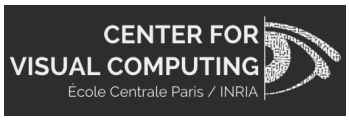


Master of Applied Mathematics: Vision and Learning, 2013 – 2014

Thesis

Dual Decomposition with Accelerated First-Order Scheme for Discrete Markov Random Field Optimization

Applied to Locally Affine Image Registration

D. Khuê Lê-Huu
Supervisor: Professor Nikos Paragios

Abstract

The problem of discrete Markov Random Field optimization, or MAP inference, can be reformulated as an integer linear program (ILP), which is NP-Hard in general. A usual method to approximately solve this ILP is to relax the integral constraints. Many MAP inference methods have been based on this linear programming relaxation. In this work, we propose a new decomposition scheme to solve the dual of this relaxed linear program, where the dependencies between any two nodes of the graph are relaxed using Lagrangian relaxation. Since the dual function is non-differentiable, subgradient methods are first used. The application on a stereo vision problem shows that the convergence rate of these methods, which is $O(1/\epsilon^2)$ in theory, is not practical. Therefore, we smooth the dual using Nesterov's method and then use optimal first-order gradient methods to optimize the obtained smooth function, which result in a better convergence rate of $O(1/\epsilon)$. The method can handle any graph structures with arbitrary potential functions. As an application, a new MRF model for locally affine image registration is proposed.

Contents

1 Introduction

Markov Random Fields (MRFs) are a class of *Probabilistic Graphical Models* (PGMs), which use a graph-based representation to encode a probability distribution, where the nodes represent random variables and the edges represent independencies between them.

Since MRFs has the ability to model soft contextual constraints between random variables, they are extremely suitable for image or scene modeling, which usually involve interactions between a subset of pixels or scene components. The very first application of MRFs in computer vision and image processing was proposed in [Geman and Geman, 1984] for the problem of image denoising/restoration. Since then, they have attracted a significant amount of computer vision research and have become a ubiquitous method for solving all kinds of vision problems, such as image denoising and restoration [Geman and Geman, 1984; Chambolle, 2005], stereo vision [Boykov et al., 2001], multi-view reconstruction [Kolmogorov and Zabih, 2002; Vogiatzis et al., 2007], image segmentation [Rother et al., 2004], optical flow and motion analysis [Glocker et al., 2008], object recognition [Felzenszwalb and Huttenlocher, 2005],... just to name a few. Refer to [Wang et al., 2013] for a survey on MRF modeling, inference and learning in computer vision.

One of the main reasons for which MRFs have become so popular is that many computer vision problems, such as the ones listed above, can be formulated as labeling problems, which can be next seen as problems of Maximum a Posteriori (MAP) inference of some MRFs. There are four main classes of MAP inference methods:

Message-passing, also known as *belief propagation* (BP), was first proposed in [Pearl, 1982] for inference on trees. The idea of message passing is iteratively improving the labeling by passing local messages between neighboring nodes. The messages are the beliefs about the local configuration. The first generalization of BP was *Loopy belief propagation* [Frey and MacKay, 1997] for the use of BP in graphs with loops, which does not provide a guarantee on the convergence and the quality of the solution. Recent generalizations of BP include *tree-reweighted message passing* (TRW) [Wainwright et al., 2005], which approximates the energy function based on a convex combination of trees and then maximizes a lower bound on the energy. However, the algorithm is not guaranteed to increase this bound and thus may not converge. Therefore, [Kolmogorov, 2006] developed a modification of this algorithm, called *sequential tree-reweighted message passing* (TRW-S), in which the lower bound is guaranteed not to decrease.

Move-making methods apply a sequence of minimizations over subsets of the label space, iteratively improving the current labeling. These include graph cut based methods such as α-*expansion* and $\alpha\beta$-*swap* [Boykov et al., 2001] for submodular, metric or semi-metric energy functions; *Quadratic pseudo-boolean optimization* (QPBO) [Kolmogorov and Rother, 2007] for non-submodular energy functions. A generalization of α-expansion was proposed in [Komodakis et al., 2008], called *Fast primal-dual* (FastPD), which optimizes both the MRF optimization problem and its dual at each iteration, leading to a significant speed up. FastPD can handle arbitrary potential functions; however, it might get stuck in sub-optimum.

Combinatorial methods see the labeling problem as an *integer linear program* (ILP) and solve it exactly using combinatorial techniques such as branch-and-bound [Otten and Dechter, 2014; Martins et al., 2011], multicut [Kappes et al., 2011], etc...

Methods in this class provide the exact integer solution, unlike the methods in the other classes that usually provide real solutions (because they use an approximation), which need further a *rounding* step to be converted to feasible integer solutions.

Convex relaxation methods approximate the original labeling problem, which is NP-Hard, based on different relaxations and then use convex optimization techniques to solve the relaxed problem. The most popular class of these methods is *linear programming (LP) relaxation*, which consists of relaxing the integer constraints in the integer linear program being equivalent to the MRF labeling problem. Note that TRW, TRW-S and FastPD can also be considered to fall into this class, since they are based on the LP relaxation. These three methods, however, might stuck in local optimum.

The method proposed in [Komodakis and Paragios, 2009; Komodakis et al., 2011] uses *dual decomposition* (DD) [Bertsekas, 1999] to decompose the dual problem into a number of subproblems which are easy to solve, leveraging the structure of the problem. The sum of the minima of these subproblems corresponds to a value of the dual objective, which is a *lowerbound* of the primal objective. This lowerbound is iteratively maximized using *projected subgradient* methods [Bertsekas, 1999] (note that minimizing the primal objective of an LP is equivalent to maximizing its dual objective).

Based on the same DD framework, [Kappes et al., 2012] proposed to update the dual objective using *bundle methods* [Bertsekas, 1999] instead of using projected subgradients.

These two DD methods are guaranteed to converge to the global optimum. However, they provide a very slow rate of convergence, namely $O(1/\epsilon^2)$ time complexity for an ϵ-accurate solution. This is mainly caused by the non-smoothness of the dual objective. Thus, in [Jojic et al., 2010], Nesterov's smoothing and accelerated first-order gradient method [Nesterov, 2005] was applied to the previous DD framework to obtain the better convergence rate of $O(1/\epsilon)$. However, as pointed out by [Savchynskyy et al., 2011], there was an inconsistency in choosing the norms in [Jojic et al., 2010], which lead to invalid complexity bounds. They also presented a similar approach, correctly choosing the norms, and moreover, dynamically adjust the Lipschitz constant and the smoothing parameter to further accelerate the algorithm.

Finally, other more complex relaxations methods have also been proposed, including the *quadratic programming relaxation* [Ravikumar and Lafferty, 2006] and the *second order cone programming relaxation* [Kumar et al., 2006]. However, as shown in [Kumar et al., 2009], the simple LP relaxation provides a better approximation than these more sophisticated methods.

Refer to [Kappes et al., 2013] and [Wang et al., 2013] for a more complete review on MAP inference methods.

In this work, we will introduce a novel dual decomposition approach for solving the relaxed LP. In this decomposition, the dependencies between any two nodes of the graph are relaxed using Lagrangian relaxation. Similar to [Komodakis and Paragios, 2009], subgradient methods are first used because the dual function is non-differentiable. The application on a stereo vision problem shows that the convergence rate of these methods, which is $O(1/\epsilon^2)$ in theory, is not practical. Thus, we smooth the dual using Nesterov's method

and then use optimal first-order gradient methods to optimize the obtained smooth function, which result in a better convergence rate of $O(1/\epsilon)$. The method can handle any graph structures with arbitrary potential functions. In addition, a new MRF model for locally affine image registration is proposed at the end.

The report is organized as follows. In the next section we give a brief introduction to PGMs and MRFs, mainly on the principle of factorizing a probability distribution and on the MAP inference problem. In section 3, it we will show how to approximate the MAP inference problem by a linear programming relaxation. In section 4, our novel dual decomposition scheme using subgradient optimization is presented, with an application on stereo disparity map estimation. Section 5 is the core of our work, where we combine a smoothing technique with optimal first-order gradient methods to solve our proposed dual problem. As an application, we propose a new MRF model for locally affine image registration in section 6.

2 Markov Random Fields and Probabilistic Graphical Models

Probabilistic graphical models (PGMs) use a graph-based representation to compactly encode a complex probability distribution over a high-dimensional space, where the conditional dependence between the variables are represented by the structure of the graph. In this graphical representation, the nodes represent the considered random variables, and the edges correspond to probabilistic interactions (i.e. dependencies/independencies) between them. These independencies allow the distribution to be represented in a factorized form, i.e. the joint distribution can be decomposed into a product of factors each depending only on a subset of the variables. (And inversely, a particular factorization of the distribution guarantees that certain independencies hold [Koller and Friedman, 2009].)

PGMs have two main classes: the first, called *Bayesian networks*, uses directed acyclic graphs, and the second, *Markov random fields* (MRFs), uses undirected graphs. Directed graphs are useful for expressing causal relationships between random variables, whereas undirected graphs are more suitable for expressing soft constraints between them. These two classes induce different factorizations for the considered distribution (and equivalently, encode different sets of independencies), as presented below.

For a distribution $p(X)$ over the random variable X, denote by $p(x)$ the distribution evaluated for the particular value x, i.e. $p(x) := p(X = x)$. It is similar for multi-dimensional or joint random variables.
Now consider the multi-dimensional (or joint) distribution $p(\mathbf{X}) = p(X_1, X_2, \ldots, X_n)$ over the random variables $\mathbf{X} = (X_1, X_2, \ldots, X_n)$. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with the set of n nodes $\mathcal{V}$, corresponding to n random variables $X_1, \ldots, X_n$, and the set of edges $\mathcal{E}$.

- If G is a directed acyclic graph, then we say that $p(\mathbf{X})$ *factorizes in* $\mathcal{G}$ if and only if $p(\mathbf{x})$ is of the form:
$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i | x_{\pi_i}) \quad \forall \mathbf{x}, \tag{1}$$
where x_{π_i} denotes the set of parents of the node X_i.
The graph $\mathcal{G}$ is thus the Bayesian network encoding the distribution $p(\mathbf{X})$.

- If G is an undirected graph: denote by $\mathcal{C}$ the set of *cliques* of $\mathcal{G}$ (a clique is a set of fully connected nodes), then we say that $p(\mathbf{X})$ *factorizes in* G if and only if $p(\mathbf{x})$ is of the form:
$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(\mathbf{x}_C) \quad \forall \mathbf{x}, \tag{2}$$
where ψ_C are some non-negative functions of the variables $\mathbf{x}_C = (x_i)_{i \in C}$ in the clique C (these functions are called *potentials*), and $Z = \sum_{\mathbf{x}} \prod_{C \in \mathcal{C}} \psi_C(\mathbf{x}_C)$ is a normalization factor (such that the left-hand side of (2) is a valid probability distribution).
The graph $\mathcal{G}$ is thus the MRF encoding the distribution $p(\mathbf{X})$.

For example, for the Bayesian network shown in Figure 1,we have the following factorization:
$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2)p(x_3|x_1, x_2)p(x_4|x_2, x_3),$$

Figure 1: An example of Bayesian network (left) and Markov random field (right).

and for the MRF, we have

$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z}\psi_1(x_1)\psi_2(x_2)\psi_3(x_3)\psi_4(x_4)\times \\ \times \psi_{13}(x_1, x_3)\psi_{23}(x_2, x_3)\psi_{24}(x_2, x_4)\psi_{34}(x_3, x_4)\psi_{234}(x_2, x_3, x_4).$$

We will focus on discrete MRFs (i.e. $\mathbf{x}$ takes value in a discrete set) in this work. More on PGMs can be found in [Koller and Friedman, 2009].

MAP Inference and Energy Minimization

A large variety of important computer vision problems, for example the ones listed previously, can be formulated as labeling problems, where one seeks to optimize some measure of the quality of the labeling. One of the reasons why MRFs are so popular is that, these labeling problems can be viewed as problems of maximum a posteriori (MAP) inference of some MRF, i.e. finding

$$\mathbf{x}_{\text{opt}} = \arg\max_{\mathbf{x}} p(\mathbf{x}) = \arg\max_{\mathbf{x}} \frac{1}{Z}\prod_{C\in\mathcal{C}} \psi_C(\mathbf{x}_C) \tag{3}$$

where one value of the variable $\mathbf{x}$ represents one labeling possible to the labeling problem. In general, this problem is known to be NP-Hard [Shimony, 1994].

This MAP inference problem can be reformulated in terms of an energy minimization problem, which is used more often in computer vision. Since the potentials ψ_C are non-negative, we can define $\psi_C(\mathbf{x}_C) = \exp\{-\theta_C(\mathbf{x}_C)\}$ (like $\psi_C(\cdot)$, we also refer $\theta_C(\cdot)$ as *clique potentials*) and the joint probability becomes

$$p(\mathbf{x}) = \frac{1}{Z}\prod_{C\in\mathcal{C}} \exp\{-\theta_C(\mathbf{x}_C)\} = \frac{1}{Z}\exp\left\{-\sum_{C\in\mathcal{C}}\theta_C(\mathbf{x}_C)\right\}.$$

We define the *energy* of the MRF by

$$E(\mathbf{x}) = \sum_{C\in\mathcal{C}} \theta_C(\mathbf{x}_C). \tag{4}$$

Clearly, the MAP inference problem (3) is equivalent to minimizing this energy:

$$\mathbf{x}_{\text{opt}} = \arg\max_{\mathbf{x}} p(\mathbf{x}) = \arg\min_{\mathbf{x}} E(\mathbf{x}). \tag{5}$$

The most common type of MRFs that is widely used in computer vision is the *pairwise* MRFs, in which the order of maximal cliques is 2 (i.e. any clique contains at most 2 nodes). The energy of a pairwise MRF factorizing in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is thus

$$E(\mathbf{x}) = \sum_{p\in\mathcal{V}} \theta_p(x_p) + \sum_{pq\in\mathcal{E}} \theta_{pq}(x_p, x_q) \tag{6}$$

The terms $\theta_p(\cdot)$ are called the *unary potentials* and the terms $\theta_{pq}(\cdot)$ are the *pairwise potentials*.

3 Linear Programming Relaxation

We consider the problem of optimizing a discrete MRF factorizing in a pairwise graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where we assume that each random variable can take values in a set of labels $\mathcal{L}$ of cardinality $d = |\mathcal{L}|$. Define $\theta_p : \mathcal{L} \to \mathbb{R}$ and $\theta_{pq} : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ the so-called unary potential and pairwise potential function for each node $p \in \mathcal{V}$ and each edge $pq \in \mathcal{E}$. The task of MRF optimization is assigning a label $l_q \in \mathcal{L}$ to each node $p \in \mathcal{V}$ such that the following MRF energy is minimized:

$$E = \sum_{p \in \mathcal{V}} \theta_p(l_p) + \sum_{pq \in \mathcal{E}} \theta_{pq}(l_p, l_q). \tag{7}$$

Define the indicator function $u_p : \mathcal{L} \to \{0, 1\}$ such that $u_p(l) = 1$ if we assign the label l to the node p (i.e. $l_p = l$) and $u_p(l) = 0$ otherwise; the indicator function $u_{pq} : \mathcal{L} \times \mathcal{L} \to \{0, 1\}$ such that $u_{pq}(l, l') = 1$ if we assign the label l to the node p, the label l' to the node q, and $u_{pq}(l, l') = 0$ otherwise. Since each node is assigned with only one label, it is straightforward that the following conditions hold:

$$\sum_{l \in \mathcal{L}} u_p(l) = 1, \qquad \forall p \in \mathcal{V}, \tag{8}$$

$$\sum_{l' \in \mathcal{L}} u_{pq}(l, l') = u_p(l), \qquad \forall pq \in \mathcal{E}, \forall l \in \mathcal{L}, \tag{9}$$

$$\sum_{l' \in \mathcal{L}} u_{pq}(l', l) = u_q(l), \qquad \forall pq \in \mathcal{E}, \forall l \in \mathcal{L}, \tag{10}$$

$$u_p(l) \in \{0, 1\}, u_{pq}(l, l') \in \{0, 1\}, \quad \forall p \in \mathcal{V}, pq \in \mathcal{E}, l \in \mathcal{L}, l' \in \mathcal{L}. \tag{11}$$

We use the notation $\{f(s)\}_{s \in \mathcal{S}}$ to denote the vector consisting of all the possible values of $f(s)$ where s is taken from the finite discrete set $\mathcal{S}$. The dimension of this vector is thus $|\mathcal{S}|$.

