

2-17-2017

Directed Acyclic Graph Continuous Max-Flow Image Segmentation for Unconstrained Label Orderings

John SH Baxter

Robarts Research Institute, jbxter@robarts.ca

Martin Rajchl

Imperial College London

A. Jonathan McLeod

Robarts Research Institute

Jing Yuan

Robarts Research Institute

Terry M. Peters

Robarts Research Institute, tpeters2@uwo.ca

Follow this and additional works at: <https://ir.lib.uwo.ca/robartspub>



Part of the [Artificial Intelligence and Robotics Commons](#), and the [Bioimaging and Biomedical Optics Commons](#)

Citation of this paper:

Baxter, John SH; Rajchl, Martin; McLeod, A. Jonathan; Yuan, Jing; and Peters, Terry M., "Directed Acyclic Graph Continuous Max-Flow Image Segmentation for Unconstrained Label Orderings" (2017). *Robarts Imaging Publications*. 21.

<https://ir.lib.uwo.ca/robartspub/21>

Directed Acyclic Graph Continuous Max-Flow Image Segmentation for Unconstrained Label Orderings

John S.H. Baxter · Martin Rajchl · A. Jonathan McLeod · Jing Yuan · Terry M. Peters

Received: date / Accepted: date

Abstract Label ordering, the specification of subset-superset relationships for segmentation labels, has been of increasing interest in image segmentation as they allow for complex regions to be represented as a collection of simple parts. Recent advances in continuous max-flow segmentation have widely expanded the possible label orderings from binary background/foreground problems to extendable frameworks in which the label ordering can be specified. This article presents Directed Acyclic Graph Max-Flow (DAGMF) image segmentation which is flexible enough to incorporate any label ordering without constraints. This framework uses augmented Lagrangian multipliers and primal-dual optimization to develop a highly parallelized solver implemented using GPGPU. This framework is validated on synthetic, natural, and medical images illustrating its general applicability.

Keywords continuous max-flow · image segmentation · convex optimization · variational optimization · ASETS

1 Introduction

Techniques in Markov Random Field estimation and solution have been immensely useful in the field of image analysis, specially image segmentation. Pseudo-boolean programming [7,27] and fast optimization techniques such as graph-cuts [8] have allowed for exact solutions to Markov Random Field models with submodular energies [6,7,30] giving them increased applicability. Binary graph-cuts have also been the basis for algorithms

such as α -expansion and α - β swap [9] which approximate solutions to NP-hard (non-sub-modular) multi-region segmentation problems [30].

Of particular interest are *extendable* algorithms, that is, a single algorithm or algorithmic framework which can address problems with an arbitrarily high number of labels, rather than a fixed number, and frameworks that address different *label orderings*, i.e. constraints on the labels in terms of subset-superset relationships. Algorithms such as α -expansion and α - β swap [9] have provided a graph-cut based framework for addressing the Potts model [33]. In terms of label orderings, this model constrains the labels to be disjoint and form a valid partition of the image. Ishikawa et al. [26] developed a framework for minimizing energies with multiple linearly ordered labels. Delong et al. [16] extended this to models composed of sub-modular inclusion terms and super-modular exclusion terms. Both frameworks maintain a sub-modular energy, allowing them to achieve global optimality. Delong et al. [17] also presented an extendable framework for hierarchical label orderings using α -expansion as a subroutine. These frameworks have *constrained* label orderings in that no framework can optimize a problem with arbitrary subset-superset relationships.

Parallel to discrete graph-cut models, continuous analogs have also gained increasing attention, using variational optimization to describe the image space not as a discrete collection of pixels but as a bounded continuum of locations. These have been shown to reduce metrication artifacts associated with graph-cut segmentation. [32,41,42] Primal-dual optimization provided an efficient framework for solving these models iteratively [11,41]. Extendable max-flow models, ones which handle an arbitrary number of labels, analogous to the above have been proposed and are described with more

J.S.H. Baxter
Robarts Research Institute, Western University, London
E-mail: jbaxter@robarts.ca

	Continuous or Discrete?	Possible label orderings	Approach	Optimality?
Billionnet & Minoux [6]	Discrete	Submodular functions	Single graph cut	Global binary
Boykov et al. [9]	Discrete	Potts model	α -expansion proposed α - β -swap proposed	Approximate binary
Ishikawa et al. [26]	Discrete	Linear ordering (Ishikawa model)	Single graph cut	Global binary
Kolmogorov & Zabih [30]	Discrete	Submodular functions	Single graph cut	Global binary
Schlesinger & Flach [37]	Discrete	Linear ordering	Single graph cut	Global binary
Pock et al. [32]	Continuous	Potts model	Primal-dual optimization Projected gradient descent	Global fuzzy
DeLong et al. [16]	Discrete	Sub/supermodular models (subset of partial orderings)	Single graph cut	Global binary
Yuan et al. [42]	Continuous	Potts model	Primal-dual optimization Augmented Lagrangian	Global fuzzy
DeLong et al. [17]	Discrete	Hierarchical models (subset of partial orderings)	α -expansion-based	Approximate binary
Bae et al. [2]	Continuous	Ishikawa model (linear ordering)	Primal-dual optimization Augmented Lagrangian	Global fuzzy & Global binary
Baxter et al. [3,34]	Continuous	Hierarchical models (subset of partial orderings)	Primal-dual optimization Augmented Lagrangian	Global fuzzy

Table 1: Extendable Max-Flow/Min-Cut Segmentation Frameworks

detail in Section 2. In these models, multiple labels are represented in separate continuous spaces connected in such a manner as to encode topological information.

Contributions: This article describes a continuous max-flow framework that allows for labels to be organized as a set of continuous spaces linked as a directed acyclic graph (DAG), which is a strict generalization of previous, more constrained algorithms [3, 34, 42]. This article also derives two solution algorithms for this framework derived from primal-dual optimization: the first based on augmented Lagrangian optimization, and the second on proximal Bregman projections. This framework is sufficiently powerful to represent any arbitrary collection of subset-superset relationships and thus can be considered to optimize *unconstrained* label orderings.

2 Previous Extendable Max-Flow Approaches

Previous approaches to continuous max-flow segmentation with an arbitrary number of labels include those with a rigidly defined topology and those where the topology can be more flexible. Both these framework and extendable discrete frameworks are summarized in Table 1. (*Continuous or Discrete?* refers to whether the problem theoretically considers the image domain to be a continuum or a discrete collection of voxels. *Optimality?* refers to whether the framework guarantees a globally optimal solution or an approximate one,

and whether that guarantee applies to a binary labeling problem or fuzzy labeling problem.)

The notation for each of the methods have been rewritten to emphasize the similarities between the different frameworks. Specifically, in each method, L represents a label or object in the image, with S as the *source label* representing the entire domain of the image Ω . The function $u_L(x)$ is the labeling function, that is, $u_L(x) \approx 1$ means that location $x \in \Omega$ is within object L and $u_L(x) \approx 0$ means that location $x \in \Omega$ is not within object L . $D_L(x)$ and $R_L(x)$ are the *data* terms and *regularization* or *smoothness* terms, respectively, which correspond to costs minimized by the optimization process. We will also use the notation \mathbb{L} to refer to the labels forming the partition of the image domain Ω .

Rigidly-Defined Topology:

The continuous Potts model [32, 42] was the first continuous max-flow model which allowed for the incorporation of more labels than the standard foreground/background and minimizes the segmentation energy:

$$\begin{aligned}
 E(u) &= \sum_{\forall L} \int_{\Omega} (D_L(x)u_L(x) + R_L(x)|\nabla u_L(x)|) dx \\
 \text{s.t.} \quad &\sum_{\forall L \in \mathbb{L}} u_L(x) = u_S(x) = 1 \\
 &u_L(x) \geq 0
 \end{aligned} \tag{1}$$

In this model, the only notion of topology is that all labels are disjoint (the only labels present are the source

label, S , and the labels forming the partition \mathbb{L}); there is no regularization considerations for grouping labels. The fuzzy segmentation problem (where $u_L(x) \in [0, 1]$) can be solved with global optimality for these problems, but the binary equivalent (where $u_L(x) \in \{0, 1\}$) is known to be only approximately optimal.

The continuous analogue to the Ishikawa model [2] took label topology into account by introducing the notion of a more complex label ordering. However, in the Ishikawa model, the only label orderings supported are linear orderings, that is, each label is either a subset or a superset of another label; no two labels can be disjoint as in the Potts model. This model minimizes the energy:

$$E(u) = \sum_{\forall L_i} \int_{\Omega} (D_{L_i}(x)(u_{L_i}(x) - u_{L_{i+1}}(x)) + R_{L_i}(x)|\nabla u_{L_i}(x)|) dx \quad (2)$$

s.t. $u_{L_0}(x) = 1$
 $u_{L_i}(x) \geq u_{L_{i+1}}(x)$

In these models, L_0 plays the role of S , and the partition is $\mathbb{L} = \{L_0 \setminus L_1, L_1 \setminus L_2, \dots, L_{N-1} \setminus L_N, L_N\}$. Similar to the continuous Potts model, the fuzzy segmentation problem (where $u_{L_i}(x) \in [0, 1]$) can be solved with global optimality. The binary equivalent (where $u_L(x) \in \{0, 1\}$) can also be optimally solved via a thresholding system on the fuzzy label values. [2]

Neither of these models allows for the label ordering itself to be specified. The Potts model only allows for completely unordered labels and the Ishikawa model only allows for fully ordered labels. Recent approaches in partially ordered labels in continuous max-flow image segmentation [35] illustrated that relaxing these constraints on extendable models could have practical significance, using a flexible label ordering to encode knowledge about the scene composition.

Flexible Topology:

Hierarchical continuous max-flow image segmentation (HMF) [3, 34] allowed for non-linear label orderings to be defined with one stipulation, that the label ordering was hierarchical. The practical implication is that two labels could either be entirely disjoint as in the Potts model or one is a superset of the other as in the Ishikawa model. This model minimizes the energy:

$$E(u) = \sum_{\forall L} \int_{\Omega} (D_L(x)u_L(x) + R_L(x)|\nabla u_L(x)|) dx$$

s.t. $u_S(x) = 1$

$$\sum_{L' \in L.C} u_{L'}(x) = u_L(x) \text{ if } L.C \neq \emptyset$$

$$u_L(x) \geq 0 \quad (3)$$

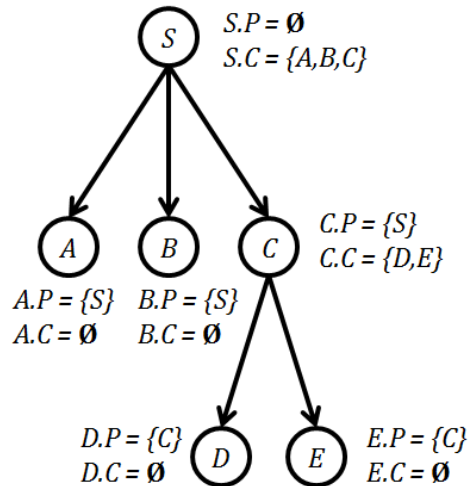


Fig. 1: Example label ordering diagram with four leaf-labels ($\mathbb{L} = \{A, B, D, E\}$) with parent and child operators, $L.P$ and $L.C$ respectively, explicitly written. This label ordering is suitable for HMF as it is a valid hierarchy; each label has exactly one parent except the source label S , representing the entire image, which has none.

