FLUId Image analysis and Description (FLUID)
Project No. FP6-513663
FET - Open Domain, Priority IST

Deliverable 2.5

Motion Estimation Based on Convex Problem Decomposition

Florian Becker, Christoph Schnörr
CVGPR group, University of Mannheim

Due date of deliverable: 31 August 2007
Actual submission date: 15 October 2007
Start date of project: 1 December 2004
Duration: 3 years

Project cofunded by the European Commission
within the Sixth Framework Programme (2002-2006).
Contents

1 Introduction .................................................. 1

2 Decomposition for Convex Optimisation Problems ....... 2
   2.1 Problem Statement ........................................ 2
   2.2 Dual Decomposition ...................................... 3
      2.2.1 The Case of two Sub-PROblems .................. 3
      2.2.2 Extension to $d > 2$ Sub-PROblems ............... 4
      2.2.3 Constrained Optimisation Problems ............... 5
   2.3 Algorithm .................................................. 5
      2.3.1 Updating the Dual Variables ..................... 6
      2.3.2 Stopping Criteria ................................... 6
      2.3.3 Solution of the Original Problem ............... 7

3 Dual Decomposition Of $\|Du + c\|_2$-type Problems .... 7
   3.1 Standard Form ........................................... 8
   3.2 Dual Decomposition ..................................... 9
   3.3 Extension for Badly Conditioned Systems ............. 11
   3.4 Decomposition ........................................... 14

4 Experiments ..................................................... 17
   4.1 Experimental Setup ...................................... 17
   4.2 Horn and Schunck ....................................... 18
      4.2.1 Discretisation .................................... 20
      4.2.2 Results .......................................... 21
   4.3 Optical Flow Estimation With Higher Order Regularisation .... 22
      4.3.1 Discretisation .................................... 22
      4.3.2 Results .......................................... 25
   4.4 Stopping Criteria ....................................... 25
   4.5 Conclusion ............................................... 25

5 Conclusion .................................................. 27
1 Introduction

In many applications of image sequence analysis of natural scenes or scientific time series data, variational approaches to motion estimation have shown superior performance, due to the ability to take into consideration noise characteristics (data term) and qualitative prior knowledge about solutions.

Regarding the computation time required by users, increasingly strong constraints have to be satisfied. On the one hand, some applications require real-time evaluations inline with the processing of other modules of a larger system. On the other hand, the amount of image measurements produced is steadily increasing, in an increasing number of disciplines (surveillance of urban scences, medical imaging, bio-science, etc.). For example, in 3D time-resolved image sequences of experimental fluid mechanics, image volumes of \(4k \times 4k \times \text{few hundred}\) are to be expected in the future.

As a consequence, despite the availability of sophisticated numerical multi-grid solvers for variational approaches that compute solutions at nearly linear time complexity relative to the number of unknowns, there is a need for parallel implementations in order to accommodate the demands of the future. In this connection, coarse-grained architectures based on off-the-shelf hardware are attractive for a broad range of users.

A basic strategy in this context is domain decomposition [SBG96, QV99], that is to subdivide the underlying image domain into adjacent or slightly overlapping subdomains, and to perform two-step iterations consisting of computing in parallel local solutions in all subdomains, that are combined in a subsequent step to obtain a unique global solution to the variational problem.

This route has been explored in our group for variational approached using first-order regularisation [KSBW05]. It turned out, however, that the classical theory of domain decomposition does not cover more sophisticated variational approaches that incorporate higher-order regularisation and robust non-quadratic norms, for example. As a consequence, we decided to pursue a more general general approach than originally planned in the proposal: Convexity of the variational approach should be the only requirement for decomposing a variational approach into subproblems, that can be solved on different machines connected with a high-speed network. Moreover, we decided not to investigate pure feasibility problems, that is to compute any solution in the intersection of convex constraint sets, but continued to focus on the more favourable property of uniqueness of solutions.

Our more general convexity assumption encompasses all approaches that can be handled by domain decomposition, as well as most of the more general approaches that proved to be useful in the literature like, for instance, higher-
order regularisation or robust norms. Particular care has been devoted to avoid singular subproblems that naturally arise when standard domain decomposition is applied to higher-order problems. We also note that, while the problem decomposition can be specified by a decomposition of the underlying domain, other decompositions of the problem variables are feasible as well, irrespective of the underlying domain.

2 Decomposition for Convex Optimisation Problems

After stating the problem and introducing some notations we describe a method for decomposing convex optimisation problems in Section 2.2. A detailed description of an iterative algorithm for solving the problem in decomposed form is given in Section 2.3.

2.1 Problem Statement

Given a convex optimisation problem,

$$\min_{u \in \mathbb{R}^n} f(u)$$

(1)

with \( f \) convex we assume that the objective function can be separated into \( d \) convex functions \( f_i \), so that

$$f(u) = \sum_{i=1}^{d} f_i(u, y).$$

The variable vector was split up into one set of private variables \( u_i \in \mathbb{R}^{n_i} \) for each sub-function and a common set of complicating variables \( y \in \mathbb{R}^{n_c} \). The latter variables are involved in more than one sub-function and are therefor also referred to as coupling variables. In contrast the private - or internal, local - variables are only involved in exactly one sub-function. The decomposed problem is then defined as

$$\min_{\{u_i\}, y} \sum_{i=1}^{d} f_i(u_i, y)$$

where we denote \( \{u_i\} := \{u_1, u_2, \ldots, u_d\} \). The number of complicating variables should be small compared to the inner ones to allow an efficient solution of the decomposing algorithms. In the trivial case of \( n_c = 0 \) each sub-problem can be solved independently and the results can be assembled easily.
2.2 Dual Decomposition

In [BXM03] the authors describe a decomposition method which makes use of the Lagrange dual problem formulation, also referred to as Lagrangian relaxation in [Ber95].

We will introduce this method for two sub-problems and then extend it to an arbitrary number of subdivisions. In Section 2.2.3 we shortly comment on an extension for constrained optimisation. Finally we describe an iterative algorithm to solve the original problem via its decomposed representation.

2.2.1 The Case of two Sub-Problems

For convenience we start by splitting up the objective function into two sub-problems ($d = 2$):

$$\min_{u_1, u_2, y} f_1(u_1, y) + f_2(u_2, y)$$

Next we introduce an own set of complicating variables $y_i$ for each sub-function $f_i$ and enforce their identity by a consistency constraint,

$$(P) \quad \min_{u, y} f_1(u_1, y_1) + f_2(u_2, y_2)$$

s.t. $y_1 = y_2$

where we use the notation $u := \{u_1, u_2\}$ and $y := \{y_1, y_2\}$. In the context of the duality theory, this is the primal problem, here denoted as (P). In order to solve an equivalent unconstrained optimisation problem we define the Lagrange primal function by making the coupling constraint implicit.

$$p(u, y) := f_1(u_1, y_1) + f_2(u_2, y_2) + \sup_\lambda \lambda^T (y_1 - y_2)$$

$$= \begin{cases} 
  f_1(u_1, y_1) + f_2(u_2, y_2) & \text{if } y_1 = y_2 \\
  +\infty & \text{if } y_1 \neq y_2 
\end{cases}$$