Define

$$\begin{aligned}
\mathbf{u}_p &= \{u_p(l)\}_{l \in \mathcal{L}} && \in \{0, 1\}^d && (12)\\
\mathbf{u}_{pq} &= \left\{u_{pq}(l, l')\right\}_{l \in \mathcal{L}, l' \in \mathcal{L}} && \in \{0, 1\}^{d^2} && (13)\\
\mathbf{u} &= \left\{\{\mathbf{u}_p\}_{p \in \mathcal{V}}, \{\mathbf{u}_{pq}\}_{pq \in \mathcal{E}}\right\} && \in \{0, 1\}^{d|\mathcal{V}| + d^2|\mathcal{E}|} && (14)\\
\boldsymbol{\theta}_p &= \{\theta_p(l)\}_{l \in \mathcal{L}} && \in \mathbb{R}^d && (15)\\
\boldsymbol{\theta}_{pq} &= \left\{\theta_{pq}(l, l')\right\}_{l, l' \in \mathcal{L}} && \in \mathbb{R}^{d^2} && (16)\\
\boldsymbol{\theta} &= \left\{\{\theta_p\}_{p \in \mathcal{V}}, \{\theta_{pq}\}_{pq \in \mathcal{E}}\right\} && \in \{0, 1\}^{d|\mathcal{V}| + d^2|\mathcal{E}|}. && (17)\\
& && && (18)
\end{aligned}$$

It is straightforward that the energy in (7) can be written in the form:

$$E(\boldsymbol{\theta}, \mathbf{u}) = \sum_{p \in \mathcal{V}} \boldsymbol{\theta}_p^\top \mathbf{u}_p + \sum_{pq \in \mathcal{E}} \boldsymbol{\theta}_{pq}^\top \mathbf{u}_{pq} \tag{19}$$

and we need to minimize this energy with respect to $\mathbf{u}$, satisfying the constraints (8), (9), (10) and (11). The set of $\mathbf{u}$ satisfying these constraints is known as the *marginal polytope* [Wainwright et al., 2005]. The MRF optimization has now become an integer linear program. If we relax the integer constraints (11) to

$$u_p(l) \geq 0, u_{pq}(l, l') \geq 0, \quad \forall p \in \mathcal{V}, pq \in \mathcal{E}, l \in \mathcal{L}, l' \in \mathcal{L}, \tag{20}$$

and keep all the other constraints, then the marginal polytope becomes the *local marginal polytope*. Let $\mathcal{U}$ denote this local marginal polytope, i.e. $\mathcal{U}$ is given by

$$\mathcal{U} = \{\mathbf{u} \mid \mathbf{u} \text{ satisfies } (8), (9), (10) \text{ and } (20)\}. \tag{21}$$

The LP-relaxed MRF problem is thus given by:

$$\begin{aligned} \text{minimize} \quad & E(\boldsymbol{\theta}, \mathbf{u}) = \sum_{p \in \mathcal{V}} \boldsymbol{\theta}_p^\top \mathbf{u}_p + \sum_{pq \in \mathcal{E}} \boldsymbol{\theta}_{pq}^\top \mathbf{u}_{pq} \\ \text{subject to} \quad & \mathbf{u} \in \mathcal{U}, \end{aligned} \tag{22}$$

We will reformulate the constraint $\mathbf{u} \in \mathcal{U}$ to get a standard look of an LP. Denote by $\mathbf{1}$ the $\mathbb{R}^d$ vector with all elements equal to 1. The constraints (8) can be re-written as $\mathbf{1}^\top \mathbf{u}_p = 1$ for any p, while (9) can be re-written as $\mathbf{D} \cdot \mathbf{u}_{pq} = \mathbf{u}_p \quad \forall pq \in \mathcal{E}$, and (10) as $\mathbf{C} \cdot \mathbf{u}_{pq} = \mathbf{u}_q \quad \forall pq \in \mathcal{E}$, where

$$\mathbf{D} = \mathbf{diag}(\mathbf{1}^\top, \mathbf{1}^\top, \ldots, \mathbf{1}^\top), \quad \mathbf{C} = \begin{bmatrix} \mathbf{diag}(\mathbf{1}) & \mathbf{diag}(\mathbf{1}) & \cdots & \mathbf{diag}(\mathbf{1}) \end{bmatrix} \tag{23}$$

where $\mathbf{D}$ has d vectors in the diagonal and $\mathbf{C}$ has d blocks $\mathbf{diag}(\mathbf{1})$ (thus the size of $\mathbf{D}$ and $\mathbf{C}$ are $d \times d^2$).
Therefore, (22) can be re-written as

$$\begin{aligned} \text{minimize} \quad & E(\boldsymbol{\theta}, \mathbf{u}) = \sum_{p \in \mathcal{V}} \boldsymbol{\theta}_p^\top \mathbf{u}_p + \sum_{pq \in \mathcal{E}} \boldsymbol{\theta}_{pq}^\top \mathbf{u}_{pq} \\ \text{subject to} \quad & \mathbf{1}^\top \mathbf{u}_p = 1 \quad \forall p \in \mathcal{V}, \\ & \mathbf{D} \cdot \mathbf{u}_{pq} = \mathbf{u}_p \quad \forall pq \in \mathcal{E}, \\ & \mathbf{C} \cdot \mathbf{u}_{pq} = \mathbf{u}_q \quad \forall pq \in \mathcal{E}, \\ & \mathbf{1} \succeq \mathbf{u} \succeq \mathbf{0}. \end{aligned} \tag{24}$$

Note that the redundant constraint $\mathbf{1} \succeq \mathbf{u}$ (which can be inferred form the other constraints) has been added. We shall see later that this is for simplicity in solving the sub-problems when doing dual decomposition.

4 Projected Subgradient Dual Decomposition

Dual decomposition (DD) is an old but quite powerful technique to solve non-linear optimization problems [Bertsekas, 1999]. In section 4.1 we present the general principle of dual decomposition, then in section 4.2 we propose a new dual decomposition scheme to solve the LP (24). Note that this decomposition scheme is different from the one in previous works [Komodakis and Paragios, 2009; Komodakis et al., 2011; Kappes et al., 2012; Jojic et al., 2010; Savchynskyy et al., 2011]. Section 4.3 presents experimental results on a problem of stereo disparity map estimation.

4.1 The principle

We briefly present the dual decomposition technique with subgradient (or supergradient) scheme. Details can be found in [Bertsekas, 1999].

The key idea of DD is to decompose a difficult large problem into smaller easier subproblems using Lagrangian relaxation, and then extract a solution by correctly combining the solutions to the sub-problems. Thus, DD has two principal components: several subproblems, called the *slaves*, and a *master* problem will act as a coordinator between the slaves.

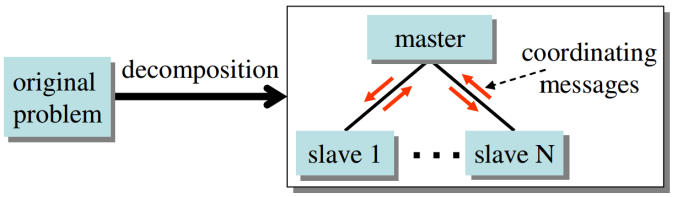


Figure 2: The principle of dual decomposition: the dual problem is decomposed into easier *slave* problems, which are coordinated by a *master* problem. Image taken from [Komodakis et al., 2011].

Consider the following convex problem:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) := \sum_{i=1}^{n} f_i(\mathbf{x}_i) \\ &\text{subject to} && \sum_{i=1}^{n} \mathbf{A}_i \mathbf{x}_i = \mathbf{0}, \end{aligned} \tag{25}$$

where, $f_i : \mathcal{X} \to \mathbb{R}$ are convex functions, $\mathcal{X} \subset \mathbb{R}^n$ is a closed convex set, $\mathbf{A}_i \in \mathbb{R}^{m \times n}$ $(i = 1, \cdots, n)$.

We relax the equality constraints using Lagrange relaxation. The *Lagrangian* is given by:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^{n} f_i(\mathbf{x}) + \boldsymbol{\lambda}^\top \left(\sum_{i=1}^{n} \mathbf{A}_i \mathbf{x}_i \right) = \sum_{i=1}^{n} \left(f_i(\mathbf{x}_i) + \boldsymbol{\lambda}^\top \mathbf{A}_i \mathbf{x}_i \right),$$

and the *(Lagrange) dual function* by:

$$g(\boldsymbol{\lambda}) = \min_{\mathbf{x} \in \mathcal{X}^n} L(\mathbf{x}, \boldsymbol{\lambda}) = \min_{\mathbf{x} \in \mathcal{X}^n} \sum_{i=1}^{n} \left(f_i(\mathbf{x}_i) + \boldsymbol{\lambda}^\top \mathbf{A}_i \mathbf{x}_i \right). \tag{26}$$

The function $f(\mathbf{x})$ can be referred as the *(Lagrange) primal function*. The maximal value of the dual function provide a lowerbound on the minimum of the primal function: $\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}) \leq \min_{\mathbf{x}} f(\mathbf{x})$. This property is called the *weak duality*. When equality holds, we say that the *strong duality* holds. For the problem (25), the condition for which the strong duality holds is that there is at least a point $\mathbf{x} \in \mathcal{X}$ such that the equality constraint holds. We suppose this is the case.

We refer to the problem (25) as the *primal problem* (or simply the primal), and the problem of maximizing the dual function as the *dual problem* (or the dual). If strong duality holds,

then instead of solving directly the primal, we can solve the dual and then recover the optimum $\mathbf{x}^*$ to the primal from the optimum $\boldsymbol{\lambda}^*$ to the dual, using $\mathbf{x}^* = \arg\min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}^*)$.

Clearly, from (26), we can see that the problem of finding the dual function decouples into n independent sub-problems, finding:

$$g_i(\boldsymbol{\lambda}) = \min_{\mathbf{x}_i \in \mathcal{X}} \left\{ f_i(\mathbf{x}_i) + \boldsymbol{\lambda}^\top \mathbf{A}_i \mathbf{x}_i \right\}. \tag{27}$$

These are the slaves. The dual problem becomes

$$\max_{\boldsymbol{\lambda}} \sum_{i=1}^{n} g_i(\boldsymbol{\lambda}), \tag{28}$$

which serves as the master. This problem is convex and so it can be solved using a sub-gradient method (note that $g(\boldsymbol{\lambda})$ is not differentiable) and the convergence to the global optimum is guaranteed. At each iteration, $\boldsymbol{\lambda}$ is updated by $\boldsymbol{\lambda} \longleftarrow \boldsymbol{\lambda} + \alpha_t \nabla g(\boldsymbol{\lambda})$, where $\nabla g(\boldsymbol{\lambda})$ denotes the supergradient of the function $g(\cdot)$ at $\boldsymbol{\lambda}$.

For a function of the form $h(\boldsymbol{\lambda}) = \min_{\mathbf{y} \in \mathcal{C}} \left\{ a(\mathbf{y}) + \boldsymbol{\lambda}^\top b(\mathbf{y}) \right\} = a(\mathbf{y}^*) + \boldsymbol{\lambda}^\top b(\mathbf{y}^*)$ where $\mathcal{C}$ is a compact set, we have

$$h(\boldsymbol{\lambda}') \leq a(\mathbf{y}^*) + (\boldsymbol{\lambda}')^\top b(\mathbf{y}^*) = h(\boldsymbol{\lambda}) + (\boldsymbol{\lambda}' - \boldsymbol{\lambda})^\top b(\mathbf{y}^*),$$

which means $b(\mathbf{y}^*)$ is a supergradient of $h(\cdot)$ at $\boldsymbol{\lambda}$. Applying this result to the dual function (26), we see that $\sum_{i=1}^{n} \mathbf{A}_i \mathbf{x}_i^*$ is a supergradient at $\boldsymbol{\lambda}$, where $\mathbf{x}_i^*$ are the optimal solutions to the slaves (27). Therefore, the update at each iteration is $\boldsymbol{\lambda} \longleftarrow \boldsymbol{\lambda} + \alpha_t \left(\sum_{i=1}^{n} \mathbf{A}_i \mathbf{x}_i^* \right)$.

Note that there are problems in which we have constraints on $\boldsymbol{\lambda}$, for example $\boldsymbol{\lambda} \in \Lambda$, some feasible set. In this case, the updated value of $\boldsymbol{\lambda}$ must be projected onto this set: $\boldsymbol{\lambda} \longleftarrow [\boldsymbol{\lambda} + \alpha_t \nabla g(\boldsymbol{\lambda})]_\Lambda$.

The presented technique was used in the dual decomposition framework proposed by [Komodakis et al., 2011].

4.2 Applied to MRF optimization

We now apply the techniques presented previously to the LP (24).