In these models, each label L is either a leaf-label or is recursively partitioned into a set of other, simpler labels $L.C$. The partition of the image is therefore $\mathbb{L} = \{L | L.C = \emptyset\}$. We will use the same notation as Baxter et al. [3] in that $L.C$ refers to the set of *children* of L and $L.P$ refers to the set of *parents* of L . The hierarchical constraint is that this label ordering can be expressed as a tree rooted at a node representing label S . (That is, for every label $L \neq S$, $L.P$ consists of a single element.) We will also use the star notation, $L.C^*$ to denote the *descendants* of L , that is, $L.C \cup \{L'.C | L' \in L.C\} \cup \{L''.C | \exists L' \in L.C (L'' \in L'.C)\} \cup \dots$. Similarly, $L.P^*$ denotes the *ancestors* of L , that is $L.P \cup \{L'.P | L' \in L.P\} \cup \{L''.P | \exists L' \in L.P (L'' \in L'.P)\} \cup \dots$

The goal of HMF solvers [3, 34] was to have a single unified framework that solved any of these models, rather than specifying a particular model. This is in contrast to earlier work in hierarchical max-flow label ordering in which the particular label ordering (and thus the number of labels) was fixed. [35] The solvers guarantee global optimality for fuzzy labels where $u_L(x) \in [0, 1]$, but as this framework is a strict generalization of the Potts model, it cannot in general guarantee an globally optimal binary labeling where $u_L(x) \in \{0, 1\}$.

The ultimate purpose of this article is to remove the constraint that $L.P$ consists of a single element, developing a solution algorithm suitable for any possible directed acyclic graph. By allowing a label to have multiple parents, regions can be grouped in a way that

is distinctly non-hierarchical, such as areas of overlap between two distinct regions, addressing a number of useful but previously insoluble label orderings such as the Venn diagram model in Section 6.1. Additionally, Section 5 demonstrates that DAGs are sufficiently powerful to represent *any* label ordering, that is, to apply a regularization term to any arbitrary grouping of labels.

3 Continuous Max-Flow Model

The Directed Acyclic Graph Max-Flow (DAGMF) segmentation model this article addresses is the minimization of the convex energy functional:

$$\begin{aligned} E(u) &= \sum_{\forall L} \int_{\Omega} (D_L(x)u_L(x) + R_L(x)|\nabla u_L(x)|) dx \\ \text{s.t. } u_S(x) &= 1 \\ \sum_{L' \in L.C} w_{(L',L)} u_{L'}(x) &= u_L(x) \text{ if } L \notin \mathbb{L} \\ u_L(x) &\geq 0 \end{aligned} \quad (4)$$

Similar to HMF, the partition $\mathbb{L} = \{L | L.C = \emptyset\}$ is the set of leaf-labels or labels with no children. Unlike HMF, $L.P$ does not have to consist of a single element, but each label can have multiple parents. The non-negative weight terms $w_{(L',L)}$ determine what weight to give the labeling function of L when calculating the labeling function of its parent label, L' . These terms must sum to 1 over the parents of each label, that is:

$$\sum_{\forall L' \in L.P} w_{(L',L)} = 1 \quad (5)$$

which will guarantee a valid partition of Ω into the leaf-labels, that is, $\sum_{\forall L \in \mathbb{L}} u_L(x) = 1$ (Shown in Sec. 3.3). It is clear that HMF is a subset of these models as $L.P$ having a single element implies $w_{(L',L)} = 1$ showing that equation (3) is a special case of equation (4). We can assume, without loss of generality, that $D_L(x) = 0$ whenever $L \notin \mathbb{L}$. This is achievable by pushing down the data terms to the labels children, i.e. $D_L(x)u_L(x) = \sum_{L' \in L.C} D_L(x)w_{(L',L)}u_{L'}(x)$.

The modeling approach is derived from those presented by Yuan et al.[41][42] and follows the same format, showing the duality of a max-flow primal formation to this minimization problem through an intermediate primal-dual optimization problem. An augmented Lagrangian framework and a proximal Bregman framework are proposed for minimizing this intermediate representation.

3.1 Primal Formulation

The primal model represents network flow maximization through a large graph with only the sink flows constrained. The dual of this formulation is the DAGMF equation (4) as we shall prove in the following section. We can write the primal model as:

$$\begin{aligned} \max_{p,q} \int_{\Omega} p_S(x) dx \\ p_L(x) &\leq D_L(x) \text{ where } L \in \mathbb{L} \\ |q_L(x)| &\leq R_L(x) \text{ where } L \neq S \\ 0 &= \text{div } q_L(x) + p_L(x) - \sum_{L' \in L.P} w_{(L',L)} p_{L'}(x) \\ &\text{where } L \neq S \end{aligned} \quad (6)$$

This is equivalent to a multi-flow problem over a large graph constructed from the image dimensions with the provided directed acyclic graph as overall architecture. Other than constraints put on the magnitude of the spatial flows, $q_L(x)$, and the capacity of the sink flows ($p_L(x)$ where $L.C = \emptyset$), the system is assumed to have infinite capacity. The *flow conservation constraint*:

$$G_L(x) = \text{div } q_L(x) + p_L(x) - \sum_{L' \in L.P} w_{(L',L)} p_{L'}(x) = 0 \quad (7)$$

ensures that no flow enters the network except through the source flow $p_S(x)$ or exits the network except through the outgoing flows $p_L(x)$ at the leaf labels $L \in \mathbb{L}$. This is a strict generalization of the hierarchical form explored by Baxter et al.[3] considering hierarchies to be a specific class of rooted DAG.

3.2 Primal-Dual Formulation

The primal model can be converted to a primal-dual model through the use of Lagrangian multipliers over the flow conservation constraint $G_L(x)$ yielding the Lagrangian:

$$\begin{aligned} \min_u \max_{p,q} \left(\int_{\Omega} p_S(x) dx + \sum_{\forall L \neq S} \int_{\Omega} u_L(x) G_L(x) dx \right) \\ p_L(x) &\leq D_L(x), \text{ where } L \in \mathbb{L} \\ |q_L(x)| &\leq R_L(x) \text{ } L \neq S . \end{aligned} \quad (8)$$

First, we must ensure that equation (8) is convex with respect to u , considering p, q to be fixed, and concave with respect to p, q with u fixed, as to meet the requirements of the minimax theorem (Ekeland & Temam

[19], pp. 176). Considering p, q as fixed, G is obviously fixed as well, implying that equation (8) is linear over u and therefore convex. It should also be noted that G is a linear function of p, q , meaning that (8) is again linear and therefore concave with respect to p and q . This implies the existence of a saddle point and the equivalence of the formulation regardless of the order of the prefix max and min operators under the minimax theorem. (Ekeland & Temam[19], pp. 176)

3.3 Dual Formulation

To convert the primal-dual model into the original minimization problem, we can find the saddle point through the optimization of each isolated variable. We proceed through the sink-flow variables, p_L , working bottom-up from the leaf-labels \mathbb{L} , then address the spatial flow variables, q_L , within each label. Starting with any leaf-label, $L \in \mathbb{L}$, we can isolate p_L in (8) giving:

$$\begin{aligned} & \min_{u_L} \max_{p_L(x) \leq D_L(x)} \int_{\Omega} u_L(x) p_L(x) dx \\ & = \min_{u_L(x) \geq 0} \int_{\Omega} u_L(x) D_L(x) dx \end{aligned} \quad (9)$$

when $u_L(x) \geq 0$. (If $u_L(x) < 0$, the function can be arbitrarily maximized by $p_L(x) \rightarrow -\infty$.) Moving onto every branch label, L , where $L.C \neq \emptyset$ and $L.P \neq \emptyset$, p_L can be isolated in (8) as:

$$\begin{aligned} & \min_{u} \max_{p_L(x)} \left(\int_{\Omega} u_L(x) p_L(x) dx \right. \\ & \quad \left. - \sum_{\forall L' \in L.C} \int_{\Omega} w_{(L,L')} u_{L'}(x) p_L(x) dx \right) = 0 \end{aligned} \quad (10)$$

at the saddle point $u_L(x) = \sum_{\forall L' \in L.C} w_{(L,L')} u_{L'}(x)$. Lastly, the source flow, p_S , can be isolated in a similar manner, that is:

$$\begin{aligned} & \min_{u} \max_{p_S(x)} \left(\int_{\Omega} p_S(x) dx \right. \\ & \quad \left. - \sum_{\forall L' \in S.C} \int_{\Omega} w_{(S,L')} u_{L'}(x) p_S(x) dx \right) = 0 \end{aligned} \quad (11)$$

at the saddle point, $1 = \sum_{\forall L' \in S.C} w_{(S,L')} u_{L'}(x)$. These constraints can be combined to yield the labeling constraints in the original formulation. The maximization of the spatial flow functions can be expressed in a well-studied form (Giusti [20], pp. 3-4) as:

$$\max_{|q_L| \leq R_L(x)} \int_{\Omega} u_L(x) \operatorname{div} q_L(x) dx = \int_{\Omega} R_L(x) |\nabla u_L(x)| dx$$

(12)

through the application of Gauss-Ostrogradsky theorem:

$$\int_{\Omega} u_L(x) \operatorname{div} q_L(x) dx = - \int_{\Omega} q_L(x) \cdot \nabla u_L(x) dx \quad (13)$$

which implies that we can express the saddle point of equation (8) as the original energy functional, (4), and therefore, finding the saddle point of (8) is equivalent to solving the DAGMF problem.

4 Solutions to the Primal-Dual Formulation

4.1 Augmented Lagrangian Approach

One way to address the primal-dual optimization problem in equation (8), is to augment the Lagrangian with a quadratic penalty term. This *augmented Lagrangian* is equivalent to the unaugmented Lagrangian, but encourages faster convergence to solutions which fulfill the optimization constraint, in this case $G_L(x) = 0$. [5] Such an augmentation yields the formula:

$$\begin{aligned} & \min_{u} \max_{p,q} \left(\int_{\Omega} p_S(x) dx + \sum_{\forall L \neq S} \int_{\Omega} u_L(x) G_L(x) dx \right. \\ & \quad \left. - \frac{c}{2} \sum_{\forall L \neq S} \int_{\Omega} G_L(x)^2 dx \right) \\ & \quad p_L(x) \leq D_L(x), \quad \forall L \in \mathbb{L} \\ & \quad |q_L(x)| \leq R_L(x), \quad L \neq S \end{aligned} \quad (14)$$

where c is the quadratic penalty parameter. Using this formula, we can iteratively maximize each component. To simplify the notation, we will denote the incoming flow to a label as:

$$\zeta_L(x) = \sum_{L' \in L.P} w_{(L',L)} p_{L'}(x) \quad (15)$$

Equation (14) can be optimized iteratively by holding each variable fixed except one which is used to maximize (in the case of $q_L(x)$, $p_L(x)$, and $p_S(x)$) or minimize (in the case of $u_L(x)$) the objective function accordingly similar to previous augmented Lagrangian approaches to continuous max-flow models [2,3,34,41,42]. Therefore, the solution steps for the augmented Lagrangian approach are:

1. Maximize (14) over the spatial flows q_L approximately for each label via iterative application of:

$$\text{Proj}_{|q_L(x)| \leq R_L(x)} \left(q_L(x) + \tau \nabla(\text{div } q_L(x) + p_L(x) - \zeta_L(x) - u_L(x)/c) \right)$$

which is a Chambolle projection iteration. [11] τ is a small positive gradient descent parameter. In practice, one iteration has been found to be sufficient and computationally preferable. [2, 3, 41, 42]

2. Maximize (14) the outgoing flow to the sink, p_L where $L.C = \emptyset$, analytically as:

$$\min\{D_L(x), \zeta_L(x) - \text{div } q_L(x) + u_L(x)/c\}$$

3. Maximize (14) the outgoing flow for each label, p_L where $L.C \neq \emptyset$ and $L.P \neq \emptyset$, analytically as:

$$\frac{1}{1 + \sum_{L' \in L.C} w_{(L,L')}^2} \left(\zeta_L(x) - \text{div } q_L(x) + u_L(x)/c + \sum_{\forall L' \in L.C} w_{(L,L')} (p_{L'}(x) + \text{div } q_{L'}(x) - \zeta_{L'}(x) + w_{(L,L')} p_L(x) - u_{L'}(x)/c) \right)$$

4. Maximize (14) over the source flow, p_S , analytically as:

$$\frac{1}{\sum_{L' \in S.C} w_{(S,L')}^2} \left(\frac{1}{c} + \sum_{\forall L' \in S.C} w_{(S,L')} (\text{div } q_{L'}(x) + p_{L'}(x) - \zeta_{L'}(x) + w_{(S,L')} p_S(x) - u_{L'}(x)/c) \right)$$

5. Minimize (14) over u_L for each label analytically as:

$$u_L(x) - c (\text{div } q_L(x) - \zeta_L(x) + p_L(x))$$

The precise pseudo-code is given in Algorithm 1. The convergence of said augmented Lagrangian algorithm is on the order of $O(1/N)$ similar to other primal-dual approaches [12]. This was shown by Hong and Luo [25] for *multi-block alternating direction method of multipliers* algorithms such as our algorithm. This convergence relies on the decomposition of cost function and constraints across each block having particular, restrictive properties which are upheld for all inter-node flows, p_L . The use of a proximal term, $-\frac{\tau}{2}|q_L(x) - q_L^{old}(x)|^2$, on the spatial flow updates corrects for not upholding all these properties. This guarantees linear convergence under weaker restrictions [25] and provides a theoretical justification for the use of only a single Chambolle iteration per update step.