The vector $\lambda \in \mathbb{R}^{nc}$ is referred to as the set of Lagrange multipliers. With the Lagrangian, which is defined as

$$L(u, y, \lambda) = f_1(u_1, y_1) + f_2(u_2, y_2) + \lambda^T (y_1 - y_2),$$

we can express (P) as

$$\min_{u, y} p(u, y) = \min_{u, y} \sup_\lambda L(u, y, \lambda).$$
The Lagrange dual problem (D) corresponding to (P) is defined as
\[
(D) \quad \max \inf_{\lambda,u,y} L(u, y, \lambda) = \max_{\lambda} d(\lambda)
\]
with the Lagrange dual function \( d(\lambda) \). The dual function is concave even if
the primary problem is not convex. A proof can be found in any standard
literature, i.e. [BV04]. Another important result from duality theory is the
weak duality relation
\[
(P) \quad p^* := \min_{u,y} p(u, y) \geq \max_{\lambda} d(\lambda) =: d^* \quad (D).
\]
For convex functions this relation holds strictly if some further conditions,
so called constraint qualifications hold. In this case the Slater’s conditions
are fulfilled trivially because we have no inequality constraints. Hence strong
duality holds:
\[
p^* = d^*
\]
This allows to solve the primal problem via the dual formulation. The dual
problem can be separated into two sub-problems embedded into a master
problem:
\[
\max_{\lambda} d(\lambda) = \max_{\lambda} \inf_{u_1, y_1} f_1(u_1, y_1) + \lambda^\top y_1 + \inf_{u_2, y_2} f_2(u_2, y_2) - \lambda^\top y_2
\]
\begin{align*}
\text{sub-problem 1} & \quad \text{sub-problem 2} \\
\text{master problem} & \quad \\
\end{align*}
Note that both sub-problems are convex optimisation problems with separate
variable sets. Hence they can be solved in parallel. Their only connection is \( \lambda \)
which is subject to optimisation in the master problem. The maximisation of
the concave function \( d(\lambda) \) is a convex optimisation problem which can be
solved for example using a method based on the subgradient.

### 2.2.2 Extension to \( d > 2 \) Sub-Problems

The previous notation can easily be extended to an arbitrary number of sub-
problems. For this purpose we introduce linear operators \( C_l \in \mathbb{R}^{(d-1) n_c \times (n_1+n_c)} \)
to represent the coupling equality constraints, i.e.
\[
\begin{pmatrix}
y_1 - y_2 \\
y_2 - y_3 \\
\vdots \\
y_{d-1} - y_d
\end{pmatrix} = \sum_{l=1}^{d} C_l \begin{pmatrix} u_l \\ y_l \end{pmatrix} = 0.
\]
Redundant constraints may be added in the same manner to improve numerical properties of the algorithm. Note that in this formulation the constraints also include the internal variable vectors which are involved in only one sub-problem by definition and hence need not to be constrained. Therefor the leftmost $n_t$ columns of the $C_l$-matrices can be assumed to be all-zero. This representation was chosen in order to simplify the notation in the following section. The primal problem becomes

$$ (P) \quad \min_{u,y} \sum_{l=1}^{d} f_l(u_l, y_l) $$

$$ \text{s.t.} \sum_{l=1}^{d} C_l \begin{pmatrix} u_l \\ y_l \end{pmatrix} = 0 $$

and in dual decomposed form

$$ (D) \quad \max_{\lambda} \sup_{u_l,y_l} \sum_{l=1}^{d} f_l(u_l, y_l) - \lambda^\top C_l \begin{pmatrix} u_l \\ y_l \end{pmatrix}. $$

In this formulation the subgradient of the dual function is given by

$$ s := \nabla_\lambda d(\lambda) = - \sum_{l=1}^{d} C_l \begin{pmatrix} u_l \\ y_l \end{pmatrix}. $$ (2)

### 2.2.3 Constrained Optimisation Problems

The described method can be extended to optimisation problems with equality and inequality constraints. As we will only consider unconstrained optimisation problems in our experiments we refer to [Ber95], [BXM03] and [CCMGB06] for details.

### 2.3 Algorithm

Solving the decomposed problem is performed in an iterative manner. In each iteration the local variables of the sub-problems are updated first. Due to the decomposition this can be done in parallel. Then the dual variables are updated. The iterative part of the algorithm ends when a stopping criteria is met. Finally the solution of the original problem is composed from the internal and complicating variables. In Algorithm 1 the complete algorithm is outlined. The remaining section deals with details of the method.
Algorithm: Dual Optimisation

Input: decomposed problem: $f_l(u_l, y_l)$, $C_l$

Output: solution $u$

$\lambda^{(0)} \leftarrow 0$ ;

$u_l^{(0)} \leftarrow 0, y_l^{(0)} \leftarrow 0 \quad l = 1, \ldots, d$ ;

$k \leftarrow 0$ ;

repeat

solve sub-problems $l = 1, \ldots, d$ in parallel:

\[
\begin{pmatrix}
  u_l^{(k+1)} \\
  y_l^{(k+1)}
\end{pmatrix} \leftarrow \arg \min_{u_l, y_l} f_l(u_l, y_l) - \lambda^{(k)} \top C_l \begin{pmatrix}
  u_l \\
  y_l
\end{pmatrix};
\]

$\lambda^{(k+1)} \leftarrow$ updated dual variables ;

$k \leftarrow k + 1$ ;

until until convergence ;

construct solution $u$ of the original problem ;

Algorithm 1: Iterative framework for dual decomposition of convex problems.

2.3.1 Updating the Dual Variables

A simple update rule for the dual variables is based on the subgradient (2):

\[
\lambda^{(k+1)} \leftarrow \lambda^{(k)} + \alpha^{(k)} \frac{s^{(k)}}{\|s^{(k)}\|}
\]

where $\alpha^{(k)}$ is a non-negative sequence. [CCMGB06] proposes to choose a sequence with the properties $\lim_{k \to \infty} \alpha^{(k)} \to 0$ and $\sum_{k=1}^{\infty} \alpha^{(k)} \to \infty$, i.e.

\[
\alpha^{(k)} = \frac{1}{a + bk}
\]

where $a$ and $b$ are some constant scalars.

In [Ber95] and [CCMGB06] further details on subgradient and alternative methods such as cutting-plane, bundle or trust region methods can be found.

2.3.2 Stopping Criteria

The most simple stopping criteria is to stop the algorithm after a predetermined number of iterations. In [CCMGB06] the relative change of the dual variables is used as an alternative test:

\[
c_D := \frac{\|\lambda^{(k+1)} - \lambda^{(k-1)}\|_2}{\|\lambda^{(k)}\|_2} \leq \epsilon_{\lambda}
\]
In our experiments we will also test the same measurement but applied to the primal variables,

\[
c_P := \frac{\|u^{(k+1)} - u^{(k-1)}\|_2}{\|u^{(k)}\|_2} \leq \epsilon_u
\]

where \(u^{(k)}\) is the recombination of the local and complicating variables to a solution of the non-decomposed problem, see next section.

