COMBINATORIAL CONTINUOUS MAXIMUM FLOW

CAMILLE COUPRIE *, LEO GRADY †, HUGUES TALBOT * , AND LAURENT NAJMAN *

Abstract. Maximum flow (and minimum cut) algorithms have had a strong impact on computer vision. In particular, graph cuts algorithms provide a mechanism for the discrete optimization of an energy functional which has been used in a variety of applications such as image segmentation, stereo, image stitching and texture synthesis. Algorithms based on the classical formulation of max-flow defined on a graph are known to exhibit metrication artefacts in the solution. Therefore, a recent trend has been to instead employ a spatially *continuous* maximum flow (or the dual min-cut problem) in these same applications to produce solutions with no metrication errors. However, known fast continuous max-flow algorithms have no stopping criteria or have not been proved to converge. In this work, we revisit the continuous max-flow problem and show that the analogous discrete formulation is *different* from the classical max-flow problem. We then apply an appropriate combinatorial optimization technique to this combinatorial continuous max-flow (CCMF) problem to find a null-divergence solution that exhibits no metrication artefacts and may be solved exactly by a fast, efficient algorithm with provable convergence. Finally, by exhibiting the dual problem of our CCMF formulation, we clarify the fact, already proved by Nozawa in the continuous setting, that the max-flow and the total variation problems are not always equivalent.

1. Introduction. Optimization methods have been used to address a wide variety of problems in computer vision. The early optimization approaches were formulated in terms of active contours and surfaces [1] and then later level sets [2]. These formulations were used to optimize energies of varied sophistication (e.g., using regional, texture, motion or contour terms [3]) but generally converged to a local minimum, generating results that were sensitive to initial conditions and noise levels. Consequently, more recent focus has been on energy formulations (and optimization algorithms) for which a global optimum can be found.

Such energy formulations typically include a term which minimizes the boundary length (or surface area in 3D) of a region or the total variation of a scalar field in addition to a data term and/or hard constraints. In this paper, we focus on image segmentation as the example application on which to base our exposition. Indeed, segmentation has played a prominent (and early) role in the development of the use of global optimization methods in computer vision, and often forms the basis of many other applications [4, 5, 6].

Graph-based approaches. The max-flow/min-cut problem on a graph is a classical problem in graph theory, for which the earliest solution algorithm goes back to Ford and Fulkerson [7]. Initial methods for global optimization of the boundary length of a region formulated the energy on a graph and relied on max-flow/min-cut methods for solution [8, 9]. It was soon realized that these methods introduced a metrication error. Metrication errors are clearly visible when gradient information is weak or missing, for example in the case of medical image segmentation or in some materials science applications like electron tomography, where weak edges are unavoidable features of the imaging modality. Metrication errors are also far more obvious in 3D than in 2D and increasingly so, as the dimensionality increases (see for example the 3D-lungs example in [10]). Furthermore, metrication artifacts are

*Université Paris-Est, Laboratoire d'Informatique Gaspard-Monge, Equipe A3SI, ESIEE Paris (93160 Noisy-le-Grand, France) (c.couprie@esiee.fr, h.talbot@esiee.fr, l.najman@esiee.fr).

†Siemens Corporate Research, Department of Imaging and Visualization, Princeton, N.J. 08540. USA) (leo.grady@siemens.com).

even more present in other applications such as surface reconstruction and rendering, where the artifacts are a lot worse than in image segmentation. Various solutions for metrication errors were proposed. One solution involved the use of a highly connected lattice with a specialty edge weighting [11], but the large number of graph edges required to implement this solution could cause memory concerns when implemented for large 2D or 3D images [10].

Continuous approaches. To avoid the metrication artefacts without increasing memory usage, one trend in recent years has been to pursue spatially *continuous* formulations of the boundary length and the related problem of total variation [12, 10, 13]. Historically, a continuous max-flow (and dual min-cut problem) was formulated by Iri [14] and then Strang [15]. Strang's continuous formulation provided an example of a *continuization* (as opposed to *discretization*) of a classically discrete problem, but was not associated to any algorithm. Work by Appleton and Talbot [10] provided the first PDE-based algorithm to find Strang's continuous max-flows and therefore optimal min-cuts. This same algorithm was also derived by Unger *et al.* from the standpoint of minimizing continuous total variation [16]. Adapted to image segmentation, this algorithm is shown to be equivalent [17] to the Appleton-Talbot algorithm and has been demonstrated to be fast when implemented on massively parallel architectures. Unfortunately, this algorithm has no stopping criteria and has not been proved to converge. Works by Pock *et al* [18], Zach *et al* [19] and Chambolle *et al* [20] present different algorithms for optimizing comparable energies for solving multilabel problems, but again, those algorithms are not proved to converge. Some other works have been presenting provably converging algorithms, but with relatively slow convergence speed. For example, Pock and Chambolle introduce in [21] a general saddle point algorithm that may be used in various applications, but needs half an hour to segment a 350×450 image on a CPU and still more than a dozen seconds on a GPU (Details are given in the experiments section).

Links and differences with total variation minimization. G. Strang [22] has shown the continuous max-flow problem for the l_2 norm to be the dual of the total variation (TV) minimization problem under some assumptions. The problem of total variations, introduced successively in computer vision by Shulman and Hervé [23] and later Rudin, Osher and Fatemi [24] as a regularizing criterion for optical flow computation and then image denoising, has been shown to be quite efficient for smoothing images without affecting contours. Moreover, a major advantage of TV is that it is a convex problem. Thus, a straightforward algorithm such as gradient descent may be applied to find a minimum solution. However, there is a need for faster methods, and significant progress have been achieved recently with primal-dual approaches [12], Nesterov's algorithm [25] and Split-Bregman/Douglas-Rachford methods [26, 27]. Most methods minimizing TV focus on image filtering as an application, and even if those methods are remarkably fast in denoising applications, segmentation problems require a lot more iterations for those algorithms to converge. As stated previously, continuous max-flow is dual with total variation *in the continuous setting under restrictive regularity conditions [28].* In fact, Nozawa [29] showed that there is a duality gap between weighted TV and weighted max-flow under some conditions in the continuous domain. Thus, it is important to note that weighted TV problems are not equivalent to the weighted maximum flow problem studied in this paper. Furthermore, many works assume that the continuous-domain duality holds algorithmically, but we show later in this paper that at least in the combinatorial case this is not true.

The previous works for solving the max-flow problem illustrate two difficulties with continuous-based formulation, that are (1) the discretization step, which is necessary for deriving algorithms, but may break continuous-domain properties; and (2) the convergence, both of the underlying continuous formulation, and the associated algorithm, which itself depend on the discretization. It is very well known that even moderately complex systems of PDEs may not converge, and even existence of solutions is sometimes not obvious [30]. Even when existence and convergence proofs both exist, sometimes algorithmic convergence may be slow in practice. For these reasons, combinatorial approaches to the maximal flow and related problems are beginning to emerge.

Combinatorial approaches. Discrete calculus [31, 32] has been used in recent years to produce a combinatorial reformulation of continuous problems onto a graph in such a manner that the solution behaves analogously to the continuous formulation (e.g., [33, 34]).

In particular, Lippert presented in [35] a combinatorial formulation of an isotropic continuous max-flow problem on a planar lattice, making it possible to obtain a provably optimal solution. However, in Lippert's work, parameterization of the capacity constraint is tightly coupled to the 4-connected squared grid, and the generalization to higher dimensions seems non-trivial. Furthermore it involves the multiplication of the number of capacity constraints by the degree of each node, thus increasing the dimension of the problem. This formulation did not lead to a fast algorithm. The author compared several general solvers, quoting an hour as their best time for computing a solution on a 300×300 lattice.

Motivation and contributions. In this paper, we pursue a combinatorial reformulation of the continuous max-flow problem initially formulated by Strang, which we term *combinatorial continuous maximum flow* (CCMF). Viewing our contribution differently, we adopt a discretization of continuous max-flow as the primary problem of interest, and then we apply fast combinatorial optimization techniques to solve the discretized version.

Our reformulation of the continuous max-flow problem produces a (divergence-free) flow problem defined on an arbitrary graph. Strikingly, CCMF are not equivalent to the discretization of continuous max-flows produced by Appleton and Talbot or that of Lippert (if for no other reason than the fact that CCMF is defined on an arbitrary graph). Moreover, *CCMF is not equivalent to the classical max-flow on a graph.* In particular, we will see that the difference lies in the fact that capacity constraints in classical max-flow restrict flows along graph edges while the CCMF capacity constraints restrict the total flow passing through the nodes.

The CCMF problem is convex. We deduce an expression of the dual problem, which allows us to employ a primal-dual interior point method to solve it, such as interior point methods have been used for graph cuts [36] and second order cone problems in general [37]. The CCMF problem has several desirable properties, including:

1. the solution to CCMF on a 4-connected lattice avoids the metrication errors. Therefore, the gridding error problems may be solved without the additional memory required to process classical max-flow on a highly-connected graph;
2. in contrast to continuous max-flow algorithms of Appleton-Talbot (AT-CMF) and equivalent, the solution to the CCMF problem can be computed with guaranteed convergence in practical time;
3. the CCMF problem is formulated on an arbitrary graph, which enables the algorithm to incorporate nonlocal edges [38, 33, 39] or to apply it to arbitrary

clustering problems defined on a graph; and
4. the algorithm for solving the CCMF problem is fast, easy to implement, compatible with multi-resolution grids and is straightforward to parallelize.

Our computation of the CCMF dual further reveals that duality between total variation minimization and maximum flow does not hold for CCMF and combinatorial total variation (CTV). The comparison between those two combinatorial problems is motivated by several interests:

1. for clarification: in the continuous domain, the duality between TV and MF holds under some regularity constraints. In the discrete anisotropic domain (i.e. Graph Cuts), this duality always holds. However, In the isotropic weighted discrete case, i.e. in the CTV, AT-CMF or CCMF cases, we are not aware of any discretization such that the duality holds. It is not obvious to realize that the duality does not hold in the discrete setting even though it may in the continuous case. We present clearly the differences between the two problems, theoretically, and in term of results;
2. to expose links between CCMF and CTV: We prove that the weak duality holds between the two problems; and
3. for efficiency: in image segmentation, the fastest known algorithms to optimize CTV are significantly slower than CCMF. Therefore, there is reason to believe that CCMF may be used to efficiently optimize energies for which TV has been previously shown to be useful (e.g., image denoising).