Note that within each step there exists a large amount of inherent parallelism allowing for *general purpose programming on graphics processing units* (GPGPU) acceleration of each step with relative ease. This solver and GPGPU accelerated variants are available open-source at www.advancedsegmentationtools.org.

4.2 Proximal Bregman Approach

The augmented Lagrangian approach for DAGMF presented in Section 4.1 may not be the most efficient solution algorithm under certain conditions, particularly due to the potentially prohibitive memory requirements of saving all the flow variables. Pseudo-flow approaches [1] use Bregman proximal projection operators in the optimization process, implicitly representing the inter-node flow variables (the flow variables p_L), and allowing for more memory efficient and possibly more computationally efficient solution algorithms. We will be using a similar approach to Bae et al.[1] to develop a novel proximal Bregman or pseudo-flow approach for addressing DAGMF problems. Such proximal point approaches to variational optimization are known to have weak convergence (i.e. for any collection of smooth functions $y_L(x)$, $L \in \mathbb{L}$, the value $\sum_{L \in \mathbb{L}} \int_{\Omega} u_L^k(x) y_L(x) dx$ converges to $\sum_{L \in \mathbb{L}} \int_{\Omega} u_L^{\infty}(x) y_L(x) dx$ where u^k is the labeling at iteration k and u^{∞} is the optimum of Eq (4).)[36] This was explicitly demonstrated for Bregman distance functions by Censor and Zenios (Censor and Zenios [10], Theorem 3.6) as well as Chen and Teboulle (Chen and Teboulle [13], Theorem 3.4).

Before developing a pseudo-flow approach, it is necessary to develop some simplifying notation. The specific notation introduced is $d_L(x)$ where $L \in \mathbb{L}$ which represents the *flow excess* of the label, L , taking into account its ancestors, and $W_{(A,B)}$ is the amount of labeling weight assigned to a descendant label A derived from a descendant label B . $d_L(x)$ can be defined recursively as:

$$d_L(x) = \begin{cases} 0, & \text{if } L = S \\ D_L(x) + \text{div } q_L(x) + \sum_{L' \in L.P} w_{(L',L)} d_{L'}(x), & \text{if } L \in \mathbb{L} \\ \text{div } q_L(x) + \sum_{L' \in L.P} w_{(L',L)} d_{L'}(x), & \text{else} \end{cases} \quad (16)$$

which mirrors the top-down process of excess flow accumulation, and $W_{(A,B)}$ can also be defined recursively as:

$$W_{(A,B)} = \begin{cases} \sum_{p \in \text{path}(A,B)} \prod_{(M,N) \in p} w_{(M,N)} & \\ \begin{cases} 1, & \text{if } A = B \\ \sum_{\forall L \in A.C} w_{(A,L)} W_{(L,B)}, & \text{if } B \in A.P^* \\ 0, & \text{else} \end{cases} & \end{cases} \quad (17)$$

which mirrors the bottom-up process of label accumulation.

With this notation, we can develop a pseudo-flow representation which is amenable to solving through a

Algorithm 1: Augmented Lagrangian solution to the DAGMF problem. This algorithm implements the steps described in Section 4.1, taking the data terms $D_L(x)$, smoothness terms $R_L(x)$, gradient descent step size τ , and quadratic penalty constant c as input, yielding the labeling $u_L(x)$ which minimizes Equation (4).

```

Topological sort the DAG into the sorted list  $\mathbb{O}$  (begins with source label  $S$ ) with reverse list  $\mathbb{O}^{-1}$  (ends with source label  $S$ );
InitializeSolution();
while not converged do
  UpdateFlows();
  for  $\forall L$  do
     $\forall x, u_L(x) \leftarrow u_L(x) - c(\text{div } q_L(x) - \zeta_L(x) + p_L(x))$ ;
  end
end

```

which uses the subroutines:

InitializeSolution()

```

Clear  $u_L(x), q_L(x)$  for all labels;
for each  $L$  in order  $\mathbb{O}^{-1}$  do
   $\forall x, p_L(x) \leftarrow \min_{L', C=\emptyset} D_{L'}(x)$ ;
   $\forall x, \zeta_L(x) \leftarrow \min_{L', C=\emptyset} D_{L'}(x)$ ;
  if  $L.C = \emptyset$  then
    if  $L \in \arg \min_{L', C=\emptyset} D_{L'}(x)$  then
       $\forall x, u_L(x) \leftarrow 1/|\arg \min_{L', C=\emptyset} D_{L'}(x)|$ ;
    else
       $\forall x, u_L(x) \leftarrow 0$ ;
    end
  end
  for each  $L' \in L.P/\{S\}$  do
     $\forall x, u_{L'}(x) \leftarrow u_{L'}(x) + w_{(L',L)}u_L(x)$ ;
  end
end

```

UpdateFlows()

```

for  $\forall L \neq S$  do
   $\forall x, q_L(x) \leftarrow \text{Proj}_{|q_L(x)| \leq R_L(x)}(q_L + \tau \nabla(\text{div } q_L(x) + p_L(x) - \zeta_L(x) - u_L(x)/c))$ ;
end
Clear  $\zeta_L(x)$  for all labels;
for each  $L$  in order  $\mathbb{O}$  do
  for each  $L' \in L.C$  do
     $\forall x, \zeta_{L'}(x) \leftarrow \zeta_{L'}(x) + w_{(L,L')}p_L(x)$ ;
  end
  if  $L.C \neq \emptyset$  and  $L.P \neq \emptyset$  then
     $\forall x, \sigma_L(x) \leftarrow \zeta(x) - \text{div } q_L(x) + u_L(x)/c$ ;
  else if  $L = S$  then
     $\forall x, \sigma_S(x) \leftarrow 1/c$ ;
  end
end
for each  $L$  in order  $\mathbb{O}^{-1}$  do
  if  $L.C = \emptyset$  then
     $\forall x, p_L(x) \leftarrow \min\{D_L(x), \zeta_L(x) - \text{div } q_L(x) + u_L(x)/c\}$ ;
    for  $L' \in L.P$  do
       $\forall x, \sigma_{L'}(x) \leftarrow \sigma_{L'}(x) + w_{(L',L)}(\text{div } q_{L'}(x) + p_{L'}(x) - \zeta_{L'}(x) + w_{L',L}p_L(x))$ ;
    end
  else if  $L = S$  then
     $\forall x, p_S(x) \leftarrow \frac{1}{\sum_{L' \in S.C} w_{(S,L')}^2} \sigma_S(x)$ ;
  else
     $\forall x, p_L(x) \leftarrow \frac{1}{1 + \sum_{L' \in L.C} w_{(L,L')}^2} \sigma_L(x)$ ;
    for  $L' \in L.P$  do
       $\forall x, \sigma_{L'}(x) \leftarrow \sigma_{L'}(x) + w_{(L',L)}(\text{div } q_{L'}(x) + p_{L'}(x) - \zeta_{L'}(x) + w_{L',L}p_L(x))$ ;
    end
  end
end

```


Algorithm 2: Proximal Bregman solution to the DAGMF problem. This algorithm implements the steps described in Section 4.2, taking the data terms $D_L(x)$, smoothness terms $R_L(x)$, gradient descent step size τ , and proximity weighting constant c as input, yielding the labeling $u_L(x)$ which minimizes Equation (4).

```

Topological sort the DAG into the sorted list  $\mathbb{O}$  (begins with source label  $S$ ) with reverse list  $\mathbb{O}^{-1}$  (ends with source label  $S$ );
 $\forall L \in \mathbb{L}, u_L(x) \leftarrow 1/|\mathbb{L}|$ ;
while not converged do
   $\forall L, \forall x, d_L(x) \leftarrow \text{div } q_L(x)$ ;
   $\forall L \in \mathbb{L}, \forall x, d_L(x) \leftarrow d_L(x) + D_L(x)$ ;
  for  $L$  in order  $\mathbb{O}/\{S\}$  do
    for  $L' \in L.C$  do
       $\forall x, d_{L'}(x) \leftarrow d_{L'}(x) + w_{(L,L')}d_L(x)$ ;
    end
  end
   $\forall L \in \mathbb{L}, \forall x, u_L(x) \leftarrow u_L(x) \exp\left(-\frac{d_L(x)}{c}\right)$ ;
   $\forall L \in \mathbb{L}, \forall x, d_L(x) \leftarrow u_L(x)$ ;
   $\forall x, a(x) \leftarrow \sum_{\forall L \in \mathbb{L}} u_L(x)$ ;
   $\forall L \in \mathbb{L}, \forall x, u_L(x) \leftarrow u_L(x)/a(x)$ ;
   $\forall L \notin \mathbb{L}, \forall x, d_L(x) \leftarrow 0$ ;
  for  $L$  in order  $\mathbb{O}^{-1}/\{S\}$  do
     $\forall x, q_L(x) \leftarrow \text{Proj}_{|q_L(x)| \leq S_L(x)}(q_L(x) - c\tau \nabla d_L(x))$ ;
    for  $L' \in L.P/\{S\}$  do
       $\forall x, d_{L'}(x) \leftarrow d_{L'}(x) + w_{(L',L)}d_L(x)$ ;
    end
  end
end

```

mixture of Bregman proximal projections and Chambolle iterations. The entropic distance metric used in this formulation is:

$$\begin{aligned} \text{dist}_g(u, v) &= \sum_{\forall L \in \mathbb{L}} \int_{\Omega} u_L(x) \ln \left(\frac{u_L(x)}{v_L(x)} \right) dx \\ &+ \sum_{\forall L \in \mathbb{L}} \int_{\Omega} (-u_L(x) + v_L(x)) dx \end{aligned} \quad (18)$$

which can be verified to be a Bregman distance (when $\forall L, u_L(x) \in [0, 1]$) using the entropy function:

$$g(u) = \sum_{\forall L \in \mathbb{L}} \int_{\Omega} (u_L(x) \ln u_L(x) - u_L(x)) dx. \quad (19)$$

The non-smooth pseudo-flow formulation for DAGMF is:

$$\max_{|q_L(x)| \leq R_L(x)} \int_{\Omega} \min_{L \in \mathbb{L}} (d_L(x)) dx \quad (20)$$

