Continuous Inference in Graphical Models with Polynomial Energies

Mathieu Salzmann
NICTA*
Canberra, Australia
mathieu.salzmann@nicta.com.au

Abstract

In this paper, we tackle the problem of performing inference in graphical models whose energy is a polynomial function of continuous variables. Our energy minimization method follows a dual decomposition approach, where the global problem is split into subproblems defined over the graph cliques. The optimal solution to these subproblems is obtained by making use of a polynomial system solver. Our algorithm inherits the convergence guarantees of dual decomposition. To speed up optimization, we also introduce a variant of this algorithm based on the augmented Lagrangian method. Our experiments illustrate the diversity of computer vision problems that can be expressed with polynomial energies, and demonstrate the benefits of our approach over existing continuous inference methods.

1. Introduction

Many computer vision problems can be expressed as the minimization of an energy that is a polynomial function of the variables that define the problem. This, for instance, is the case of non-rigid shape recovery [27], or shape-from-shading [5]. With the availability of techniques such as homotopy continuation [19, 11], or Groebner basis [4, 21, 10] to solve systems of polynomial equations, one could think of solving such minimization problems by finding the zeros of the gradient of the energy. However, existing polynomial system solvers can only cope with small numbers of variables, which precludes following this approach for most realistic computer vision problems.

A popular approach to tackling large computer vision problems consists in exploiting the structure of the variables of interest, and, in particular, the fact that these variables form a graph with cliques of reasonably small sizes (i.e., of size two in many cases). A solution to the task at hand is then obtained by performing inference in the corresponding graphical model. For discrete variables, many inference methods have been proposed (e.g., [24, 14, 8]). More recently, several inference techniques for continuous variables have been introduced. However, these methods either are restricted to modeling very specific energies (e.g., Gaussian belief propagation [2]), or do not offer convergence guarantees for general graphs (e.g., nonparametric belief propagation [22], particle belief propagation [7, 15], fusion moves [12]).

In this paper, we introduce an approach to performing inference in graphical models with continuous variables and whose energies are polynomial functions. By exploiting the structure of the graph, our method lets us cope with much larger numbers of variables than polynomial system solvers. Furthermore, it comes with convergence guarantees for general graphs, while allowing us to handle more general energies than Gaussian belief propagation.

More specifically, we follow a dual decomposition approach [1, 9] and split the graph encoding the global problem into subsets of its variables. Since the global energy is a polynomial function, so are the energies of the individual subgraphs. However, these energies now depend on much smaller numbers of variables. Therefore, we can exploit polynomial system solvers to find the minimum energy configuration of each subgraph by computing the zeros of its energy gradient. The standard iterative dual decomposition procedure can be employed to encourage the solutions of the subgraphs to agree on their shared variables. This just introduces an additional linear term in the original subgraph energies, which thus remain polynomial. Since the optimal solution for each subgraph can be obtained, our method inherits the convergence guarantees of dual decomposition, even though the global energy is non-convex. Furthermore, it can easily be parallelized and can inherently handle graphs with cycles and with cliques of size larger than two. Finally, we introduce a variant of our algorithm that exploits the augmented Lagrangian method to speed up the optimization process.

To illustrate the diversity of scenarios that our approach can handle, we applied it to four problems: Non-rigid image registration, deformable surface 3D reconstruction, shape-

*NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the ARC through the ICT Centre of Excellence program.

from-shading and optical flow estimation. Our experiments demonstrate the effectiveness of our approach, as well as its benefits over particle belief propagation and fusion moves.

2. Related Work

Polynomial functions arise in many computer vision tasks. For instance, minimal problems [21, 10] are commonly formulated in terms of polynomial equations, and addressed with Groebner basis solvers [4]. Similarly, shape-from-shading has been expressed as the solution to a polynomial system [5] obtained by homotopy continuation [19, 11]. Nonetheless, polynomial solvers can only handle small numbers of variables. As a consequence, approximate formulations have been designed [5].

Alternatively, inference methods in graphical models have been widely used to minimize energy functions of many variables by accounting for their graph structure. While most of the literature assumes discrete variables (e.g., [24, 14, 8]), several methods have been proposed to address the continuous case. For instance, Gaussian belief propagation [2] minimizes quadratic energies, which form a special case of the ones handled by our approach. Instead of quadratic energies, piecewise convex functions were recently utilized [30]. Our approach can cope with non-convex energies without requiring piecewise approximations. Kernel belief propagation [20] can make use of general energies. However, it cannot exploit known potentials, but instead requires training data to learn the potentials. Nonparametric belief propagation [22] can also exploit general energies, but approximates the messages passed between the graph nodes as mixtures of Gaussians, thus modifying the original problem.

Currently, to the best of our knowledge, the most successful approaches to performing continuous inference in graphical models are particle (convex) belief propagation (PCBP) [7, 15] and Fusion Moves [12]. The former iteratively samples particles to form candidate variable assignments and finds a MAP estimate of the resulting discrete problem with (convex) belief propagation. The latter iteratively generates two candidate solutions and combines them by solving a binary labeling problem. PCBP has recently been applied to deformable surface 3D reconstruction [18] and stereo matching [29]. Fusion Moves have been employed for optical flow estimation [12] and non-rigid shape recovery [27]. While both approaches have shown good performance, they lack convergence guarantees for general graphs. Fusion Moves guarantee that the combined solution is at least as good as the two candidate ones. However, generating good candidates remains an open problem.

In this paper, we tackle the problem of performing continuous inference in graphical models where the energy function is polynomial. Since our method is based on dual decomposition [1, 9], it inherits its convergence guarantees, while still allowing us to model a large class of energies. Note that other decomposition methods (e.g., [28, 17]) have also been proposed to minimize different convex energies, or non-convex ones by convex approximations [6].