In the next section, we review the formulation of continuous max-flow, derive the CCMF formulation and its dual and then provide details of the fast and provably convergent algorithm used to solve the new CCMF problem.

2. Method. Our combinatorial reformulation of the continuous maximum flow problem leads to a formulation on a graph which is *different* from the classical max-flow algorithm. Before proceeding to our formulation, we review the continuous max-flow and the previous usage of this formulation in computer vision.

2.1. The continuous max-flow (CMF) problem. First introduced by Iri [14], Strang presents in [15] an expression of the continuous maximum flow problem

$$\begin{aligned} &\max F_{st}, \\ \text{s.t.} \quad &\nabla \cdot \overrightarrow{F} = 0, \\ &||\overrightarrow{F}|| \leq g. \end{aligned} \tag{2.1}$$

Here we denote by F_{st} the total amount of flow going from the source location s to the sink location t, $\overrightarrow{F}$ is the flow, and g is a scalar field representing the local metric distortion. The solution to this problem is the exact solution of the geodesic active contour (GAC) (or surface) formulation [40]. In order to solve the problem (2.1), the Appleton-Talbot algorithm (AT-CMF) [10] solves the following partial differential equation system:

$$\begin{aligned} \frac{\partial P}{\partial \tau} &= -\nabla \cdot \overrightarrow{F}, \\ \frac{\partial \overrightarrow{F}}{\partial \tau} &= -\nabla P, \\ \text{s. t. } &||\overrightarrow{F}|| \leq g. \end{aligned} \tag{2.2}$$

Here P is a potential field similar to the excess value in the Push-Relabel maximum flow algorithm [41]. AT-CMF is effectively a simple continuous computational fluid dynamics (CFD) simulation with non-linear constraints. It uses a forward finite-difference discretization of the above PDE system subject to a Courant-Friedrich-Levy (CFL) condition, also seen in early level-sets methods. At convergence, the potential function P approximates an indicator function, with 0 values for the background labels, and 1 for the foreground, and the flow $\overrightarrow{F}$ has zero-divergence almost everywhere. However, there is no guarantee of convergence for this algorithm and, in practice, many thousands of iterations can be necessary to achieve a binary P, which can be very slow.

Although Appleton-Talbot is a continuous approach, applying this algorithm to image processing involves a discretization step. The capacity constraint $||\overrightarrow{F}|| \leq g$ is interpreted as $\max {F_{x(i)}}^2 + \max {F_{y(i)}}^2 \leq g_i^2$, with $F_{x(i)}$ the outgoing flow of edges linked to node i along the x axis, and $F_{y(i)}$ the outgoing flow linked to node i along the y axis. We notice that the weights are associated with point locations (pixels), which will correspond later to a node-weighted graph.

In the next section, we use the operators of discrete calculus to reformulate the continuous max-flow problem on a graph and show the surprising result that the graph formulation of the continuous max-flow leads to a different problem from the classical max-flow problem on a graph.

2.2. A discrete calculus formulation. Before establishing the discrete calculus formulation of the continuous max-flow problem, we specify our notation. A graph consists of a pair $G = (V, E)$ with vertices $v \in V$ and edges $e \in E \subseteq V \times V$. A transport graph $G(V, E)$ comprises two additional nodes, named a source s and a sink t, and additional edges linking some nodes to the sink and some to the source. Including the source and the sink nodes, the cardinalities of G are given by $n = |V|$ and $m = |E|$. An edge, e, spanning two vertices, v_i and v_j, is denoted by e_{ij}. In this paper we deal with weighted graphs that include weights on both the edges and nodes. An edge weight is a value assigned to each edge which may be viewed as a capacity in the context of a maximum flow problem. The weight of an edge, e_{ij}, is denoted by $\tilde{g}_{ij}$. In this work, we assume that $\tilde{g}_{ij} \in \mathbb{R}$ and $\tilde{g}_{ij} > 0$, and use $\tilde{g}$ to denote the vector of $\mathbb{R}^m$ containing the $\tilde{g}_{ij}$ for every edge e_{ij} of G. In addition to edge weights, we may also assign weights to nodes. The weight of node v_i is denoted by g_i. In this work, we also assume that $g_i \in \mathbb{R}$ and $g_i > 0$. We use g to denote the vector of $\mathbb{R}^n$ containing the g_{ij} for every edge e_{ij} of G. We define a flow through edge e_{ij} as F_{ij} where $F_{ij} \in \mathbb{R}$ and use the vector $F \in \mathbb{R}^m$ to denote the flows through all edges in the graph . Each edge is assumed to be oriented, such that a positive flow on edge e_{ij} indicates the direction of flow from v_i to v_j, while a negative flow indicates the direction of flow from v_j to v_i.

The incidence matrix is a key operator for defining a combinatorial formulation of the continuous max-flow problem. Specifically, the incidence matrix $A \in \mathbb{R}^{m \times n}$ is known to define the discrete calculus analogue of the gradient, while A^T is known to define the discrete calculus analogue of the divergence (see [32] and the references therein). The incidence matrix maps functions on nodes (a scalar field) to functions on edges (a vector field) and may be defined as

$$A_{e_{ij}v_k} = \begin{cases} +1 & \text{if } i = k, \\ -1 & \text{if } j = k, \\ 0 & \text{otherwise,} \end{cases} \tag{2.3}$$

for every vertex v_k and edge e_{ij}. In our formulation of continuous max-flow, we use the expression $|A|$ to denote the matrix formed by taking the absolute value of each entry individually.

Given these definitions, we now produce a discrete (combinatorial) version of the continuous max-flow of (2.1) on a transport graph. As in [32, 33, 34], the continuous vector field indicating flows may be represented by a vector on the edge set, F. Additionally, the combinatorial divergence operator allows us to write the first constraint in (2.1) as $A^T F = 0$. The second constraint in (2.1) involves the comparison of the point-wise norm of a vector field with a scalar field. Therefore, we can follow [32, 33, 34] to define the point-wise ℓ_2 norm of the flow field F as $\sqrt{|A^T|F^2}$. In our notations here, as in the rest of the paper, $F^2 = F \cdot F$ denotes a element-wise product, "$\cdot$" denoting the Hadamard (element-wise) product between the vectors, and the square root of a vector is also here and in the rest of the paper the vector composed of the square roots of every elements.

Putting these pieces together, we obtain

$$\begin{aligned} &\max F_{st}, \\ &\text{s.t.} \quad A^T F = 0, \\ &\quad |A^T|F^2 \leq g^2. \end{aligned} \tag{2.4}$$

Compare this formulation to the classical max-flow problem on a graph, given in our notation as

$$\begin{aligned} &\max F_{st}, \\ &\text{s.t.} \quad A^T F = 0, \\ &\quad |F| \leq \tilde{g}. \end{aligned} \tag{2.5}$$

By comparing these formulations of the traditional max-flow with our combinatorial formulation of the continuous max-flow, it is apparent that the key difference between the classical formulation and our combinatorial continuous max-flow is in the capacity constraints. In both formulations, the flow is defined through edges, but in the classical case the capacity constraint restricts flow through an edge while the CCMF formulation restricts the amount of flow passing through a node by taking in account the neighboring flow values. This contrast-weighting applied to nodes (pixels) has been a feature of several algorithms with a continuous formulation, including geodesic active contours [40], CMF [10], TVSeg [16], and [21]. On the other hand, the problem defined in (2.4) is defined *on an arbitrary graph* in which contrast weights (capacities) are almost always defined on *edges*. The node-weighted capacities fit Strang's formulation of a scalar field of constraints in the continuous max-flow formulation and therefore our graph formulation of Strang's formulation carries over these same capacities. The most important point here is not having expressed the capacity on the nodes of the graph but rather how the flow is enforced to be bounded by the metric in each node. Bounding the norm of neighboring flow values in each node simulates a closer behavior of the continuous setting than if no contribution of neighboring flow was made as in the standard max-flow problem. Figure 2.1 illustrates the relationship of the edge capacity constraints in the classical max-flow problem to the node capacity constraints in the CCMF formulation. The null-divergence constraint is also essential in our formulation since it permits us to obtain constant partitions almost everywhere in the dual solution introduced in the next section, in particular binary ones.

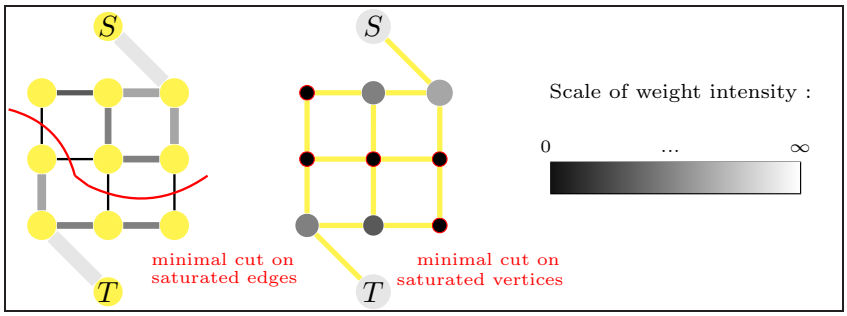


FIG. 2.1. *The difference between classical max-flow on a graph with the combinatorial continuous max-flow (CCMF) on a graph is that classical max-flow uses edge-weighted capacities while CCMF uses node-weighted capacities. This difference is manifest in the different solutions obtained for both algorithms and the algorithms required to find a solution. Specifically, the solution to the CCMF problem on a lattice does not exhibit metrication bias.*

In the context of image segmentation, the vector g varies inversely to the image gradient. We propose to use, as in [16],

$$g = \exp(-\beta||\nabla I||_2), \tag{2.6}$$

where I indicates the image intensity. For simplicity, this weighting function is defined for greyscale images, but g could be used to penalize changes in other relevant image quantities, such as color or texture. Before addressing the solution of the CCMF problem, we consider the dual of the CCMF problem and, in particular, the sense in which it represents a minimum cut. Since the cut weights are present on the nodes rather than the edges, we must expect the minimum cut formulation to be different from the classical minimum cut on a graph.