We first relax the two equality constraints involving $\mathbf{u}_{pq}$ in (24) (i.e. the coupling constraints) using Lagrangian method. The Lagrangian is

$$\begin{aligned} L(\mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\nu}) &= E(\boldsymbol{\theta}, \mathbf{u}) + \sum_{pq \in \mathcal{E}} \boldsymbol{\lambda}_{pq}^\top (\mathbf{D} \cdot \mathbf{u}_{pq} - \mathbf{u}_p) + \sum_{pq \in \mathcal{E}} \boldsymbol{\nu}_{pq}^\top (\mathbf{C} \cdot \mathbf{u}_{pq} - \mathbf{u}_q) \\ &= E(\boldsymbol{\theta}, \mathbf{u}) - \sum_{pq \in \mathcal{E}} \boldsymbol{\lambda}_{pq}^\top \mathbf{u}_p - \sum_{pq \in \mathcal{E}} \boldsymbol{\nu}_{pq}^\top \mathbf{u}_q + \sum_{pq \in \mathcal{E}} \left(\boldsymbol{\lambda}_{pq}^\top \mathbf{D} \cdot \mathbf{u}_{pq} + \boldsymbol{\nu}_{pq}^\top \mathbf{C} \cdot \mathbf{u}_{pq} \right). \end{aligned}$$

If we convert the MRF into a directed graph, where each edge becomes directed (in any manner), then it is seen that $\sum_{pq \in \mathcal{E}} = \sum_{p \in \mathcal{V}} \sum_{q \in \mathrm{Ch}(p)} = \sum_{q \in \mathcal{V}} \sum_{p \in \mathrm{Pa}(q)}$ where $\mathrm{Ch}(p)$ and $\mathrm{Pa}(p)$ respectively denote the set of children and the set of parents of the node p. Therefore, we have

$$\sum_{pq\in\mathcal{E}} \boldsymbol{\lambda}_{pq}^\top \mathbf{u}_p = \sum_{p\in\mathcal{V}} \sum_{q\in \mathrm{Ch}(p)} \boldsymbol{\lambda}_{pq}^\top \mathbf{u}_p = \sum_{p\in\mathcal{V}} \left(\sum_{q\in\mathrm{Ch}(p)} \boldsymbol{\lambda}_{pq} \right)^\top \mathbf{u}_p$$

$$\sum_{pq\in\mathcal{E}} \boldsymbol{\nu}_{pq}^\top \mathbf{u}_q = \sum_{q\in\mathcal{V}} \sum_{p\in \mathrm{Pa}(q)} \boldsymbol{\nu}_{pq}^\top \mathbf{u}_q = \sum_{p\in\mathcal{V}} \sum_{q\in\mathrm{Pa}(p)} \boldsymbol{\nu}_{qp}^\top \mathbf{u}_p = \sum_{p\in\mathcal{V}} \left(\sum_{q\in\mathrm{Pa}(p)} \boldsymbol{\nu}_{qp} \right)^\top \mathbf{u}_p$$

Denote $\boldsymbol{\lambda}_p = -\sum_{q\in\mathrm{Ch}(p)} \boldsymbol{\lambda}_{pq}$ and $\boldsymbol{\nu}_p = -\sum_{q\in\mathrm{Pa}(p)} \boldsymbol{\nu}_{qp}$ we have

$$L(\mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \sum_{p\in\mathcal{V}} (\boldsymbol{\theta}_p + \boldsymbol{\lambda}_p + \boldsymbol{\nu}_p)^\top \mathbf{u}_p + \sum_{pq\in\mathcal{E}} (\boldsymbol{\theta}_{pq} + \mathbf{D}^\top \boldsymbol{\lambda}_{pq} + \mathbf{C}^\top \boldsymbol{\nu}_{pq})^\top \mathbf{u}_{pq}. \tag{29}$$

To obtain the *dual function* $g(\boldsymbol{\lambda}, \boldsymbol{\nu})$, we minimize $L(\mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\nu})$ subject to $\mathbf{1}^\top \mathbf{u}_p = 1 \quad \forall p \in \mathcal{V}$ and $\mathbf{1} \succeq \mathbf{u} \succeq \mathbf{0}$:

$$\begin{aligned} g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = & \min_{\mathbf{1}^\top \mathbf{u}_p = 1, \mathbf{u}_p \succeq \mathbf{0}\ \forall p} \left\{ \sum_{p\in\mathcal{V}} (\boldsymbol{\theta}_p + \boldsymbol{\lambda}_p + \boldsymbol{\nu}_p)^\top \mathbf{u}_p \right\} \\ & + \min_{\mathbf{1}\succeq \mathbf{u}_{pq} \succeq \mathbf{0}\ \forall pq\in\mathcal{E}} \left\{ \sum_{pq\in\mathcal{E}} (\boldsymbol{\theta}_{pq} + \mathbf{D}^\top \boldsymbol{\lambda}_{pq} + \mathbf{C}^\top \boldsymbol{\nu}_{pq})^\top \mathbf{u}_{pq} \right\} \\ = & \sum_{p\in\mathcal{V}} \min_{\mathbf{1}^\top \mathbf{u}_p = 1, \mathbf{u}_p \succeq \mathbf{0}} (\boldsymbol{\theta}_p + \boldsymbol{\lambda}_p + \boldsymbol{\nu}_p)^\top \mathbf{u}_p + \sum_{pq\in\mathcal{E}} \min_{\mathbf{1}\succeq \mathbf{u}_{pq} \succeq \mathbf{0}} (\boldsymbol{\theta}_{pq} + \mathbf{D}^\top \boldsymbol{\lambda}_{pq} + \mathbf{C}^\top \boldsymbol{\nu}_{pq})^\top \mathbf{u}_{pq}. \end{aligned}$$

Therefore, the slave problems are

$$g_p(\boldsymbol{\lambda}_p, \boldsymbol{\nu}_p) = \min_{\mathbf{1}^\top \mathbf{u}_p = 1, \mathbf{u}_p \succeq \mathbf{0}} (\boldsymbol{\theta}_p + \boldsymbol{\lambda}_p + + \boldsymbol{\nu}_p)^\top \mathbf{u}_p, \quad \forall p \in \mathcal{V}, \tag{30}$$

$$g_{pq}(\boldsymbol{\lambda}_{pq}, \boldsymbol{\nu}_{pq}) = \min_{\mathbf{1}\succeq \mathbf{u}_{pq} \succeq \mathbf{0}} (\boldsymbol{\theta}_{pq} + \mathbf{D}^\top \boldsymbol{\lambda}_{pq} + \mathbf{C}^\top \boldsymbol{\nu}_{pq})^\top \mathbf{u}_{pq}, \quad \forall pq \in \mathcal{E}. \tag{31}$$

And the master problem is maximizing

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \sum_{p\in\mathcal{V}} g_p(\boldsymbol{\lambda}_p, \boldsymbol{\nu}_p) + \sum_{pq\in\mathcal{E}} g_{pq}(\boldsymbol{\lambda}_{pq}, \boldsymbol{\nu}_{pq}) \tag{32}$$

over the set

$$\Lambda = \left\{ \{\{\boldsymbol{\lambda}_p\}, \{\boldsymbol{\lambda}_{pq}\}, \{\boldsymbol{\nu}_p\}, \{\boldsymbol{\nu}_{pq}\}\} \;\middle|\; \boldsymbol{\lambda}_p + \sum_{q\in\mathrm{Ch}(p)} \boldsymbol{\lambda}_{pq} = \boldsymbol{\nu}_q + \sum_{p\in\mathrm{Pa}(q)} \boldsymbol{\nu}_{pq} = \mathbf{0} \quad \forall pq \in \mathcal{E} \right\} \tag{33}$$

We can view the above decomposition scheme as a decomposition of the problem into local problems on each node and edge (Figure 3), since the slaves (30) and (31) involved only node and edge variables. Note that the decomposition scheme proposed by [Komodakis et al., 2011] does not include the above scheme.

Solving the slaves

The solution $\bar{\mathbf{u}}_p$ and $\bar{\mathbf{u}}_{pq}$ to the slaves (30) and (31) are straightforward:

$$\bar{\mathbf{u}}_p[i] = 1 \text{ and } \bar{\mathbf{u}}_p[j] = 0 \quad \forall j \neq i, \text{ where } i = \arg\min_i (\boldsymbol{\theta}_p + \boldsymbol{\lambda}_p + \boldsymbol{\nu}_p)[i], \tag{34}$$

$$\bar{\mathbf{u}}_{pq}[i] = \begin{cases} 0 & \text{if } (\boldsymbol{\theta}_{pq} + \mathbf{D}^\top \boldsymbol{\lambda}_{pq} + \mathbf{C}^\top \boldsymbol{\nu}_{pq})[i] \geq 0, \\ 1 & \text{if } (\boldsymbol{\theta}_{pq} + \mathbf{D}^\top \boldsymbol{\lambda}_{pq} + \mathbf{C}^\top \boldsymbol{\nu}_{pq})[i] < 0 \end{cases} \quad \forall i. \tag{35}$$

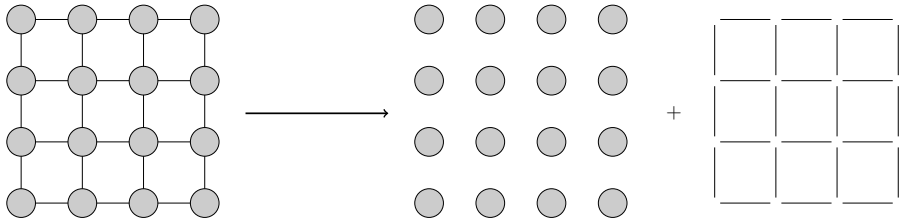


Figure 3: The problem is decomposed into local sub-problems at each node and each edge. Here we use a grid graph for illustration, the decomposition is of course valid for any other structures.

Solving the master

Note that the dual function (32) can be re-written as $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \min \left\{ a(\mathbf{u}) + \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\nu} \end{bmatrix}^\top b(\mathbf{u}) \right\}$ where

$$b(\mathbf{u}) = \begin{bmatrix} \{\mathbf{u}_p\}_{p\in\mathcal{V}} \\ \{\mathbf{y}_{pq}\}_{pq\in\mathcal{E}} \\ \{\mathbf{u}_p\}_{p\in\mathcal{V}} \\ \{\mathbf{z}_{pq}\}_{pq\in\mathcal{E}} \end{bmatrix}, \quad \mathbf{y}_{pq} := \mathbf{D}\cdot\mathbf{u}_{pq}, \quad \mathbf{z}_{pq} := \mathbf{C}\cdot\mathbf{u}_{pq}.$$

Hence, if $\bar{\mathbf{u}}$ is an optimal solution to the slave problems, then $b(\bar{\mathbf{u}})$ is a supergradient of g at $(\boldsymbol{\lambda}, \boldsymbol{\nu})$. Therefore, at each iteration, $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ are updated by:

$$\boldsymbol{\lambda}_p \longleftarrow [\boldsymbol{\lambda}_p + \alpha_t \bar{\mathbf{u}}_p]_\Lambda, \qquad \boldsymbol{\nu}_p \longleftarrow [\boldsymbol{\nu}_p + \alpha_t \bar{\mathbf{u}}_p]_\Lambda \tag{36}$$

$$\boldsymbol{\lambda}_{pq} \longleftarrow [\boldsymbol{\lambda}_{pq} + \alpha_t \bar{\mathbf{y}}_{pq}]_\Lambda, \qquad \boldsymbol{\nu}_{pq} \longleftarrow [\boldsymbol{\nu}_{pq} + \alpha_t \bar{\mathbf{z}}_{pq}]_\Lambda \tag{37}$$

where $\alpha_t > 0$ is the step size at iteration t, and $[\mathbf{a}]_\Lambda$ denotes the projection of $\mathbf{a}$ on the set Λ. This projection is given by Lemma 1 below.

Lemma 1 *The Euclidean projection* $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\nu}})$ *of a given point* $(\mathbf{a}, \mathbf{b})$ *on the set* Λ *is given by:*

$$\bar{\boldsymbol{\lambda}}_{pq} = \mathbf{a}_{pq} - \frac{1}{|\mathrm{Ch}(p)|+1}\left(\mathbf{a}_p + \sum_{q\in\mathrm{Ch}(p)} \mathbf{a}_{pq}\right), \quad \forall pq \in \mathcal{E}, \tag{38}$$

$$\bar{\boldsymbol{\lambda}}_p = -\left(\sum_{q\in\mathrm{Ch}(p)} \bar{\boldsymbol{\lambda}}_{pq}\right), \quad \forall p \in \mathcal{V}; \tag{39}$$

$$\bar{\boldsymbol{\nu}}_{pq} = \mathbf{b}_{pq} - \frac{1}{|\mathrm{Pa}(q)|+1}\left(\mathbf{b}_q + \sum_{p\in\mathrm{Pa}(q)} \mathbf{b}_{pq}\right), \quad \forall pq \in \mathcal{E}, \tag{40}$$

$$\bar{\boldsymbol{\nu}}_q = -\left(\sum_{p\in\mathrm{Pa}(q)} \bar{\boldsymbol{\nu}}_{pq}\right), \quad \forall q \in \mathcal{V}. \tag{41}$$

Proof See Appendices, page 34. □

Using this lemma, it is easy to show that the updating rules (36) and (37) are reduced to

$$\boldsymbol{\lambda}_p \longleftarrow \boldsymbol{\lambda}_p + \alpha_t \Delta\boldsymbol{\lambda}_p, \qquad \boldsymbol{\nu}_p \longleftarrow \boldsymbol{\nu}_p + \alpha_t \Delta\boldsymbol{\nu}_p$$

$$\boldsymbol{\lambda}_{pq} \longleftarrow \boldsymbol{\lambda}_{pq} + \alpha_t \Delta\boldsymbol{\lambda}_{pq}, \qquad \boldsymbol{\nu}_{pq} \longleftarrow \boldsymbol{\nu}_{pq} + \alpha_t \Delta\boldsymbol{\nu}_{pq}$$

where

$$\Delta\boldsymbol{\lambda}_p = \mathbf{u}_p - \frac{\mathbf{u}_p + \sum_{r\in \mathrm{Ch}(p)} \mathbf{y}_{pr}}{|\mathrm{Ch}(p)| + 1}, \qquad \Delta\boldsymbol{\nu}_p = \mathbf{u}_p - \frac{\mathbf{u}_p + \sum_{r\in \mathrm{Pa}(p)} \mathbf{z}_{pr}}{|\mathrm{Pa}(p)| + 1}, \tag{42}$$

$$\Delta\boldsymbol{\lambda}_{pq} = \mathbf{y}_{pq} - \frac{\mathbf{u}_p + \sum_{r\in \mathrm{Ch}(p)} \mathbf{y}_{pr}}{|\mathrm{Ch}(p)| + 1}, \qquad \Delta\boldsymbol{\nu}_{pq} = \mathbf{z}_{pq} - \frac{\mathbf{u}_p + \sum_{r\in \mathrm{Pa}(p)} \mathbf{z}_{pr}}{|\mathrm{Pa}(p)| + 1}. \tag{43}$$

Moreover, instead of updating $(\boldsymbol{\lambda}, \boldsymbol{\nu})$, and then updating $\boldsymbol{\theta}_p + \boldsymbol{\lambda}_p + \boldsymbol{\nu}_p$ and $\boldsymbol{\theta}_{pq} + \mathbf{D}^\top\boldsymbol{\lambda}_{pq} + \mathbf{C}^\top\boldsymbol{\nu}_{pq}$ to solve the slave problems, we can directly update $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_{pq}$ using $\boldsymbol{\theta}_p \longleftarrow \boldsymbol{\theta}_p + \alpha_t\Delta\boldsymbol{\lambda}_p + \alpha_t\Delta\boldsymbol{\nu}_p$ and $\boldsymbol{\theta}_{pq} \longleftarrow \boldsymbol{\theta}_{pq} + \alpha_t\mathbf{D}^\top\Delta\boldsymbol{\lambda}_{pq} + \alpha_t\mathbf{C}^\top\Delta\boldsymbol{\nu}_{pq}$.

Finally, we have the following algorithm:

Algorithm

1. Solve the node slave problems:

   (a) Find an i such that $\boldsymbol{\theta}_p[i] = \min_j \boldsymbol{\theta}_p[j]$.

   (b) Get the solution

   $$\bar{\mathbf{u}}_p[i] = 1 \text{ and } \bar{\mathbf{u}}_p[j] = 0 \quad \forall j \neq i. \tag{44}$$

2. Solve the edge slave problems:

   $$\bar{\mathbf{u}}_{pq}[i] = \begin{cases} 0 & \text{if } \boldsymbol{\theta}_{pq}[i] \geq 0, \\ 1 & \text{if } \boldsymbol{\theta}_{pq}[i] < 0 \end{cases} \quad \forall i. \tag{45}$$

3. Compute $\bar{\mathbf{y}}_{pq} = \mathbf{D}\cdot\bar{\mathbf{u}}_{pq}$ and $\bar{\mathbf{z}}_{pq} = \mathbf{C}\cdot\bar{\mathbf{u}}_{pq}$ and then $\Delta\boldsymbol{\lambda}$ and $\Delta\boldsymbol{\nu}$ using (42) and (43).