3. Our Approach

In this section, we present our approach to performing inference in graphical models with continuous variables and polynomial energies. We first introduce our main algorithm and then propose a variant to speed up minimization. Finally, we discuss some properties of our method.

3.1. Dual Decomposition with Polynomial Energies

Let $\mathbf{x} \in \mathbb{R}^D$ be the vector containing the D continuous random variables of the problem of interest. We consider the case where these random variables form a graph. Thus the energy of a specific realization of these variables can be expressed as

$$E(\mathbf{x}) = \sum\nolimits_i f_i(\mathbf{x}_{\alpha_i}) \ , \tag{1}$$

where α_i represents either a group of variables, or a single variable. The functions f_i may encode any local energy, such as arising from image observations, or regularizers. However, we assume that they are polynomial in $\mathbf{x}$.

Our goal is to find the configuration $\mathbf{x}^*$ that corresponds to the minimum energy $E(\mathbf{x}^*)$. To achieve this while accounting for the graph structure, we make use of dual decomposition [1, 9]. Dual decomposition proceeds by introducing auxiliary variables $\{\mathbf{x}^i_{\alpha_i}\}$ and formulating the minimization of $E(\mathbf{x})$ as

$$\begin{aligned} \min_{\{\mathbf{x}^i_{\alpha_i}\},\mathbf{x}} \ & \sum\nolimits_i f_i(\mathbf{x}^i_{\alpha_i}) \\ \text{s. t.} \quad & \mathbf{x}^i_{\alpha_i} = \mathbf{x}_{\alpha_i} \ , \ \forall i \ . \end{aligned} \tag{2}$$

The Lagrange dual function of this problem can be written as

$$g(\{\boldsymbol{\lambda}^i\}) = \min_{\{\mathbf{x}^i_{\alpha_i}\},\mathbf{x}} \sum_i \left(f_i(\mathbf{x}^i_{\alpha_i}) + \left(\mathbf{x}^i_{\alpha_i} - \mathbf{x}_{\alpha_i}\right)^T \boldsymbol{\lambda}^i \right) , \tag{3}$$

where $\boldsymbol{\lambda}^i$ is the vector of Lagrange multipliers for the constraints on the variables involved in f_i. The original variables $\mathbf{x}$ can be eliminated from $g(\{\boldsymbol{\lambda}^i\})$ by explicitly performing the minimization with respect to $\mathbf{x}$. This yields the constraints

$$\{\boldsymbol{\lambda}^i\} \in \Lambda = \left\{ \{\boldsymbol{\lambda}^i\} \Big| \sum_{i|j\in\alpha_i} \boldsymbol{\lambda}^i_j = 0 \ , \ \forall j \in [1, D] \right\} \ ,$$

where $\boldsymbol{\lambda}^i_j$ is the multiplier in $\boldsymbol{\lambda}^i$ that corresponds to the constraint on $\mathbf{x}_j$. This lets us write the Lagrange dual problem

$$\max_{\{\boldsymbol{\lambda}^i\}\in\Lambda} \sum\nolimits_i g^i(\boldsymbol{\lambda}^i) \ , \tag{4}$$

whose objective function has now decoupled into subproblems, or slaves, of the form

$$g^i(\boldsymbol{\lambda}^i) = \min_{\mathbf{x}^i_{\alpha_i}} f_i(\mathbf{x}^i_{\alpha_i}) + \left(\mathbf{x}^i_{\alpha_i}\right)^T \boldsymbol{\lambda}^i \ . \tag{5}$$

Following standard practice, we use a projected subgradient method to solve the problem in Eq. 4. It can be shown [9] that the subgradient $\nabla g^i(\boldsymbol{\lambda}^i) = \bar{\mathbf{x}}^i_{\alpha_i}(\boldsymbol{\lambda}^i)$, where $\bar{\mathbf{x}}^i_{\alpha_i}(\boldsymbol{\lambda}^i)$ is the optimal solution to the i^{th} slave. This yields an iterative algorithm where, at each iteration t, the optimal solution to each slave is computed given the current values of $\{\boldsymbol{\lambda}^i\}$, and each $\boldsymbol{\lambda}^i$ is updated as $\boldsymbol{\lambda}^i = \left[\boldsymbol{\lambda}^i + \eta_t \bar{\mathbf{x}}^i_{\alpha_i}(\boldsymbol{\lambda}^i)\right]_\Lambda$, where η_t is the step size of the subgradient ascent, and $[\cdot]_\Lambda$ denotes the projection onto the set Λ. This projection can be achieved by subtracting from each variable in $\bar{\mathbf{x}}^i_{\alpha_i}(\boldsymbol{\lambda}^i)$ its mean value over all slaves that contain this variable. Similarly, the primal solution can be obtained by averaging over the variables shared across multiple slaves. Under some conditions discussed below, dual decomposition with subgradient ascent is guaranteed to converge to the global optimum of Eq. 4. Note, however, that this translates into global convergence of the original problem only when the variables shared by the subproblems agree and when the duality gap is zero. We observed that, in practice, this occurred very often.

Convergence of dual decomposition is conditioned on the ability to compute the optimal solution to each slave in Eq. 5. To this end, we search for a solution such that the gradient of the energy vanishes. For each slave, this yields the system

$$\left\{ \begin{array}{rcl} \partial f_i / \partial \mathbf{x}^i_{\alpha_i,1} + \boldsymbol{\lambda}^i_1 & = & 0 \\ & \vdots & \\ \partial f_i / \partial \mathbf{x}^i_{\alpha_i,|\alpha_i|} + \boldsymbol{\lambda}^i_{|\alpha_i|} & = & 0 \end{array} \right. , \tag{6}$$

where $\mathbf{x}^i_{\alpha_i,j}$ is the j^{th} variable of slave i, and $|\alpha_i|$ denotes the number of variables in this slave.