### 2.3.3 Solution of the Original Problem

As the last step the solution of the original problem is composed from the partial solutions of the sub-problems. The internal variables can directly be mapped to their analogue. In the decomposed formulation the complicating variables are represented by several instances. Due to the coupling constraint they do not differ in a large scale. So we may choose the values from just one instance or the arithmetic mean of all appearances, i.e.

\[
y^{(k)} \leftarrow \frac{1}{d} \sum_{l=1}^{d} y_l^{(k)} \\
u^{(k)} \leftarrow \left( \begin{array}{c} (u_1^{(k)})^\top \cdots (u_d^{(k)})^\top (y^{(k)})^\top \end{array} \right)^\top
\]

### 3 Dual Decomposition Of \(\|Du+c\|^2_2\)-type Problems

In this section we demonstrate, how convex optimisation problems of the form

\[
\min_{u \in \mathbb{R}^n} \frac{1}{2} \|Du + c\|_2^2 \\
\text{with } D \in \mathbb{R}^{m \times n}, c \in \mathbb{R}^m
\]

can be decomposed and solved via the dual decomposition method described in Section 2. The decision to focus on this class of problems is motivated by the matter of fact, that several important problems from image processing can be mapped to this form: For example the discretisation of the variational approach to image recovery [CL97],

\[
\min_{u} \frac{1}{2} \int_{\Omega} (Au - u_0)^2 \, dx + \lambda \|\nabla u\|_2^2 \, dx
\]
with the disturbed image $u_0$ and a known transformation operator $A$. In this report we apply the described method to two methods for motion estimation: The classical approach by Horn and Schunck [HS81],

$$
\min_u \frac{1}{2} \int_{\Omega} \| \nabla g^T u + g_t \|^2_2 + \lambda \| \nabla u_1 \|^2_2 + \lambda \| \nabla u_2 \|^2_2 \, dx
$$

where $u(x) := (u_1(x) \ u_2(x))^T$ and a method with higher order regularisation described in [YSM07],

$$
\min_u \frac{1}{2} \int_{\Omega} \| \nabla g^T u + g_t \|^2_2 + \lambda_1 \| \nabla \text{div} \ u \|^2_2 + \lambda_2 \| \nabla \text{curl} \ u \|^2_2 \, dx
$$

$$
+ \frac{1}{2} \lambda_3 \int_{\partial \Omega} \| \partial_n u \|^2_2 \, dx.
$$

In Section 4 we will report experimental results for applying the described method to the latter two approaches.

### 3.1 Standard Form

For simplicity we will assume in the following sections that the problem is given in a form that makes the independent structure explicit:

$$
f(u) = \frac{1}{2} \| Du + c \|^2_2
$$

$$
= \frac{1}{2} \left\| \begin{pmatrix}
D_{1,I} & 0 & \cdots & 0 & D_{1,C} \\
0 & D_{2,I} & 0 & 0 & D_{2,C} \\
\vdots & \ddots & \ddots & \vdots & \vdots \\
0 & \cdots & 0 & D_{d,I} & D_{d,C}
\end{pmatrix}
\begin{pmatrix}
u_1 \\
\vdots \\
u_d \\
y
\end{pmatrix}
+ \begin{pmatrix}
c_1 \\
c_2 \\
\vdots \\
c_d
\end{pmatrix}
\right\|^2_2
$$

(4)

The variables are arranged into private variables $u_l \in \mathbb{R}^{n_l}$ and complicating ones $y \in \mathbb{R}^{nc}$ and stacked together into $u := (u_1^T, \cdots, u_d^T, y^T)^T$ so $n = n_c + \sum_{l=1}^{d} n_l$. The matrices $D_{l,I} \in \mathbb{R}^{n_l \times n_l}$ and $D_{l,C} \in \mathbb{R}^{n_l \times n_c}$ describe the involvement of the private respectively complicating variables in sub-problem $l$. The vectors $c_l \in \mathbb{R}^{n_l}$ hold the constant value for each sub-problem.

In Section 3.4 we describe in detail a method to bring a problem of the general form $\frac{1}{2} \| Du + c \|^2_2$ with arbitrary $D$ and $c$ into this standard notation.
3.2 Dual Decomposition

In the representation (4) the objective function decomposes into

\[ f(u) = \frac{1}{2} \| Du + c \|_2^2 \]

\[ = \frac{1}{2} \sum_{l=1}^{d} \left( \begin{array}{c} (D_{l,I} \ y_l) \\ (D_{l,C} \ y_l) \end{array} \right) \left( \begin{array}{c} u_l \\ y_l \end{array} \right)^\top \left( \begin{array}{c} u_l \\ y_l \end{array} \right) + \left( \begin{array}{c} c_l \\ c_l \end{array} \right) \left( \begin{array}{c} 1 \\ 1 \end{array} \right) \]

\[ = \sum_{l=1}^{d} \frac{1}{2} \left( \sum_{i=1}^{d} \left( \begin{array}{c} u_i \\ y_i \end{array} \right)^\top \left( \begin{array}{cc} A_l & D_{l,I} \\ C_l \ y_l & D_{l,C} \ y_l \end{array} \right) \left( \begin{array}{c} u_i \\ y_i \end{array} \right) + \left( \begin{array}{c} c_i \\ c_l \end{array} \right) \right) \]

\[ = \sum_{l=1}^{d} \frac{1}{2} \left( \sum_{i=1}^{d} \left( \begin{array}{c} u_i \\ y_i \end{array} \right)^\top A_l \left( \begin{array}{c} u_i \\ y_i \end{array} \right) - \left( \begin{array}{c} u_i \\ y_i \end{array} \right)^\top b_l + \text{const} = \sum_{l=1}^{d} f_l(u_l, y_l) + \text{const} \]

For clarity we will omit the constant values in the following description. Each \( f_l \) is a quadratic function in the internal variables \( u_l \) and the complicating variables \( y_l \). Due to the way \( A_l \) is constructed, it is guaranteed to be symmetric and positive semi-definite and so the function \( f_l \) is convex:

\[ x^\top A_l x = \| (D_{l,I} \ D_{l,C}) x \|_2^2 \geq 0 \quad \forall x \in \mathbb{R}^{n_l + n_c} \Rightarrow A_l \succeq 0 \]

However this does not rule out poorly conditioned or even rank-deficient matrices. In Section 3.3 we will propose a way to extend the algorithm in order to overcome this problem. In the following we assume the matrices \( A_l \) to be strictly positive definite and hence regular matrices.