2.3. The CCMF dual problem. Classically, the max-flow problem is dual to the min-cut problem, allowing a natural geometric interpretation of the objective function. In order to provide the same interpretation, we now consider the dual problem to the CCMF and show that we optimize a node-weighted minimum cut. In the following proposition, we denote the element-wise quotient of two vectors $u = [u_1, \ldots, u_k]$ and $v = [v_1, \ldots, v_k]$ by $u \cdot /v = [u_1/v_1, \ldots, u_k/v_k]$. We also denote $\mathbf{1}^n$ a unit vector $[1, \ldots, 1]$ of size n, and recall that the square exponent v^2 of a vector v represents the resulting vector of the element-wise mulitiplication $v \cdot v$.

PROPOSITION 2.1. *In a transport graph G with m edges, n nodes, we define a vector c of $\mathbb{R}^m$, composed of zeros except for the element corresponding to the source/sink edge which is* 1. *Let λ and ν be two vectors of $\mathbb{R}^n$. The CCMF problem*

$$\begin{aligned} \max_{F \in \mathbb{R}^m} \quad & c^T F, \\ s.\ t. \quad & A^T F = 0, \\ & |A^T| F^2 \leq g^2. \end{aligned} \tag{2.7}$$

has for dual

$$\begin{aligned} \min_{\lambda \in \mathbb{R}^n,\ \nu \in \mathbb{R}^n} \quad & \lambda^T g^2 + \frac{1}{4} \Big(\mathbf{1}^n \cdot /(|A|\lambda) \Big)^T \Big((c + A\nu)^2 \Big), \\ s.\ t. \quad & \lambda \geq 0, \end{aligned} \tag{2.8}$$

equivalently written in (2.15)*, and the optimal solution* (F^*, λ^*, ν^*) *verifies*

$$\max_F c^T F = c^T F^* = 2{\lambda^*}^T g^2, \tag{2.9}$$

and the n *following equalities*

$$\lambda^* \cdot |A^T| \Big((c + A\nu^*) \cdot / (|A|\lambda^*) \Big)^2 = 4\lambda^* \cdot g^2. \tag{2.10}$$

Proof. The Lagrangian of (2.7) is

$$L(F, \lambda, \nu) = (c^T + \nu^T A^T)F + \lambda^T |A^T| F^2 - \lambda^T g^2,$$

with $\lambda \in \mathbb{R}^n$ and $\nu \in \mathbb{R}^n$ two Lagrange multipliers. We have to find F in the dual function is such as $\nabla_F L(F, \lambda, \nu) = 0$.

$$\nabla_F L(F, \lambda, \nu) = c + A\nu + 2|A|\lambda \cdot F = 0 \;\Leftrightarrow\; F = (c + A\nu) \cdot / (-2|A|\lambda). \tag{2.11}$$

Substituting F in the Lagrangian function , we obtain

$$L(\lambda, \nu) = (c^T + \nu^T A^T) \Big((c + A\nu) \cdot / (-2|A|\lambda) \Big) + \lambda^T |A^T| \Big((c + A\nu) \cdot / (-2|A|\lambda) \Big)^2 - \lambda^T g^2.$$

The Lagrangian may be simplified as:

$$L(\lambda, \nu) = -\Big(\mathbf{1}^n \cdot / (4|A|\lambda) \Big)^T \Big((c + A\nu)^2 \Big) - \lambda^T g^2.$$

The dual of (2.7) is also

$$\begin{aligned} \min_{\lambda, \nu} \; & \lambda^T g^2 + \frac{1}{4} \Big(\mathbf{1}^n \cdot / (|A|\lambda) \Big)^T \Big((c + A\nu)^2 \Big), \\ \text{s. t.} \quad & \lambda \geq 0. \end{aligned} \tag{2.12}$$

Furthermore, the KKT optimality conditions give

$$\lambda^* \cdot \Big(|A^T| {F^*}^2 - g^2 \Big) = 0. \tag{2.13}$$

Using the expression of F^* from equation (2.11) , and substituting it in (2.13), we obtain

$$\lambda^* \cdot |A^T| \Big((c + A\nu^*) \cdot / (|A|\lambda^*) \Big)^2 = 4\lambda^* \cdot g^2.$$

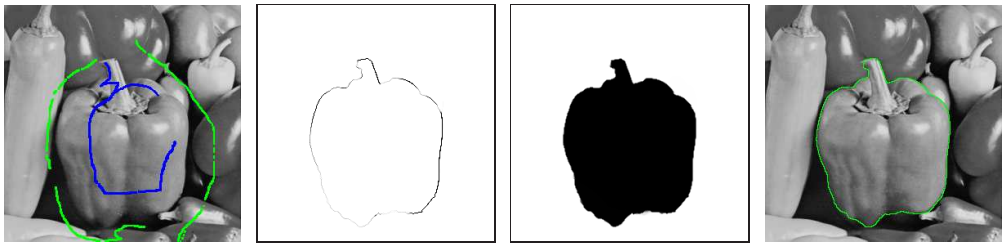
Taking the sum of right hand side and left hand side,

$$(c + A\nu^*)^2 \cdot / (|A|\lambda^*) = 4{\lambda^*}^T g^2. \tag{2.14}$$

Finally, if we substitute (2.14) in the dual expression (2.8), we have

$$\max_F \; c^T F = c^T F^* = 2{\lambda^*}^T g^2.$$

□



(a) (b) λ (c) ν (d) Threshold ν at .5

FIG. 2.2. *The dual problem to CCMF is a node-weighted minimum cut in which the variable λ is a weighted indicator vector labeling boundary nodes and the variable ν is a nearly binary vector indicating the source/sink regions. As a result, the contours of ν are slightly blurry. This is due to the equilibrium effect between the two dual variables. In practice, as λ is nonzero only in presence of a contour, ν is binary almost everywhere, except on a very thin line.*

The expression of the CCMF dual may be written in summation form as

$$\min_{\lambda,\nu} \sum_{v_i \in V} \overbrace{\lambda_i g_i^2}^{\text{weighted cut}} + \overbrace{\frac{1}{4} \sum_{e_{ij} \in E \setminus \{s,t\}} \frac{(\nu_i - \nu_j)^2}{\lambda_i + \lambda_j}}^{\text{smoothness term}} + \overbrace{\frac{1}{4} \frac{(\nu_s - \nu_t - 1)^2}{\lambda_s + \lambda_t}}^{\text{source/sink enforcement}} \quad \text{s. t. } \lambda_i \geq 0 \quad \forall i \in V. \tag{2.15}$$

Interpretation: The optimal value λ^* is a weighted indicator of the saturated vertices (a vertex v_i is saturated if $|A^T|_i F^2 = g_i^2$ where $|A^T|_i$ indicates the ith row of $|A^T|$):

$$\lambda^*(v_i) \begin{cases} > 0 & \text{if } |A^T|_i F^2 = g(v_i)^2, \\ = 0 & \text{otherwise.} \end{cases} \tag{2.16}$$

The variables ν_s and ν_t are not constrained to be set to 0 and 1, only their difference is constrained to be equal to one. Thus their value range between a constant and a constant plus one. Let us call this constant δ, and without loss of generality one can consider that $\delta = 0$. The term ν is at optimality a weighted indicator of the source/sink/saturated vertices partition:

$$\nu^*(v_i) = \begin{cases} 0 + \delta & \text{if } v_i \in S, \\ \text{a number between } (0+\delta) \text{ and } (1+\delta) & \text{if } |A^T|_i F^2 = g(v_i)^2, \\ 1 + \delta & \text{if } v_i \in T. \end{cases}$$

The expression (2.9) of the CCMF dual shows that the problem is equivalent to finding a minimum weighted cut defined on the nodes.

Finally, the "weighted cut" is recovered in (2.15), and the "smoothness term" is compatible with large variations of ν at the boundary of objects because of a large denominator (λ) in the contour area. An illustration of optimal λ and ν on an image is shown on Fig. 2.2.

2.4. Solving the Combinatorial Continuous Max-Flow problem. When considering how to optimize the CCMF problem (2.4), the first key observation is that the constraints bound a convex feasible region.

The real valued functions $f_i : \mathbb{R}^m \to \mathbb{R}$ defined for every node $i = 1, \ldots, n$ as $f_i(F) = |A^T|_i F^2 - g_i^2$ are non-negative quadratic, so convex functions. We define the vector $f(F)$ of $\mathbb{R}^n$ as $f(F) = [f_1(F), \ldots, f_n(F)]^T$.

Since the constraints are convex, the CCMF problem may be solved with a fast primal dual interior point method (see [42]), which we now review in the specific context of CCMF.

Algorithm 1: Primal dual interior point algorithm

Data: $F \in \mathbb{R}^m = 0, \lambda \in (\mathbb{R}_+^*)^n, \nu \in \mathbb{R}^n$, $\mu > 1, \epsilon > 0$.

Result: F solution to the CCMF problem (2.7) such as F_{st} is maximized under the divergence free and capacity constraints, and ν, λ solution to the CCMF dual problem (2.9).

repeat

1. Compute the surrogate duality gap $\widehat{\eta} = -f(F)^T \lambda$ and set $t = \mu n / \widehat{\eta}$.
2. Compute the primal-dual search direction Δ_y such as $M\Delta_y = r$.
3. Determine a step length $s > 0$ and set $y = y + s\Delta_y$. $\quad (y = [F, \lambda, \nu]^T)$

until $||r_p||_2 \leq \epsilon$, $||r_d||_2 \leq \epsilon$, *and* $\widehat{\eta} \leq \epsilon$.