Using the equivalence between Eq's (4),(8) and (6), we can introduce constraints on $u_L(x)$ present in Eq. (4) into Eq. (8). Thus, the pseudo-flow formulation can be derived from Eq. (8) through the following series of steps. First, we must switch the order of the max and min operators, which as stated in Section 3.2 is allowable, giving us:

$$\begin{aligned} \max_{p,q} \min_u \int_{\Omega} \left(p_S(x) + \sum_{\forall L} u_L(x) G_L(x) \right) dx \\ \forall L \in \mathbb{L}, p_L(x) \leq D_L(x) \text{ and } \forall L \neq S, |q_L(x)| \leq R_L(x) \end{aligned}$$

(21)

By explicitly introducing the constraints $u_S(x) = 1$ and $u_L(x) = \sum_{L' \in L.C} w_{(L,L')} u_{L'}(x)$ from Eq. (4) (along with expanding $G_L(x)$ according to its definition in Eq. (7)) we can group together terms allowing us to eliminate the variable $p_S(x)$, yielding:

$$\begin{aligned} \max_{p,q} \min_u \int_{\Omega} \left(\sum_{\forall L} u_L(x) \text{div } q_L(x) \right. \\ \left. + \sum_{\forall L \in \mathbb{L}} u_L(x) p_L(x) \right) dx \\ \forall L \in \mathbb{L}, p_L(x) \leq D_L(x) \text{ and } \forall L \neq S, |q_L(x)| \leq R_L(x) \\ u_S(x) = 1, \text{ and } \forall L \notin \mathbb{L}, u_L(x) = \sum_{L' \in L.C} w_{(L,L')} u_{L'}(x) \end{aligned} \quad (22)$$

By explicitly introducing the constraint $u_L(x) \geq 0$, which implies that $\max_{p_L(x) \leq D_L(x)} u_L(x) p_L(x) = u_L(x) D_L(x)$, we can eliminate the remaining inter-node flow variables $p_L(x)$ yielding:

$$\begin{aligned} \max_q \min_u \int_{\Omega} \left(\sum_{\forall L} u_L(x) \text{div } q_L(x) \right. \\ \left. + \sum_{\forall L \in \mathbb{L}} u_L(x) D_L(x) \right) dx \\ \forall L \neq S, |q_L(x)| \leq R_L(x) \text{ and } \forall L, u_L(x) \geq 0 \\ u_S(x) = 1, \text{ and } \forall L \notin \mathbb{L}, u_L(x) = \sum_{L' \in L.C} w_{(L,L')} u_{L'}(x) \end{aligned}$$

(23)

Since $u_L(x) = \sum_{L' \in L.C} w_{(L,L')} u_{L'}(x)$ we can use the definition of $d_L(x)$ to simplify the optimization, removing every branch-label's $u_L(x)$ from the objective function, which gives us:

$$\begin{aligned} & \max_q \min_u \int_{\Omega} \sum_{\forall L \in \mathbb{L}} u_L(x) d_L(x) dx \\ & \forall L \neq S, |q_L(x)| \leq R_L(x) \text{ and } \forall L, u_L(x) \geq 0 \\ & u_S(x) = 1, \text{ and } \forall L \notin \mathbb{L}, u_L(x) = \sum_{L' \in L.C} w_{(L,L')} u_{L'}(x) \end{aligned} \quad (24)$$

Note that by the construction of the DAG, $u_S(x) = 1$ and $u_L(x) = \sum_{L' \in L.C} w_{(L,L')} u_{L'}(x)$ together imply $\sum_{\forall L \in \mathbb{L}} u_L(x) = 1$. This allows us to remove all branch-label's $u_L(x)$ from the constraint set, replacing them with the constraint $\sum_{\forall L \in \mathbb{L}} u_L(x) = 1$. This removes $u_L(x)$ entirely from the optimization problem, except in the case of leaf-labels, yielding the equation:

$$\begin{aligned} & \max_q \min_u \int_{\Omega} \sum_{\forall L \in \mathbb{L}} u_L(x) d_L(x) dx \\ & \forall L \neq S, |q_L(x)| \leq R_L(x) \\ & \sum_{\forall L \in \mathbb{L}} u_L(x) = 1, \text{ and } \forall L, u_L(x) \geq 0 \\ & = \max_{|q_L(x)| \leq R_L(x)} \int_{\Omega} \min_{L \in \mathbb{L}} (d_L(x)) dx . \end{aligned} \quad (25)$$

We can combine this representation with the entropic distance to yield an improved labeling $u(x)$ proximal to $v(x)$ as:

$$\begin{aligned} & \arg \min_u \max_q \int_{\Omega} \sum_{\forall L \in \mathbb{L}} u_L(x) d_L(x) dx + c \text{dist}_g(u, v) \\ & \forall L \neq S, |q_L(x)| \leq R_L(x) \\ & \sum_{\forall L \in \mathbb{L}} u_L(x) = 1, \text{ and } \forall L, u_L(x) \geq 0 \end{aligned} \quad (26)$$

given a constant non-negative proximity weighting parameter, $c \geq 0$. This proximal projection operation can be solved analytically for all leaf-labels $L \in \mathbb{L}$ by placing a Lagrange multiplier on the constraint $\sum_{\forall L \in \mathbb{L}} u_L(x) = 1$ which yields the formula:

$$u_L(x) = \frac{v_L(x) \exp\left(-\frac{d_L(x)}{c}\right)}{\sum_{\forall L' \in \mathbb{L}} v_{L'}(x) \exp\left(-\frac{d_{L'}(x)}{c}\right)} \quad (27)$$

which fulfills the constraint $u_L(x) \geq 0$ provided that $v_L(x) \geq 0$. The labeling functions at the leaf-labels can be propagated upwards using the labeling constraints

$u_L(x) = \sum_{L' \in L.C} w_{(L,L')} u_{L'}(x)$ to get the value of $u_L(x)$ where $L \notin \mathbb{L}$. A corollary to this is that this label update equation approaches the non-smooth pseudo-flow representation, Eq. (20), as $c \rightarrow 0$.

Lastly, we must consider updating the spatial flows, once again by finding the gradient of equation (26) with respect to $\text{div } q_L(x)$. Doing so yields another Chambolle iteration [11] with positive gradient descent parameter τ :

$$q_L(x) \leftarrow \text{Proj}_{|q_L(x)| \leq R_L(x)} (q_L(x) - c\tau q'_L(x)) \quad (28)$$

where $q'_L(x)$ is:

$$q'_L(x) = \begin{cases} \nabla \left(v_L(x) \exp\left(-\frac{d_L(x)}{c}\right) \right) & \text{if } L \in \mathbb{L} \\ \nabla \left(\sum_{L' \in \mathbb{L}} W_{(L,L')} v_{L'}(x) \exp\left(-\frac{d_{L'}(x)}{c}\right) \right) & \text{else} \end{cases} \quad (29)$$

Combining these processes together, the solution steps for the proximal Bregman approach are:

1. Calculate the values of $d_L(x)$ in a top-down manner.
2. Minimize (26) over u_L where $L \in \mathbb{L}$ using Eq. (27)
3. Maximize (26) over q_L for each label using the Chambolle iteration Eq. (28). This can be done efficiently in a bottom-up manner. As with the augmented Lagrangian approach, one iteration has been found to be sufficient for proximal Bregman approaches to continuous max-flow. [1]

The precise pseudo-code is given in Algorithm 2.

Again, each step displays a large amount of inherent parallelism allowing for GPGPU acceleration of each step with relative ease. The formulation also implicitly represents the labeling and inter-node flows, requiring much lower memory use which make it more efficient for implementing large problems with constrained GPU memory resources. This solver is available open-source at www.advancedsegmentationtools.org.

4.3 Discretization and Memory Consumption

Both Algorithm 1 and 2 are formulated in a way that is agnostic to how Ω is discretized. In our implementations, Ω is discretized into a grid similar to previous continuous max-flow approaches [2, 3, 34, 41, 42].

The memory requirements for both algorithms are dominated by the space required buffers storing the primal/dual optimization variables, intermediate optimization variables, and input data and regularization terms. Each of these grows linearly with the size of the image in our discretization approach. Given a model with N_L leaf-labels and N_B branch-labels (labels that are not the source S or a leaf-label), the number of

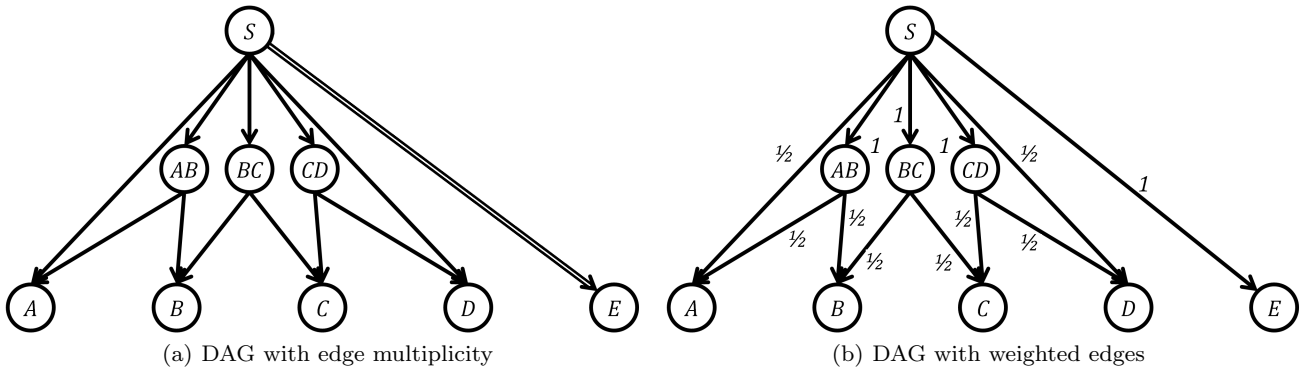


Fig. 2: DAG for segmentation into labels $\mathbb{L} = \{A, B, C, D, E\}$ in which label groups $\mathbb{G} = \{AB, BC, CD\}$ are regularized. Note that this would be impossible in a hierarchical model since the regularization groups conflict with each other. Fig. 2(a) shows the intermediate multi-edged, unweighted DAG. Fig. 2(b) shows this DAG with weights explicitly recorded rather than through multiplicity which is used by the solution algorithms.

buffers required by the augmented Lagrangian algorithm (Algorithm 1) is $5N_L + 6N_B + 2$ buffers for 2D images and $6N_L + 7N_B + 2$ buffers for 3D volumes. The proximal Bregman (Algorithm 2) requires fewer buffers, specifically $4N_L + 3N_B + 1$ buffers for 2D images and $5N_L + 4N_B + 1$ for 3D volumes. The proximal Bregman approach therefore uses between 25% to 50% memory required for the augmented Lagrangian algorithm for segmentation problems with a large number of labels, depending on how many are leaf-labels versus branch-labels. This decrease can play a large role in the feasibility of large segmentation problems, especially when implemented using the limited memory space available to the GPU.

5 Arbitrary Region Regularization

With the increasing complexity of part-whole and mutual exclusion relationships being integrated into optimization based segmentation, it is important to show the capability of DAGMF to address the regularization of arbitrarily defined label structures, that is, the incorporation of any label ordering. First, we must express the segmentation in terms of its partition set, \mathbb{L} , and a finite set of grouped-labels $\mathbb{G} \subset 2^{\mathbb{L}}$, each a union of a set of leaf-labels, defining the groups we want to apply a regularization term to. For brevity, we will consider the segmentation problem to be stated in this form.

To show how arbitrary region regularization can be implemented with DAGMF, we must consider the construction of a DAG with associated transformations on smoothness parameters. First, create a graph with one vertex corresponding to each element of \mathbb{L} and one for each element of \mathbb{G} and one for the source node, S .