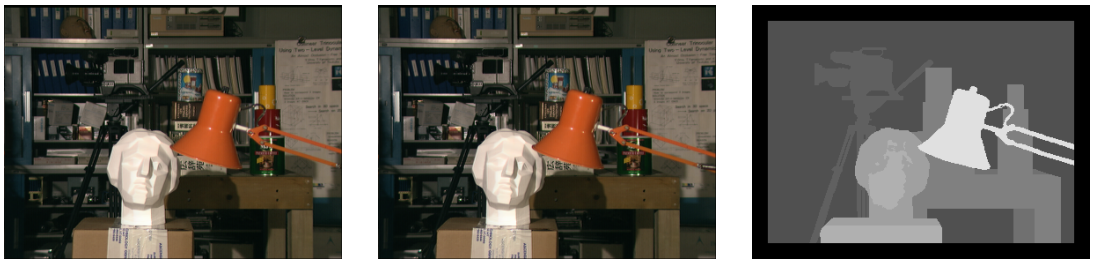
4. Update the slave parameters:

   $$\boldsymbol{\theta}_p \longleftarrow \boldsymbol{\theta}_p + \alpha_t\Delta\boldsymbol{\lambda}_p + \alpha_t\Delta\boldsymbol{\nu}_p$$
   $$\boldsymbol{\theta}_{pq} \longleftarrow \boldsymbol{\theta}_{pq} + \alpha_t\mathbf{D}^\top\Delta\boldsymbol{\lambda}_{pq} + \alpha_t\mathbf{C}^\top\Delta\boldsymbol{\nu}_{pq}.$$

5. Go back to 1. and repeat until convergence.

4.3 Results

We apply our algorithm for computing the disparity maps from the well-known Tsukuba stereo images (Figure 4).



(a) Left image (b) Right image (c) Disparity ground-truth

Figure 4: Images used for the experiments.

We use a grid graph with 4-connectivity, each node corresponds to each pixel of the disparity map and there is an edge connecting any pair of neighboring pixels. The sizes of the images are $w = 384, h = 288$. Thus, the number of nodes is $|\mathcal{V}| = wh = 110592$ and the number of edges is $|\mathcal{E}| = w(h-1) + h(w-1) = 220512$.

The MAP inference consists of assigning each node with a label that corresponds to the disparity of the corresponding pixel. The MRF energy is given by

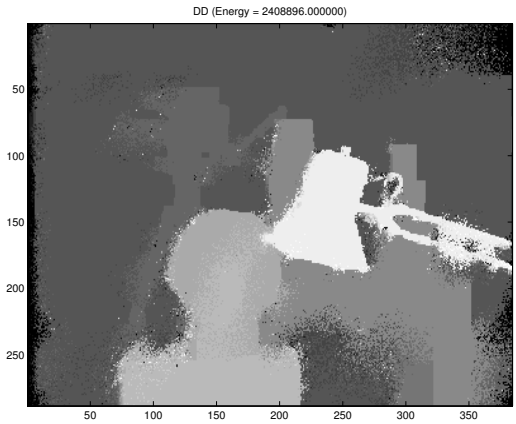
$$E = \sum_{p \in \mathcal{V}} \theta_p(d_p) + \sum_{pq \in \mathcal{E}} \theta_{pq}(d_p, d_q), \tag{46}$$

where d_p is the disparity assigned to the node p, taking values from the set of labels $\mathcal{L} = \{0, 1, \ldots, 16\}$. Denote by I_1 the left grayscale image and by I_2 the right grayscale image.
We define the potentials using truncated absolute difference functions:

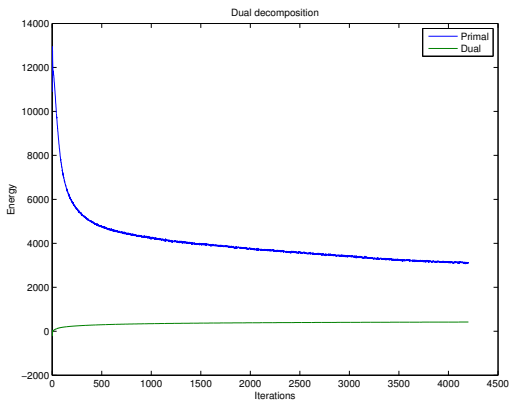
$$\begin{aligned} \theta_p(d_p) &= \min(|I_1(x,y) - I_2(x - d_p, y)|, \sigma) \quad (p = (x,y)) \\ \theta_{pq}(d_p, d_q) &= w_{pq} \min(|d_p - d_q|, \tau). \end{aligned}$$

The unary potentials penalize solutions that are inconsistent with the observed data, they are also called the *data terms*; whereas the pairwise potentials enforces spatial coherence and often called the *smoothness terms*. This model was discussed for example in [Zhang and Seitz, 2007]. We can choose for example $\sigma = 18$, $\tau = 2$ and $w_{pq} = 10 \quad \forall pq \in \mathcal{E}$.

The results are shown in Figure 5.



(a) Estimated disparity map



(b) The values of the primal and the dual.

Figure 5: Results by subgradient dual decomposition.

Theoretically, we know that the algorithm will converge after a big enough number of iterations. Indeed, from the results, we observe that the dual and primal tend to convergence. However, the convergence rate of subgradient methods is $O(1/\epsilon^2)$ and in our example, this is just too loose.

5 Smoothing and Accelerated First-order Gradient Methods

The reason we had to use subgradient methods instead of classical gradient methods was because the (convex) dual function (32) is not differentiable. As we have seen, subgradient methods have a poor convergence rate, which is theoretically $O(1/\epsilon^2)$ [Bertsekas, 1999]. A method to improve the convergent rate is smoothing the dual, and then using classical gradient descent methods, which can achieve $O(1/\epsilon)$ [Bertsekas, 1999]. Or even better, if the gradient of the smoothed objective function is Lipschitz continuous, then a class of accelerated first-order methods can be applied to have $O(\sqrt{L/\epsilon})$ rate (where L is the Lipschitz constant). The combination of smoothing and accelerated first-order method was pioneered by [Nesterov, 2005] for a class of functions. A unified framework for a more general class of functions was introduced in [Beck and Teboulle, 2012].

The strategy has been applied to MRF optimization by several authors, for example by [Jojic et al., 2010] and [Savchynskyy et al., 2011] for the dual decomposition framework proposed in [Komodakis and Paragios, 2009]. In this work, we apply it to our proposed dual decomposition approach.

In Section 5.1, the smoothing technique in [Nesterov, 2005] will be presented. Section 5.2 will show how to apply this technique to our previous MRF dual decomposition approach. In Section 5.3 we show how to apply accelerated first-order methods to minimize the obtained smooth function. Finally, Section 5.4 we apply the method to the previous stereo example to show that it completely outperforms the projected subgradient dual decomposition that we previously proposed in Section 4.

5.1 Smoothing technique

We briefly present the technique proposed by [Nesterov, 2005], which provides a smooth approximation for any function $f : E_1 \to \mathbb{R}$ of the form:

$$f(x) = \max_{u \in Q} \left\{ \langle Ax, u \rangle_{E_2} - \phi(u) \right\} \tag{47}$$

where:

- E_1, E_2 are finite-dimensional real vector spaces;
- Q is a bounded closed convex set in E_2;
- $\phi(\cdot)$ is a continuous convex function on Q;
- $A : E_1 \to E_2^*$ a *linear map* from E_1 to the *dual space* E_2^* of E_2. The *adjoint operator* $A^* : E_2 \to E_1^*$ is defined by $\langle A^* u, x \rangle_{E_1} = \langle Ax, u \rangle_{E_2} \quad \forall x \in E_1, \forall u \in E_2$. The norm of A is given by:

$$\|A\| = \sup_{x \in E_1, u \in E_2} \left\{ \langle Ax, u \rangle_{E_2} : \|x\|_{E_1} = \|u\|_{E_2} = 1 \right\}.$$

For simplicity, the results and derivations are presented for the particular case $E_1 = (\mathbb{R}^n, \|\cdot\|_p)$ and $E_2^* = (\mathbb{R}^m, \|\cdot\|_q)$, to which the application to our problem is limited. We recall the following fundamental results in functional analysis (see for example [Higham, 1992]):

- The dual norm of ℓ_1 norm is ℓ_∞ norm and vice-versa.
- The dual norm of ℓ_p norm ($p > 1$) is $\ell_{p'}$ norm with $\frac{1}{p} + \frac{1}{p'} = 1$.

Thus, we have $E_1^* = \left(\mathbb{R}^n, \|\cdot\|_{\frac{p}{p-1}}\right)$ and $E_2 = \left(\mathbb{R}^m, \|\cdot\|_{\frac{q}{q-1}}\right)$. (We use the convention that $\frac{p}{p-1} = \infty$ if $p = 1$.)

In this case, the linear map A from E_1 to E_2^*, i.e. from $(\mathbb{R}^n, \|\cdot\|_p)$ to $(\mathbb{R}^m, \|\cdot\|_q)$, can be represented by an $m \times n$ matrix $\mathbf{A}$, called the *representing matrix*. The norm of this linear map is the induced norm of the representing matrix:

$$\|A\| = \|\mathbf{A}\|_{pq} = \sup_{\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}} \frac{\|\mathbf{A}\mathbf{x}\|_q}{\|\mathbf{x}\|_p}.$$

Now, we seek a smooth approximation of the function $f : \mathbb{R}^n \to \mathbb{R}$ of the form:

$$f(\mathbf{x}) = \max_{\mathbf{u} \in Q} \left\{ (\mathbf{A}\mathbf{x})^\top \mathbf{u} - \phi(\mathbf{u}) \right\} \tag{48}$$

where $\mathbf{A}$ is an $\mathbb{R}^{m \times n}$ matrix, $Q \subset E_2 = \left(\mathbb{R}^m, \|\cdot\|_{\frac{q}{q-1}}\right)$ a bounded closed convex set, $\phi : Q \to \mathbb{R}$ is a continuous convex function. **We will show later that the (additive) inverse of the dual of our MRF optimization problem can be written in this form.**

Let $d : Q \to \mathbb{R}$ be a function with the following properties:

- $d(\cdot)$ is continuous and strongly convex with convexity parameter $\sigma > 0$.
- $\min_{\mathbf{u} \in Q} d(\mathbf{u}) = 0$. Thus, from the previous property we have $d(\mathbf{u}) \geq \frac{\sigma}{2} \|\mathbf{u} - \mathbf{u}_0\|^2$ where $\mathbf{u}_0 = \arg\min_{\mathbf{u} \in Q} d(\mathbf{u})$ is the *prox center* of Q.

The function $d(\cdot)$ is called a **prox function** of the set Q [Nesterov, 2005]. Note that we have equipped E_2 with the norm $\|\cdot\|_{\frac{q}{q-1}}$, thus, the inequality in the latter property should read $d(\mathbf{u}) \geq \frac{\sigma}{2} \|\mathbf{u} - \mathbf{u}_0\|^2_{\frac{q}{q-1}}$.

Define

$$\boxed{f_\mu(\mathbf{x}) = \max_{\mathbf{u} \in Q} \left\{ (\mathbf{A}\mathbf{x})^\top \mathbf{u} - \phi(\mathbf{u}) - \mu d(\mathbf{u}) \right\} = (\mathbf{A}\mathbf{x})^\top \mathbf{u}_\mu(\mathbf{x}) - \phi(\mathbf{u}_\mu(\mathbf{x})) - \mu d(\mathbf{u}_\mu(\mathbf{x}))} \tag{49}$$

where $\mathbf{u}_\mu(\mathbf{x}) = \arg\max_{\mathbf{u} \in Q} \left\{ (\mathbf{A}\mathbf{x})^\top \mathbf{u} - \phi(\mathbf{u}) - \mu d(\mathbf{u}) \right\}$. We have the following results.

Theorem 1 (Smoothing) *[Nesterov, 2005]*

1. *The function $f_\mu(\mathbf{x})$ is well defined and continuously differentiable at any $\mathbf{x} \in \mathbb{R}^n$. Moreover, this function is convex and its gradient*

$$\nabla f_\mu(\mathbf{x}) = \mathbf{A}^\top \mathbf{u}_\mu(\mathbf{x})$$

is Lipschitz continuous with constant $L_\mu = \frac{\|\mathbf{A}\|^2_{pq}}{\mu\sigma}$, i.e.

$$\|\nabla f_\mu(\mathbf{x}) - \nabla f_\mu(\mathbf{y})\|_p^* \leq \frac{\|\mathbf{A}\|^2_{pq}}{\mu\sigma} \|\mathbf{x} - \mathbf{y}\|_p \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

2. Denote $D = \max_{\mathbf{u}\in Q} d(\mathbf{u})$. *Then for any* $\mathbf{x} \in \mathbb{R}^n$ *we have:*

$$f_\mu(\mathbf{x}) \leq f(\mathbf{x}) \leq f_\mu(\mathbf{x}) + \mu D. \tag{50}$$

From the inequality (50) we see that $f_\mu(\mathbf{x})$ is an ϵ-accurate approximation of $f(\mathbf{x})$ with $\epsilon = \mu D$.

5.2 Applied to smooth the dual of our MRF optimization problem

In subsection 5.2.1, we will first reformulate the MRF dual problem so that the previous smoothing technique can be applied. Next, in section 5.2.2, we will detail how to apply the previous smoothing technique to our new dual problem.

5.2.1 The MRF dual problem

Recall that our MRF optimization problem reduces to the linear program (24). For the purpose of smoothing (that will become clear later), we will slightly reformulate that LP in the following form:

$$\begin{aligned}
\text{minimize} \quad & E(\boldsymbol{\theta}, \mathbf{u}) = \sum_{p\in\mathcal{V}} \boldsymbol{\theta}_p^\top \mathbf{u}_p + \sum_{pq\in\mathcal{E}} \boldsymbol{\theta}_{pq}^\top \mathbf{u}_{pq} \\
\text{subject to} \quad & \mathbf{1}^\top \mathbf{u}_p = 1 \quad \forall p \in \mathcal{V}, \\
& \mathbf{1}^\top \mathbf{u}_{pq} = 1 \quad \forall pq \in \mathcal{E}, \\
& \mathbf{D} \cdot \mathbf{u}_{pq} = \mathbf{u}_p \quad \forall pq \in \mathcal{E}, \\
& \mathbf{C} \cdot \mathbf{u}_{pq} = \mathbf{u}_q \quad \forall pq \in \mathcal{E}, \\
& \mathbf{u} \geq \mathbf{0}.
\end{aligned} \tag{51}$$

Note that the redundant constraint $\mathbf{1}^\top \mathbf{u}_{pq} = 1 \quad \forall pq \in \mathcal{E}$ (which can be inferred from the other constraints) has been added. The dimension of $\mathbf{u}$ is $m = d\,|\mathcal{V}| + d^2\,|\mathcal{E}|$. (We recall that d is the number of labels.)

It is straightforward that there exists a unique matrix $\mathbf{A}$ such that:

$$\mathbf{A}^\top \mathbf{u} = \begin{bmatrix} \{-\mathbf{D}\mathbf{u}_{pq} + \mathbf{u}_p\}_{pq\in\mathcal{E}} \\ \{-\mathbf{C}\mathbf{u}_{pq} + \mathbf{u}_q\}_{pq\in\mathcal{E}} \end{bmatrix}. \tag{52}$$

(We will later explain how to construct $\mathbf{A}$ at page 18.) Thus, the third and fourth equality constraints in the primal problem (51) can be re-written as $\mathbf{A}^\top \mathbf{u} = \mathbf{0}$. Let us also re-write the other constraints as $\mathbf{u} \in Q$, where

$$Q = \left\{ \mathbf{u} \;\middle|\; \mathbf{1}^\top \mathbf{u}_p = 1 \quad \forall p \in \mathcal{V}, \quad \mathbf{1}^\top \mathbf{u}_{pq} = 1 \quad \forall pq \in \mathcal{E}, \quad \mathbf{u} \geq \mathbf{0} \right\}, \tag{53}$$

which is a bounded closed convex set in $\mathbb{R}^m$.