Since each function f_i is polynomial, so is its gradient, and thus Eq. 6 is a polynomial system. As long as each slave depends on a reasonable number of variables, this system can be solved using standard techniques. In particular, we make use of homotopy continuation methods [19, 11]. Homotopy continuation methods proceed by first replacing the original polynomial system $\mathcal{P}(\mathbf{x}) = 0$ with a simpler one $\mathcal{P}_0(\mathbf{x}) = 0$ that has the same degree and the same number of roots. $\mathcal{P}_0(\mathbf{x})$ is then gradually transformed into $\mathcal{P}(\mathbf{x})$ by homotopy, while the solutions are traced from those of $\mathcal{P}_0(\mathbf{x})$ to those of $\mathcal{P}(\mathbf{x})$ by a numerical continuation method. Although relying on numerical methods, homotopy continuation guarantees to find all the complex roots of a polynomial system with probability 1. More details can be found in [19]. In our experiments, we employed the solver HOM4PS-2.0 [11]. While HOM4PS-2.0 is the fastest solver available, it still requires 9 hours to solve a system of 20 quadratic equations in 20 variables. Therefore, for many computer vision problems, it would not be directly applicable to the original task.

In general, the system of Eq. 6 has multiple roots. Many of these roots can be discarded based on the fact that we are not interested in complex values. The variable assignment corresponding to the global minimum can then be obtained by comparing the energy value of the remaining roots.

Convergence of dual decomposition also depends on the properties of the sequence of subgradient step sizes η_t [1, 9]. In our experiments, we considered two such sequences: The adaptive rule introduced in [9], and a non-summable diminishing step length rule [1]. The adaptive rule can be expressed as

$$\eta_t = \frac{g^*_t + \delta_t - g_t}{\|\nabla g_t\|^2} \ , \text{ with } \delta_{t+1} = \left\{ \begin{array}{ll} \tau_0 \delta_t & \text{if } g_t \text{ improved by } \delta_t \\ \max(\tau_1 \delta_t, \delta) & \text{otherwise} \end{array} \right. , \tag{7}$$

where g_t and ∇g_t denote the dual function value and subgradient at iteration t, respectively, and g^*_t is the best dual value obtained so far. In our experiments, $\tau_0 = 1.5$, $\tau_1 = 0.75$, and $\delta = 0.01\delta_0$. The initial value δ_0 is problem dependent. As non-summable diminishing step length rule, we used

$$\eta_t = \frac{\eta_0}{(1 + t/t_0)\,\|\nabla g_t\|} \ , \tag{8}$$

where η_0 is problem dependent, and $t_0 = 50$ in practice.

3.2. Augmented Lagrangian Formulation

While dual decomposition with subgradient ascent guarantees convergence to the global optimum of Eq. 4, many iterations often are required to reach this optimum. To speed up the process, we follow an approach based on the Alternating Direction Method of Multipliers (ADMM), which exploits the augmented Lagrangian of the original problem [13]. This boils down to introducing a quadratic penalty in the Lagrange dual function, which yields

$$\tilde{g}(\{\boldsymbol{\lambda}^i\}) = \min_{\{\mathbf{x}^i_{\alpha_i}\},\mathbf{x}} \sum\nolimits_i f_i(\mathbf{x}^i_{\alpha_i}) + \left(\mathbf{x}^i_{\alpha_i}\right)^T \boldsymbol{\lambda}^i + \frac{\rho_t}{2} \left\|\mathbf{x}^i_{\alpha_i} - \mathbf{x}_{\alpha_i}\right\|^2 ,$$

where we implicitly accounted for the constraint $\{\boldsymbol{\lambda}^i\} \in \Lambda$, and where ρ_t is a time-varying weight. With this new term that involves the original variables $\mathbf{x}$, the objective of the corresponding dual problem does not decouple anymore. Therefore, minimization with respect to $\{\mathbf{x}^i_{\alpha_i}\}$ and $\mathbf{x}$ is performed in an alternating manner. Given a fixed $\mathbf{x}$, the dual problem decomposes into slaves of the form

$$\tilde{g}^i(\boldsymbol{\lambda}^i) = \min_{\mathbf{x}^i_{\alpha_i}} f_i(\mathbf{x}^i_{\alpha_i}) + \left(\mathbf{x}^i_{\alpha_i}\right)^T \boldsymbol{\lambda}^i + \frac{\rho_t}{2} \left\|\mathbf{x}^i_{\alpha_i} - \mathbf{x}_{\alpha_i}\right\|^2 \ .$$

Since the additional penalty term is quadratic, the slave energies remain polynomial. Their global minimum can be

Algorithm 1 DD-Poly / ADMM-Poly

Initialize $\boldsymbol{\lambda}_i = 0\ ,\ \forall i$
Initialize δ_0, or η_0
Initialize ρ_0 and ρ_1 (ADMM-Poly only)
for $t = 1$ to #iters **do**
  Compute ρ_t from Eq. 10 (ADMM-Poly only)
  for $i = 1$ to #slaves **do**
    Find the roots of the system in Eq. 6, or Eq. 9
    Find the lowest energy root $\bar{\mathbf{x}}^i_{\alpha_i}$ among the roots
  end for
  Compute η_t from Eq. 7, or Eq. 8
  $\boldsymbol{\lambda}^i = \left[\boldsymbol{\lambda}^i + \eta_t \bar{\mathbf{x}}^i_{\alpha_i}(\boldsymbol{\lambda}^i)\right]_\Lambda\ ,\ \forall i$
  Compute $\mathbf{x}$ by averaging the shared variables in $\{\bar{\mathbf{x}}^i_{\alpha_i}\}$
end for

obtained by solving polynomial equations of the form

$$\partial f_i / \partial \mathbf{x}^i_{\alpha_i,j} + \boldsymbol{\lambda}^i_j + \rho_t(\mathbf{x}^i_{\alpha_i,j} - \mathbf{x}_{\alpha_i,j}) = 0\ , \qquad (9)$$

where $\mathbf{x}_{\alpha_i,j}$ is fixed, and finding the solution with the lowest energy among the roots. With fixed $\{\mathbf{x}^i_{\alpha_i}\}$, the augmented Lagrangian has a closed-form solution for $\mathbf{x}$, which corresponds to the primal value of the original problem, and can thus be obtained by averaging over the variables shared across multiple slaves.