Create an edge from the source vertex to each vertex corresponding to an element of \mathbb{G} . For each element $G \in \mathbb{G}$, create an edge from vertex corresponding to G to each vertex corresponding to an element $L \in G$. Now, find the vertex with the maximal in-degree which corresponds to an element in \mathbb{L} and call this in-degree r . Add sufficient edges from the source vertex to each vertex in \mathbb{L} to ensure that the in-degree for each is r . Figure 2(a) shows an example of this for the problem $\mathbb{L} = \{A, B, C, D, E\}$ with $\mathbb{G} = \{AB, BC, CD\}$.

This process yields an unweighted multi-graph in which the multiplicity of any given edge is indicate of the weight to be assigned to the corresponding edge in the weighted DAG proportional to adjacent incoming edges. Taking the example used in Figure 2(a) and applying the required normalization yields that shown in Figure 2(b).

To determine the appropriate smoothness terms, we note that for each vertex associated with element $G \in \mathbb{G}$, we have the labeling constraint:

$$u_G(x) = \sum_{L \in G} \frac{1}{r} u_L(x) \quad (30)$$

from the construction of the graph. This means that whatever smoothness term that is desired for G must be multiplied by r to account for the factor of $1/r$.

This is not the only way to create a DAGMF model for a given problem, and many segmentation problems defined in such a way can be implemented by multiple DAGMF structures. In that respect, the DAGMF structure is not necessarily unique and optimization of the structure to improve performance may be possible.

6 Experiments

6.1 Synthetic Image: Venn Diagram

To test the efficacy of the segmentation algorithm and to highlight the improvements of arbitrary region regularization made possible through DAGMF, we constructed the example segmentation given in Figure 4. This image involves two overlapping objects, each of which is regularized by the proposed DAGMF model using the structure showing in Figure 3, but such a regularization scheme is not possible in HMF, as the dark yellow intersection has more than one parent region. (In HMF, the intersection can be denoted either as part of the red square or green circle but not both.) The data term for each leaf-label is:

$$D_L(x) = |I(x) - \bar{I}_L| \quad (31)$$

where $I(x)$ is the RGB intensity of the pixel and \bar{I}_L is the RGB intensity of the label L prior to the addition of noise. The regularization terms $R_L(x)$ were all constants, tuned individually to improve the performance of each method. Note that the DAGMF reconstruction substantially improves the crispness of the area where all four regions meet which is heavily distorted in the other three segmentation techniques.

6.2 Medical Images - Brain Tissue Segmentation

To demonstrate the applicability of this technique in medical image segmentation, consider segmentation of the brain into background (K), external cerebrospinal fluid (eCSF), cortical gray matter (cGM), white matter (WM), subcortical gray matter (sGM) and ventricles (V). In this example, we are using the BrainWeb [14] database to provide a realistic digital phantom, providing an exact ground truth labeling. The data terms (shown in Fig. 7) were developed using a simple Bayesian framework which includes both intensity and localization components and are representative of those one would derive for this segmentation problem on clinical images [31]. The DAG used for segmentation is provided in Figure 6. The smoothness terms were all constants, that is, $R_L(x) = \alpha_L$, meaning that no contrast sensitive terms were used to localize edges, but only uniform regularization was used to penalize longer boundaries between regions. The data terms, shown in Figure 7, follow the Bayesian formula:

$$D_L(x) = -\ln P(I(x)|x \in L) + \text{dist}(x, R_L) \quad (32)$$

where $P(I(x)|x \in L)$ is the probability of voxel x having intensity $I(x)$ given it is in label L , and $\text{dist}(x, R_L)$ is an

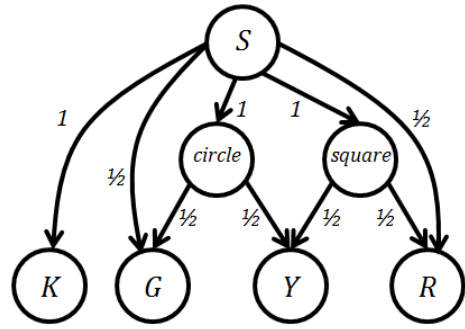
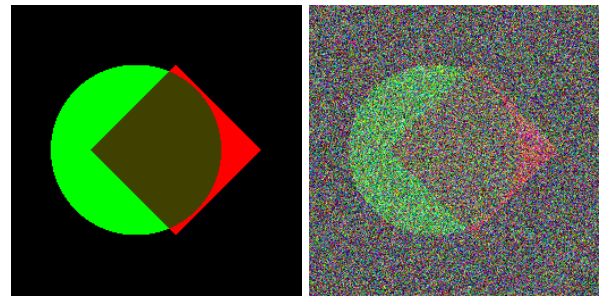
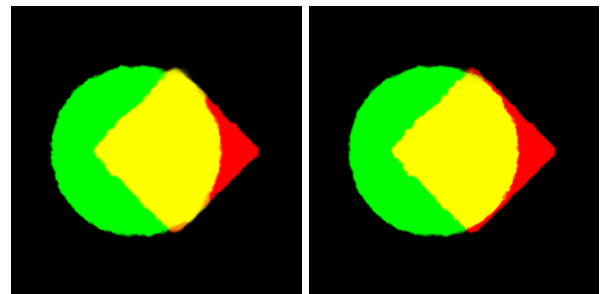


Fig. 3: Directed acyclic graph and weights used for DAGMF segmentation shown in Figure 4. The nodes *circle* and *square* denote the labels associated with the union of green with yellow and red with yellow respectively.



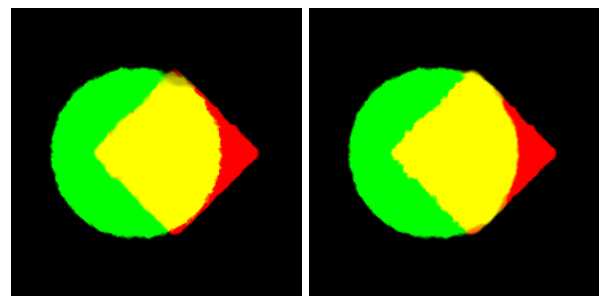
(a) Original

(b) Noisy



(c) Potts (DSC 94.6%)

(d) DAGMF (DSC 95.4%)



(e) HMF #1 (DSC 95.0%)

(f) HMF #2 (DSC 94.9%)

Fig. 4: Synthetic image (a) polluted with noise (b) and reconstructed using a Potts model (c), DAGMF (d) and HMF models with either the red square (e) or green circle (f) regularized. Weighted DSC is given for each segmentation.

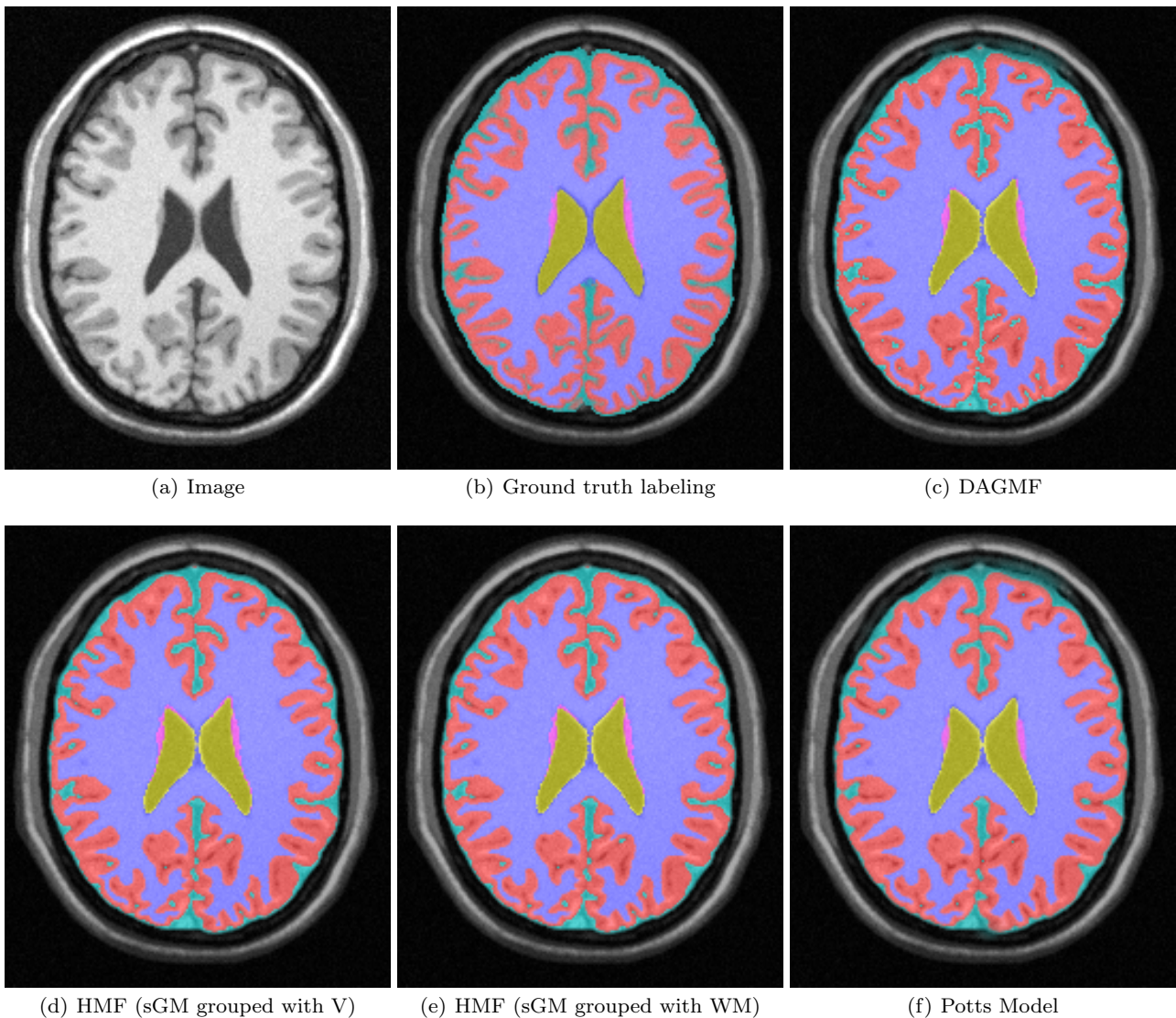


Fig. 5: Brain tissue segmentation using DAGMF using data terms in Fig. 7 and constant smoothness terms. Note the improvement in the pink subcortical gray matter region.

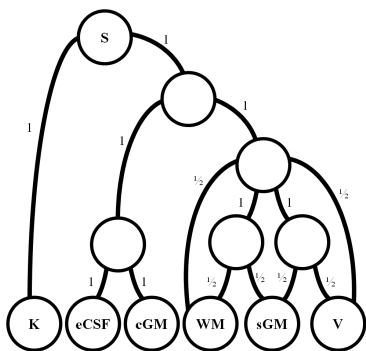


Fig. 6: DAG representing the brain tissue segmentation problem in Fig. 5.

estimated distance between x and the region associated with label L . The segmentation is shown in Figure 5.

The construction of the DAG is motivated by the grouping of adjacent regions with similar regularization requirements (such as the grouping of the eCSF and cGM) or by grouping regions that, when grouped, have a significantly less tortuous boundary (such as the grouping of all brain regions into a common superlabel opposed to the background or K label). Notably, a Venn diagram model (similar to Fig. 3) represents the subcortical labels, using the geometrical intuition that the subcortical gray matter could be interpreted spatially as the overlap of the WM and V regions.