Now, we will relax the constraint $\mathbf{A}^\top \mathbf{u} = \mathbf{0}$ using Lagrangian relaxation. Denote the dual variables by

$$\mathbf{x} = \begin{bmatrix} \{\boldsymbol{\lambda}_{pq}\}_{pq\in\mathcal{E}} \\ \{\boldsymbol{\nu}_{pq}\}_{pq\in\mathcal{E}} \end{bmatrix} \in \mathbb{R}^n, \quad n = 2d\,|\mathcal{E}|\,.$$

Then the Lagrangian is $L(\mathbf{u}, \mathbf{x}) = E(\mathbf{u}) - \mathbf{x}^\top \mathbf{A}^\top \mathbf{u}$ (which can also have the form (29)) and the dual function is

$$g(\mathbf{x}) = \min_{\mathbf{u} \in Q} L(\mathbf{u}, \mathbf{x}) = -\max_{\mathbf{u} \in Q} \left\{ (\mathbf{A}\mathbf{x})^\top \mathbf{u} - E(\mathbf{u}) \right\}.$$

Define:

$$f(\mathbf{x}) = \max_{\mathbf{u} \in Q} \left\{ (\mathbf{A}\mathbf{x})^\top \mathbf{u} - E(\mathbf{u}) \right\} \tag{54}$$

we have $g(\mathbf{x}) = -f(\mathbf{x})$ and thus maximizing $g(\mathbf{x})$ is equivalent to minimizing $f(\mathbf{x})$, which has the form (48) (with $\phi(\cdot) = E(\cdot)$) and thus can be approximated using the smoothing technique previously presented.

5.2.2 Smoothing the dual

To smooth $f(\cdot)$, we need to find a prox function $d(\cdot)$ of the set Q, defined by (53). Then from (49), a smooth approximation of f is then given by:

$$f_\mu(\mathbf{x}) = \max_{\mathbf{u} \in Q} \left\{ (\mathbf{A}\mathbf{x})^\top \mathbf{u} - E(\mathbf{u}) - \mu d(\mathbf{u}) \right\} = \max_{\mathbf{u} \in Q} \left\{ -L(\mathbf{u}, \mathbf{x}) - \mu d(\mathbf{u}) \right\} \tag{55}$$

We use the prox function defined in the following lemma.

Lemma 2 *The function*

$$d(\mathbf{u}) = \sum_{p \in \mathcal{V}} \left(\log d + \sum_{i=1}^{d} u_p^i \log u_p^i \right) + \sum_{pq \in \mathcal{E}} \left(2 \log d + \sum_{i=1}^{d^2} u_{pq}^i \log u_{pq}^i \right) \tag{56}$$

is a prox function, with respect to the ℓ_1 norm, of the set Q defined by (53), with convexity parameter

$$\sigma = \frac{1}{|\mathcal{V}| + |\mathcal{E}|}. \tag{57}$$

Proof See Appendices, page 35. □

Note that in using this prox function, **we automatically equip Q (or E_2) with the norm** ℓ_1, or equivalently, we set $q = \infty$. (Recall that the norm equipped to E_2 is $\|\cdot\|_{\frac{q}{q-1}}$.)

The optimal solution $\mathbf{u}_\mu(\mathbf{x})$ of (55) can be computed:

$$u_p^i = \frac{\exp(a_p^i)}{\sum_{j=1}^{d} \exp(a_p^j)} \quad i = 1, \ldots, d \tag{58}$$

$$u_{pq}^i = \frac{\exp(a_{pq}^i)}{\sum_{j=1}^{d^2} \exp(a_{pq}^j)} \quad i = 1, \ldots, d^2. \tag{59}$$

and finally the smooth approximation is:

$$\boxed{f_\mu(\mathbf{x}) = \mu \sum_{p \in \mathcal{V}} \log \left(\sum_{i=1}^{d} \exp(a_p^i) \right) + \mu \sum_{pq \in \mathcal{E}} \log \left(\sum_{i=1}^{d^2} \exp(a_{pq}^i) \right) - \mu D} \tag{60}$$

where

$$\mathbf{a}_p = -\frac{1}{\mu}(\boldsymbol{\theta}_p + \boldsymbol{\lambda}_p + \boldsymbol{\nu}_p)$$
$$\mathbf{a}_{pq} = -\frac{1}{\mu}(\boldsymbol{\theta}_{pq} + \mathbf{D}^\top \boldsymbol{\lambda}_{pq} + \mathbf{C}^\top \boldsymbol{\nu}_{pq})$$

and

$$D = \max_{\mathbf{u}\in Q} d(\mathbf{u}) = \sum_{p\in\mathcal{V}} \log d + \sum_{pq\in\mathcal{E}} 2\log d = (|\mathcal{V}| + 2\,|\mathcal{E}|)\log d. \tag{61}$$

Refer to the Appendices, page 36 for the calculation details.

Now, according to Theorem 1 (page 15), we have the following results:

1. The function $f_\mu(\mathbf{x})$ is well defined and continuously differentiable at any $\mathbf{x} \in \mathbb{R}^n$. Moreover, this function is convex and its gradient

$$\nabla f_\mu(\mathbf{x}) = \mathbf{A}^\top \mathbf{u}_\mu(\mathbf{x}) = \begin{bmatrix} \{-\mathbf{D}\mathbf{u}_\mu(\mathbf{x})_{pq} + \mathbf{u}_\mu(\mathbf{x})_p\}_{pq\in\mathcal{E}} \\ \{-\mathbf{C}\mathbf{u}_\mu(\mathbf{x})_{pq} + \mathbf{u}_\mu(\mathbf{x})_q\}_{pq\in\mathcal{E}} \end{bmatrix} \quad \text{using (52)} \tag{62}$$

   is Lipschitz continuous, with respect to the norm $\|\cdot\|_p$, with constant

$$L_\mu = \frac{\|\mathbf{A}\|_{p,\infty}^2}{\mu\sigma} = \frac{(|\mathcal{V}| + |\mathcal{E}|)\,\|\mathbf{A}\|_{p,\infty}^2}{\mu}. \tag{63}$$

2. For any $\mathbf{x} \in \mathbb{R}^n$ we have:

$$f_\mu(\mathbf{x}) \le f(\mathbf{x}) \le f_\mu(\mathbf{x}) + \mu D \tag{64}$$

   where $D = (|\mathcal{V}| + 2\,|\mathcal{E}|)\log d$.

Computation of the Lipschitz constant

The Lipschitz constant is computed using (63), which requires the computation of $\|\mathbf{A}\|_{p,\infty}^2$. We are particularly interested in the ℓ_1, ℓ_2 and ℓ_∞ norms. We have the following results for the induced norm $\|\mathbf{A}\|_{p,q}$ (see [Higham, 1992; Drakakis and Pearlmutter, 2009] for example):

$$\|\mathbf{A}\|_{1,\infty} = \max_{1\le i\le m, 1\le j\le n} |a_{ij}| \tag{65}$$
$$\|\mathbf{A}\|_{2,\infty} = \max_{1\le i\le m} \|\mathbf{a}_i\|_2 \tag{66}$$
$$\|\mathbf{A}\|_{\infty,\infty} = \max_{1\le i\le m} \|\mathbf{a}_i\|_1 \tag{67}$$

where $\mathbf{a}_i$ is the i-th row of $\mathbf{A}$.

The matrix $\mathbf{A}$ is defined by (52). We construct it as follows. Define $\mathbf{A}^\top = \begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_2 \\ \mathbf{A}_2 & \mathbf{B}_2 \end{bmatrix}$ such that

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 \end{bmatrix} \mathbf{u} = \{-\mathbf{D}\mathbf{u}_{pq} + \mathbf{u}_p\}_{pq\in\mathcal{E}}$$
$$\begin{bmatrix} \mathbf{A}_2 & \mathbf{B}_2 \end{bmatrix} \mathbf{u} = \{-\mathbf{C}\mathbf{u}_{pq} + \mathbf{u}_q\}_{pq\in\mathcal{E}}$$

where $\mathbf{A}_1$ and $\mathbf{A}_2$ have $|\mathcal{E}| \times |\mathcal{V}|$ blocks, each block is a $d \times d$ matrix; $\mathbf{B}_1$ and $\mathbf{B}_2$ have $|\mathcal{E}| \times (|\mathcal{V}| + |\mathcal{E}|)$ blocks, each block is a $d \times d^2$ matrix.
For any (directed) edge pq, let e_{pq} denotes its edge number, $1 \leq e_{pq} \leq |\mathcal{E}|$ (the edges are numbered from 1 to $|\mathcal{E}|$). We set

$$\mathbf{A}_1[e_{pq}, p] = \mathbf{I}_d, \quad \mathbf{B}_1[e_{pq}, e_{pq}] = -\mathbf{D}, \quad \mathbf{A}_2[e_{pq}, q] = \mathbf{I}_d, \quad \mathbf{B}_2[e_{pq}, e_{pq}] = -\mathbf{C} \quad \forall pq \in \mathcal{E},$$

where $\mathbf{I}_d$ is the d-dimensional identity matrix and $\mathbf{M}[i, j]$ denotes the (i, j) block of a block matrix $\mathbf{M}$. (We recall that $\mathbf{D}$ and $\mathbf{C}$ are defined by (23), thus $\mathbf{A}$ contains only $0, 1$ and -1.) Clearly, $\mathbf{A}$ also encodes the structure of the graph.

With this construction of the matrix $\mathbf{A}$, we can easily have

$$\|\mathbf{A}\|_{1,\infty} = 1, \quad \|\mathbf{A}\|_{2,\infty} = \max_{p \in \mathcal{V}} \sqrt{|\mathcal{N}(p)|}, \quad \|\mathbf{A}\|_{\infty,\infty} = \max_{p \in \mathcal{V}} |\mathcal{N}(p)| \tag{68}$$

where $\mathcal{N}(p)$ is the set of neighboring nodes of node p.

5.3 Accelerated first-order methods

Once the objective has been smoothed, we obtain a convex and continuously differentiable function with Lipschitz continuous gradient. A class of very efficient methods, called *accelerated first-order methods* and pioneered by [Nesterov, 1983], are very suitable for this type of function. "Accelerated" because they provide an improved convergence rate of $O(\sqrt{L/\epsilon})$ (where L is the Lipschitz constant), compared to the classical gradient method with $O(L/\epsilon)$. These methods are also often called *optimal first-order methods* since the complexity bound that they provide $O(\sqrt{L/\epsilon})$ is the best complexity bound that one can obtain using only first-order information, as proved by [Nesterov, 1988].

An extension for minimizing composite functions (of the form $F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$ where f is smooth convex, g is convex but possibly non-smooth) was presented in [Nesterov et al., 2007]. More recently, [Beck and Teboulle, 2009] presented Fast Iterative Shrinkage-Thresholding Algorithm (FISTA), which is an extension of [Nesterov, 1983] for handling composite functions (the two are equivalent when $g = 0$).

It should be noted that the level of applicability of these methods are not the same, depending on the norm of the Lipschitz continuity of the gradient. For example, the methods in [Nesterov, 1988, 2005; Nesterov et al., 2007] can handle any norms, whereas [Nesterov, 1983] and FISTA [Beck and Teboulle, 2009] can be applied only for ℓ_2 norm.

Overview and unified analysis can be found in [Nesterov, 2004] and [Tseng, 2008].

We will use FISTA [Nesterov, 1983; Beck and Teboulle, 2009] for simplicity. Since FISTA can only handle ℓ_2, we choose $p = 2$.

The algorithm is presented below (refer to [Beck and Teboulle, 2009] for details), where $f(\mathbf{x})$ is convex and continuously differentiable, the gradient of f is Lipschitz continuous, with respect to the norm ℓ_2, with Lipschitz constant L. If L is not known or computable, or if it is just too loose (i.e. too large), then we use a backtracking step, as shown in the next algorithm.

FISTA with constant step size

- Input: $L > 0, \mathbf{y} = \mathbf{x}_0 \in \mathbb{R}^n, t_0 = 1,$.
- Repeat for $k = 1, 2, \ldots$:
  1. $\mathbf{x}_k = \mathbf{y} - \frac{1}{L}\nabla f(\mathbf{y})$
  2. $t_k = \frac{1+\sqrt{1+4t_{k-1}^2}}{2}$
  3. $\mathbf{y} = \mathbf{x}_k + \frac{t_{k-1}-1}{t_k}(\mathbf{x}_k - \mathbf{x}_{k-1})$

FISTA with backtracking

- Input: $L_0 > 0, \beta > 1, \mathbf{y} = \mathbf{x}_0 \in \mathbb{R}^N, t_0 = 1,$.
- Repeat for $k = 1, 2, \ldots$:
  1. $L_k = L_{k-1}$
  2. $\mathbf{x}_k = \mathbf{y} - \frac{1}{L_k}\nabla f(\mathbf{y})$
  3. While $f(\mathbf{x}_k) > f(\mathbf{y}) + \nabla f(\mathbf{y})^\top(\mathbf{x}_k - \mathbf{y}) + \frac{L_k}{2}\|\mathbf{x}_k - \mathbf{y}\|_2^2$
     (a) $L_k = \beta L_k$
     (b) $\mathbf{x}_k = \mathbf{y} - \frac{1}{L_k}\nabla f(\mathbf{y})$
     End
  4. $t_k = \frac{1+\sqrt{1+4t_{k-1}^2}}{2}$
  5. $\mathbf{y} = \mathbf{x}_k + \frac{t_{k-1}-1}{t_k}(\mathbf{x}_k - \mathbf{x}_{k-1})$

5.4 Applied to the Tsukuba stereo problem

We use the same grid-graph as previous. Recall that $|\mathcal{V}| = 110592, |\mathcal{E}| = 220512, d = 16$. Thus, if we choose $\mu = 10^{-4}$, then it is guaranteed that we will get an ϵ accurate solution, with $\epsilon = \mu D_2 = \mu\left(|\mathcal{V}| + 2\,|\mathcal{E}|\right)\log d \simeq 153$.

The Lipschitz constant (63) becomes

$$L_\mu = \frac{\left(|\mathcal{V}| + |\mathcal{E}|\right)\|\mathbf{A}\|_{2,\infty}^2}{\mu} = \frac{\left(|\mathcal{V}| + |\mathcal{E}|\right) \times \max_{p\in\mathcal{V}} \times |\mathcal{N}(p)|}{\mu} = \frac{\left(|\mathcal{V}| + |\mathcal{E}|\right) \times 4}{\mu} = 4 \times (110592 + 220512) \times 10^4. \tag{69}$$

This value is too loose. Therefore, the FISTA algorithm with backtracking was used. The results are shown in Figure 7. The optimal energy of this problem is about 485. After 500 iterations, the energy reached 576. We observe clearly that with smoothing and accelerated first-order methods, the convergence has been significantly improved over subgradient methods. In future work, we plan to study different smoothing schemes as well as different optimal first-order methods, and perform a comparison with state-of-the art methods.

