 Test section

We will now briefly demonstrate how to produce references and citations. The easiest way to consistently refer to equations and floats, is to use the macro `\cref`. For example we can refer to one equation, such as eq. (9.1); we can refer to multiple equations, such as eqs. (9.2) and (9.3); or we can even refer to different kinds of objects at the same time, such as fig. 9.1, table 9.1, and listing 9.1.

When we want to cite an academic paper or book, the standard option is to use the macro `\cite`.[**hipster**] By default, this produces a footnote citation, since these are unintrusive yet informative. However, if you prefer any other way of formatting the citations, then you just have to comment out the redefinitions of `\cite` and `\cites` from `preamble/macro.tex`, and then tweak the arguments to BibLaTeX in the file `preamble/include.tex` until you're satisfied. It is also possible to give optional arguments to the citations.[**statistics**] This includes the possibility of adding text to the footnote citations.[**feynman**] Furthermore, using commands like `\textcite`, it is also possible to cite **feynman** inline. Finally, we can also cite multiple sources at once using the `\cites` macro.[**hipster**, **statistics**, **haskell**][1]

Note that all the papers and books that we cited above, have to be declared in the BibTeX bibliography file `library.bib`. The format of the bibliography entries should hopefully be clear from the included examples; if not, then as usual, Google is your friend.

[1] This one is just a regular footnote, not a citation.

Bibliography

[1] M. J. Black and P. Anandan. "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields". In: *Computer Vision and Image Understanding* 63.1 (1996), pp. 75 –104. ISSN: 1077-3142. DOI: http://dx.doi.org/10.1006/cviu.1996.0006 (cit. on pp. 4, 6).

[2] T. F. Chan and P. Mulet. "On the Convergence of the Lagged Diffusivity Fixed Point Method in Total Variation Image Restoration". In: *SIAM Journal on Numerical Analysis* 36.2 (1999), pp. 354–367. DOI: 10.1137/S0036142997327075. eprint: http://dx.doi.org/10.1137/S0036142997327075 (cit. on p. 20).

[3] B. K. Horn and B. G. Schunck. *Determining Optical Flow*. Tech. rep. Cambridge, MA, USA, 1980 (cit. on pp. 3, 4, 11).

[4] H. H. Nagel and W Enkelmann. "An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 8.5 (May 1986), pp. 565–593. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1986.4767833 (cit. on p. 17).

[5] D. Shulman and J. Y. Herve. "Regularization of discontinuous flow fields". In: *Visual Motion, 1989.,Proceedings. Workshop on*. 1989, pp. 81–86. DOI: 10.1109/WVM.1989.47097 (cit. on p. 19).

[6] D. Sun, S. Roth, J. P. Lewis, and M. J. Black. "Computer Vision – ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part III". In: ed. by D. Forsyth, P. Torr, and A. Zisserman. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. Chap. Learning Optical Flow, pp. 83–97. ISBN: 978-3-540-88690-7. DOI: 10.1007/978-3-540-88690-7_7 (cit. on p. 21).

[7] H. Zimmer, A. Bruhn, and J. Weickert. "Optic flow in harmony". In: *International Journal of Computer Vision* 93.3 (2011), pp. 368–388 (cit. on pp. 9, 10, 21, 23).