According to the dual decomposition method we can write down the primal problem as

\[ \min_{\{u_l, y_l\}} \sum_{l=1}^{d} \left( \frac{1}{2} \left( \begin{array}{c} u_l \\ y_l \end{array} \right)^\top A_l \left( \begin{array}{c} u_l \\ y_l \end{array} \right) - b_l^\top \left( \begin{array}{c} u_l \\ y_l \end{array} \right) \right) \]

s.t. \[ \sum_{l=1}^{d} C_l \left( \begin{array}{c} u_l \\ y_l \end{array} \right) = 0 \]
With the dual variables \( \lambda \in \mathbb{R}^d, \) the Lagrangian is given by

\[
L(\{u_l, y_l\}, \lambda) = \sum_{l=1}^{d} f_l(u_l, y_l) - \lambda^\top \left( \sum_{l=1}^{d} C_l \left( y_l \right) \right)
\]

\[
= \sum_{l=1}^{d} \left( \frac{1}{2} \begin{pmatrix} u_l \\ y_l \end{pmatrix}^\top A_l \begin{pmatrix} u_l \\ y_l \end{pmatrix} - \left( b_l + C_l^\top \lambda \right)^\top \begin{pmatrix} u_l \\ y_l \end{pmatrix} \right)
\]

and the dual problem of (5) is

\[
\max_{\lambda} \inf_{\{u_l, y_l\}} L(\{u_l, y_l\}, \lambda)
\]

\[
= \max_{\lambda} \sum_{l=1}^{d} \left[ \inf_{u_l, y_l} \left( \frac{1}{2} \begin{pmatrix} u_l \\ y_l \end{pmatrix}^\top A_l \begin{pmatrix} u_l \\ y_l \end{pmatrix} - \left( b_l + C_l^\top \lambda \right)^\top \begin{pmatrix} u_l \\ y_l \end{pmatrix} \right) \right] = \max_{\lambda} \sum_{l=1}^{d} d_l(\lambda) = \max_{\lambda} d(\lambda).
\]

For this class of problems, the sub-problems

\[
\min_{u_l, y_l} \frac{1}{2} \begin{pmatrix} u_l \\ y_l \end{pmatrix}^\top A_l \begin{pmatrix} u_l \\ y_l \end{pmatrix} - \left( b_l + C_l^\top \lambda \right)^\top \begin{pmatrix} u_l \\ y_l \end{pmatrix}
\]

are unconstrained, quadratic convex problems which can be solved efficiently. To find an optimal value it is sufficient that the first derivative vanishes. This reduces to a linear program,

\[
A_l \begin{pmatrix} u_l \\ y_l \end{pmatrix} = \left( b_l + C_l^\top \lambda_l \right)
\]

which can be solved i.e. with a conjugate gradient method, see i.e. [Ber95].

Maximising the master problem objective function can be accomplished by a method based on the subgradient,

\[
s := \nabla_\lambda d(\lambda) = -\sum_{l=1}^{d} C_l \left( \overline{u}_l(\lambda) \right) \overline{y}_l(\lambda)
\]

where we use the explicit formulation for the optimal primal variables,

\[
\begin{pmatrix} \overline{u}_l(\lambda) \\ \overline{y}_l(\lambda) \end{pmatrix} := A_l^{-1} \left( b_l + C_l^\top \lambda_l \right)
\]

10
Assuming $\|s\| > 0$, a line search for a good step scale $\alpha \geq 0$ can be solved optimally, i.e.

$$
\alpha^* = \arg \max_\alpha d(\lambda + \alpha s).
$$

Therefore we insert the solution of the sub-problems into the corresponding part of the dual function,

$$
d_l(\lambda) = f_l(\overline{\mu}_l(\lambda), \overline{y}_l(\lambda)) - \lambda^T C_l \left( \frac{\overline{\mu}_l(\lambda)}{\overline{y}_l(\lambda)} \right)
$$

$$
= \frac{1}{2} \left( b_l + C_i^T \lambda \right)^T A_i^{-1} A_i \left( b_l + C_i^T \lambda \right) - \left( b_l + C_i^T \lambda \right)^T A_i^{-1} \left( b_l + C_i^T \lambda \right)
$$

$$
= - \frac{1}{2} \left( b_l + C_i^T \lambda \right)^T A_i^{-1} \left( b_l + C_i^T \lambda \right)^T.
$$

For a line search for $\alpha$ on the dual function $d(\lambda)$ being a sum of concave quadratic functions in $\lambda$ it is sufficient that the first derivative vanishes:

$$
\frac{d}{d\alpha} d(\lambda + \alpha s) = \frac{d}{d\alpha} \sum_{l=1}^d d_l(\lambda + \alpha s)
$$

$$
= \frac{d}{d\alpha} \sum_{l=1}^d - \frac{1}{2} \left( b_l + C_i^T (\lambda + \alpha s) \right)^T A_i^{-1} \left( b_l + C_i^T (\lambda + \alpha s) \right)
$$

$$
= - \sum_{l=1}^d \left( s^T C_i A_i^{-1} \left( b_l + C_i^T \lambda \right) + \alpha s^T C_i A_i^{-1} C_i^T s \right) = 0
$$

$$
\Rightarrow \alpha^* = - \frac{\sum_{l=1}^d \left( b_l + C_i^T \lambda \right)^T A_i^{-1} C_i^T s}{\sum_{l=1}^d s^T C_i A_i^{-1} C_i^T s} = - \frac{\sum_{l=1}^d \left( b_l + C_i^T \lambda \right)^T r_l}{\sum_{l=1}^d s^T C_i r_l}
$$

However it is necessary to solve the linear problem

$$
A_l r_l = C_i^T s
$$

for $l = 1, \ldots, d$ which is of the same size as those involved in the solution of the sub-problems.

### 3.3 Extension for Badly Conditioned Systems

The decomposition we proposed guarantees the sub-problem matrices $A_l$ to be symmetric and positive semidefinite. However depending on the problem and the actual splitting, badly conditioned or singular matrices may result. The condition number induced by the $\| \cdot \|_2$-matrix norm,

$$
\kappa(A) := \|A\|_2 \|A^{-1}\|_2
$$
which is the ratio between the largest and smallest singular value of $A$ (see [GL96]) gives a hint how conjugate gradient perform on solving linear systems involving $A$. Experiments showed that the condition numbers of the subproblem matrices directly influence the convergence of the whole approach. The smaller the values the faster the method approaches the optimal solution. For values of about $10^7$ or larger the algorithm diverges.

We propose a framework to improve the numerical properties of the algorithm. It allows to modify the matrix $A_l$ by adding a positive semidefinite symmetric matrix $B_l$ without changing the solution. In Section 4 we show how the available information about the problem structure can be exploited to build this matrices and thereby improve the convergence properties of the algorithm.

For the description of the method we assume that the original problem was decomposed as described in Section 3.1. We will start by modifying the update step for the primal variables and then show that the overall procedure still solves the same problem.