The primal dual interior point (PDIP) algorithm iteratively computes the primal F and dual λ, ν variables so that the Karush-Kuhn-Tucker (KKT) optimality conditions are satisfied. This algorithm solves the CCMF problem (2.7) by applying Newton's method to a sequence of a slightly modified version of the KKT conditions. The steps of the PDIP procedure are given in Algorithm 1. Specifically for CCMF, the system $M\Delta_y = r$ system may be written

$$\begin{bmatrix} \sum_{i=1}^n \lambda_i \nabla^2 f_i(F) & Df(F)^T & A \\ -\mathrm{diag}(\lambda) Df(F) & -\mathrm{diag}(f(F)) & 0 \\ A^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta_F \\ \Delta_\lambda \\ \Delta_\nu \end{bmatrix} = - \begin{bmatrix} r_d = c + Df(F)^T \lambda + A\nu \\ r_c = -\mathrm{diag}(\lambda) f(F) - (1/t) \\ r_p = A^T F \end{bmatrix} \tag{2.17}$$

with r_d, r_c, and r_p representing the dual, central, and primal residuals. Additionally, the derivatives are given by

$$\nabla f_i(F) = 2|A^T|_i \cdot F, \quad Df(F) = \begin{bmatrix} \nabla f_1(F)^T \\ \vdots \\ \nabla f_n(F)^T \end{bmatrix}, \quad \nabla^2 f_i(F) = \mathrm{diag} \begin{bmatrix} 2|A^T|_{i1} \\ \vdots \\ 2|A^T|_{im} \end{bmatrix},$$

with "$\cdot$" denoting the Hadamard (element-wise) product between the vectors.

Consequently, the primary computational burden in the CCMF algorithm is the linear system resolution required by (2.17). In the result section we present execution times obtained in our experiments using a conventional CPU implementation.

Observe that, although this linear system is large, it is very sparse and is straightforward to parallelize on a GPU [43, 44, 45], for instance using an iterative GPU solver. If it does not necessarily imply a faster solution, the asymptotic complexity of modern iterative and modern direct solvers is about the same and, in our experience, there has been a strong improvement in the performance of a linear system solve when going from a direct solver CPU solution to a conjugate gradient solver GPU solution, especially as the systems get larger.

3. Comparison between CCMF and existing related approaches.

3.1. Min cut. As detailed in Section 2.3, there is a difference between the CCMF and the classic max flow formulation in the capacity constraint. Therefore, the dual

problems are different. As proved by Ford and Fulkerson [7], and formalized for example in [36], the min cut problem writes with our notations of section 2.3

$$\min_u \tilde{g}^T |Au + c|. \tag{3.1}$$

We note that the ℓ_1 norm of the gradient of the solution u of the above min-cut expression turns into an ℓ_2 norm of the gradient of the solution ν in the CCMF dual (2.8).

3.2. Primal Dual Total variations. Unger *et al.* [17] proposed to optimize the following primal dual TV formulation

$$\begin{gathered}\min_u \max_F \sum_{i,j} (u_{i+1,j} - u_{i,j}) F^1_{i,j} + (u_{i,j+1} - u_{i,j}) F^2_{i,j} + \text{data fidelity}, \\ \text{s.t. } \sqrt{(F^1_{i,j})^2 + (F^2_{i,j})^2} \leq g_{i,j}.\end{gathered} \tag{3.2}$$

We can note that the CCMF flow capacity constraint in a 4-connected lattice is similar to the constraint in (3.2) with a slight modification, and would be with finite-element notations

$$\sqrt{(F^1_{i-1,j})^2 + (F^2_{i,j-1})^2 + (F^1_{i,j})^2 + (F^2_{i,j})^2} \leq g_{i,j}. \tag{3.3}$$

In contrast to this finite element discretization, the provided CCMF formulation of continuous max-flow is defined arbitrary graphs. Furthermore, we note that the optimization procedure employed to solve (3.2) generalizes Appleton-Talbot's algorithm.

3.3. Combinatorial total variations. We now compare the dual CCMF problem with the Combinatorial Total Variation (CTV) problem. We note that the CCMF dual is different than CTV and discuss the weak duality of the two problems.

We recall the continuous expression of total variation given by Strang

$$\begin{gathered}\min_u \iint_\Omega g \, ||\nabla u|| \;\; \mathrm{d}x \, \mathrm{d}y, \\ \text{s. t.} \quad \int_\Gamma uf \, \mathrm{ds} = 1,\end{gathered} \tag{3.4}$$

where Ω represents the image domain and Γ the boundary of Ω. The source and sink membership is represented by f, such that $f(x, y) > 0$ if (x, y) belongs to the sink, $f(x, y) < 0$ if (x, y) belongs to the source, and $f = 0$ otherwise. Considering a transport graph G, and a vector u defined on the nodes, this continuous problem may be written with combinatorial operators

$$\begin{gathered}\min_{u \in \mathbb{R}^n} g^T \sqrt{|A^T|(Au)^2}, \\ \text{s. t.} \;\; u_s - u_t = 1,\end{gathered} \tag{3.5}$$

where the square root operator of a vector $v = [v_1, \ldots, v_k]$ is an element-wise square root operator $\sqrt{v} = [\sqrt{v_1}, \ldots, \sqrt{v_k}]$. Another way to write the same problem is

$$\begin{gathered}\min_u \sum_{v_i \in V} g_i \sqrt{\sum_{e_{ij} \in E} (u_i - u_j)^2}, \\ \text{s. t. } u_s - u_t = 1.\end{gathered} \tag{3.6}$$

We note that in these equations the capacity g must be defined on the vertices. Although we describe this energy as "combinatorial total variation", due to its derivation in discrete calculus terms, it is important to note that this formulation is very similar to the discretization which has appeared previously in the literature from Gilboa and Osher [46], and Chambolle [12] (if we allow $g = 1$ everywhere).

3.3.1. CCMF and CTV are not dual. Strang proved that the continuous max-flow problem is dual to the total variation problem under some assumptions on g. Remarkably, this duality is not observed in the discrete case in which the CCMF dual and CTV problems are given by

$$\begin{array}{c} \min_{\lambda,\nu}\ \lambda^T g^2 + \frac{1}{4}\left(\mathbf{1}^n \cdot /(|A|\lambda)\right)^T\left((c+A\nu)^2\right), \\ \text{s. t. } \lambda \geq 0, \end{array} \quad \not\equiv \quad \begin{array}{c} \min_u g^T\sqrt{|A^T|(Au)^2}, \\ \text{s. t. } u_s - u_t = 1. \end{array}$$

We note that the analytic expressions are different and also not equivalent. The duality of the classical max-flow problem with the minimum cut holds because in the expression of the Lagrangian function, is possible to deduce a value of λ by substituting it into the dual problem so that it only depends on ν. However, λ in the dual CCMF problem (2.9) depends on several values of neighboring λ and ν. Thus, the CCMF dual problem cannot be simplified by removing a variable, for example by identifying ν and u in the CTV problem. Said differently, combining a null-divergence constraint with an *anisotropic* capacity constraint in the classical max-flow formulation allows the duality to hold; in contrast, in the CCMF formulation, combining a null-divergence constraint with an *isotropic* capacity constraint does not allow such duality to hold.

Numerically we can also show on examples that the value $\max F_{st}$ of the flow optimizing CCMF is not equal to the minimum value of CTV (See Table 3.1).

3.3.2. Theoretical links between CCMF and CTV. Even if CCMF is not dual with CTV, the two problems are weakly dual.

In the combinatorial setting the weak duality property is given by

$$||F|| \leq g \Rightarrow F^T(Au) \leq g^T||Au||. \tag{3.7}$$

The next property shows that the norm $||F|| = |A^T|F^2$ verifies weak duality.

PROPOSITION 3.1. *Let G be a transport graph, F a flow in G verifying the capacity constraint $\sqrt{|A^T|F^2} \leq g$. Let u be a vector of $\mathbb{R}^n$ (defined on nodes of G). Then*

$$F^T Au \leq g^T\sqrt{|A^T|(Au)^2}.$$

Proof. Since we know that $\sqrt{|A^T|F^2} \leq g$, the following statement is true

$$\sqrt{|A^T|F^2}^T\sqrt{|A^T|(Au)^2} \leq g^T\sqrt{|A^T|(Au^2}.$$

We can now show that

$$F^T Au \leq \sqrt{(|A^T|F^2)}^T\sqrt{|A^T|(Au)^2}.$$

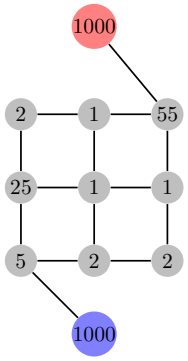
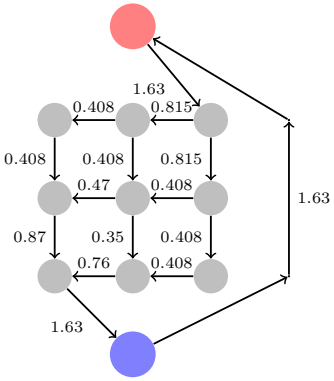
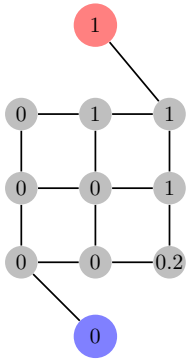
Transport graph with weights g on the nodes	**Combinatorial Continuous Max-Flow (CCMF)**	**Combinatorial Total Variation (CTV)**
	$\max_F \quad F_{st},$ s. t. $\quad A^T F = 0,$ $\quad\quad\; \lvert A^T\rvert F^2 \leq g^2.$	$\min_u \; g^T \sqrt{\lvert A^T\rvert (Au)^2},$ s. t. $\quad u_s - u_t = 1.$
		
	$F_{st} = 1.63$	$CTV(u) = \sum_i g_i \sqrt{\sum_j (u_i - u_j)^2}$ $= 2 + 1 \times \sqrt{2} + 1 \times \sqrt{2}$ $+1 \times \sqrt{1 + 0.8^2}$ $+2 \times \sqrt{0.8^2 + 0.2^2}$ $+2 \times 0.2$ $= 8.16.$

TABLE 3.1

Example illustrating the difference between optimal solutions of combinatorial continuous maximum flow problem (CCMF), and combinatorial total variation (CTV).