The primary benefit of DAGMF in the context of medical image segmentation in particular is that the

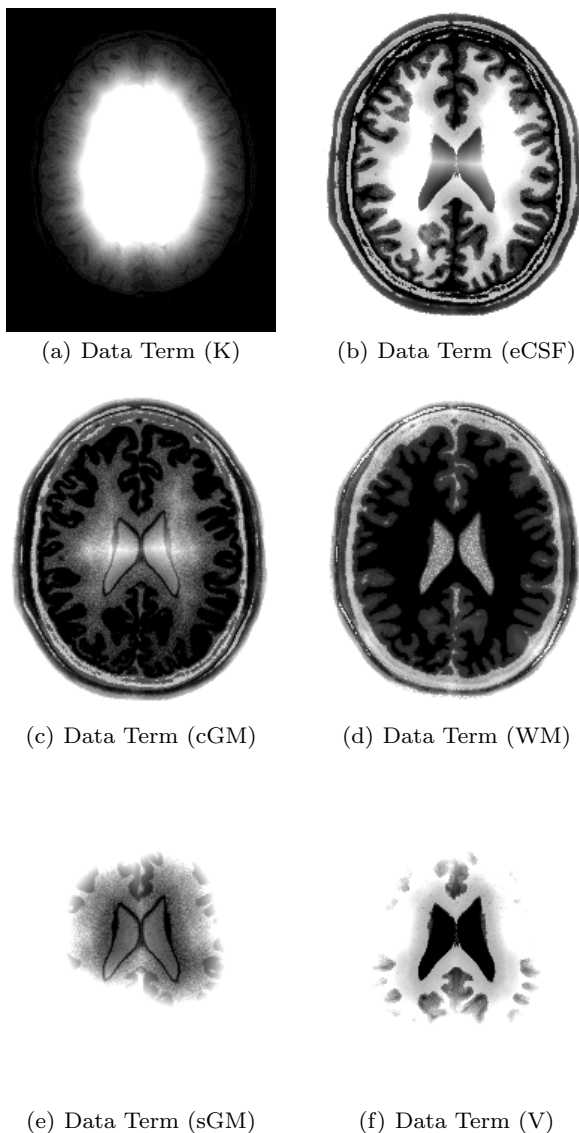


Fig. 7: Bayesian data terms used in Fig. 5.

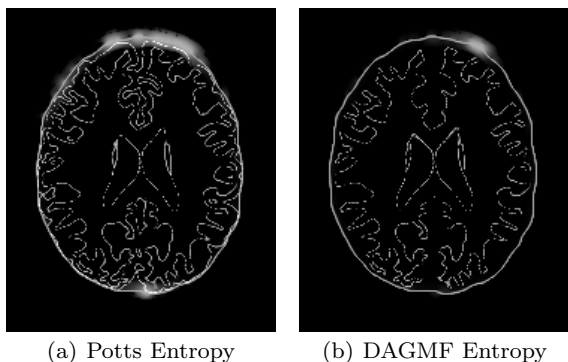


Fig. 8: Segmentation uncertainty (entropy) from Figure 5. The Potts model has much higher uncertainty in the background segmentation around the frontal lobe.

Region	DAGMF	HMF 1	HMF 2	Potts
K	99.2%	99.2%	99.2%	98.8%
eCSF	74.9%	76.5%	76.5%	72.5%
cGM	91.1%	91.0%	91.0%	90.6%
WM	95.9%	95.7%	95.7%	95.9%
sGM	81.3%	76.1%	77.3%	76.6%
V	95.7%	96.1%	95.9%	95.1%

Table 2: Dice coefficient for segmentations in Fig. 5.

regularization of any label group allows for the segmentation algorithm designers to incorporate multiple regularization schemes without prohibiting others. Analogous to Section 6.1, HMF could be used to regularize sGM with either V or WM, but not both. DAGMF allowed for both regularization schemes to be implemented simultaneously. This improved the segmentation accuracy of the subcortical gray matter over the Potts model and both HMF models as highlighted in Table 2, despite having the same data terms and uniform regularization.

In addition, the DAGMF result had a lower entropy, implying that the segmentation was more certain than that of Potts as shown in Figure 8. The entropy of the fuzzy labeling at each voxel can be used as a surrogate for the uncertainty of the binary labeling marginalized at each voxel. Note however that this is only an approximation, as the theoretical basis of segmentation uncertainty in (both binary and fuzzy) continuous max-flow models has yet to be investigated.

6.3 Natural Images: Scene Decomposition

Natural image segmentation tasks that could benefit from unconstrained label orderings include scene decomposition [24]. As shown by Delong et al. [16], geometric context often requires non-hierarchical regularization schemes. This segmentation problem is composed of five distinct regions (F -front, T -top, B -bottom, L -left, R -right) as shown in Figure 9(a). Strong priors are available for the T , B , L and R regions but often not the F region. For regularization purposes, one could consider F as the intersection of each of the four side regions, taking advantage of their priors and encourage a more central position in the segmented image. In previous approaches, such as HMF, such regularization would not be possible. The DAG representing this regularization structure is given in Figure 9(b). Unlike previous discrete approaches [16], the image is partitioned into the label regions $\{T, B, L, R\}$ thus preventing errors such as labelling a single pixel as being in both L and R regions.

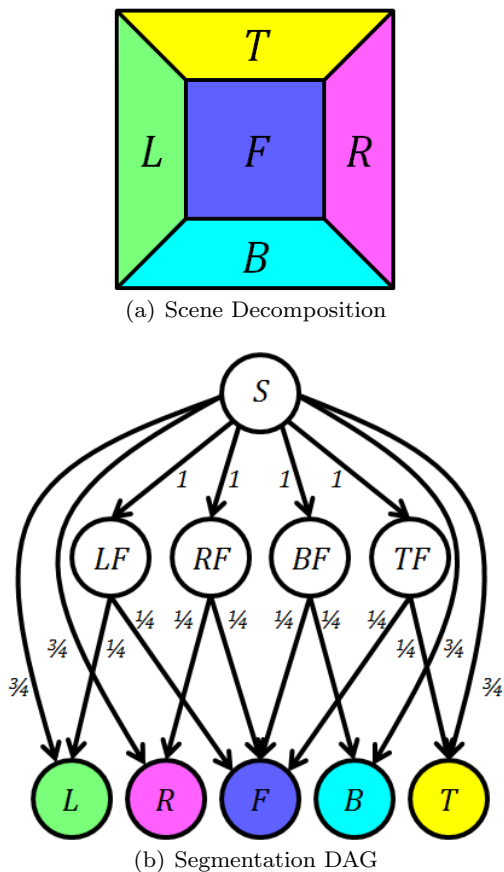


Fig. 9: Segmentation structure used in scene decomposition into F -front, T -top, B -bottom, L -left, R -right. The color code corresponds to that used in Figure 10.

An example segmented image using this framework is shown in Figure 10. These images were collected from the Geometric Context dataset [24] and the data terms used were derived from the Surface Layout classification framework [24]. Specifically, the data terms for each region in the partition, i.e. labels $L \in \mathbb{L}$, are:

$$D_L(x) = -\ln P_L(x) \quad (33)$$

where $P_L(x)$ is the likelihood estimate of pixel x belonging to label L as determined by the Surface Layout classification framework [24]. The regularization terms, $R_L(x)$ were all the same constant for the regions LF , RF , BF , and TF and were zero for all end-labels.

For quantitative validation, this scene decomposition model was applied to the Stanford indoor image dataset [15] (48 images) using the classifiers developed by Hoiem et al. [24] as the basis for the data terms. The regularization terms, given in Table 3, determined by an exhaustive search on a subset of the database (first 4 images). Qualitative segmentation results for DAGMF as well as the continuous Potts model [42] and HMF



Fig. 10: Example outdoor scene segmentation. Accuracy rate is given for each segmentation. The color code for the segmented images are shown in Figure 9.

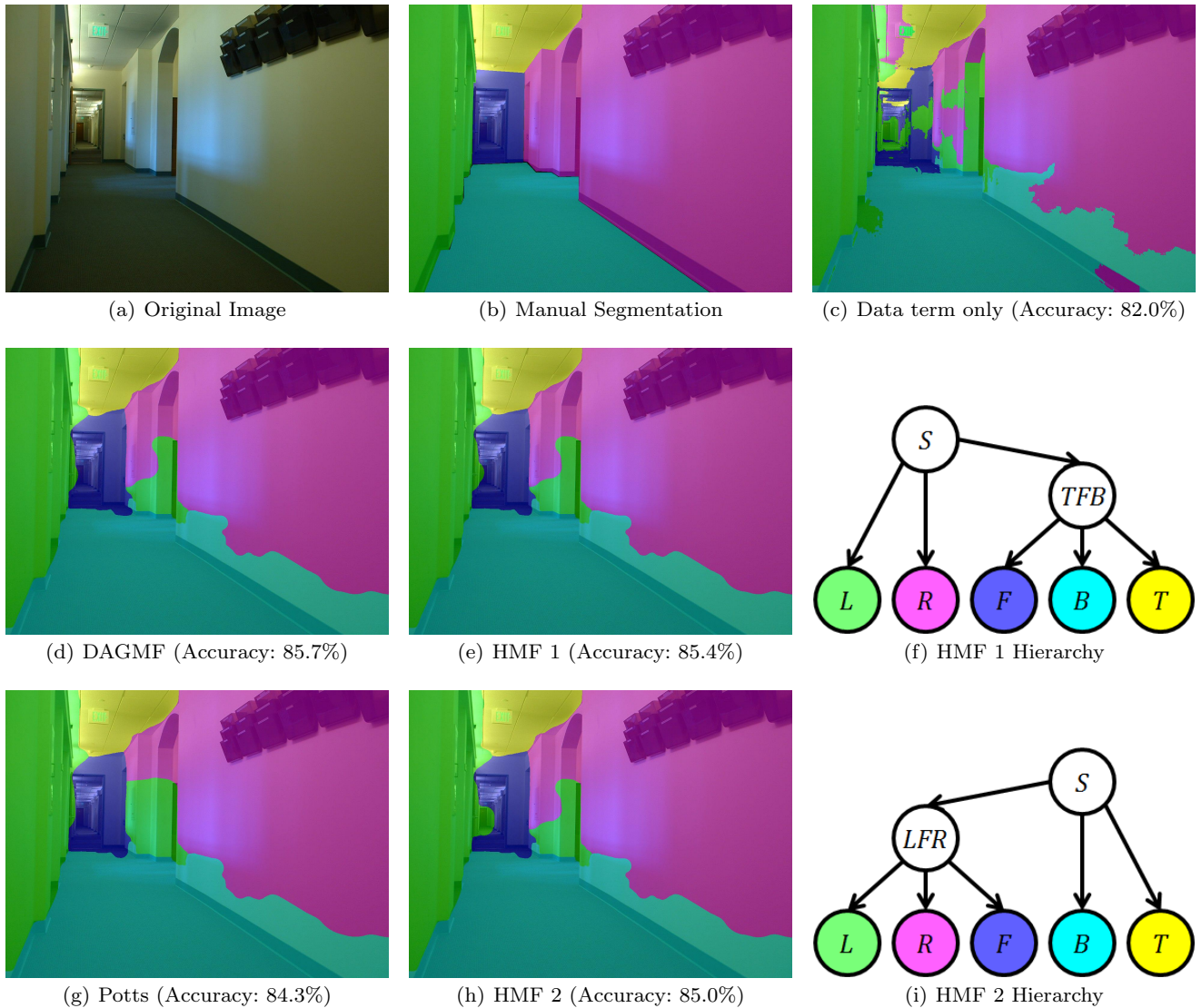


Fig. 11: Example scene segmentation from the Stanford indoor dataset [15]. Accuracy rates are given for each segmentation. Label orderings used in the first and second HMF segmentation are shown in (f) and (i) respectively. The color code for the segmented images are shown in Figure 9 and in (f) and (i).

Model - Label	Value
Potts	10
HMF 1 - TFB	5
HMF 1 - $L \in \mathbb{L}$	7.5
HMF 2 - LFR	5
HMF 2 - $L \in \mathbb{L}$	7.5
DAGMF - LF, RF, BF, TF	5
DAGMF - $L \in \mathbb{L}$	9

Table 3: Value of the constant regularization terms used in the various max-flow models.