Instead of (7) we minimise the modified objective function

$$
\tilde{f}_l (u_l, y_l) := f (u_l, y_l) + \frac{1}{2} \left\| B_l \begin{pmatrix} u_l - u_l^{(k)} \\ y_l - y_l^{(k)} \end{pmatrix} \right\|_2^2
$$

$$
= \frac{1}{2} \begin{pmatrix} u_l \\ y_l \end{pmatrix}^T (A_l + B_l) \begin{pmatrix} u_l \\ y_l \end{pmatrix} - \begin{pmatrix} u_l \\ y_l \end{pmatrix}^T \begin{pmatrix} b_l + B_l \begin{pmatrix} u_l^{(k)} \\ y_l^{(k)} \end{pmatrix} \end{pmatrix} + \frac{1}{2} \left\| B_l \begin{pmatrix} u_l^{(k)} \\ y_l^{(k)} \end{pmatrix} \right\|_2^2.
$$

Due to our assumption that $B_l$ is positive semidefinite this is still a convex optimisation problem and hence it is sufficient to solve the linear system

$$(A_l + B_l) \begin{pmatrix} u_l \\ y_l \end{pmatrix} = b_l + C_l^T \lambda^{(k)} + B_l \begin{pmatrix} u_l^{(k)} \\ y_l^{(k)} \end{pmatrix}$$

in each iteration and sub-problem. The update step for the primary variables becomes

$$
\begin{pmatrix} u_l^{(k+1)} \\ y_l^{(k+1)} \end{pmatrix} \leftarrow \text{solution of } (A_l + B_l) \begin{pmatrix} u_l \\ y_l \end{pmatrix} = b_l + C_l^T \lambda^{(k)} + B_l \begin{pmatrix} u_l^{(k)} \\ y_l^{(k)} \end{pmatrix}.
$$

Together with the unmodified update step for the dual variables and some $\alpha$

$$
\lambda^{(k+1)} \leftarrow \lambda^{(k)} + \alpha s,
$$
we can write one iteration of the modified algorithm in the form $Mx^{(k+1)} = N x^{(k)} + f$:

$$
\begin{pmatrix}
A_1 + B_1 & 0 & \cdots & 0 & 0 \\
0 & \ddots & \ddots & \vdots & \vdots \\
\vdots & \ddots & \ddots & 0 & \vdots \\
0 & \cdots & 0 & A_d + B_d & 0 \\
C_1 & \cdots & \cdots & C_d & \alpha^{-1} I
\end{pmatrix}
\begin{pmatrix}
u_1^{(k+1)} \\
y_1^{(k+1)} \\
\vdots \\
u_d^{(k+1)} \\
y_d^{(k+1)} \\
\lambda^{(k+1)}
\end{pmatrix}
= 
\begin{pmatrix}
u_1^{(k)} \\
y_1^{(k)} \\
\vdots \\
u_d^{(k)} \\
y_d^{(k)} \\
\lambda^{(k)}
\end{pmatrix}
\begin{pmatrix}
b_1 \\
\vdots \\
b_d \\
0
\end{pmatrix}
$$

which reveals the algorithm as an instance of iterative methods for linear systems. More specific, this is a splitting method (see [GL96]) for solving the linear problem $(M - N)x = f$, or

$$
\begin{pmatrix}
A_1 & 0 & \cdots & 0 & -C_1^T \\
0 & \ddots & \ddots & \vdots & \vdots \\
\vdots & \ddots & \ddots & 0 & \vdots \\
0 & \cdots & 0 & A_d & -C_d^T \\
C_1 & \cdots & \cdots & C_d & 0
\end{pmatrix}
\begin{pmatrix}
u_1 \\
y_1 \\
\vdots \\
u_d \\
y_d \\
\lambda
\end{pmatrix}
= 
\begin{pmatrix}
b_1 \\
\vdots \\
b_d \\
0
\end{pmatrix}
$$

which are exactly the Karush-Kuhn-Tucker first order optimality conditions for the primal and dual optimal variables of the decomposed problem (5) respectively (6):

$$
\left(\nabla_{u_i} L\right)(\{u_i^*, y_i^*, \lambda^*\}) = A_i \left(\begin{pmatrix} u_i^* \\ y_i^* \end{pmatrix}\right) - b_i - C_i^T \lambda = 0 \quad \forall i = 1, \ldots, d
$$

$$
\nabla_{\lambda} L(\{u_i^*, y_i^*, \lambda^*\}) = - \sum_{i=1}^{d} C_i \left(\begin{pmatrix} u_i^* \\ y_i^* \end{pmatrix}\right) = 0
$$

This shows, that the method with the modified objective functions still solves the same problem as the original one if it converges.
To sum up: Choosing a suitable $B_l$ allows to modify the sub-problems so that their numerical properties improve. We showed that if the algorithm converges, it does to a solution of the original problem. In the experimental Section 4 we will give an example how to derive a set of modifying matrices from the problem structure.

### 3.4 Decomposition

For decomposing the objective function we first identify each row of $D \in \mathbb{R}^{n \times m}$ and $c \in \mathbb{R}^n$ as a single quadratic term:

$$f(u) = \frac{1}{2} \|Du + c\|^2 = \frac{1}{2} \left\| \begin{pmatrix} D_{1,*} \\ \vdots \\ D_{n,*} \end{pmatrix} u + \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \right\|^2$$

$$= \sum_{l=1}^n \frac{1}{2} (D_{i,*} u + c_i)^2$$

where $D_{i,*}$ denotes the row $i$ of the matrix $D$ and $c_i$ the elements of $c$.

Next we assign each term $i$ to the sub-functions $l \in \{1, \ldots, d\}$ by choosing a weight vector $w_l \in [0,1]^n$, so $\sum_{l=1}^d w_l = e$ where $e \in \mathbb{R}^n$ is an all-one-vector. This generalises "hard" decomposition methods (i.e. $w_l \in \{0,1\}^n$) by allowing to assign each term to more than one sub-function and with different impact. The choice of the weights can be based on geometrical cues but also on more abstract decomposition methods such as graph cuts.

The sets $\mathcal{I}$ and $\mathcal{I}_1, \ldots, \mathcal{I}_d \subset \mathcal{I}$ will represent the indices of all terms in $D$, $\mathcal{I} := \{1, \ldots, n\}$, respectively those terms involved in sub-function $l$, i.e. $\mathcal{I}_l := \{i \in \mathcal{I} | w_{l,i} > 0\}$ where $w_{l,i}$ is the $i$-th element of the vector $w_l$. We order the elements $\mathcal{I}_l \in \mathcal{I}$ by associating the set with the relation “≤” and write $\mathcal{I}_1 \leq \mathcal{I}_2 \leq \cdots \leq \mathcal{I}_d$.

With $W_l := \text{diag}(w_l)$ the objective function decomposes into:

$$f(u) = \sum_{l=1}^d \sum_{i \in \mathcal{I}_l} \frac{1}{2} w_{l,i} (D_{i,*} u + c_i)^2$$

$$= \sum_{l=1}^d \frac{1}{2} \left\| W_l^{\frac{1}{2}} (Du + c) \right\|^2$$

For the further description we introduce the extension operator $R_{\mathcal{I} \rightarrow \mathcal{J}} := (\delta_{\mathcal{I},\mathcal{J}})_{i,j} \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ with $\mathcal{J} \subset \mathcal{I}$. Here $\delta_{i,j} := \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$ denotes the
Kronecker delta function. The transposed version $R_{\mathcal{J} \to \mathcal{I}}^\top$ can be regarded as a restriction operator which extracts a subset of rows $\mathcal{J}$ out of $\mathcal{I}$ from a linear operator $D \in \mathbb{R}^{[\mathcal{I}] \times m}$: $R_{\mathcal{J} \to \mathcal{I}}^\top D = D_{\mathcal{J}, \cdot}$. Furthermore the following equation holds:

$$R_{\mathcal{J} \to \mathcal{I}} R_{\mathcal{J} \to \mathcal{I}}^\top = \left( \sum_{k=1}^{[\mathcal{J}]} \delta_{i_k, \mathcal{J}_k} \delta_{i, \mathcal{J}_k} \right)_{i,j} = \text{diag} \left( (\delta_{i, \mathcal{J}})_{i=1, \ldots, [\mathcal{I}]} \right)$$

where $\delta_{i \in \mathcal{I}} := \begin{cases} 1 & i \in \mathcal{I} \\ 0 & i \notin \mathcal{I} \end{cases}$. We will use this operator to reduce the size of the matrix and vector involved in each sub-function: Due to the definition of $\mathcal{T}$, the equality

$$W_l^{\top} R_{\mathcal{T} \to \mathcal{I}}^\top R_{\mathcal{T} \to \mathcal{I}}^\top W_l^{\top} = W_l^{\top} \text{diag} \left( (\delta_{i, \mathcal{T}})_{i=1, \ldots, [\mathcal{I}]} \right) W_l^{\top}$$

$$= W_l^{\top} \text{diag} \left( (\delta_{i, \mathcal{T}, >0})_{i=1, \ldots, [\mathcal{I}]} \right) W_l^{\top} = W_l$$

holds. Then we can define $P_l := R_{\mathcal{T} \to \mathcal{I}}^\top W_l^{\top}$ and rewrite the objective as

$$f(u) = \sum_{l=1}^{d} \frac{1}{2} (Du + c)^\top W_l (Du + c)$$

$$= \sum_{l=1}^{d} \frac{1}{2} (Du + c)^\top R_{\mathcal{T} \to \mathcal{I}}^\top R_{\mathcal{T} \to \mathcal{I}}^\top W_l^{\top} (Du + c)$$

$$= \sum_{l=1}^{d} \frac{1}{2} \left\| R_{\mathcal{T} \to \mathcal{I}}^\top W_l^{\top} (Du + c) \right\|_2^2 = \sum_{l=1}^{d} \frac{1}{2} \left\| P_l (Du + c) \right\|_2^2 .$$

In the next step we make the reduced number of variables involved in $P_l Du$ explicit. Analog to the index list for the terms, $\mathcal{J} := \{1, \ldots, m\}$ represents the variables $u$ in the original problem. Then we define the sets $\mathcal{J}_l \subset \mathcal{J}$ of variables with non-zero contribution to $P_l Du$, i.e. $\mathcal{J}_l := \{ j \in \mathcal{J} | P_l e_j \neq 0 \}$ where $e_j \in \mathbb{R}^m$ is a vector with an one at position $j$ and zero elsewhere. The number of sub-functions in which a variable occurs is given by the entries of the diagonal matrix

$$\text{diag}(v) = V := \sum_{l=1}^{d} R_{\mathcal{J}_l \to \mathcal{J}} R_{\mathcal{J}_l \to \mathcal{J}}^\top = \text{diag} \left( \sum_{l=1}^{d} \delta_{\mathcal{J}_l, \mathcal{J}} \right)_{j=1, \ldots, [\mathcal{J}]} .$$

For each sub-function $l$ we identify the complicating variables by $\mathcal{J}_C^l := \{ j \in \mathcal{J}_l | v_j > 1 \}$ and the internal ones by $\mathcal{J}_I^l := \{ j \in \mathcal{J}_l | v_j = 1 \}$. We arrange all complicating variables $\mathcal{J}_C := \bigcup_{l=1}^{d} \mathcal{J}_C^l$ into $y := R_{\mathcal{J}_C \to \mathcal{J}}^\top u$ and denote the internal variables of sub-function $l$ with $u_l := R_{\mathcal{J}_I^l \to \mathcal{J}}^\top u$.  

15
With this notations we can define a linear operator that reduces the variable vector to a smaller variant which only contains the complicating and the involved inner variables for one sub-problem:

$$
\begin{pmatrix}
R_{J^l_1 \to J}^T \\
R_{J^l \to J}^T
\end{pmatrix} u =
\begin{pmatrix}
u_t \\
y_t
\end{pmatrix}
$$

The sets $J_C$ and $J^l_1$ are disjoint by construction and $J^l \subset J_C \cup J^l_1$ so that the following equality holds:

$$
P_l D \left( \begin{pmatrix} R_{J^l_1 \to J}^T \\ R_{J^l \to J}^T \end{pmatrix} \right)^\top \left( \begin{pmatrix} R_{J^l_1 \to J}^T \\ R_{J^l \to J}^T \end{pmatrix} \right) = P_l D \left( \begin{pmatrix} R_{J^l_1 \to J}^T \\ R_{J^l \to J}^T \end{pmatrix} \right)^\top \left( \begin{pmatrix} R_{J^l_1 \to J}^T \\ R_{J^l \to J}^T \\ R_{J^l \to J}^T \end{pmatrix} \right)
$$

$$
= P_l D \left( \text{diag} \left( (\delta_j)_{j \in J^l_1} \right) + \text{diag} \left( (\delta_j)_{j \in J_C} \right) \right)
$$

$$
= P_l D \text{diag} \left( (\delta_j)_{j \in J^l} \right) = P_l D
$$

Now we can bring the objective function into the form assumed in Section 3.

$$
f(u) = \sum_{l=1}^d \frac{1}{2} \left\| P_l D u + P_l c \right\|^2 + \sum_{l=1}^d \frac{1}{2} \left\| P_l D \left( \begin{pmatrix} R_{J^l_1 \to J}^T \\ R_{J^l \to J}^T \end{pmatrix} \right)^\top \left( \begin{pmatrix} R_{J^l_1 \to J}^T \\ R_{J^l \to J}^T \end{pmatrix} \right) u + P_l c \right\|^2
$$

$$
= \sum_{l=1}^d \frac{1}{2} \left\| P_l DR_{J^l_1 \to J} P_l DR_{J^l \to J} \left( \begin{pmatrix} R_{J^l_1 \to J}^T \\ R_{J^l \to J}^T \end{pmatrix} \right) u + P_l c \right\|^2
$$

$$
= \sum_{l=1}^d \frac{1}{2} \left\| \begin{pmatrix} D_{1,l} & \cdots & 0 \\
0 & D_{2,l} & \cdots \\
\vdots & \ddots & \ddots \\
0 & \cdots & D_{d,l}
\end{pmatrix} \begin{pmatrix} u_1 \\
\vdots \\
u_d
\end{pmatrix} + \begin{pmatrix} c_1 \\
\vdots \\
c_d
\end{pmatrix} \right\|^2
$$

However usually not every complicating variable is involved in every sub-function. We use this fact to further reduce the complexity of the sub-
problems: We define $Q_l := \begin{pmatrix} R_{g_{l+1}}^T - \mathcal{J} \\ R_{g_{l-1}}^T - \mathcal{J} \end{pmatrix}$ and get

$$f(u) = \sum_{l=1}^d \frac{1}{2} \| P_l Du + c \|_2^2 = \sum_{l=1}^d \frac{1}{2} \| P_l DQ_l^T Q_l u + P_l c \|_2^2$$

$$= \sum_{l=1}^d \frac{1}{2} \| D_l \begin{pmatrix} u_l \\ y_l \end{pmatrix} + c_l \|_2^2$$

$$= \sum_{l=1}^d \frac{1}{2} \begin{pmatrix} u_l \\ y_l \end{pmatrix}^\top D_l^\top D_l \begin{pmatrix} u_l \\ y_l \end{pmatrix} + \begin{pmatrix} u_l \\ y_l \end{pmatrix}^\top D_l^\top c_l + c_l^\top c_l$$

$$= \sum_{l=1}^d \frac{1}{2} \begin{pmatrix} u_l \\ y_l \end{pmatrix}^\top A_l \begin{pmatrix} u_l \\ y_l \end{pmatrix} = \begin{pmatrix} u_l \\ y_l \end{pmatrix}^\top b_l + \text{const}$$

In this form only those complicating variables are included that are really involved in each sub-function.