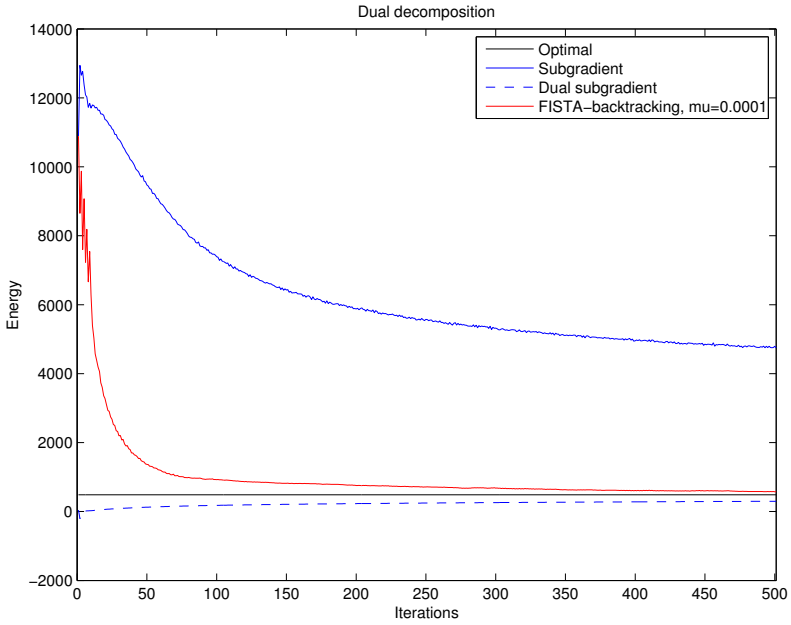
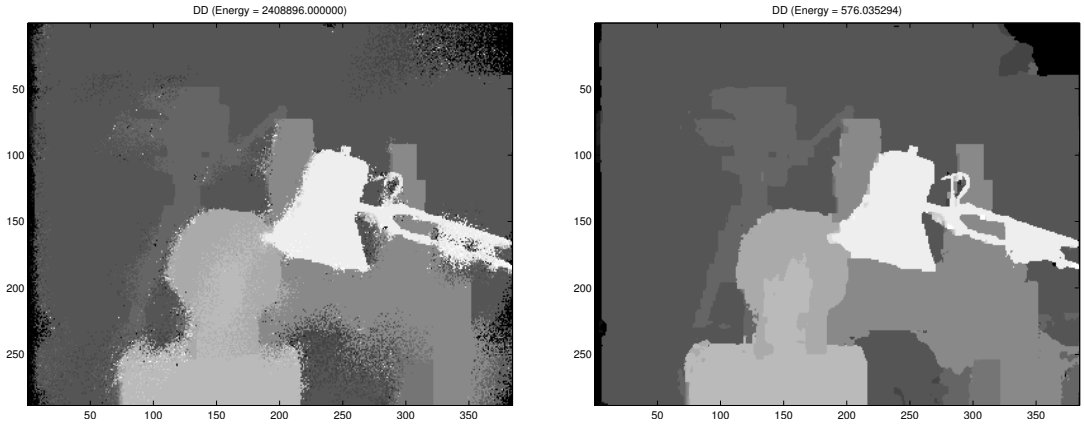


Figure 6: Energy



(a) Subgradient, after 5000 iterations

(b) Smoothing, after 500 iterations

Figure 7: Results

5.5 Related work

Several authors tried to improve the convergence rate of the dual decomposition framework proposed by [Komodakis and Paragios, 2009], where subgradient methods were used. [Jojic et al., 2010] used the smoothing and accelerated gradient method by [Nesterov, 2005] for higher-order graphs. [Savchynskyy et al., 2011] used the the smoothing method of [Nesterov, 2005] in combination with an accelerated gradient method by [Nesterov, 2004], for binary grid-graphs with two acyclic slaves.

6 Locally Affine Image Registration Using Markov Random Fields

Image registration is a fundamental task that has been extensively studied and applied in medical image analysis [Paragios et al., 2014]. The task of image registration is to find a spatial transformation T aligning two (or a set of) images. Given two images I_1 and I_2, one seeks a spatial transformation T such that $T(I_1)$ and I_2 are matched. This problem is often formulated as a minimization problem:

$$\min_{T} \quad \mathcal{S}(T(I_1), I_2) + \mathcal{R}(T)$$

where $\mathcal{S}(\cdot, \cdot)$ is some similarity measure that quantifies the level of alignment between the images, and $\mathcal{R}(\cdot)$ is a regularization term that may favor any specific properties in the solution that the user requires, and seeks to tackle the difficulty associated with the ill-posedness of the problem [Sotiras et al., 2013]. We called I_1 the *floating image* and I_2 the *fixed image*.

The type of the transformation usually defines the name of the corresponding registration task. *Parametric registration* considers the transformation as a parameterized model where each parameter defines a degree of freedom of the deformation. An example is affine registration with 6 degrees of freedom. These models often offer a good compromise between performance and computational complexity. In *non-parametric registration*, also called *dense* or *deformable registration*, each pixel has its individual transformation, which is more challenging and often requires hight computational cost.

One approach for reducing computational complexity is to restrict the transformation to be of low degree of freedoms, by either considering the dense deformation as a combination of parametric deformations (such as piecewise affine [Pitiot et al., 2003; Commowick et al., 2008] or poly-affine [Arsigny et al., 2003, 2006]), or using control-points interpolated deformation models (such as Free Form Deformations [Sederberg and Parry, 1986; Rueckert et al., 1999]).

A complete survey on many aspects (models, similarity measures, optimization methods, etc...) on deformable medical image registration can be found in [Sotiras et al., 2013].

In this work, we propose a new locally affine model for image registration, in which each pixel is supposed to be deformed under a possible number of affine deformations and the optimal transformation will be computed using a discrete MRF formulation. This is an ongoing work and only some parts of it have been done. In the next section, we will present the general idea of the model and explain how the problem can be reformulated as a labeling problem. In section 6.2, the construction the MRF model for this problem will be drawn, and in section 6.3, a method for the affine deformation step will be presented. Since this is still an ongoing work, we have not produced any experimental results yet.

6.1 The idea

We assume that the local deformation around any pixel is affine, i.e. the neighborhood of any pixel $\mathbf{p} = (x, y)$ in I_1 can be perfectly registered to the image I_2 by an affine transformation, under which $\mathbf{p}$ will be matched to its (true) corresponding pixel $\mathbf{p}' = (x', y')$ in I_2 (we assume there is no occlusion) (Figure 8). Convention: a pixel $\mathbf{p}$ may denote its image

coordinates $\mathbf{p} = (x, y)$ or its homogeneous coordinates $(x, y, 1)$, depending on the context.

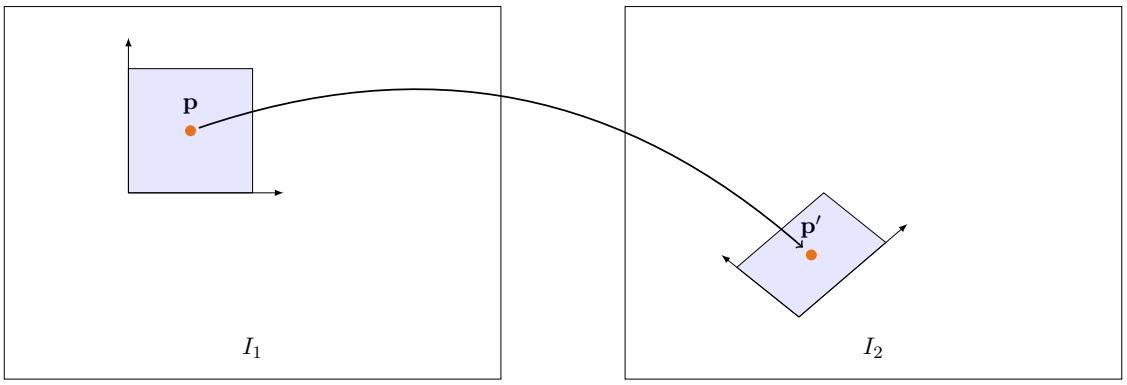


Figure 8: We assume that the local deformation around any pixel is affine.

Now consider the following problem: given a patch (of some radius) centered at $\mathbf{p}$ in I_1, register it to I_2. This problem is called (affine) *template matching* in the Computer Vision literature [Korman et al., 2013] (the size of the patch is considered small compared to the image, hence the name *template*).

Clearly, if we can solve exactly this template matching problem for every pixel in I_1, then the flow field can be trivially obtained. Now suppose we use an iterative method, which is fast but the quality of the solution depends on the initialization (an example of such a method is the well-known Lucas-Kanade algorithm [Lucas and Kanade, 1981]). A possible solution to this initialization-sensitivity problem is running the algorithm with different initializations and then choosing the best one. (More robust methods exist for solving the template matching problem, for example [Korman et al., 2013], which does not rely on initialization and is guaranteed to find an approximation to the global optimum. These will be investigated in future work.)

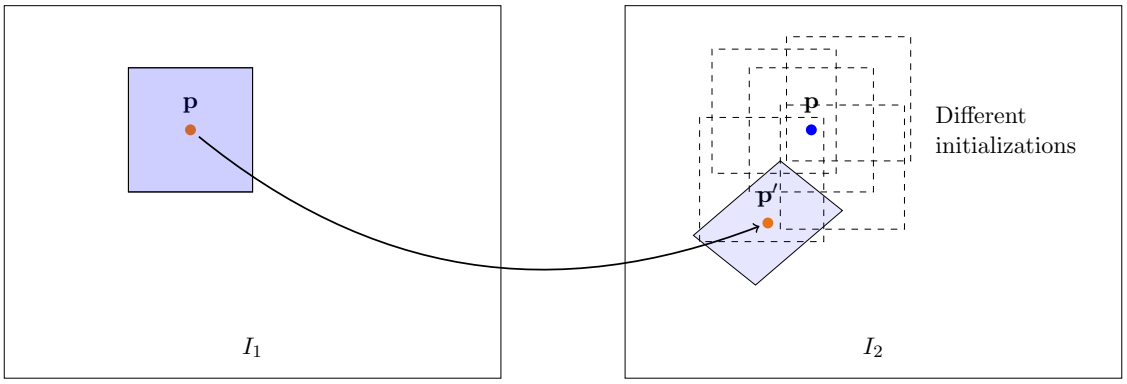


Figure 9: For template matching algorithms that are sensitive to initialization, a possible solution is trying with different initializations and then choosing the best one.

Denote $P_1(\mathbf{p})$ a patch centered at $\mathbf{p}$ in I_1. For every pixel $\mathbf{p}$, we move this patch around $\mathbf{p}$ by a vector $\mathbf{d}$ and use the new patch as initialization for the template matching algorithm (the new patch is centered at $\mathbf{p} + \mathbf{d}$). Denote $\mathbf{A}_p^{\mathbf{d}}$ **the affine transformation matrix returned by the algorithm, and** $e_p^{\mathbf{d}}$ **the corresponding matching error**.

Denote $\mathcal{D}$ the pre-defined set of $\mathbf{d}$ and suppose that $\mathcal{D}$ is the same for all pixels. For example, $\mathcal{D}$ can be defined by sampling along the main coordinate axes, as shown in Figure 10. In this case,

$$\mathcal{D} = \{\ldots, (-2\delta, 0), (-\delta, 0), (0, 0), (\delta, 0), \ldots, (0, \delta), (0, -\delta) \ldots\}.$$

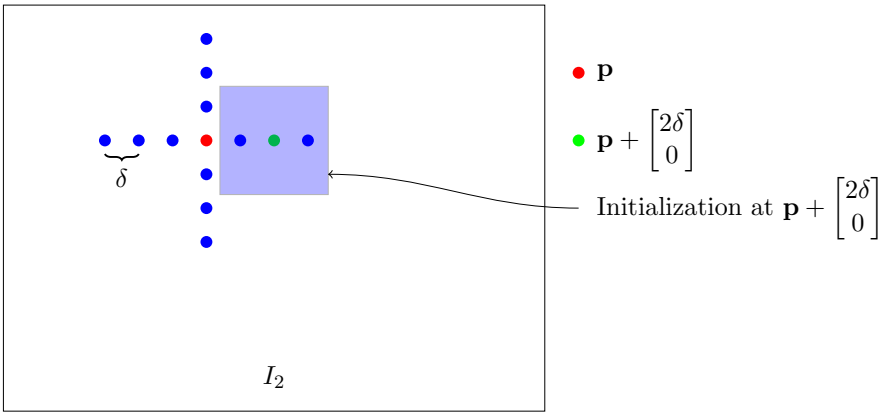


Figure 10: Sampling the position of the initialization along the main coordinate axes.

Note that moving the patch by $\mathbf{d}$ and use it as initialization is equivalent to initializing the transformation matrix $\mathbf{A}$ at $\mathbf{I} + \mathbf{d}$, where $\mathbf{I}$ is the identity matrix. Recall that any affine transformation matrix is of the form

$$\mathbf{A} = \begin{bmatrix} r_1 & r_2 & t_1 \\ r_3 & r_4 & t_2 \\ 0 & 0 & 1 \end{bmatrix}.$$

Now for each pixel $\mathbf{p}$, we have $|\mathcal{D}|$ candidates of transformation matrix $\mathbf{A}_p^{\mathbf{d}}$, where $\mathbf{d} \in \mathcal{D}$. Under $|\mathcal{D}|$ possible affine transformations, a pixel $\mathbf{p}$ in I_1 is mapped to $|\mathcal{D}|$ pixels in I_2 (two or more of these pixels may overlap). These $|\mathcal{D}|$ affine transformations are defined by $|\mathcal{D}|$ matrices $\mathbf{A}_p^{\mathbf{d}_p}, \mathbf{d}_p \in \mathcal{D}$.

We need to choose for each pixel the most appropriate affine transformation from the set of possible ones, such that at the end the difference between the two images are minimized. Clearly, this is a multi-labeling problem where each label is assigned to each possible affine transformations, and thus, can be solved using MRFs.

6.2 MRF modeling

Recall that we need to choose for each pixel an affine transformation among the $|\mathcal{D}|$ possible ones, thus we can define the set of labels as $\mathcal{L} = \{1, \ldots, |\mathcal{D}|\}$, where each label corresponds to a translation vector $\mathbf{d}$ in $\mathcal{D}$, and each pixel defines a node in the MRF. For simplicity, we use a grid graph as usual.

The labeling problem can be solved by minimizing the following MRF energy:

$$\min_{l_p \in \mathcal{L}, p \in \mathcal{V}} E := \sum_{p \in \mathcal{V}} \theta_p(l_p) + \sum_{pq \in \mathcal{E}} \theta_{pq}(l_p, l_q). \tag{70}$$

The variables and functions are defined below.

Unary potentials penalize solutions that are inconsistent with the observed data (thus they are also called *data terms*). If we consider the pixels independently, then the optimal assignments should be the ones that minimize the matching error. Thus we can define the data terms as:

$$\theta_p(l_p) = e_p^{\mathbf{d}_p}. \tag{71}$$

Pairwise potentials penalize displacement changes between adjacent pixels (thus they are also called *smoothness terms*). For any adjacent pixels $\mathbf{p}$ and $\mathbf{q}$, their corresponding displacements $\mathbf{u}_p$ and $\mathbf{u}_q$ should not be too different. If $\mathbf{p}$ and $\mathbf{q}$ are assigned l_p and l_q respectively, then their corresponding displacements are

$$\mathbf{u}_p(l_p) = \mathbf{A}_p^{\mathbf{d}_p}\mathbf{p} - \mathbf{p}, \quad \mathbf{u}_q(l_q) = \mathbf{A}_q^{\mathbf{d}_p}\mathbf{q} - \mathbf{q}.$$

Hence, we can define the smoothness terms as increasing functions of $\|\mathbf{u}_p - \mathbf{u}_q\|$, for example

$$\theta_{pq}(l_p, l_q) = w_{pq} \|\mathbf{u}_p(l_p) - \mathbf{u}_q(l_q)\| \tag{72}$$

where w_{pq} is some weighting coefficient.

6.3 Affine template matching and Lucas-Kanade algorithm

In this section we present the Lucas-Kanade method [Lucas and Kanade, 1981] for doing affine registration. An in-depth discussion on Lucas-Kanade method and its variants was given in [Baker and Matthews, 2004], on which our presentation is based on.