(n=48)	Accuracy Rate	
	Original	Improvement over DTO
DTO	84.4% \pm 6.7%	
Potts	85.6% \pm 7.1%	1.2% \pm 1.2%
HMF 1	85.7% \pm 6.9%	1.3% \pm 1.1%
HMF 2	85.7% \pm 7.0%	1.3% \pm 1.0%
DAGMF	85.9% \pm 7.0%	1.5% \pm 1.2%

Table 4: Accuracy rates for segmentations in the Stanford indoor dataset such as that shown in Fig. 11. DTO refers to the “data term only” method.

models [3,34] are shown in Figure 11 taken from the testing dataset with quantitative results in Table 4. The “data term only” (DTO) results reflect the accuracy of the classifiers developed by Hoiem et al. [24].

The use of simple constant regularization improves the accuracy slightly overall, with all regularized methods showing a statistically significant improvement over the data-term-only method. The more representative DAGMF model has slightly higher accuracy than the previous max-flow models, however, due to the size of the dataset and similarity between methods, no statistical significance was detected. This is largely due to the data-term generation, which was based on super-pixels and thus assigns a constant data term to a local region, the boundaries of which were not considered in the regularization. The results for each max-flow model could readily be improved by creating more complex regularization terms cognizant of these boundaries.

6.4 Natural + Synthetic Images: Hue Reconstruction

Another application which is not currently possible with existing continuous max-flow methods is hue denoising in color images. The problem of hue reconstruction is of particular interest as the hue displays distinctly cyclic behaviour, meaning that it is especially ill-suited for Ishikawa models [26] more traditionally used for image reconstruction. In addition, hue offers a color property that is, in theory, invariant to lighting and atmospheric conditions but is sensitive to RGB additive noise especially at low brightness and saturation.

The first step in hue reconstruction with DAGMF is the construction of an appropriate DAG. In this scenario, the DAG can follow a relatively simple two-layer formula. The bottom layer consists of N nodes representing the discrete bins the hues are grouped into. The second layer is a regularization layer with N nodes, each with $M < N/2$ edges to the lower layer. These edges are arranged in a cyclic manner, each regularization node being connected to M consecutive hues. The result is truncated linear regularization. Between two hue nodes the regularization grows linearly with the distance between them if less than M , else, the regularization is constant. Fully linear regularization can be achieved by setting $M = \lfloor N/2 \rfloor$. An example of these types of DAGs is given in Figure 12. Note that in this case, using the Potts model for hue reconstruction can be expressed as a limiting case as this framework with $M = 1$.

In Figure 13, RGB additive noise was applied to a synthetic image with 50% saturation and 25-75% brightness. Reconstructing the hue using a DAGMF hue reconstruction model ($N = 36, M = 16$) allowed for linear regularization across hues resulting in a 79% decrease

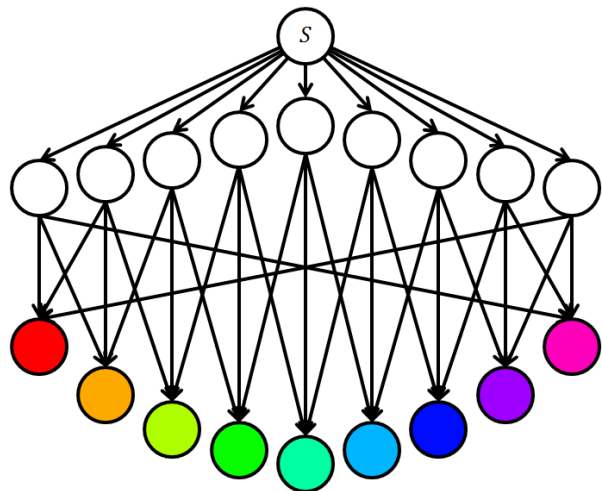


Fig. 12: Example DAG for hue reconstruction with $N = 9$ discrete hues and a truncated linear model of width $M = 3$. Although not shown, the weight of each edge on the top level is 1, and $1/M$ on the bottom layer.

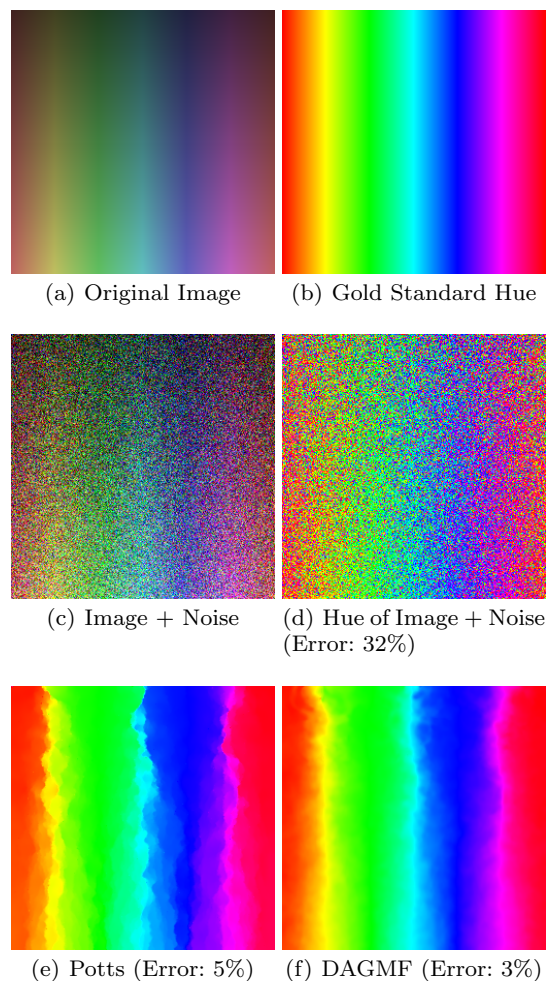


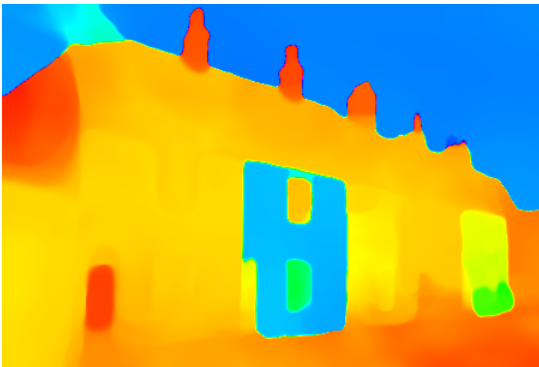
Fig. 13: Hue reconstruction on synthetic image with corresponding normalized hue error.



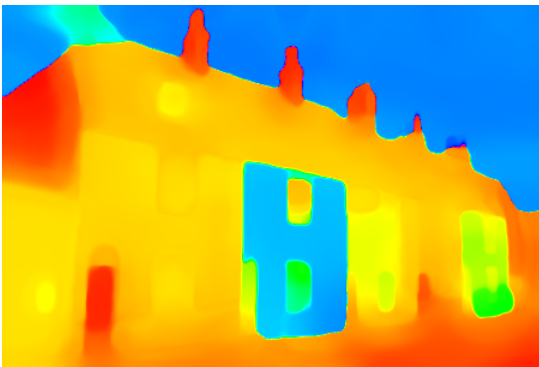
(a) Original Photograph



(b) Original Hue



(c) Potts Model



(d) DAGMF Model

Fig. 14: Example hue reconstruction on natural images with DAGMF model ($N = 36$, $M = 16$).

in the hue error, compared to 64% using a Potts model with the same data terms and regularization values. The data term used was:

$$D_L(x) = |I(x) - \bar{I}_L| \quad (34)$$

where $I(x)$ is the RGB-valued intensity of pixel x and \bar{I}_L is the RGB-valued intensity of the highest saturation colour corresponding to hue L .

The same pipeline can be used on natural images to robustly estimate hue in the presence of noise and lighting effects. Figure 14 displays this hue reconstruction on natural images in the presence of shadows and atmospheric perspective effects. Compared to the Potts model, the cyclic DAGMF model retains much more detail at the same level of regularization, preserving features such as smaller windows and doors.

6.5 Medical Images - MRI Phase Reconstruction

Similar to hue reconstruction in Section 6.4, phase reconstruction requires that the field being reconstructed is represented cyclically, which is not suitable to current continuous max-flow solvers. Often, in order to perform simple MR phase processing, the original phase, represented in the interval $[-\pi, \pi]$, is first unwrapped to the interval $(-\infty, \infty)$ minimizing the differences in phase over a local neighbourhood [39]. However, phase unwrapping techniques can be error-prone in the presence of noise [38], necessitating filtering approaches that do not rely on unwrapped phase. [18] Phase smoothing is used to extract residual or high-pass phase components for MR imaging types dependent on phase information such as susceptibility weighted imaging [23]. Figure 15 demonstrates the application of the previous cyclic DAGMF model (Figure 12) with 40 equally-spaced phase labels to reconstructing the background phase in a 3T cranial MRI, using the data term:

$$D_L(x) = |\theta(x) - \theta_L| \quad (35)$$

where $\theta(x)$ is the phase measured at voxel x and θ_L is the phase associated with label L . Figure 15(d) displays the residual phase information after smoothing.

7 Discussion and Future Work

From a theoretical point of view, DAGMF illustrates that any label ordering is possible to incorporate into a continuous max-flow segmentation paradigm using primal-dual optimization with augmented Lagrangian multipliers. Thus, further work in continuous max-flow theory should focus on the development of more specific solvers for segmentation problem sub-classes. One

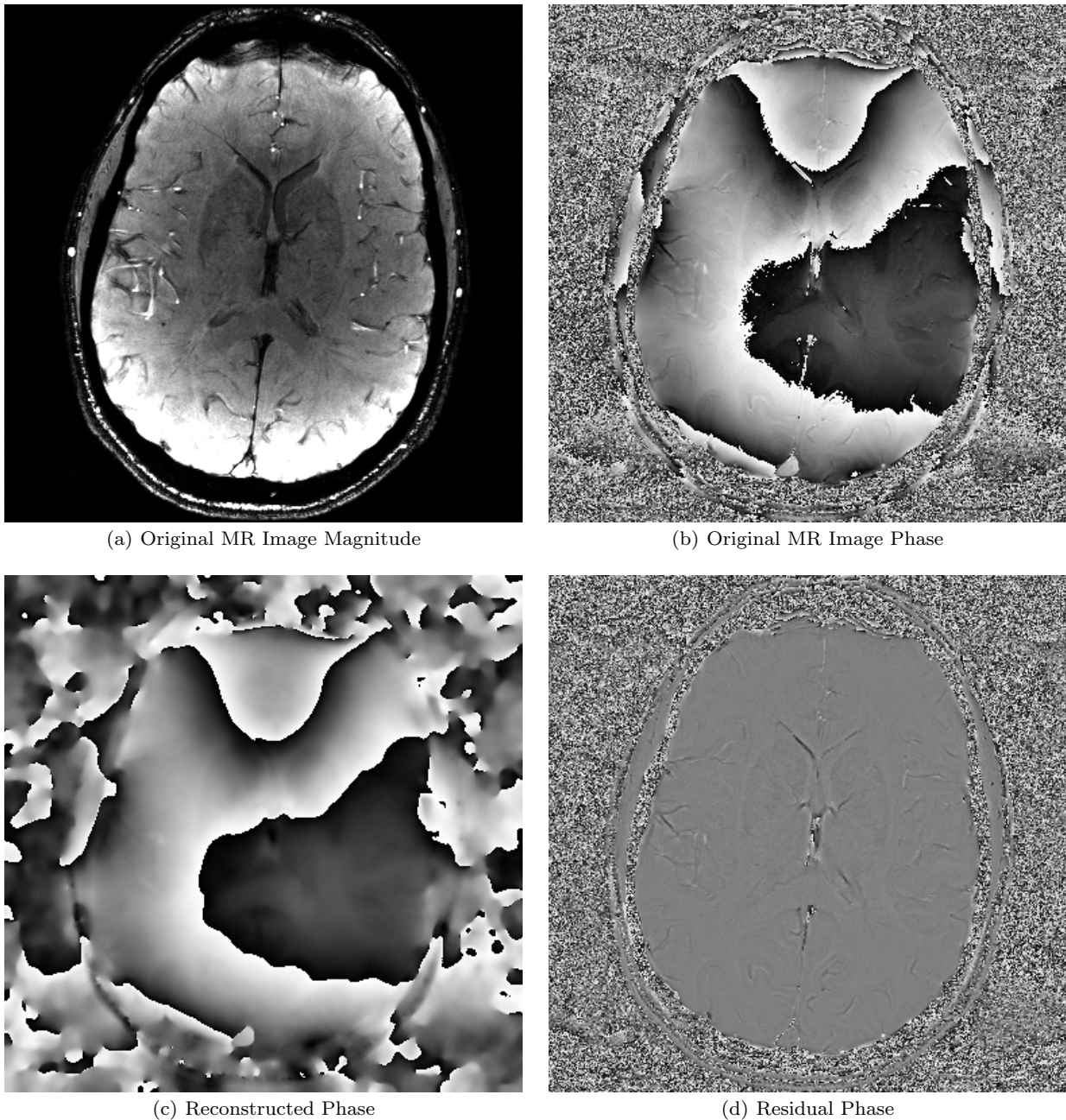


Fig. 15: Phase reconstruction of an 3T cranial MRI using 40 equal-spaced phase labels in a cyclic DAGMF model.