4 Experiments

To verify the theory we applied the method proposed in Section 3 to two variational motion estimation methods based on the optical flow constraint as data term: the well known approach by Horn and Schunck [HS81] in Section 4.2 and the approach by Yuan et al. [YSM07] in Section 4.3 which uses higher order regularisation.

We start by describing the common experimental setup in Section 4.1. Then for both approaches we explain the used discretisation method and how to cast the approach to the standard form $\frac{1}{2} \| Du + c \|_2^2$. Finally we show the results for solving the two optimisation problems via their dual decomposed form and evaluate the performance of two stopping criteria.

4.1 Experimental Setup

In order to gain an image pair $g_1, g_2$, a synthetic particle image as described in [CW05] was symmetrically warped by a vector field $-\frac{1}{2} v$ respectively $+\frac{1}{2} v$ using cubic spline interpolation. The resulting images are 500 by 500 pixels, see Figure 1(a). The vector field is defined by an affine function in the image coordinates, i.e. $v(x) = Ax + b$. In order to keep the displacement magnitudes small enough for the optical flow constraint we scaled the vector
field to \( \|v\|_2 \leq 1 \). The ground truth is depicted in Figure 1(b) using the colour representation described in Figure 1(c).

In all approaches we chose a geometrical based method to determine \( w_l \) and to assign the terms to the sub-functions. We defined overlapping rectangular areas \( \Omega_i \) within the grid and assigned each term to sub-function \( l \) if the positions \( x_j \) of all involved variables fall within \( \Omega_i \). Terms assigned to more than one sub-functions were equally weighted.

All experiments were implemented in MATLAB. The sub-problems were solved as linear programs with a conjugate gradient method. For the master problem we applied a subgradient method with the update rule

\[
\lambda^{(k+1)} \leftarrow \lambda^{(k)} + \alpha^{(k)} \frac{s}{\|s\|_2^2}
\]

and \( \alpha^{(k)} = \frac{1}{a + bk} \). Both the primal variables \( u_l, y_l \) as well as the dual variables \( \lambda \) were initialised with all-zero vectors.

The quality of the result \( u \) was not measured with respect to the ground truth of the data. Instead the solution of the non-decomposed problem \( v \) was used as reference in order to verify the equivalence of the solution of the original and the decomposed problem. For every grid position \( x \in \Omega \) we determine the Euclidean error \( e(x) := \|u(x) - v(x)\|_2 \) and denote the mean error, the standard deviation and the maximum error by

\[
\mu_e := \frac{1}{|\Omega|} \sum_{x \in \Omega} e(x)
\]

\[
\sigma_e := \sqrt{\frac{1}{|\Omega| - 1} \sum_{x \in \Omega} (e(x) - \mu_e)^2}
\]

\[
\max e := \max_{x \in \Omega} e(x)
\]

The results and error measurements were recorded for each iteration of the algorithm. Reaching the maximum number of iterations - here ten - was the only stopping criteria used. In Section 4.4 we will verify the performance of the alternative ones defined in Section 2.3.2.

### 4.2 Horn and Schunck

In [HS81], Horn and Schunck proposed the following variational approach to estimate the optical flow:

\[
\min_u \int_\Omega \frac{1}{2} \| \nabla g^T u + g_t \|_2^2 + \frac{1}{2} \lambda \| \nabla u_1 \|_2^2 + \frac{1}{2} \lambda \| \nabla u_2 \|_2^2 \, dx
\]

where \( u(x) := (u_1(x) \ u_2(x))^T \).
Figure 1: Experimental data
4.2.1 Discretisation

For our experiments we used a regular grid with square cells for both data and variables as depicted in Figure 2(a). Assuming the given image is \( n \times m \) in size, the variables are stacked together column- and then component-wise. Denoting the component \( k \) of the vector at position \((i, j)\) as \( u_k(i, j) \) we define the variable vector as

\[
u := \begin{pmatrix} u_1(1, 1) \\ \vdots \\ u_1(n, 1) \\ u_1(1, 2) \\ \vdots \\ u_1(n, m) \\ u_2(1, 1) \\ \vdots \\ u_2(n, 1) \\ u_2(1, 2) \\ \vdots \\ u_2(n, m) \end{pmatrix} \in \mathbb{R}^{2nm}
\]

We use a simple finite difference scheme for the estimation of the variable gradients which yields the following matrices:

\[
D_{\partial_x} := \begin{pmatrix} I_m \otimes D_n & 0 \\ 0 & I_m \otimes D_n \end{pmatrix} \in \mathbb{R}^{(n-1)n \times 2nm} \quad \text{derivative in x-direction}
\]

\[
D_{\partial_y} := \begin{pmatrix} D_n \otimes I_m & 0 \\ 0 & D_n \otimes I_m \end{pmatrix} \in \mathbb{R}^{(n-1)m \times 2nm} \quad \text{derivative in y-direction}
\]

with \( D_n := \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(n-1) \times n} \)

and the identity matrix \( I_n \in \mathbb{R}^{n \times n} \).

The spatial image derivatives \( g_x(i, j) \) and \( g_y(i, j) \) were estimated from \( \frac{1}{2}(g_1 + g_2) \) using binomial FIR filters,

\[
\frac{1}{8} \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{pmatrix} \quad \text{respectively} \quad \frac{1}{8} \begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix}
\]
and a simple difference $g_t = g_2 - g_1$ was used for the temporal derivatives. For the optical flow constraint (OFC) these values were arranged as

$$D_{\text{OFC}} := \text{diag} (g_x(1,1), \ldots, g_x(n,m), g_y(1,1), \ldots, g_y(n,m))$$

and $c_{\text{OFC}} := g_t$

Altogether the objective function for the approach by Horn and Schunck in discrete form reads

$$f(u) := \frac{1}{2} \|D_{\text{OFC}}u + c_{\text{OFC}}\|_2^2 + \frac{1}{2} \lambda \|D_{\partial_x} u\|_2^2 + \frac{1}{2} \lambda \|D_{\partial_y} u\|_2^2 .$$

To make the decomposition method described in Section 3 applicable, we rewrite this function as

$$f(u) = \frac{1}{2} \left\| \begin{pmatrix} D_{\text{OFC}} \sqrt{\lambda} D_{\partial_x} & \left( \begin{array}{c} c_{\text{OFC}} \\ 0 \\ 0 \end{array} \right) \end{pmatrix} u \right\|_2^2 .$$

For the decomposition we defined four subdivisions marked blue, red, green and yellow in Figure 2(b) which overlap by one grid unit.