Using summation notation, then by the Cauchy-Schwartz inequality,

$$\sum_{i\in V}\sum_{e_{ij}\in E} F_{ij}(u_i - u_j) \leq \sum_{i\in V} \sqrt{\left(\sum_{e_{ij}\in E} {F_{ij}}^2\right)\left(\sum_{e_{ij}\in E} (u_i - u_j)^2\right)}.$$

We conclude that

$$F^T Au \leq \sqrt{(|A^T|F^2)}^T \sqrt{|A^T|(Au)^2} \leq g^T \sqrt{|A^T|(Au)^2}.$$

□

In terms of energy value, this proposition means that the CCMF energy, that is to say the flow, is always smaller than the combinatorial total variation.

PROPOSITION 3.2. *Let F be a compatible flow verifying the constraints in* (2.4),

and u a vector of $\mathbb{R}^n$ such as $u_s - u_t = 1$. Then,

$$F_{st} \leq g^T \sqrt{A^T (Au)^2}. \tag{3.8}$$

Proof. In the combinatorial setting, the Green formula gives

$$F^T Au = u^T A^T F.$$

Let a be a vector of $\mathbb{R}^n$ defined for each node v_i as

$$a_i = \begin{cases} -1 & \text{if } v_i \text{ belongs to the sink,} \\ 1 & \text{if } v_i \text{ belongs to the source,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

We can provide the following equivalent formulation of the CCMF problem (2.4) using an incidence matrix A of the transport graph deprived of the sink-source edge:

$$\begin{aligned} &\max F_{st}, \\ \text{s.t.} \quad &A^T F = F_{st} a, \\ &|A^T| F^2 \leq g. \end{aligned} \tag{3.10}$$

As F verifies the divergence free constraint $A^T F = F_{st} a$,

$$u^T A^T F = u^T F_{st} a,$$

and as the equation $u_s - u_t = 1$ may be written equivalently $a^T u = 1$, we conclude that

$$u^T F_{st} a = F_{st},$$

and by weak duality (Property 3.1),

$$F_{st} \leq g^T \sqrt{A^T (Au)^2}.$$

□

This property should be of interest for extending CCMF to applications other than segmentation, which is outside the scope of this paper. In the next section, we present the performance of our approach in the context of segmentation.

4. Results. We now present applications of Combinatorial Continuous Maximum Flow in image segmentation. To be used as a segmentation algorithm, three solutions are possible: the dual variable ν may be used directly if the user needs a matted result, otherwise ν may be thresholded, or finally an isoline or isosurface may be extracted from ν.

In the introduction we discussed how several works are related to CMF. Some are equivalent or slight modifications of AT-CMF for non-segmentation applications [16, 17, 18, 19]. As in the original AT-CMF work, none of these come with a convergence proof. In contrast, the work of [47] and [35] are provably convergent but both are very slow, and do not generalize easily to 3D image segmentation. Consequently, it seems reasonable to compare our segmentation method to the original AT-CMF as representative of the continuous approach, as well as graph cuts, which represent the

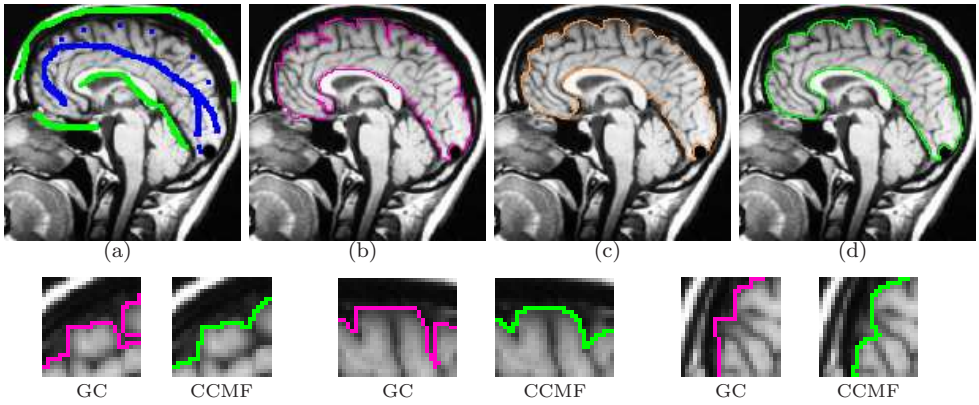


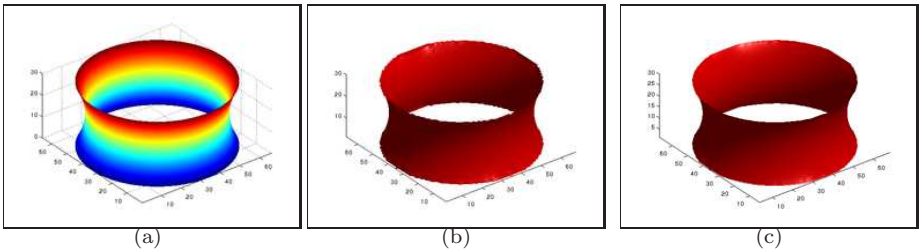
FIG. 4.1. *Brain segmentation. (a) Original image with foreground and background seeds. (b,c,d) Segmentation obtained with (b) graph cuts (GC), (c) AT-CMF, threshold of P (obtained after 10000 iterations), (d) CCMF, threshold of ν (15 iterations).*

purely discrete case. Finally, even if the total variation problem is different than the CMF problem (existence of a duality gap in the continuous [29]), the two problems are related enough to be compared. Specifically, we will also compare with Chambolle and Pock's recent work [21] which presented an algorithm for optimizing a TV-based energy for image segmentation.

Our validation is intended to establish three properties of the CCMF algorithm. First, we establish that the CCMF does avoid the metrication artifacts exhibited by conventional graph cuts (on a 4-connected lattice). This property is established by examples on a natural image and the recovery of the classical catenoid structure as the minimal surface spanning two rings. Second, we compare the convergence of the CCMF algorithm to the AT-CMF algorithm to show that the CCMF algorithm converges more quickly and in a more stable fashion. Finally, we establish that our formulation of the CMF problem does not degrade segmentation performance on a standard database. In fact, because of the reduction in metrication error our algorithm gives improved numerical results.For this experiment, we use the GrabCut database to compare the quality of the segmentation algorithms. In addition to the above tests, we demonstrate through examples that the CCMF algorithm is also flexible enough to incorporate prior (unary) terms and to operate in 3D.

4.1. Metrication artifacts and minimal surfaces. We begin by comparing the CCMF segmentation result with the classical max-flow algorithm (graph cuts). Figure 4.1 shows the segmentation of a brain, in which the contours obtained by graph cuts are noticeably blocky in the areas of weak gradient, while the contours obtained by both AT-CMF and CCMF are smooth.

In the continuous setting, the maximum flow computed in a 3D volume produces a minimal surface. The CCMF formulation may be also recognized as a minimal surface problem. In the dual formulation, the objective function is equivalent to a weighted sum of surface nodes. In [10], Appleton and Talbot compared the surfaces obtained from their algorithm with the analytic solution of the catenoid problem to demonstrate that their algorithm was a good approximation of the continuous minimal surface and was not creating discretization artifacts. The catenoid problem arises from consideration of two circles with equal radius whose centers lie along the z axis. The



(a) (b) (c)

FIG. 4.2. *The catenoid test problem: The source is constituted by two full circles and sink by the remaining boundary of the image. (a) Surface computed analytically, (b) isosurface of P obtained by CMF, (c) isosurface of ν obtained by CCMF. The root mean square error (RSME) has been computed to evaluate the precision of the results to the surface computed analytically. The RSME for CMF is 1.98 and for CCMF 0.75. The difference between those results is due to the fact that the CMF algorithm enforces exactly the source and sink points, leading to discretization around the disks. In contrast, the boundary localized around the seeds of ν is smooth, composed of grey levels. Thus the resulting isosurface computed by CCMF is more precise.*

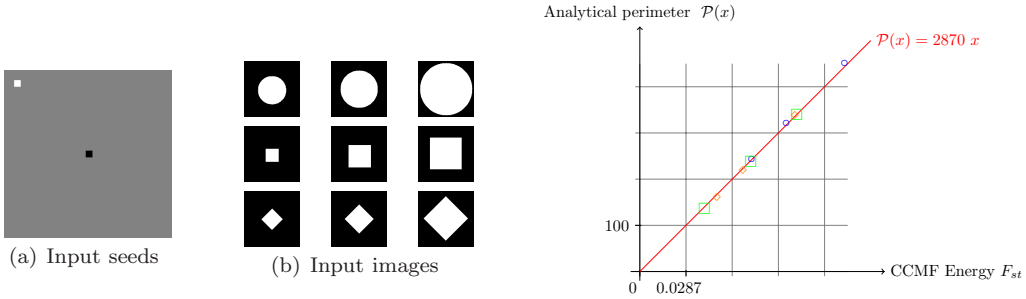
minimal surface which forms to connect the two circles is known as a catenoid. The catenoid appears in nature, for example by creating a soap bubble between two rings. In order to demonstrate that CCMF is also finding a minimal surface, we performed the same catenoid experiment as was in [10]. The results are displayed in Figure 4.2, where we show that CCMF approximates the analytical solution of the catenoid with even greater fidelity than the AT-CMF example.

In order to verify that the solution obtained by CCMF approximates perimeters of planar objects in 2D, we segmented several shapes – squares, discs – with known perimeters, scaled at different size, and compared the analytic perimeters with the energies obtained by CCMF. The results are reported in Fig. 4.3, where we observe that the true boundary length is proportional to the energy obtained with CCMF.