While, for some values of ρ_t and η_t, and for convex energies, ADMM has convergence guarantees, they do not apply to general polynomial energies. In practice, we observed good convergence of the algorithm, especially when the augmented Lagrangian penalty was activated only after several iterations of the algorithm. This inspired us the rule

$$\rho_t = \rho_0 + \frac{\rho_1 - \rho_0}{1 + \exp(-\gamma(t - t_0))}\ , \qquad (10)$$

with $\gamma = 1/4$, $t_0 = 50$ and $\rho_0 = 0$ (or a small value) in our experiments. Note that we found this strategy to be more effective than a warm start approach that would begin with a large ρ and progressively decrease it until it reaches 0. We also found that the ADMM approach was more stable with η_t following the rule in Eq. 8 rather than the rule in Eq. 7. The methods of Sections 3.1 (DD-Poly) and 3.2 (ADMM-Poly) are described in Algorithm 1.

3.3. Properties and Extension

We now discuss some properties of our approach, as well as its extension to non-polynomial functions.

Graphs with cycles: As opposed to most existing approaches, the convergence guarantees of DD-Poly remain unchanged by the presence of cycles in the graph.

Higher order cliques: The cliques in the graph may be of order higher than two without affecting the convergence guarantees of DD-Poly. The bottleneck comes from the speed of the polynomial system solver, which depends on the number of variables and on the degree of the polynomials. In practice, we used cliques of size 18, 5 and 4.

Parallel computation: Dual decomposition naturally supports parallel computation. The slave problems can be solved in parallel on individual cores. Only $\{\boldsymbol{\lambda}^i\}$ and $\mathbf{x}$ must be computed at a global level.

Non-polynomial energies: A straightforward extension of our method to non-polynomial energies can be achieved by approximating such functions with polynomials. The most common polynomial function approximation methods are the Taylor and Chebyshev expansions. Instead of such global approximations, better accuracy may be achieved by approximating the slave energies with piecewise polynomial functions. To illustrate this, let us consider the case of a single variable x split into 3 intervals. Following a first-order Taylor expansion, a non-polynomial function $f(x)$ can be approximated as

$$f(x) \approx \begin{cases} f(x_1) + \left.\frac{df}{dx}\right|_{x_1}(x - x_1) & \text{if } x < l_x \\ f(x_2) + \left.\frac{df}{dx}\right|_{x_2}(x - x_2) & \text{if } l_x \leq x \geq u_x \\ f(x_3) + \left.\frac{df}{dx}\right|_{x_3}(x - x_3) & \text{if } x > u_x \end{cases}\ ,$$

where x_1, x_2 and x_3 are three values at which the function was evaluated, and l_x and u_x define the intervals. The resulting three polynomials can be solved independently, and solutions outside the domain of each polynomial discarded. The final x can then be obtained by comparing the remaining solutions. In general, when each slave depends on multiple variables, the solution space must be divided into hypercubes inside which the polynomial approximations are computed. Note that the global optimum for a hypercube may not be a local optimum of the approximate energy, but a local optimum on a boundary of the hypercube, or the approximate energy value at a corner of the hypercube. Solutions on the boundaries can be computed by fixing a subset of the variables to the bound of their interval and finding the local optima of the resulting polynomial function. Solutions at corners are simply obtained by evaluating the approximate energy with all variables set to interval bounds.

4. Experimental Evaluation

In this section, we demonstrate the effectiveness and generality of our method on 4 computer vision problems that exhibit different properties of our algorithms.

4.1. Non-rigid Image Registration

Non-rigid image registration consists in finding the warping that transforms a reference image to an input image depicting, for instance, the same object undergoing deformations. Here, we consider the case where the observed

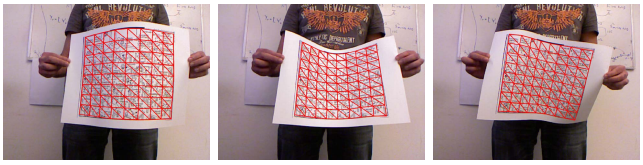

Figure 1. **Non-rigid registration.** Recovered 2D mesh overlaid on the original images. All images were treated independently.

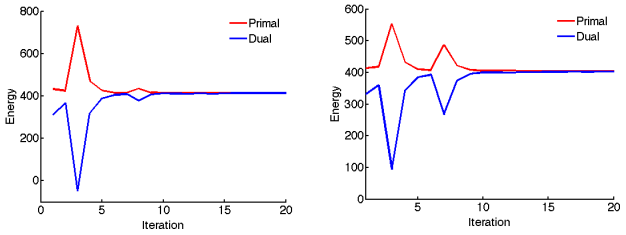

Figure 2. **Energy evolution during registration.** Primal and dual energies as a function of the iteration of DD-Poly. These plots correspond to the 1st and 2nd images in Fig. 1.

object is a non-rigid surface, which we model as a triangulated mesh, as suggested in [16]. Given the 2D locations of the mesh vertices in the reference image, the goal is to find these locations in the input image. To this end, similarly as [16, 25], we assume that we are given correspondences between keypoints in the reference and input images.