### 4.2.2 Results

For the experiments we used the described image data and chose the only regularisation parameter as $\lambda = 0.1$. The parameters for the dual variable update rule were set to $a = -300$ and $b = 400$. 

![Figure 2: Horn and Schunck: discretisation and decomposition](image)
The results are depicted in Figure 3. The error plot over the iterations one to ten in Figure 3(c) shows a steep drop of all error measurements within few iterations. After ten iterations they reach $\mu_e = 3.57 \cdot 10^{-6}$, $\sigma_e = 4.04 \cdot 10^{-5}$ and max $e = 0.0028$. According to Figure 3(b) the errors of the order of max $e$ are located at only few positions at the artificial borders especially where all four areas meet.

4.3 Optical Flow Estimation With Higher Order Regularisation

The second approach proposed by [YSM07] uses higher order regularisation of the vector field:

$$
\min_u \frac{1}{2} \int_\Omega \left( \left\| \nabla g \top u + g_1 \right\|_2^2 + \lambda_1 \left\| \nabla \text{div } u \right\|_2^2 + \lambda_2 \left\| \nabla \text{curl } u \right\|_2^2 \right) dx
+ \frac{1}{2} \lambda_3 \int_{\partial \Omega} \left\| \partial_n u \right\|_2^2 dx
$$

4.3.1 Discretisation

For the discretisation we used the framework based on mimetic differences described in [HS97a] and [HS97b]. For details on the problem discretisation we refer to [YSM07]. In Figure 4(a) the underlying variable grid is depicted. Using the notation introduced in this paper the discrete version of the objective function is

$$
\frac{1}{2} \left\| G u + \partial_t g \right\|_{H^2_v}^2 + \frac{1}{2} \lambda_1 \left\| G \text{div } u \right\|_{H^2_s}^2 + \frac{1}{2} \lambda_2 \left\| G \text{curl } u \right\|_{H^2_e}^2 + \frac{1}{2} \lambda_3 \left\| \mathbb{P} u \right\|_{bc}^2
$$

where $G = \overline{G} \text{diag}(g)$, $\overline{G}$, $\text{Div}$, $\text{Curl}$ and $\mathbb{P}$ are linear operators. Here $g = \frac{1}{4}(g_1 + g_2)$ and $g_1 = g_2 - g_1$, both in a vector representation, i.e. the columns stacked together. Due to their definition [YSM07] the $H^2_v$, $H^2_s$, $H^2_e$ and $bc$-norms are equivalent to the Euclidean norm. Hence we can rewrite the objective function with the $\| \cdot \|_2$-norm and stack the single terms together,

$$
f(u) = \frac{1}{2} \left\| \begin{pmatrix}
G \\
\sqrt{\lambda_1} G \text{div} \\
\sqrt{\lambda_2} G \text{curl} \\
\sqrt{\lambda_3} \mathbb{P}
\end{pmatrix} u + \begin{pmatrix}
\partial_t g \\
0 \\
0
\end{pmatrix} \right\|_2^2
$$

and get a representation presumed in Section 3.
(a) solution of decomposed problem
(b) errors after ten iterations, $\max e = 0.0028$

(c) mean, maximum error and standard deviation (logarithmic scale)

Figure 3: Horn and Schunck: results and error measurements
(a) variable grid: each variable is represented by a dot, location of boundary terms are shaded

(b) decomposed problem: the problem was decomposed into four overlapping regions

(c) modified decomposed problem: additional boundary terms were added for each region at the artificial boundaries

Figure 4: Higher order regularisation: discretisation and decomposition
4.3.2 Results

First experiments showed that a decomposition by the proposed method leads to convex sub-function but with badly conditioned matrices $A_l$. This causes a divergent behaviour of the algorithm.

In order to improve the problem properties we imposed additional terms $\frac{1}{7}\|\partial_n u\|_2^2$ on the artificial boundaries. Thereby each modified sub-problems has boundary regularisation terms at all its boundaries just as if it was a smaller instance of the original one. In Figure 4(c) the location of the boundary terms are represented as shaded areas. In contrast Figure 4(b) shows how the unmodified grid decomposition where each sub-problem is only regularised by boundary terms at the borders of the original problem.

The discrete forms of the new regularisation terms were concentrated into the matrices $B_l$ and applied as described in Section 3.3. As a result of this modification the condition number dropped from $\rho(A_l) \approx 10^8$ to $\rho(A_l + B_l) \approx 10^5$ and the algorithm converged quickly.

The problem parameters were set to $\lambda_1 = \lambda_2 = 0.4$ and $\lambda_3 = 0.2$. The algorithm parameters were chosen by hand, $a = 900$ and $b = 100$. The four areas overlap by two grid cells.

The plot over the two error measurements in Figure 5(c) shows that the three error measurements reduce drastically within the first iteration and reach the order of their final values, $\mu_e = 1.22 \cdot 10^{-1}$, $\sigma_e = 6.39 \cdot 10^{-1}$ and $\max e = 0.02$. Due to Figure 5(b) the errors are mainly located around the artificial boundaries.

4.4 Stopping Criteria

In Section 2.3.2 we defined two stopping criteria, $c_P$ and $c_D$. Both measure the relative change of the primal respectively dual variables. For both experiments their history is plotted in Figure 6 together with the mean error $\mu_e$ and maximum error $\max e$. Especially $c_P$ gives an excellent estimation of the mean error up to an some constant factor - with low computational costs.

4.5 Conclusion

We demonstrated the results of applying the dual decomposition method to two important motion estimation methods based on the optical flow constraints and regularisation terms of different orders. In the case of the functional proposed in [YSM07] we explained how knowledge about the problem structure can be exploited to improve the numerical properties of the algorithm.
(a) solution of decomposed problem  (b) errors after ten iterations, $e_{\text{max}} = 0.02$

(c) mean, maximum error and standard deviation (logarithmic scale)

**Figure 5:** Higher order regularisation: Results
The numerical results showed that a high accuracy - compared to the non-decomposed solution - is reached within few iterations. Most errors are located within the proximity of the artificial boundaries.

Comparison of two stopping criteria showed that for this experiments the change of the primal variable is a good estimation of the evolution of the quality of the solution with a low computational burden.

5 Conclusion

In view of upcoming and existing computational challenges we described a decomposition method for a large class of convex optimisation problems which makes use of the corresponding dual Lagrangian problems. The resulting sub-problems are also convex by construction and hence can be efficiently solved using existing, reliable methods. A framework was worked out that allows to improve numerical properties of the underlying optimisation problems.

The proposed method was applied and validated for two important optimisation problems for motion estimation. Results showed that the algorithm reaches a high accuracy solution within few iterations only.
Figure 6: Error plots and measurements used as stopping criteria.
References