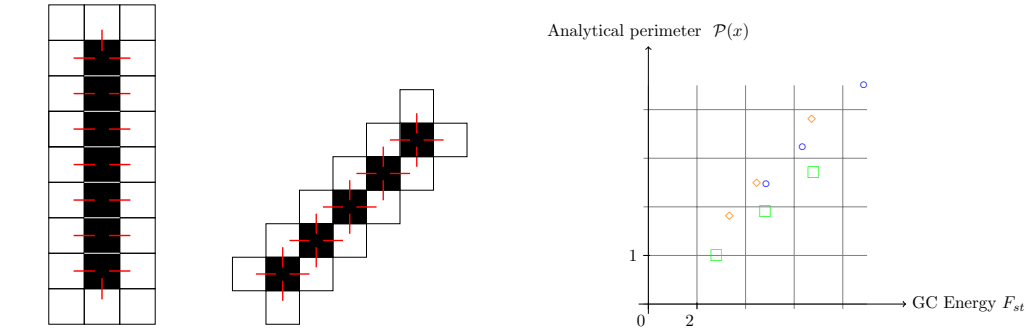
4.2. Stability, convergence and speed. We may compare the segmentation results using ν to Appleton-Talbot's result using P. We recall that AT-CMF solves the partial differential equation system (2.3) in order to solve the continuous maximum flow problem (2.1), but no proof was given for convergence. The potential function P approximates an indicator function, with 0 values for the background labels, and 1 for the foreground. It can be difficult to know when to stop the AT-CMF algorithm, since the iterations used to solve the AT-CMF algorithm may oscillate, as displayed in Figure 4.4 on a synthetic image. In contrast, the CCMF algorithm is guaranteed to converge and smoothly approaches the optimum solution.

We also compare segmentation results with results obtained by the recent algorithm of Chambolle and Pock [21] optimizing a total variation based energy using a dual variable. In comparison to our work, the dual variable is not a flow (no constraint on the divergence and no proof of convergence toward a minimal surface as defined in [15] or [10]). In the literature, the continuous-domain duality between TV and max-flow is assumed, but in computational practice they are really different. Nevertheless, even as the method of Chambolle-Pock is solving a different problem, we performed numerical comparative tests. The problem of Chambolle and Pock (now denoted CP-TV) for binary segmentation optimizes

$$\min_{u\in[0,1]^n} \max_{F\in\mathbb{R}^m, ||F||_\infty \leq 1} F^T(Au) + g^T u + \text{ hard constraints attachment}, \tag{4.1}$$



(a) Input seeds (b) Input images (c) CCMF energy vs true perimeters



GC cut cost : 16 GC cut cost : 20

(d) Graph cut cost for lines of same length (e) GC energy vs true perimeters

FIG. 4.3. *Comparison of perimeters computed analytically and with Graph cuts and CCMF. To estimate the boundary length with CCMF, segmentations have been produced using the input images in (b) with the foreground and background seeds that appears in (a). (c) : plot showing the linear relation between the true perimeters and the energies obtained with CCMF. The symbols used to represent the dots correspond to the objects segmented in the images. A similar relation of proportionality is obtained with CP-TV. (d) : value of the cut obtained for a diagonal and a vertical line of same length using graph cuts. The difference observed explains (e) : plot showing the nonlinear relation between the true perimeters and the energies obtained with Graph cuts.*

where $A \in \mathbb{R}^{m\times n}$ is the incidence matrix of the graph of n nodes and m edges defined on the image, $g \in \mathbb{R}^n$ is the metric on the nodes, defined in equation (2.6).

The CCMF algorithm is faster than both AT-CMF and CP-TV for 2D image segmentation. We have implemented CCMF in Matlab. We used an implementation of Appleton-Talbot's algorithm in `C++` provided by the authors, and a Matlab software implementing CP-TV on CPU, also provided by the respective authors. The three implementations make use of multi-threaded parallelization, and the CPU times reported here were computed on a Intel Core 2 Duo (CPU 3.00GHz) processor, with 2 Gb of RAM. The CCMF average computation time on a 321×481 image is 181 seconds after 21 iterations. For AT-CMF, 80000 iterations require 547 seconds. For CP-TV, the average number of iterations needed for convergence is 36000, requiring an average time of 1961 seconds on CPU, and the median computation time is 1406 seconds. Note that a GPU implementation of CP-TV, also provided by the authors, achieves a speed gain factor of about 100 on a NVidia Quadro 3700. Although it was not pursued here, it should be noted that the CCMF optimization approach could also fully benefit from parallelization (e.g., on a GPU) because the core computation is to solve a linear system of equations. Realizing the benefit of a GPU

solution would require the use of an iterative algorithm such as conjugate gradients or multigrid [43, 44], which would make comparisons to the direct solver used here less immediate. However, previous examples in the literature have suggested that this change in solver has not created difficulties in order to realize the benefits of a GPU architecture [45].

Sometimes, it may not always be necessary to wait so long for the complete convergence of CCMF (or AT-CMF) to obtain acceptable results. In cases where images exhibiting sufficiently strong gradients, one iteration may be enough to obtain a satisfying segmentation. On such an image, one iteration of CCMF, and 100 iterations of AT-CMF show acceptable approximate results reached only after about 2 seconds for either algorithm. However, one cannot always rely on strong edges, and the power of our CCMF formulation is to behave in a predictable fashion even when edges are weak.

We may also compare the computation time of CCMF to the optimization of total variation using the Split Bregman method [26]. Optimizing TV on a 100×100 image requires 5000 iterations and takes 23 seconds with a sequential implementation of Split Bregman in C. On the same image, CCMF required only 17 iterations to reach the convergence criteria $||r_d|| < 1$ and $||\widehat{\eta}|| < 2$, taking 4.7 seconds with a Matlab implementation (using a 2-threaded solver). We conclude that TV-based methods appear to be quite slow in the context of image segmentation, although we employed the Split Bregman algorithm which is known as one of the fastest algorithms for TV optimization for image denoising, and the very recent CP-TV algorithm. This difference in speed between the denoising and segmentation applications is due to the very large number of iterations required to convergence to a binary segmentation.

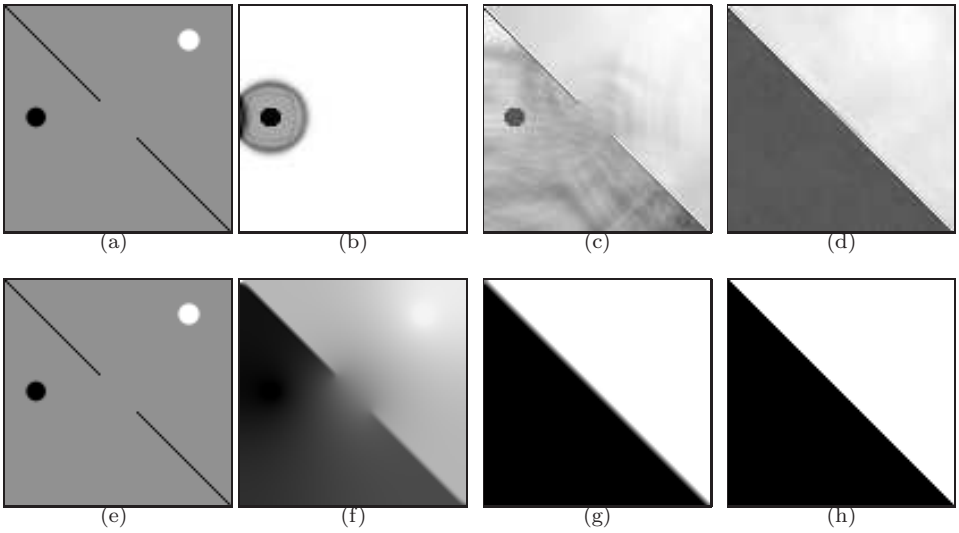


FIG. 4.4. *Segmentation of an artificial image with AT-CMF (top row) and CCMF (bottom row). Top row (AT-CMF): (a) Image where the black and white discs are seeds. AT-CMF result stopped in (b) after 100 iterations, (c) 1000 iterations, (d) 10000 iterations. Bottom row (CCMF): (e) Image where the black and white discs are seeds, CCMF result ν after (f) 1 iterations, (g) ν after 15 iterations iterations and (h) threshold of the final ν.*

	Dice Coeff.	GC	AT-CMF	CP-TV	CCMF
First set of seeds	mean	95.2	94.9	95.2	95.3
	median	96.2	96.1	96.3	96.3
	stand. dev.	4.3	4.1	4.1	4.1
Second set of seeds	mean	89.3	89.7	89.1	89.5
	median	91.2	92.0	91.7	92.3
	stand. dev.	8.4	7.7	8.5	8.6

TABLE 4.1

Dice coefficient (percentage) computed between the segmentation mask and the ground truth image (provided by GrabCut database). The lines above the double bar show results for the first set of seeds, and the lines below the double bar show results obtained with the second set of seeds.

4.3. Segmentation quality. In this experiment, we compared our CCMF formulation to the AT-CMF formulation and conventional graph cuts on the problem of image segmentation, to determine if there were strong performance differences between the formulations. We expect that there would not be, since in principle all three formulations are trying to "minimize the cut", but have slightly different definitions of the cut length. The primary advantage that we expect from our formulation is in the reduction of metrication artifacts (as compared to conventional graph cuts) and speed/convergence (as compared to AT-CMF).