More specifically, let $\mathbf{x} \in \mathbb{R}^{2N}$ be the vector containing the 2D coordinates of the N mesh vertices. We encode a keypoint in terms of the barycentric coordinates with respect to the mesh facet inside which it lies, and assume that these barycentric coordinates remain constant. We then search for the vertex locations that best satisfy the correspondences, while yielding a smooth surface. Assuming that the mesh forms a regular grid, smoothness can be encouraged by penalizing the curvature of triplets of aligned vertices [16]. Registration can then be expressed as

$$\min_{\mathbf{x}} \sum_i \left\| c_1^i \mathbf{x}_{j_i}^1 + c_2^i \mathbf{x}_{j_i}^2 + c_3^i \mathbf{x}_{j_i}^3 - \begin{pmatrix} u_i \\ v_i \end{pmatrix} \right\|^2 + w_s \sum_{(i,j,k)\in\mathcal{T}} \|\mathbf{x}_i - 2\mathbf{x}_j + \mathbf{x}_k\|^2,$$

where c_k^i is the k^{th} barycentric coordinate of keypoint i, $\mathbf{x}_{j_i}^k$ is the k^{th} vertex of facet j_i, which contains keypoint i, (u_i, v_i) are the coordinates of keypoint i in the input image, and $\mathcal{T}$ is the set of triplets of aligned vertices.

To demonstrate the effectiveness of our DD-Poly algorithm at solving this problem, we employed the paper sequence of [26][1]. Since our approach does not require any initialization, each frame was treated independently. To remove the outliers in the provided SIFT correspondences, we followed the technique of [25] that observed that inliers form a 2D subspace from which outliers are distant. Since the energy function is quadratic, the polynomial system for each slave can be solved quickly. This let us use large cliques of 9 vertices arranged in a 3×3 grid. In Fig. 1, we show our registration results for some frames of the sequence. Note that the estimated mesh fits the surface well. More interestingly, Fig. 2 depicts typical evolutions of the primal and dual energies. The zero primal-dual gap guarantees global convergence of our DD-Poly algorithm. Note that this quadratic formulation makes convergence fast even without the augmented Lagrangian term. Such a quadratic energy could be handled by Gaussian BP, which shows that the problems addressed by Gaussian BP form a subset of those we can tackle.

4.2. Deformable Surface 3D Reconstruction

We now tackle the problem of reconstructing a deformable surface in 3D from a monocular image. Similarly as [26, 27], we assume that distances between neighboring points on the surface remain constant as the surface deforms. We follow the template-based formulation of [27], where a reference surface is given, and where the shape is parametrized in terms of the depth of image points.

More specifically, let $\mathbf{x} \in \mathbb{R}^N$ be the vector containing the depths of N image points representing the surface. Furthermore, let $\mathbf{q}_i = \mathbf{A}^{-1}(u_i, v_i, 1)^T$ be the line-of-sight of point i at image coordinates (u_i, v_i), with $\mathbf{A}$ the matrix of known internal camera parameters. Assuming isometric deformations, 3D reconstruction can be expressed as

$$\min_{\mathbf{x}} \sum_{(i,j)\in\mathcal{N}} \left(\|x_i \mathbf{q}_i - x_j \mathbf{q}_j\|^2 - d_{i,j}^2 \right)^2 , \qquad (11)$$

where x_i is the unknown depth of point i, $\mathcal{N}$ is the set of neighboring points whose distance must remain constant, and $d_{i,j}$ is the known distance between point i and point j.

As a first experiment, we made use of synthetic data consisting of a 3×3 mesh deformed isometrically by rotating neighboring facets around their common edge. We generated 100 different random deformations, and projected the mesh vertices to 2D using a known camera. We then added zero-mean Gaussian noise with a standard deviation of 2 pixels to these projections. The advantage of using such a small number of points is that the true global minimum of the energy can be obtained by homotopy continuation with all the variables. As a consequence, we report the difference between this ground-truth energy and the energies obtained with our algorithms and with baseline methods. For our algorithms, we used cliques of 4 points forming a square in the 3×3 grid. Fig. 3 depicts the histograms of relative energies for our algorithms (DD-Poly and ADMM-Poly), as well as for PCBP and Fusion Moves[2]. ADMM-Poly yields energies that are always very close to the best ones. The energies of DD-Poly are higher due to its slower convergence and to the use of a fixed maximum number of iterations. Note that, with DD-Poly, the adaptive step size rule yields slightly lower energies. Fusion Moves and PCBP yield,

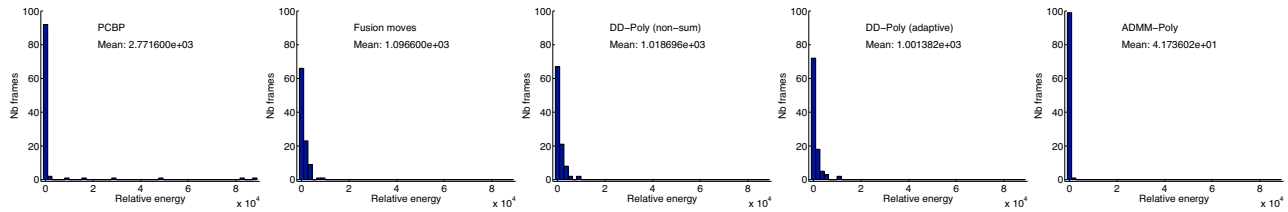

Figure 3. **Deformable 3D reconstruction from noisy data.** Histograms of relative energies w.r.t. the optimal energy found by homotopy continuation. We compare the results of PCBP, Fusion Moves, our DD-Poly algorithm (with non-summable step length and adaptive step size rules), and our ADMM-Poly algorithm. Note that ADMM-Poly yields significantly lower energies, as best seen from the mean.