We want to align a template $T(\mathbf{x})$ to an input image $I(\mathbf{x})$, where $\mathbf{x} = (x, y)$. Let $\mathbf{s}$ denote the vector of parameters and $\mathbf{W}(\mathbf{x}; \mathbf{s})$ denote the parameterized set of allowed warps. The warp $\mathbf{W}(\mathbf{x}; \mathbf{s})$ takes the pixel $\mathbf{x}$ in the template T and maps it to the sub-pixel location $\mathbf{W}(\mathbf{x}; \mathbf{s})$ in the image I.

The goal of the Lucas-Kanade algorithm is to minimize the sum of squared errors between two images: the template T and the image I warped back onto the coordinate frame of the template, i.e. minimizing

$$f(\mathbf{s}) = \sum_{\mathbf{x} \in T} [I(\mathbf{W}(\mathbf{x}; \mathbf{s})) - T(\mathbf{x})]^2 \tag{73}$$

with respect to $\mathbf{s}$. This is a non-linear optimization problem, and the Lucas-Kanade method solves it using *steepest descent method*. At each iteration, the parameters are updated by $\mathbf{s} \longleftarrow \mathbf{s} + \Delta\mathbf{s}$ where $\Delta\mathbf{s}$ is a descent direction. The direction $\Delta\mathbf{s}$ is a *steepest descent direction* if it decreases the objective function the most, i.e. it minimizes $f(\mathbf{s}+\Delta\mathbf{s})$.

Consider the first-order Taylor expansion on $[I(\mathbf{W}(\mathbf{x}; \mathbf{s}))$ we have

$$f(\mathbf{s} + \Delta\mathbf{s}) = \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}; \mathbf{s} + \Delta\mathbf{s})) - T(\mathbf{x})]^2 \tag{74}$$

$$\approx \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x}; \mathbf{s})) + \nabla I(\mathbf{W}(\mathbf{x}; \mathbf{s}))^\top \cdot \frac{\partial \mathbf{W}}{\partial \mathbf{s}}(\mathbf{x}; \mathbf{s}) \cdot \Delta\mathbf{s} - T(\mathbf{x})]^2 \tag{75}$$

where $\nabla I(\mathbf{x}) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)^\top$ denotes the gradient of image I at $\mathbf{x} = (x, y)$, $\frac{\partial \mathbf{W}}{\partial \mathbf{s}}$ is the Jacobian of the warp.

Minimizing the above quantity by setting the derivative with respect to $\Delta\mathbf{s}$ to zero:

$$2\sum_{\mathbf{x}}\left(\nabla I(\mathbf{W}(\mathbf{x};\mathbf{s}))^\top\frac{\partial\mathbf{W}}{\partial\mathbf{s}}(\mathbf{x};\mathbf{s})\right)^\top\left(I(\mathbf{W}(\mathbf{x};\mathbf{s}))+\nabla I(\mathbf{W}(\mathbf{x};\mathbf{s}))^\top\frac{\partial\mathbf{W}}{\partial\mathbf{s}}(\mathbf{x};\mathbf{s})\Delta\mathbf{s}-T(\mathbf{x})\right)=\mathbf{0},$$

we get

$$\Delta\mathbf{s}=\mathbf{H}^{-1}\sum_{\mathbf{x}}\left(\nabla I(\mathbf{W}(\mathbf{x};\mathbf{s}))^\top\frac{\partial\mathbf{W}}{\partial\mathbf{s}}(\mathbf{x};\mathbf{s})\right)^\top(T(\mathbf{x})-I(\mathbf{W}(\mathbf{x};\mathbf{s}))) \tag{76}$$

where $\mathbf{H}$ is the Hessian matrix:

$$\mathbf{H}=\sum_{\mathbf{x}}\left(\nabla I(\mathbf{W}(\mathbf{x};\mathbf{s}))^\top\frac{\partial\mathbf{W}}{\partial\mathbf{s}}(\mathbf{x};\mathbf{s})\right)^\top\left(\nabla I(\mathbf{W}(\mathbf{x};\mathbf{s}))^\top\frac{\partial\mathbf{W}}{\partial\mathbf{s}}(\mathbf{x};\mathbf{s})\right). \tag{77}$$

The Lucas-Kanade algorithm is resumed as below.

Initialize $\mathbf{s}$ and repeat until $\|\Delta\mathbf{s}\|\leq\epsilon$:

1. Warp I with $\mathbf{W}(\mathbf{x};\mathbf{s}$ to compute $I(\mathbf{W}(\mathbf{x};\mathbf{s}))$
2. Compute the error image $T(\mathbf{x})-I(\mathbf{W}(\mathbf{x};\mathbf{s}))$
3. Warp the gradient ∇I with $\mathbf{W}(\mathbf{x};\mathbf{s})$ to obtain $\nabla I(\mathbf{W}(\mathbf{x};\mathbf{s}))$
4. Evaluate the Jacobian $\frac{\partial\mathbf{W}}{\partial\mathbf{s}}$ at $(\mathbf{x};\mathbf{s})$
5. Compute $\nabla I(\mathbf{W}(\mathbf{x};\mathbf{s}))^\top\frac{\partial\mathbf{W}}{\partial\mathbf{s}}(\mathbf{x};\mathbf{s})$
6. Compute the Hessian matrix using (77)
7. Computing the steepest descent direction $\Delta\mathbf{s}$ using (76)
8. Update $\mathbf{s}\longleftarrow\mathbf{s}+\Delta\mathbf{s}$.

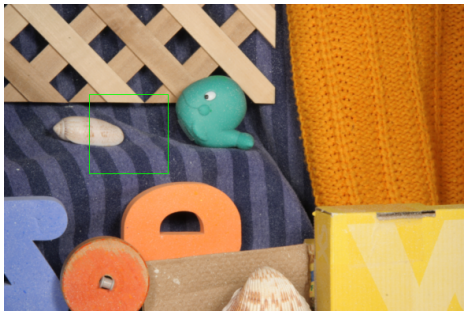
Note that we are particularly interested in affine transformation. Suppose that the transformation matrix is $\begin{bmatrix}s_1 & s_2 & s_3\\ s_4 & s_5 & s_6\end{bmatrix}$, then we have $\mathbf{s}=(s_1,s_2,s_3,s_4,s_5,s_6)$. The warping is

$$\mathbf{W}(\mathbf{x};\mathbf{s})=\begin{bmatrix}s_1 & s_2 & s_3\\ s_4 & s_5 & s_6\end{bmatrix}\begin{bmatrix}x\\ y\\ 1\end{bmatrix}=\begin{bmatrix}s_1x+s_2y+s_3\\ s_4x+s_5y+s_6\end{bmatrix}$$

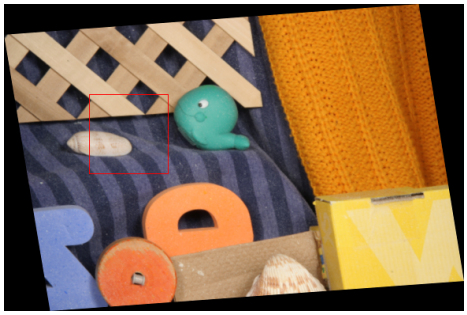
and thus the Jacobian of the warp is given by

$$\frac{\partial\mathbf{W}}{\partial\mathbf{s}}(\mathbf{x};\mathbf{s})=\begin{bmatrix}x & y & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & x & y & 1\end{bmatrix}.$$

Figure 11 where we transform an image with an affine transformation and extract a small template, then we use the Lucas-Kanade algorithm to transform the template back to the image.

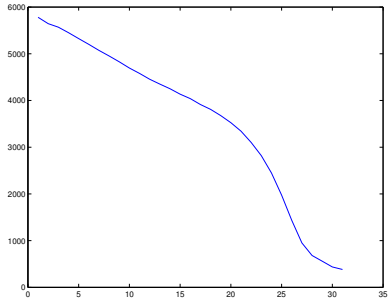


(a) Testing image I with an initialization for Lucas-Kanade agorithm.

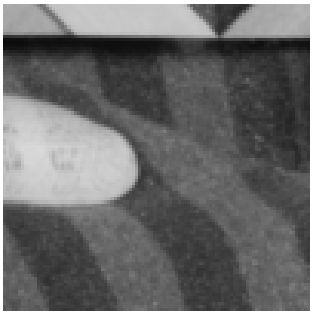


(b) A template T is extracted from a transformed image from I.

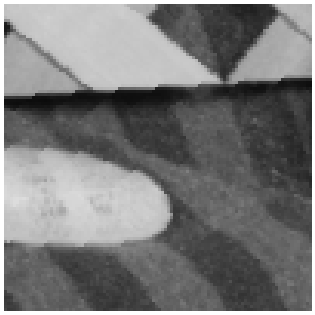
(c) Obtained final transformation.



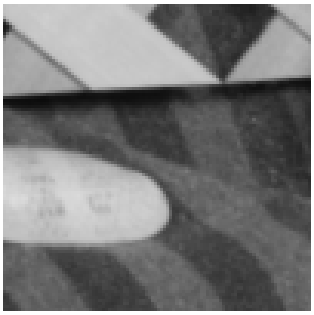
(d) The objective function $f(\mathbf{s})$ per iteration.



(e) Initial warped image.



(f) Final warped image.



(g) Extracted template.

Figure 11: A test for Lucas-Kanade algorithm.

7 Conclusion and Future Work

In this work, we have proposed a new decomposition scheme to solve the MRF optimization problem. Unlike the previous dual decomposition scheme, we relax the dependencies between any two nodes of the graph using Lagrangian relaxation. The relaxed dual problem is next smoothed using Nesterov's method and then optimized using optimal first-order gradient methods. The algorithm is guaranteed to converge to the global optimum of the relaxed linear program, with a convergence rate of $O(1/\epsilon)$. Moreover, the method can handle any graph structures with arbitrary potential functions. As an application, we have proposed a new MRF model for locally affine image registration, which is still an ongoing work.

In future work, we plan to:

- Try different smoothing schemes with different optimal first-order methods;
- Produce more experimental results, on different kinds of problems, such as image segmentation, 3D reconstruction, etc...;
- Evaluate the performance of the method, compared to the state-of-the-art;
- Complete and evaluate the proposed MRF model for locally affine image registration.

References

Arsigny, V., Commowick, O., Pennec, X., and Ayache, N. (2006). A log-euclidean polyaffine framework for locally rigid or affine registration. In Pluim, J., Likar, B., and Gerritsen, F., editors, *Biomedical Image Registration*, volume 4057 of *Lecture Notes in Computer Science*, pages 120–127. Springer Berlin Heidelberg.

Arsigny, V., Pennec, X., and Ayache, N. (2003). Polyrigid and polyaffine transformations: A new class of diffeomorphisms for locally rigid or affine registration. In Ellis, R. and Peters, T., editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003*, volume 2879 of *Lecture Notes in Computer Science*, pages 829–837. Springer Berlin Heidelberg.

Baker, S. and Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255.

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31.

Beck, A. and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202.

Beck, A. and Teboulle, M. (2012). Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580.

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific, Belmont, MA.

Boykov, Y. and Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112 vol.1.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239.

Bruhn, A., Weickert, J., and Schnrr, C. (2005). Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231.

Chambolle, A. (2005). Total variation minimization and a class of binary mrf models. In Rangarajan, A., Vemuri, B., and Yuille, A., editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 3757 of *Lecture Notes in Computer Science*, pages 136–152. Springer Berlin Heidelberg.

Commowick, O., Arsigny, V., Isambert, A., Costa, J., Dhermain, F., Bidault, F., Bondiau, P.-Y., Ayache, N., and Malandain, G. (2008). An efficient locally affine framework for the smooth registration of anatomical structures. *Medical Image Analysis*, 12(4):427 – 441.

Drakakis, K. and Pearlmutter, B. A. (2009). On the calculation of the $l_2 \to l_1$ induced matrix norm. *International Journal of Algebra*, 3(5-8):231–240.

Felzenszwalb, P. and Huttenlocher, D. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79.

Frey, B. J. and MacKay, D. J. C. (1997). A revolution: Belief propagation in graphs with cycles. In *In Neural Information Processing Systems*, pages 479–485. MIT Press.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741.

Glocker, B., Paragios, N., Komodakis, N., Tziritas, G., and Navab, N. (2008). Optical flow estimation with uncertainties through dynamic mrfs. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

Higham, N. (1992). Estimating the matrixp-norm. *Numerische Mathematik*, 62(1):539–555.

Hirschmuller, H. and Scharstein, D. (2009). Evaluation of stereo matching costs on images with radiometric differences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1582–1599.

Jojic, V., Gould, S., and Koller, D. (2010). Accelerated dual decomposition for map inference. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 503–510.

Kappes, J., Savchynskyy, B., and Schnorr, C. (2012). A bundle approach to efficient map-inference by lagrangian relaxation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1688–1695.

Kappes, J., Speth, M., Andres, B., Reinelt, G., and Schn, C. (2011). Globally optimal image partitioning by multicuts. In Boykov, Y., Kahl, F., Lempitsky, V., and Schmidt, F., editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 6819 of *Lecture Notes in Computer Science*, pages 31–44. Springer Berlin Heidelberg.

Kappes, J. H., Andres, B., Hamprecht, F. A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B. X., Lellmann, J., Komodakis, N., and Rother, C. (2013). A comparative study of modern inference techniques for discrete energy minimization problems. In *CVPR*. Oral.

Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

Kolmogorov, V. (2006). Convergent tree-reweighted message passing for energy minimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1568–1583.

Kolmogorov, V. and Rother, C. (2007). Minimizing nonsubmodular functions with graph cuts-a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(7):1274–1279.

Kolmogorov, V. and Zabih, R. (2002). Multi-camera scene reconstruction via graph cuts. In Heyden, A., Sparr, G., Nielsen, M., and Johansen, P., editors, *Computer Vision ECCV 2002*, volume 2352 of *Lecture Notes in Computer Science*, pages 82–96. Springer Berlin Heidelberg.

Komodakis, N. and Paragios, N. (2009). Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2985–2992.

Komodakis, N., Paragios, N., and Tziritas, G. (2011). Mrf energy minimization and beyond via dual decomposition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):531–552.

Komodakis, N., Tziritas, G., and Paragios, N. (2008). Performance vs computational efficiency for optimizing single and dynamic mrfs: Setting the state of the art with primal-dual strategies. *Comput. Vis. Image Underst.*, 112(1):14–29.

Korman, S., Reichman, D., Tsur, G., and Avidan, S. (2013). Fast-match: Fast affine template matching. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2331–2338.

Kumar, M., Torr, P. H. S., and Zisserman, A. (2006). Solving markov random fields using second order cone programming relaxations. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1045–1052.

Kumar, M. P., Kolmogorov, V., and Torr, P. H. S. (2009). An analysis of convex relaxations for map estimation of discrete mrfs. *J. Mach. Learn. Res.*, 10:71–106.

Lee, K. J., Yun, I. D., and Lee, S. U. (2013). Adaptive large window correlation for optical flow estimation with discrete optimization. *Image and Vision Computing*, 31(9):631 – 639.

Lempitsky, V., Roth, S., and Rother, C. (2008). Fusionflow: Discrete-continuous optimization for optical flow estimation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

Lempitsky, V., Rother, C., Roth, S., and Blake, A. (2010). Fusion moves for markov random field optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1392–1405.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Martins, A. F. T., Figueiredo, M. A. T., Aguiar, P. M. Q., Smith, N. A., and Xing, E. P. (2011). An augmented lagrangian approach to constrained MAP inference. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 169–176.

Nesterov, Y. (1983). A method of solving a convex programming problem with convergence rate o (1/k2). *Soviet Mathematics Doklady*, 27(2):372–376.