Our experiment consists of testing the Graph Cut, AT-CMF, CP-TV, and Combinatorial Continuous Max-Flow algorithms on a database with the same seeds. We used the Microsoft 'GrabCut' database available online [48], which is composed of fifty images provided with seeds. From the seed images provided, it is possible to extract two different possible markers for the background seed. Examples of such seeds are shown in Fig. 4.5(a). Different convergence criteria are available for CCMF, such as the duality gap and norms of the residuals. However, for the CMF algorithm, we do not have any satisfying criteria. Bounding the number of non binary occurrences of P does not mean that the convergence is reached, because of possible oscillations. An intermediate result after ten thousand iterations may be significantly different from the result reached after one hundred thousand. Consequently, we have run the AT-CMF algorithm until we were convinced to have reached convergence, *i.e.* when P was nearly binary and did not change significantly when we doubled the number of iterations. For half of the images in the GrabCut database, AT-CMF algorithm required more than 20000 iterations to reach convergence, for a third of the images, more than 80000 iterations, and for 1/4 of the images, more than 160000 iterations. Binary convergence was still not reached after even 500000 iterations for the rest of the images (1/4), but we stopped the computation anyway. The Chambolle-Pock TV based algorithm for binary segmentation is provably convergent and benefits also from several possible convergence criteria. The convergence criteria used is a ratio of changed pixels from two successive intervals of iterations being smaller than an epsilon that we fixed to 10^{-7} in order to obtain satisfying results. In practice, 1/4 of the images converged in less than 60000 iterations and for the rest of them we increased the maximum number of iterations up to 200000. In contrast, CCMF only needed 21 iterations on average, and never more than 27, to reach the convergence criteria $||r_d|| < 1$ and $||\widehat{\eta}|| < 2$. We noted that for all methods, segmentation quality degraded quickly if these conditions were not respected.

Figure 4.3 displays the performance results for these algorithms. The Dice coefficient is a similarity measure between sets (segmentation and ground truth), ranging

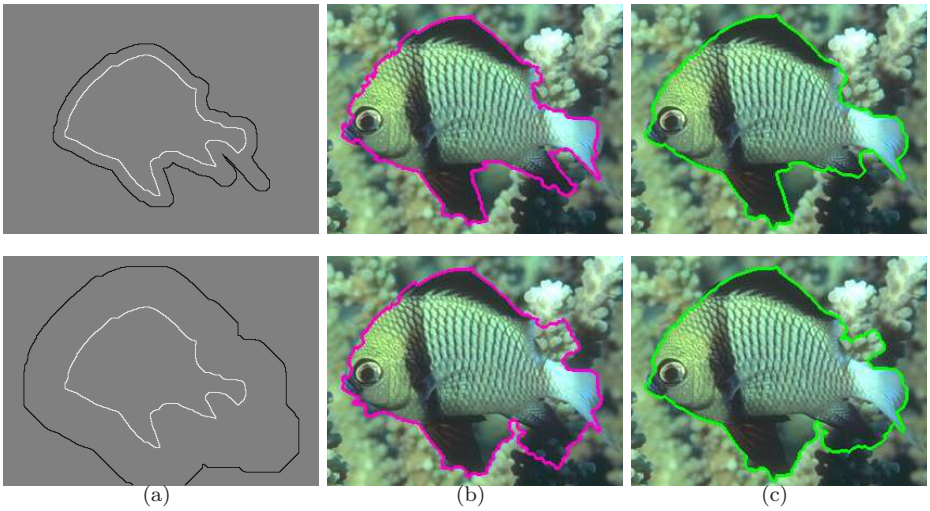


FIG. 4.5. *(a) : Seed images belonging to the first (top) and second (bottom) set of seeds provided by the GrabCut database. (b) Segmentations by Graph Cuts computed with seeds from the two sets, (c) Segmentations by CCMF.*

from 0 to 1.0 for no match and a perfect match respectively. All the tested algorithms show very good results, with a Dice coefficient of 0.95–0.97 for the well positioned seeds, and 0.89–0.92 for the second set of seeds, further away from the objects. The CCMF and AT-CMF results are really close, and the mean is better than the GC and the CP-TV results.

4.4. Extensions. The primary focus of our experiments were to demonstrate that our CCMF algorithm achieves a solution which does not exhibit metrication artifacts, that it is fast with provable convergence and that the segmentation quality is high. In the remainder of this section, we show that the algorithm can also incorporate unary terms and be equally applied in 3D. In fact, since CCMF is defined for an arbitrary graph, it could even be used to perform clustering in this more generalized framework. However, the benefit of avoiding gridding artifacts is less meaningful when performing clustering on an arbitrary graph, and therefore we omit any examples of this nature.

4.4.1. Unary terms. A simple modification of the transport graph G permits the use of unary terms to automatically specify objects to be segmented. This approach is similar to the use of unary terms in the graph cuts computation [8]. The placement of unary terms for adding data attachment constraints in the max-flow problem is performed by adding edges linking every node of the lattice to the source and to the sink. In the case of the CCMF problem, as weights are defined on the nodes, we need to add intermediary nodes between the grid and source on the one hand, and the grid and sink on the other. However, since these intermediary nodes have just two edges incident on them, these node weights are equivalent to edge weights, meaning that our construction is equivalent to the use of unary terms for AT-CMF that was pursued by Unger *et al.* [17]. In our case, we used weighted intermediary nodes simply to keep the CCMF framework consistent. Considering that the original lattice is composed of n nodes, we add for each node an "upper" node linked to the source and a "lower" node linked to the sink. An example of this construction is shown in

Fig. 4.6.

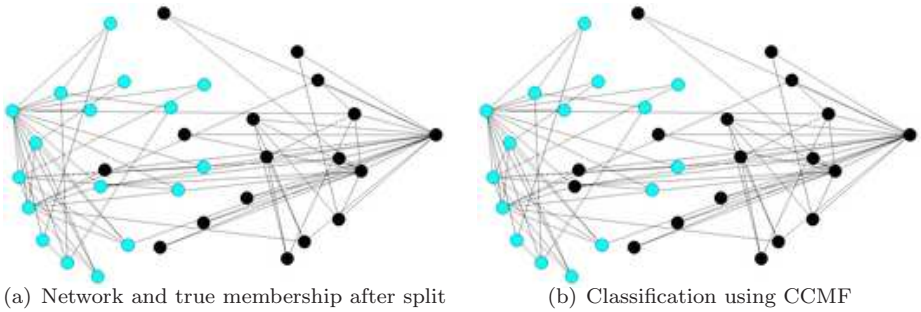


FIG. 4.6. *Example of graph construction to incorporate unary terms for unsupervised segmentation. Each node is connected to a source node and sink node through an intermediary node which is weighted to reflect the strength of the positive and negative unary term. This construction is then applied with a simple appearance prior to automatically segment two images.*

The weights of the additional nodes may be set to reflect the node priors for a particular application. For image segmentation, given mean background and foreground color BC and FC, we can set the capacities of the background nodes to $g_i = \exp(-\beta(BC_i - I_i)^2)$ and foreground nodes to $g_i = \exp(-\beta(FC_i - I_i)^2)$. Examples of results using these appearance priors are shown on Fig. 4.6.

4.4.2. Classification. As the general CCMF formulation is defined on arbitrary graphs, it is possible to employ it in tasks beyond image segmentation. For instance CCMF may be employed to solve general problems of graph clustering, even on networks with no meaningful embedding such as social networks. We show a possible application in a classification example, considering the social network studied by Zachary [49]. After the split of a university karate club due to a conflict between its two leaders, Zachary's goal was to study if it was possible to predict the side joined by the members, based only on the social structure of the club. Classically, the graph is built by associating each member to a node, and edges link two members when special affinities are known between them. As weights are associated here to the strength of coupling between members, different strategies are possible for assigning the weights onto the nodes, which is necessary for using CCMF. For example, a given node/member may be assigned by the mean of affinity with its neighboring members in the graph. This weighting strategy works well in this example (See Fig. 4.7). Another strategy would be to compute the solution in the dual graph, that is in the graph where nodes are replaced by edges and (weighted) edges are replaced by (weighted) nodes. We have shown in an example the ability of CCMF for classification, a task that can not be treated with finite element-based methods such as AT-CMF. Evaluating the performances of CCMF for classification may be a topic for future research.

4.4.3. 3D segmentation. For 3D image segmentation, the minimal surface properties of CCMF generate good quality results, as shown in Fig 4.8. The CCMF formulation applies equally well in 2D or 3D, since CCMF is formulated on an arbitrary graph (which may be a 2D lattice, a 3D lattice or an even more abstract graph). In 3D, our CCMF implementation is suffering from memory limitations in the direct



(a) Network and true membership after split (b) Classification using CCMF

FIG. 4.7. *Zachary's karate club network. The two leaders in conflict are represented by the left hand side and right hand side nodes. Just one node is misclassified with CCMF. However, this node is known classically to be an unusual situation within the social network (see [49]) which is not captured by any known algorithm.*

solver we used, limiting its performances. Future work will address this issue using a dedicated solver.

5. Conclusion. In this paper, we have presented a new combinatorial formulation of the continuous maximum flow problem and a solution using an interior point convex primal-dual algorithm. This formulation allows us to optimize the maximum flow problem as well as its dual for which we provide an interpretation as a minimal surface problem. This new combinatorial expression of continuous max-flow avoids blockiness artifacts compared to graph cuts. Furthermore, the formulation of CMF on a graph reveals that it is actually the fact that the capacity constraints are applied to point-wise norms of flow *through nodes* that allows us to avoid metrication errors, as opposed to the conventional graph cut capacity constraint *through individual edges*. Additionally, it was shown that our CCMF formulation provides a better approximation to the analytic catenoid than the conventional AT-CMF discretization of the continuous max-flow problem. Contrary to graph cuts, we showed that CCMF estimates the true boundary length of objects. Finally, unlike finite-element based methods such as AT-CMF, the CCMF formulation is expressed for arbitrary graphs, and thus can be employed in a large variety of tasks such as classification.

We provide in this paper an exact analytic expression of the dual problem, convergence of the algorithm being guaranteed by the convexity of the problem. In terms of speed, when an approximate solution is sufficient, our implementation of CCMF in Matlab is competitive to the Appleton-Talbot approach, which uses a system of PDEs, and a `C++` implementation. The Appleton-Talbot algorithm has the significant drawback of not providing a criterion for convergence. In practice, this translates to long computation times when convergence is difficult for the AT-CMF algorithm. In contrast, our algorithm in this case is much faster, as we observe that the number of iterations required for convergence does not vary much for CCMF (less than a factor of two). The CCMF algorithm is simple to implement, and may be applied to arbitrary graphs. Furthermore, it is straightforward to add unary terms to perform unsupervised segmentation.