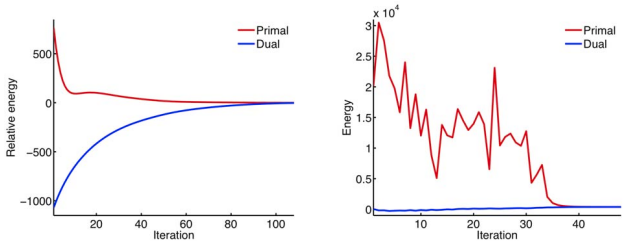

Figure 4. **Typical energy evolution during 3D reconstruction.** Primal and dual energies for ADMM-Poly on (left) a synthetic image, and (right) the first image of Fig. 5.

on average, higher energies than our algorithms. While PCBP gives low energies for most cases, the sampling procedure sometimes misses the optimal region of the space and thus yields high energy configurations. The average runtimes of the methods are 0.12s for Fusion Moves, 56.6s for PCBP, 117.8s for DD-Poly and 77.9s for ADMM-Poly. Note that, while slower than Fusion Moves, ADMM-Poly has a runtime comparable to PCBP for lower relative energy. Fig. 4(left) depicts typical energy curves for ADMM-Poly. Note that the duality gap went down to zero.

As a second 3D reconstruction experiment, we employed the same data as for non-rigid 2D registration. More precisely, we used our registration results as image locations of the mesh vertices, and estimated the depth of these points by solving the problem in Eq. 11. Our reconstruction results are shown in Fig. 5 and energy evolution curves in Fig. 4(right). Ground-truth 3D for this data is provided as Kinect depth measurements. Following [26], we computed reconstruction errors as the mean distance between the Kinect 3D points corresponding to the inlier SIFT features and our reconstruction of the same points estimated using barycentric coordinates. We obtained an average reconstruction error of 5.57mm, which compares favorably to the 7.23mm of [26].

4.3. Shape-from-Shading

We now demonstrate the use of our approach on shape-from-shading. Following [5], we parametrize the problem in terms of the unknown depth $x_{i,j}$ of each image pixel $(u_{i,j}, v_{i,j})$, where i and j account for the image row and column, respectively. Assuming an orthographic camera model, a 3D point is expressed as $\mathbf{v}_{i,j} = (u_{i,j}, v_{i,j}, x_{i,j})^T$. The image is decomposed into groups of five pixels of the form $\mathcal{G}_{ij} = \{(i,j),(i,j-1),(i,j+1),(i-1,j),(i+1,j)\}$. The normal to a triangle k composed of points $\mathbf{v}_{i,j}$, $\mathbf{v}_{i,j+1}$ and $\mathbf{v}_{i+1,j}$ can be expressed as $\mathbf{n}_k = (p_k, q_k, 1)^T$, with

$$p_k = \frac{(v_{i,j+1} - v_{i,j})(x_{i+1,j} - x_{i,j}) - (v_{i+1,j} - v_{i,j})(x_{i,j+1} - x_{i,j})}{(u_{i,j+1} - u_{i,j})(v_{i+1,j} - v_{i,j}) - (u_{i+1,j} - u_{i,j})(v_{i,j+1} - v_{i,j})} ,$$

and similarly for q_k from the cross production equation. Following the Lambertian model, the intensity of a facet can be written as $I_k = (l_1 p_k + l_2 q_k + l_3)/\sqrt{p_k^2 + q_k^2 + 1}$, where $\mathbf{l} = (l_1, l_2, l_3)^T$ is the known, normalized light direction. By summing over all the triangles in all the groups, shape from shading can be expressed as

$$\min_{\mathbf{x}} \sum_{\mathcal{G}_{ij},k} \left((p_k^2 + q_k^2 + 1)I_k^2 - (l_1 p_k + l_2 q_k + l_3)^2\right)^2 , \quad (12)$$

where $\mathbf{x}$ is the vector of all unknown depths, and where the intensity equation has been squared to make it a polynomial.

In [5], the shading equation was solved independently for each group by homotopy continuation, and a globally coherent solution found by solving a constraint satisfaction problem. As acknowledged in [5], this approach cannot handle noise due to the constraint satisfaction problem. In our dual decomposition formalism, however, the pixels shared by multiple groups are iteratively enforced to agree on their depth, which lets us handle noise. To overcome the global depth ambiguity, we added an L_2 regularizer with very low weight (i.e., 10^{-8}) that encourages the depth of the center of each group to be close to zero. Furthermore, to choose amongst the multiple discrete solutions of each group, we followed a greedy procedure that covers the image row by row. For stability, we used a low ρ_0 in ADMM-Poly. Note that, here, cliques must be of size at least 3, which makes methods such as PCBP less practical.

We performed experiments on the well-known vase image. To evaluate the effectiveness of our method at finding the global minimum of the energy, we first used a very coarse 4×4 version of this image, which let us compute

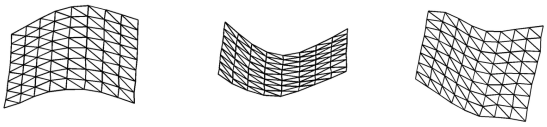

Figure 5. **3D reconstruction from real images.** Side view of our reconstructed meshes for the same images as in Fig. 1.