Nesterov, Y. (1988). On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonomika i Mateaticheskie Metody*, 24:509–517.

Nesterov, Y. (2004). *Introductory lectures on convex optimization: A basic course*, volume 87. Springer.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152.

Nesterov, Y. et al. (2007). Gradient methods for minimizing composite objective function.

Otten, L. and Dechter, R. (2014). Anytime and/or depth-first search for combinatorial optimization. In OSullivan, B., editor, *Principles and Practice of Constraint Programming*, volume 8656 of *Lecture Notes in Computer Science*, pages 933–937. Springer International Publishing.

Paragios, N., Duncan, J., and Ayache, N. (2014). *Handbook of Biomedical Imaging: Methodologies and Clinical Research*. Springer.

Pearl, J. (1982). Reverend bayes on inference engines: a distributed hierarchical approach. In *in Proceedings of the National Conference on Artificial Intelligence*, pages 133–136.

Pitiot, A., Malandain, G., Bardinet, E., and Thompson, P. (2003). Piecewise affine registration of biological images. In Gee, J., Maintz, J., and Vannier, M., editors, *Biomedical Image Registration*, volume 2717 of *Lecture Notes in Computer Science*, pages 91–101. Springer Berlin Heidelberg.

Ravikumar, P. and Lafferty, J. (2006). Quadratic programming relaxations for metric labeling and markov random field map estimation. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 737–744, New York, NY, USA. ACM.

Roth, S. and Black, M. (2007). On the spatial statistics of optical flow. *International Journal of Computer Vision*, 74(1):33–50.

Rother, C., Kolmogorov, V., and Blake, A. (2004). Grabcut -interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (SIGGRAPH)*.

Rueckert, D., Sonoda, L. I., Hayes, C., Hill, D. L. G., Leach, M. O., and Hawkes, D. J. (1999). Nonrigid registration using free-form deformations: Application to breast mr images. *IEEE Transactions on Medical Imaging*, 18:712–721.

Savchynskyy, B., Schmidt, S., Kappes, J., and Schnorr, C. (2011). A study of nesterov's scheme for lagrangian decomposition and map labeling. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1817–1823. IEEE.

Sederberg, T. W. and Parry, S. R. (1986). Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4):151–160.

Shimony, S. E. (1994). Finding {MAPs} for belief networks is np-hard. *Artificial Intelligence*, 68(2):399 – 410.

Sotiras, A., Davatzikos, C., and Paragios, N. (2013). Deformable medical image registration: A survey. *Medical Imaging, IEEE Transactions on*, 32(7):1153–1190.

Sun, D., Roth, S., and Black, M. (2014). A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106(2):115–137.

Tseng, P. (2008). On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*.

Vogiatzis, G., Hernandez, C., Torr, P. H. S., and Cipolla, R. (2007). Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2241–2246.

Wainwright, M., Jaakkola, T., and Willsky, A. (2005). Map estimation via agreement on trees: message-passing and linear programming. *Information Theory, IEEE Transactions on*, 51(11):3697–3717.

Wang, C., Komodakis, N., and Paragios, N. (2013). Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117(11):1610 – 1627.

Zhang, L. and Seitz, S. (2007). Estimating optimal parameters for mrf stereo from a single image pair. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(2):331–342.

Appendices

Euclidean projection on Λ

We give a proof for Lemma 1, stated at page 11.

Convention: if a vector $\mathbf{v}$ has an index p or pq, then it has the form

$$\mathbf{v} = \left\{ \{\mathbf{v}_p\}_{p\in\mathcal{V}}, \{\mathbf{v}_{pq}\}_{pq\in\mathcal{E}} \right\}.$$

Given $(\mathbf{a}, \mathbf{b})$, we need to find $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ in the set Λ such that $\|\boldsymbol{\lambda} - \mathbf{a}\|^2 + \|\boldsymbol{\nu} - \mathbf{b}\|^2$ is minimized. Clearly, it suffices to minimize $\|\boldsymbol{\lambda} - \mathbf{a}\|^2$ and $\|\boldsymbol{\nu} - \mathbf{b}\|^2$ separately, since there is no coupling constraint between $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$.

Note that

$$\begin{aligned}
\|\boldsymbol{\lambda} - \mathbf{a}\|^2 &= \sum_{p\in\mathcal{V}} \left(\|\boldsymbol{\lambda}_p - \mathbf{a}_p\|^2 + \sum_{q\in\mathrm{Ch}(p)} \|\boldsymbol{\lambda}_{pq} - \mathbf{a}_{pq}\|^2 \right) \\
&= \sum_{p\in\mathcal{V}} \left(\left\| - \sum_{q\in\mathrm{Ch}(p)} \boldsymbol{\lambda}_{pq} - \mathbf{a}_p \right\|^2 + \sum_{q\in\mathrm{Ch}(p)} \|\boldsymbol{\lambda}_{pq} - \mathbf{a}_{pq}\|^2 \right) \\
&= \sum_{p\in\mathcal{V}} F_p,
\end{aligned}$$

where

$$F_p = \left\| \sum_{q\in\mathrm{Ch}(p)} \boldsymbol{\lambda}_{pq} + \mathbf{a}_p \right\|^2 + \sum_{q\in\mathrm{Ch}(p)} \|\boldsymbol{\lambda}_{pq} - \mathbf{a}_{pq}\|^2 .$$

Clearly, it suffices to minimize each F_p independently.
Using Lemma 3 below, we see that F_p attains its minimum if and only if

$$\boldsymbol{\lambda}_{pq} = \mathbf{a}_{pq} - \frac{1}{|\mathrm{Ch}(p)| + 1} \left(\mathbf{a}_p + \sum_{q\in\mathrm{Ch}(p)} \mathbf{a}_{pq} \right).$$

Similarly, we can deduce that $\|\boldsymbol{\nu} - \mathbf{b}\|^2$ is minimized if and only if

$$\boldsymbol{\nu}_{pq} = \mathbf{b}_{pq} - \frac{1}{|\mathrm{Pa}(q)| + 1} \left(\mathbf{b}_q + \sum_{p\in\mathrm{Pa}(q)} \mathbf{b}_{pq} \right) \quad \forall pq \in \mathcal{E}.$$

Finally, since $(\boldsymbol{\lambda}, \boldsymbol{\nu}) \in \Lambda$, the terms $\boldsymbol{\lambda}_p$ and $\boldsymbol{\nu}_q$ can be computed using

$$\begin{aligned}
\boldsymbol{\lambda}_p &= - \left(\sum_{q\in\mathrm{Ch}(p)} \boldsymbol{\lambda}_{pq} \right), \quad \forall p \in \mathcal{V}, \\
\boldsymbol{\nu}_q &= - \left(\sum_{p\in\mathrm{Pa}(q)} \boldsymbol{\nu}_{pq} \right), \quad \forall q \in \mathcal{V}.
\end{aligned}$$

The lemma is proved.

Lemma 3 *Given $n+1$ vectors $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_n$ in $\mathbb{R}^d$. Then the minimum value of*

$$F = \left\| \sum_{i=1}^{n} \mathbf{u}_i + \mathbf{a}_0 \right\|^2 + \sum_{i=1}^{n} \|\mathbf{u}_i - \mathbf{a}_i\|^2$$

over $\mathbf{u}_1, \ldots, \mathbf{u}_n \in \mathbb{R}^d$ is attained if and only if $\mathbf{u}_i = \mathbf{a}_i - \frac{1}{n+1}\mathbf{s}, i = 1, \ldots, n$, where

$$\mathbf{s} = \sum_{i=0}^{n} \mathbf{a}_i.$$

PROOF Denote $\mathbf{y}_i = \mathbf{u}_i - \mathbf{a}_i, i = 1, 2, \ldots, n$, we have

$$F = \left\| \sum_{i=1}^{n} \mathbf{y}_i + \mathbf{s} \right\|^2 + \sum_{i=1}^{n} \|\mathbf{y}_i\|^2$$

Applying the Lemma 4 below for $n+1$ vectors we get

$$\begin{aligned} F &= \left\| -\sum_{i=1}^{n} \mathbf{y}_i - \mathbf{s} \right\|^2 + \sum_{i=1}^{n} \|\mathbf{y}_i\|^2 \\ &\geq \frac{1}{n+1} \left\| -\sum_{i=1}^{n} \mathbf{y}_i - \mathbf{s} + \mathbf{y}_1 + \mathbf{y}_2 + \cdots + \mathbf{y}_n \right\|^2 \\ &= \frac{1}{n+1} \|\mathbf{s}\|^2 . \end{aligned}$$

Equality holds if and only if $-\sum_{i=1}^{n} \mathbf{y}_i - \mathbf{s} = \mathbf{y}_1 = \mathbf{y}_2 = \cdots = \mathbf{y}_n$, which yields $\mathbf{y}_i = \frac{-1}{n+1}\mathbf{s}$ or $\mathbf{u}_i = \mathbf{a}_i - \frac{1}{n+1}\mathbf{s}$ for $i = 1, \ldots, n$. □

Lemma 4 *For any n vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n \in \mathbb{R}^d$, the following inequality holds*

$$\|\mathbf{u}_1\|^2 + \|\mathbf{u}_2\|^2 + \cdots + \|\mathbf{u}_n\|^2 \geq \frac{1}{n} \|\mathbf{u}_1 + \mathbf{u}_2 + \cdots + \mathbf{u}_n\|^2 \tag{78}$$

and equality occurs if and only if $\mathbf{u}_1 = \mathbf{u}_2 = \cdots = \mathbf{u}_n$.

PROOF Denote $f(\mathbf{u}) = \|\mathbf{u}\|^2$, then the above inequality becomes

$$f(\mathbf{u}_1) + f(\mathbf{u}_2) + \cdots + f(\mathbf{u}_n) \geq n f\left(\frac{\mathbf{u}_1 + \mathbf{u}_2 + \cdots + \mathbf{u}_n}{n} \right), \tag{79}$$

which is clearly true by Jensen's inequality since f is convex. Also by Jensen's inequality, equality holds if and only if $\mathbf{u}_1 = \mathbf{u}_2 = \cdots = \mathbf{u}_n$. □

Prox function of the set Q

The proof of Lemma 2 (page 17). Indeed, it is straightforward by Lemma 5 and Lemma 6 below.

Lemma 5 *The function*

$$d(\mathbf{w}) = \log n + \sum_{i=1}^{n} w_i \log w_i$$

is a prox function, with respect to the ℓ_1 norm, of the set

$$\left\{ \mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{R}^n \mid \mathbf{w} \geq 0, \mathbf{1}^\top \mathbf{w} = 1 \right\},$$

with convexity parameter 1.

PROOF See [Nesterov, 2005]. □

Lemma 6 *Let $d_i(\cdot)$, $i = 1, 2, \ldots, n$, be n continuous and strongly convex functions with respect to the L^1 norm on $\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_n$, respectively, and with convexity parameters $\sigma_1, \sigma_2, \ldots, \sigma_n$, respectively. Define the function*

$$d(\mathbf{w}) = d_1(\mathbf{w}_1) + d_2(\mathbf{w}_2) + \cdots + d_n(\mathbf{w}_n)$$

where $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n) \in \mathcal{W} = \mathcal{W}_1 \times \mathcal{W}_2 \times \cdots \times \mathcal{W}_n$. Then $d(\cdot)$ is continuous and strongly convex, with respect to the ℓ_1 norm, on $\mathcal{W}$, with convexity parameter

$$\sigma = \frac{1}{\frac{1}{\sigma_1} + \frac{1}{\sigma_2} + \cdots + \frac{1}{\sigma_n}}.$$

If we consider the ℓ_2 norm instead of ℓ_1, then $\sigma = \min_i(\sigma_i)$.

PROOF Straightforward by definition:

$$d(\mathbf{u}) \geq d(\mathbf{v}) + \nabla d(\mathbf{v})^\top (\mathbf{u} - \mathbf{v}) + \frac{\sigma}{2} \|\mathbf{u} - \mathbf{v}\|_1^2$$

and Cauchy-Schwarz inequality. □

Smooth approximation of f

We want to compute f_μ given by (55):

$$f_\mu(\mathbf{x}) = \max_{\mathbf{u} \in \mathcal{U}} \{-L(\mathbf{u}, \mathbf{x}) - \mu d_2(\mathbf{u})\} = -L(\mathbf{u}_\mu(\mathbf{x}), \mathbf{x}) - \mu d_2(\mathbf{u}_\mu(\mathbf{x})).$$

where

$$\mathbf{u}_\mu(\mathbf{x}) = \arg\min_{\mathbf{u} \in \mathcal{U}} \{L(\mathbf{u}, \mathbf{x}) \mu d_2(\mathbf{u})\}.$$

We will prove the results given by (58), (59) and (60):

$$u_p^i = \frac{\exp(a_p^i)}{\sum_{j=1}^d \exp(a_p^j)} \quad i = 1, \ldots, d$$

$$u_{pq}^i = \frac{\exp(a_{pq}^i)}{\sum_{j=1}^{d^2} \exp(a_{pq}^j)} \quad i = 1, \ldots, d^2$$

$$f_\mu(\mathbf{x}) = \mu \sum_{p \in \mathcal{V}} \log \left(\sum_{i=1}^{d} \exp(a_p^i) \right) + \mu \sum_{pq \in \mathcal{E}} \log \left(\sum_{i=1}^{d^2} \exp(a_{pq}^i) \right) - \mu D$$

where

$$\mathbf{a}_p = -\frac{1}{\mu}(\boldsymbol{\theta}_p + \boldsymbol{\lambda}_p + \boldsymbol{\nu}_p)$$

$$\mathbf{a}_{pq} = -\frac{1}{\mu}(\boldsymbol{\theta}_{pq} + \mathbf{D}^\top \boldsymbol{\lambda}_{pq} + \mathbf{C}^\top \boldsymbol{\nu}_{pq})$$

$$D = (|\mathcal{V}| + 2\,|\mathcal{E}|) \log d.$$

Indeed, from the expression (29) of $L(\mathbf{u}, \mathbf{x})$ and the expression (56) of $d(\mathbf{u})$, we have

$$\begin{aligned} L(\mathbf{u}, \mathbf{x}) + \mu d(\mathbf{u}) = & \sum_{p \in \mathcal{V}} \left\{ (\boldsymbol{\theta}_p + \boldsymbol{\lambda}_p + \boldsymbol{\nu}_p)^\top \mathbf{u}_p + \mu \sum_{i=1}^{d} u_p^i \log u_p^i \right\} \\ & + \sum_{pq \in \mathcal{E}} \left\{ (\boldsymbol{\theta}_{pq} + \mathbf{D}^\top \boldsymbol{\lambda}_{pq} + \mathbf{C}^\top \boldsymbol{\nu}_{pq})^\top \mathbf{u}_{pq} + \mu \sum_{i=1}^{d^2} u_{pq}^i \log u_{pq}^i \right\} \\ & + \mu(|V| + 2\,|E|) \log d. \end{aligned}$$

Using Lemma 7 below, the results are straightforward.

Lemma 7 *The problem*

$$\begin{aligned} & \textit{minimize} && h(\mathbf{w}) := -\mathbf{a}^\top \mathbf{w} + \sum_{i=1}^{n} w_i \log w_i \\ & \textit{subject to} && \mathbf{w} \geq \mathbf{0}, \quad \mathbf{1}^\top \mathbf{w} = 1 \end{aligned}$$

has the solution

$$w_i^* = \frac{\exp(a_i)}{\sum_{j=1}^{n} \exp(a_j)} \quad i = 1, \ldots, n,$$

and the optimal value is

$$h(\mathbf{w}^*) = -\log \left(\sum_{j=1}^{n} \exp(a_j) \right).$$

Proof Solve the KKT systems. □