We have also compared our algorithm to known weighted combinatorial TV minimization methods (CTV optimized with split-Bregman, CP-TV). We have shown that our results are generally better for segmentation than combinatorial TV-based methods, and that our implementation on CPU is much faster and more predictable.

Specifically, the number of iterations required for convergence does not vary much for CCMF, whereas it can vary of more than a factor of ten for CP-TV. The deep study of the relationships between CCMF and combinatorial TV reveals that, in contrast with expectations from their duality under restrictive assumptions in the continuous domain, their duality relationship is only weak in the combinatorial setting. In fact max-flow and total variation are different problems with different constraints, yielding different algorithms and different results. One key difference between max-flow and total variation is that max-flow algorithms were developed as segmentation algorithms, following the graph cuts framework. One consequence is that max-flow formulations have a null divergence objective, which is not present in total variation formulations. Null divergence can be viewed as a consequence or a necessity in order to obtain constant partitions almost everywhere, in particular binary ones. This difference is important because in the proposed CCMF framework we *impose* a tight null divergence constraint, which to our knowledge is novel for an isotropic formulation (graph-cuts have always had this constraint). AT-CMF and similar frameworks achieve null divergence if and when convergence is achieved. There is no null divergence constraint at all in total variation frameworks. As a consequence, while in practice it is possible to compare total variation and max flow formulations, they should be treated differently. However, the strong computational performance and segmentation quality results of CCMF as compared to TV (using the strongest known optimizations methods) suggests that it may be advantageous to apply our CCMF formulation to problems for which TV has proven effective (such as filtering).

Several further optimizations of CCMFs are possible, for instance using multigrid implementations, the possibility to use GPU to solve the iterative linear system, the use of a dedicated solver for the particular sparse linear system involved in the computation. We also plan to compare the efficiency obtained with the interior point method to a first order algorithm for solving the CCMF problem.

Ultimately, we hope to employ combinatorial continuous maximum flow as a powerful segmentation algorithm which avoids metrication artifacts and provides a fast solution with provable convergence. Furthermore, we intend to explore the potential of CCMF to optimize other energy functions for which graph cuts or total variation have proved useful, such as surface reconstruction or efficient convex filtering.

REFERENCES

[1] Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. IJCV **1** (1988) 321–331

[2] Sethian, J.A.: Level Set Methods and Fast Marching Methods. Cambridge University Press (1999)

[3] Paragios, N.: Geodesic Active Regions and Level Set Methods : Contributions and Applications in Artificial Vision. PhD thesis, INRIA Sophia Antipolis (2000)

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

[5] Ishikawa, H.: Exact optimization for Markov random fields with convex priors. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 1333–1336

[6] Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation part ii: Levelable functions, convex priors and non-convex cases. J. of Math. Imaging and Vis. **26** (2006) 277–291

[7] Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Canadian Journal of Mathematics **8** (1956) 399–404

[8] Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. ICCV'01 **1** (2001) 105–112 vol.1

[9] Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? PAMI **26** (2004) 147–159

[10] Appleton, B., Talbot, H.: Globally minimal surfaces by continuous maximal flows. PAMI **28** (2006) 106–118
[11] Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. ICCV'03 (2003) 26–33
[12] Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20** (2004) 89–97
[13] Nikolova, M., Esedoglu, S., Chan, T.F.: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM JAM **66** (2006) 1632–1648
[14] Iri, M.: Theory of flows in continua as approximation to flows in networks. Survey of Mathematical Programming (1979)
[15] Strang, G.: Maximum flows through a domain. Math. Prog. (1983) 123–143
[16] Unger, M., Pock, T., Trobin, W., Cremers, D., Bischof, H.: TVSeg - Interactive total variation based image segmentation. In: BMVC. (2008)
[17] Unger, M., Pock, T., Bischof, H.: Interactive globally optimal image segmentation. Technical Report ICG-TR-08/02, Graz University of Technology, Gratz, Austria (2008)
[18] Pock, T., Schoenemann, T., Graber, G., Bischof, H., Cremers, D.: A convex formulation of continuous multi-label problems. In: ECCV '08. (2008) 792–805
[19] Zach, C., Gallup, D., Frahm, J.M., Niethammer, M.: Fast global labeling for real-time stereo using multiple plane sweeps. In: Vis. Mod. Vis. (2008)
[20] Chambolle, A., Cremers, D., Pock, T.: A convex approach for computing minimal partitions. Tech. Rep. 649, Ecole Polytechnique CMAP (2008)
[21] Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. CMAP-685 (2010)
[22] Strang, G.: Maximum flows and minimum cuts in the plane. Journal of Global Optimization (2008)
[23] D. Shulman, J.H.: Regularization of discontinuous flow fields. Proceedings Workshop on Visual Motion (1989) 81–86
[24] Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D **60** (1992) 259–268
[25] Nesterov, Y.: Gradient methods for minimizing composite objective function. CORE Discussion Papers 2007076, Universit catholique de Louvain, Center for Operations Research and Econometrics (CORE) (2007)
[26] Goldstein, T., Osher, S.: The split bregman method for l1-regularized problems. SIIMS **2** (2009) 323–343
[27] Zhang, X., Burger, M., Bresson, X., Osher, S.: Bregmanized nonlocal regularization for deconvolution and sparse reconstruction (2009) [Submitted].
[28] Nozawa, R.: Max-flow min-cut theorem in an anisotropic network. Osaka J. Math **27** (1990) 805–842
[29] Nozawa, R.: Examples of max-flow and min-cut problems with duality gaps in continuous networks. Mathematical Programming **63** (1994) 213–234 10.1007/BF01582067.
[30] Fefferman, C.L.: Existence and smoothness of the Navier-Stokes equation (2000) http://www.claymath.org/millennium/Navier-Stokes_Equations/navierstokes.pdf.
[31] Hirani, A.N.: Discrete exterior calculus. PhD thesis, Cal. Tech (2003)
[32] Grady, L.J., Polimeni, J.R.: Discrete Calculus: Applied Analysis on Graphs for Computational Science. Springer (2010)
[33] Elmoataz, A., Lézoray, O., Bougleux, S.: Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. IEEE Trans. on Image Processing **17** (2008) 1047–1060
[34] Grady, L., Alvino, C.: The piecewise smooth Mumford-Shah functional on an arbitrary graph. IEEE Trans. on Image Proc. **18** (2009) 2547–2561
[35] Lippert, R.A.: Discrete approximations to continuum optimal flow problems. Studies in Applied Mathematics **117** (2006) 321–333
[36] Bhusnurmath, A., Taylor, C.J.: Graph cuts via ℓ_1 norm minimization. IEEE Transactions on Pattern Analysis and Machine Intelligence **30** (2008) 1866–1871
[37] Goldfarb, D., Yin, W.: Second-order cone programming based methods for total variation image restoration. SIAM J. on Scientific Computing **27** (2005) 622–645
[38] Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. CVPR 2005 **2** (2005) 60–65
[39] Grady, L., Schwartz, E.L.: Faster graph-theoretic image processing via small-world and quadtree topologies. In: Proc. of CVPR. Volume 2., IEEE (2004) 360–365
[40] Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. IJCV **22** (1997) 61–79
[41] Goldberg, A.V., Tarjan, R.E.: A new approach to the maximum flow problem. STOC '86,

Theory of computing (1986) 136–146
[42] Boyd, S., Vandenberghe, L.: Convex Optimization. CUP (2004)
[43] Bolz, J., Farmer, I., Grinspun, E., Schröder, P.: Sparse matrix solvers on the GPU: Conjugate gradients and multigrid. In: ACM Transactions on Graphics. Volume 22 of SIGGRAPH. (2003) 917–924
[44] Kruger, J., Westermann, R.: Linear algebra operators for GPU implementation of numerical algorithms. In: SIGGRAPH. (2003) 908–916
[45] Grady, L., Schiwetz, T., Aharon, S., Westermann, R.: Random walks for interactive organ segmentation in two and three dimensions: Implementation and validation. In: Proc. In'l Conf. Medical Image Computing and Computer-Assisted Intervention. (2005) 773–780
[46] Gilboa, G., Osher, S.: Nonlocal linear image regularization and supervised segmentation. Multiscale Modeling and Simulation **6** (2007) 595–630
[47] Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the Mumford-Shah functional. In: Int. Conf. Comp. Vis. (2009)
[48] Rother, C., Kolmogorov, V., Blake, A.: "GrabCut" — Interactive foreground extraction using iterated graph cuts. In: SIGGRAPH. (2004) 309–314
[49] Zachary, W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research **33** (1977) 452–473

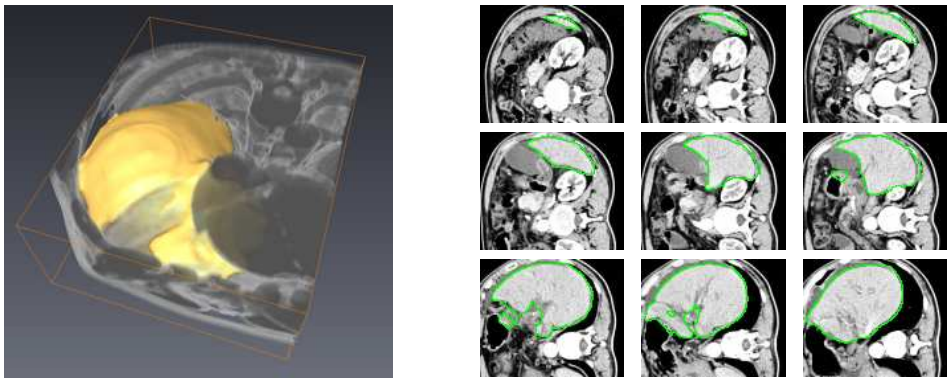


FIG. 4.8. *Liver segmentation in 3D by CCMF.*