	Noise	Energy	Rel. err	GT err	Time[s] (1 core)	Time[s] (2 cores)
HC	0%	0	0	0	2502	N/A
Ours		0	0	0	7.5	4.6
HC	1%	0.00034	0	0.56	2740	N/A
Ours		0.00043	0.0095	0.56	959	643
HC	10%	0.0143	0	0.49	2734	N/A
Ours		0.0183	0.0172	0.48	954	651

Table 1. **Shape from shading: Energies, errors and runtimes.** Comparison of energies, depth errors and runtimes between homotopy continuation (HC) and ADMM-Poly (Ours) on a 4×4 version of the vase image for noise with std of 0%, 1% and 10%. Errors are given relative to the homotopy continuation reconstructions (Rel. err) and with respect to ground-truth (GT err).

the true optimal solution by homotopy continuation. We added zero-mean Gaussian noise with standard deviations of 0%, 1% and 10% to the image intensities. In Table 1, we report the energy and depth error with respect to the homotopy continuation solution and with respect to ground-truth. The depth error is computed as the mean absolute difference between the recovered depths and the ground-truth ones divided by the maximum depth. Since a global depth ambiguity still remains, we centered all the shapes at zero depth. Similarly, to handle the global concave-convex ambiguity, we report the lowest error for these two discrete cases. Note that the shapes obtained with ADMM-Poly are very close to the best possible shapes obtained by homotopy continuation. Note also that even with a single core, our approach is faster than homotopy continuation on the full image. We noticed that, in shape-from-shading, the results are more sensitive to the values of ρ_0, ρ_1 and η_0 than in our other experiments for which finding the right order of magnitude was sufficient. This seems to be due to the presence of multiple discrete solutions. Fortunately, the primal-dual gap indicates convergence, which makes parameter tuning possible without using ground-truth. This sensitivity to the parameters can be mitigated by introducing a smoothness prior similar to the one used for non-rigid registration. Fig. 6 depicts our results on finer-scale surfaces, with energy curves shown in Fig. 7. Note that, with 10% noise, the duality gap did not go down to zero.

4.4. Optical Flow Estimation

Finally, we applied our approach to the problem of optical flow estimation from a pair of images I^t, I^{t+1}. Note that the real purpose of this experiment is not to address optical flow, but rather to illustrate the use of our method with non-polynomial energies. Following the brightness constancy assumption [3], we search for a displacement (x_i^u, x_i^v) at each pixel (u_i, v_i), such that $I^{t+1}(u_i + x_i^u, v_i + x_i^v) = I^t(u_i, v_i)$. While the image is, of course, not a polynomial function, it is common to linearize it and write $I^{t+1}(u_i + x_i^u, v_i + x_i^v) = I^t(u_i, v_i) + I_u^t(u_i, v_i)x_i^u + I_v^t(u_i, v_i)x_i^v + I_t^t(u_i, v_i)$, where I_u, I_v and I_t are the u, v and temporal image derivatives, respectively. By dropping the dependence on (u_i, v_i) for notational convenience, optical flow estimation can be expressed as

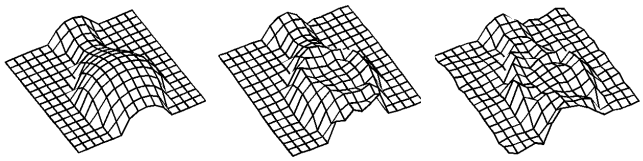

Figure 6. **Shape from shading.** Side view of our reconstructed vase surfaces for noise standard deviations of 0%, 1% and 10%. Depth errors are 0.0036, 0.112 and 0.133, respectively.

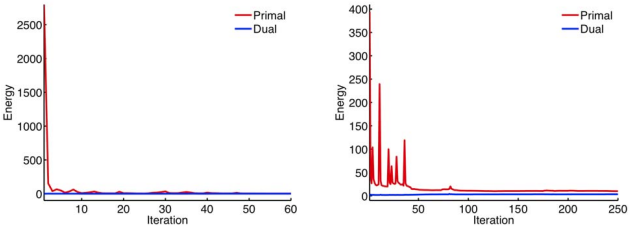

Figure 7. **Energy evolution in shape-from-shading.** Primal and dual energies for ADMM-Poly with 1% and 10% noise.

$$\min_{\mathbf{x}} \sum_i \left(I_u^t x_i^u + I_v^t x_i^v + I_t^t\right)^2 + \sum_{(i,j)\in\mathcal{N}} w_r\big(\phi(x_i^u, x_j^u) + \phi(x_i^v, x_j^v)\big)\,, \tag{13}$$

where $\mathbf{x}$ is the flow vector of all pixels, and $\phi(\cdot,\cdot)$ is a regularizer on the flow of neighboring pixels. Here, we employ the regularizer $\phi(x_i^u, x_j^u) = \ln(1+(x_i^u - x_j^u)^2/(2\sigma^2))$ based on the Lorentzian robust estimator [3].

Due to the regularizer, the energy is not a polynomial function. However, for a clique of two neighboring pixels, its gradient yields constraints of the form

$$2(I_u^t x_i^u + I_v^t x_i^v + I_t^t)I_u^t + 2w_r \frac{x_i^u - x_j^u}{2\sigma^2 + (x_i^u - x_j^u)^2} = 0\,.$$

Multiplying this equation by $(2\sigma^2 + (x_i^u - x_j^u)^2)$ gives a degree 3 polynomial, which therefore makes our method applicable. This shows that, strictly speaking, our method does not only apply to polynomial energies, but rather to polynomial gradients. The image term illustrates the use of a polynomial approximation to a non-polynomial function.