area of future work could be the development of a dedicated continuous max-flow optimization framework for hue and phase reconstruction (Sections 6.4 and 6.5 respectively) which takes advantage of the high degree of inherent symmetry in the problem definition and resultant DAG.

As stated in Section 5, there are multiple ways to construct a DAG that are mathematically equivalent but require different computation time and memory amounts. In addition, for certain subclasses of prob-

lems, a DAG may not be the most computationally efficient structure over which to perform max-flow. For problems in co-segmentation of multiple images simultaneously [22, 28] or segmentation based on a partially-annotated multi-atlas [29], other max-flow structures can take advantage of symmetry in the energy functional to reduce the number of, and simplify the interaction between, labellings ultimately creating more efficient algorithms. That being said, if that symmetry is removed or modified, such frameworks are no longer

applicable. DAGMF provides, in such scenarios, an initial framework in which accuracy can be evaluated separate from computability concerns, decoupling the processes of model creation and solver optimization. Our future work along these lines consists of being able to take regularization descriptions similar to those given in Section 5 and automatically optimize the structure used for maximum computational efficiency while ensuring mathematical equivalence.

Another area of future work is the incorporation of star convexity constraints [21,40] into a subset of the labels, allowing for both topological and shape information to be optimized for simultaneously. These shape constraints have already been integrated into binary [43] and hierarchical [4] continuous max-flow formulations.

8 Conclusions

This article presents a novel segmentation framework called Directed Acyclic Graph Max-Flow (DAGMF) segmentation which optimizes the variational continuous max-flow problem over a directed acyclic graph with each node representing a continuous labeling space. Two solution algorithms were proposed: one based on augmented Lagrangian multipliers in which the labeling constraints result from the characterization of the saddle-points; and one based on proximal Bregman projections in which the labeling constraints are explicitly enforced, but minimizes memory requirements by implicitly representing inter-node flow. These multiple solution algorithms allow for this framework to be adapted to the available computational resources.

This framework is a generalization of earlier extendable continuous max-flow approaches and is shown to be flexible enough to optimize any possible label ordering. The general applicability of this framework was demonstrated using a variety of synthetic, natural, and medical images where previous approaches could only approximate the required label ordering. This algorithm is highly parallelizable and has been implemented using GPGPU acceleration to ensure high performance.

Acknowledgements The authors would like to acknowledge Zahra Hosseini, Maria Drangova, and Ravi Menon's laboratory at the Robarts Research Institute Imaging Laboratories for their assistance in collecting and processing MRI phase data. John S.H. Baxter and A. Jonathan McLeod were funded through the Natural Sciences and Engineering Research Council of Canada.

References

1. Bae, E., Tai, X.C., Yuan, J.: Maximizing flows with message-passing: Computing spatially continuous min-cuts. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 15–28. Springer (2015)
2. Bae, E., Yuan, J., Tai, X.C., Boykov, Y.: A fast continuous max-flow approach to non-convex multi-labeling problems. In: *Efficient Algorithms for Global Optimization Methods in Computer Vision*, pp. 134–154. Springer (2014)
3. Baxter, J.S., Rajchl, M., Yuan, J., Peters, T.M.: A continuous max-flow approach to general hierarchical multi-labeling problems. arXiv preprint arXiv:1404.0336 (2014)
4. Baxter, J.S., Yuan, J., Drangova, M., Peters, T.M., Inoue, J.: Shape complexes in continuous max-flow segmentation. In: *SPIE Medical Imaging*, pp. 978,434–978,434. International Society for Optics and Photonics (2016)
5. Bertsekas, D.P.: *Nonlinear programming* (1999)
6. Billionnet, A., Minoux, M.: Maximizing a supermodular pseudoboolean function: A polynomial algorithm for supermodular cubic functions. *Discrete Applied Mathematics* **12**(1), 1–11 (1985)
7. Boros, E., Hammer, P.L.: Pseudo-boolean optimization. *Discrete applied mathematics* **123**(1), 155–225 (2002)
8. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **26**(9), 1124–1137 (2004)
9. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **23**(11), 1222–1239 (2001)
10. Censor, Y., Zenios, S.A.: Proximal minimization algorithm with d-functions. *Journal of Optimization Theory and Applications* **73**(3), 451–464 (1992)
11. Chambolle, A.: An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision* **20**(1-2), 89–97 (2004)
12. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision* **40**(1), 120–145 (2011)
13. Chen, G., Teboulle, M.: Convergence analysis of a proximal-like minimization algorithm using bregman functions. *SIAM Journal on Optimization* **3**(3), 538–543 (1993)
14. Cocosco, C.A., Kollokian, V., Kwan, R.K.S., Pike, G.B., Evans, A.C.: Brainweb: Online interface to a 3d mri simulated brain database. In: *NeuroImage* (1997)
15. Delage, E., Lee, H., Ng, A.Y.: A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 2418–2428. IEEE (2006)
16. Delong, A., Boykov, Y.: Globally optimal segmentation of multi-region objects. In: *International Conference on Computer Vision (ICCV)*, pp. 285–292. IEEE (2009)
17. Delong, A., Gorelick, L., Veksler, O., Boykov, Y.: Minimizing energies with hierarchical costs. *International journal of computer vision* **100**(1), 38–58 (2012)
18. Denk, C., Rauscher, A.: Susceptibility weighted imaging with multiple echoes. *Journal of Magnetic Resonance Imaging* **31**(1), 185–191 (2010)
19. Ekeland, I., Temam, R.: *Convex Analysis and Variational Problems*. SIAM (1976)

20. Giusti, E.: Minimal surfaces and functions of bounded variation. 80. Birkhauser (1984)
21. Gulshan, V., Rother, C., Criminisi, A., Blake, A., Zisserman, A.: Geodesic star convexity for interactive image segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3129–3136. IEEE (2010)
22. Guo, F., Yuan, J., Rajchl, M., Svenningsen, S., Capaldi, D.P., Sheikh, K., Fenster, A., Parraga, G.: Globally optimal co-segmentation of three-dimensional pulmonary 1 h and hyperpolarized 3 he mri with spatial consistence prior. *Medical image analysis* **23**(1), 43–55 (2015)
23. Haacke, E.M., Xu, Y., Cheng, Y.C.N., Reichenbach, J.R.: Susceptibility weighted imaging (swi). *Magnetic Resonance in Medicine* **52**(3), 612–618 (2004)
24. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. *International Journal of Computer Vision* **75**(1), 151–172 (2007)
25. Hong, M., Luo, Z.Q.: On the linear convergence of the alternating direction method of multipliers. arXiv preprint arXiv:1208.3922 (2012)
26. Ishikawa, H.: Exact optimization for markov random fields with convex priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **25**(10), 1333–1336 (2003)
27. Ivănescu, P.L.: Some network flow problems solved with pseudo-boolean programming. *Operations Research* **13**(3), 388–399 (1965)
28. Jang, J., Kim, H.W., Kim, Y.S.: Co-segmentation of inter-subject brain magnetic resonance images. In: IEEE International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 80–84. IEEE (2014)
29. Koch, L.M., Rajchl, M., Tong, T., Passerat-Palmbach, J., Aljabar, P., Rueckert, D.: Multi-atlas segmentation as a graph labelling problem: Application to partially annotated atlas data. In: *Information Processing in Medical Imaging*, pp. 221–232. Springer (2015)
30. Kolmogorov, V., Zabini, R.: What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **26**(2), 147–159 (2004)
31. van der Lijn, F., den Heijer, T., Breteler, M.M., Niessen, W.J.: Hippocampus segmentation in mr images using atlas registration, voxel classification, and graph cuts. *Neuroimage* **43**(4), 708–720 (2008)
32. Pock, T., Chambolle, A., Cremers, D., Bischof, H.: A convex relaxation approach for computing minimal partitions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 810–817. IEEE (2009)
33. Potts, R.B.: Some generalized order-disorder transformations. In: *Mathematical proceedings of the cambridge philosophical society*, vol. 48, pp. 106–109. Cambridge Univ Press (1952)
34. Rajchl, M., Baxter, J.S., McLeod, A.J., Yuan, J., Qiu, W., Peters, T.M., Khan, A.R.: Hierarchical max-flow segmentation framework for multi-atlas segmentation with kohonen self-organizing map based gaussian mixture modeling. *Medical Image Analysis* (2015)
35. Rajchl, M., Yuan, J., White, J., Ukwatta, E., Stirrat, J., Nambakhsh, C., Li, F.P., Peters, T.M., et al.: Interactive hierarchical-flow segmentation of scar tissue from late-enhancement cardiac mr images. *Medical Imaging, IEEE Transactions on* **33**(1), 159–172 (2014)
36. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization* **14**(5), 877–898 (1976)
37. Schlesinger, D., Flach, B.: Transforming an arbitrary minsum problem into a binary one. TU, Fak. Informatik (2006)
38. Su, X., Chen, W.: Reliability-guided phase unwrapping algorithm: a review. *Optics and Lasers in Engineering* **42**(3), 245–261 (2004)
39. Thomas, B., Somasundaram, S., Thamburaj, K., Kesavadas, C., Gupta, A.K., Bodhey, N.K., Kapilamoorthy, T.R.: Clinical applications of susceptibility weighted mr imaging of the brain—a pictorial review. *Neuroradiology* **50**(2), 105–116 (2008)
40. Veksler, O.: Star shape prior for graph-cut image segmentation. In: *Computer Vision—ECCV 2008*, pp. 454–467. Springer (2008)
41. Yuan, J., Bae, E., Tai, X.C.: A study on continuous max-flow and min-cut approaches. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2217–2224. IEEE (2010)
42. Yuan, J., Bae, E., Tai, X.C., Boykov, Y.: A continuous max-flow approach to potts model. In: *Computer Vision—ECCV 2010*, pp. 379–392. Springer (2010)
43. Yuan, J., Qiu, W., Ukwatta, E., Rajchl, M., Sun, Y., Fenster, A.: An efficient convex optimization approach to 3d prostate mri segmentation with generic star shape prior. *Prostate MR Image Segmentation Challenge, MICCAI* **7512** (2012)