We performed experiments on the Rubber Whale image of the Middlebury benchmark. All image operations were performed using the code of [23]. We used two image pyramid levels, and at each one, kept the first order Taylor image approximation fixed, but updated it for the next level. We also employed an additional quadratic regularizer to encourage the flow at each pixel to remain small. Although not strictly necessary, we noticed that this regularizer sped up convergence. For computational reasons, we only computed the flow at a maximum resolution of a quarter of the original resolution. Higher resolutions could be handled by making use of more CPUs. To compare our results against

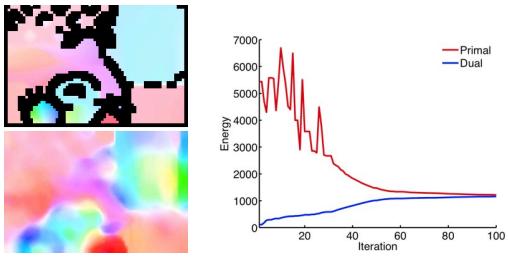

Figure 8. **Optical flow estimation.** Low-resolution ground-truth and ADMM-Poly flow estimate for the Rubber Whale image.

ground-truth, we resampled our flow to the full resolution. Our ADMM-Poly algorithm achieved an average end-point error of 0.716. To give an intuition of this number, running the code of [23] with the energy in Eq. 13 and at the same pyramid levels as us yields an average end-point-error of 0.721. In Fig. 8, we show the low-resolution ground-truth flow and the flow recovered with ADMM-Poly, as well as the energy curves at the highest pyramid level. Note that the different regions of the image are correctly identified.

5. Conclusion

In this paper, we have introduced an approach to performing continuous inference in graphical models with polynomial energies. To this end, we have followed a dual decomposition strategy, where each slave problem can be solved exactly using homotopy continuation. We have shown that our approach can be applied to a wide variety of computer vision problems. In the future, we plan to study how problems where the slaves have multiple discrete solutions can be better handled, for instance by keeping track of all these solutions. We also intend to study in more details the use of polynomial function approximations.

References

[1] D. Bertsekas. Nonlinear Programming. *Athena Scientific*, 1999.

[2] D. Bickson. Gaussian Belief Propagation: Theory and Application. In *Arxiv preprint arXiv:0811.2518*, 2008.

[3] M. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. In CVIU, 1996.

[4] D. Cox, J. Little and D. OShea. Using Algebraic Geometry. *Springer-Verlag*, 2005.

[5] A. Ecker and A.J. Jepson. Polynomial Shape from Shading. In *CVPR*, 2010.

[6] S. Heber, R. Ranftl and T. Pock. Approximate Envelope Minimization for Curvature Regularity. In *ECCV Workshops*, 2012.

[7] A. Ihler and D. McAllester. Particle Belief Propagation. In *AISTATS*, 2009.

[8] V. Kolmogorov. Convergent Tree-reweighted Message Passing for Energy Minimization. In *PAMI*, 2006.

[9] N. Komodakis, N. Paragios and G. Tziritas. MRF Energy Minimization and Beyond via Dual Decomposition. In *PAMI*, 2010.

[10] Z. Kukelova, M. Bujnak and T. Pajdla. Automatic Generator of Minimal Problem Solvers. In *ECCV*, 2008.

[11] T.L. Lee, T.Y. Li and C.H. Tsai. HOM4PS-2.0: A software package for solving polynomial systems by the polyhedral homotopy continuation method. In *Computing*, 2008.

[12] V. Lempitsky, C. Rother, S. Roth and A. Blake. Fusion Moves for Markov Random Field Optimization. In *PAMI*, 2010.

[13] A. Martins, M. Figueiredo, P. Aguiar, N. Smith and E. Xing. An Augmented Lagrangian Approach to Constrained MAP Inference. In *ICML*, 2011.

[14] T. Meltzer, C. Yanover and Y. Weiss. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *ICCV*, 2005.

[15] J. Peng, T. Hazan, D. McAllester and R. Urtasun. Convex Max-Product Algorithms for Continuous MRFs with Applications to Protein Folding. *ICML*, 2011.

[16] J. Pilet, V. Lepetit and P. Fua. Real-Time Non-Rigid Surface Detection. In *CVPR*, 2005.

[17] N. Pustelnik, C. Chaux and J. Pesquet. Parallel proximal algorithm for image restoration using hybrid regularization. In *TIP*, 2011.

[18] M. Salzmann and R. Urtasun. Beyond Feature Points: Structured Prediction for Monocular Non-rigid 3D Reconstruction. In *ECCV*, 2012.

[19] A.J. Sommese and C.W. Wampler. The Numerical Solution of Systems of Polynomials Arising in Engineering and Science. In *World Scientific*, 2005.

[20] L. Song, A. Gretton, D. Bickson, Y. Low and C. Guestrin. Kernel Belief Propagation. In *AISTATS*, 2011.

[21] H. Stewenius, D. Nister, F. Kahl and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *CVPR*, 2005.

[22] E.B. Sudderth, A.T. Ihler, M. Isard, W.T. Freeman and A.S. Willsky. Nonparametric belief propagation. In *Communications of the ACM*, 2010.

[23] D. Sun, S. Roth and M. Black. Secrets of Optical Flow Estimation and Their Principles. In *CVPR*, 2010.

[24] M.F. Tappen and W.T. Freeman. Comparison of Graph Cuts with Belief Propagation for Stereo, using Identical MRF Parameters. In *ICCV*, 2003.

[25] Q.-H. Tran, T.-J. Chin, G. Carneiro, M. Brown and D. Suter. In Defense of RANSAC for Outlier Rejection in Deformable Registration. In *ECCV*, 2012.

[26] A. Varol, M. Salzmann, P. Fua and R. Urtasun. A Constrained Latent Variable Model. In *CVPR*, 2012.

[27] S. Vicente and L. Agapito. Soft inextensibility constraints for template-free non-rigid reconstruction. In *ECCV*, 2012.

[28] A. Wedel, T. Pock, C. Zach, D. Cremers and H. Bischof. An Improved Algorithm for TV-L1 Optical Flow. In *Dagstuhl Motion Workshop*, 2008.

[29] K. Yamaguchi, T. Hazan, D. McAllester and R. Urtasun. Continuous Markov Random Fields for Robust Stereo Estimation. In *ECCV*, 2012.

[30] C. Zach and P. Kohli. A Convex Discrete-Continuous Approach for Markov Random Fields. In *ECCV*, 2012.