Graph Cut Matching In Computer Vision

Toby Collins (s0455374@sms.ed.ac.uk)
February 2004

Introduction

Many of the problems that arise in early vision can be naturally expressed in terms of energy minimization. For example, a large number of computer vision problems attempt to assign labels (such as intensity, disparity, segmentation regions) to pixels based on noisy measurements. In the presence of these uncertainties, finding the best labelling may therefore be seen as an optimization problem. In the last few years, minimum cut/maximum network flow algorithms have emerged as an elegant and increasingly useful tool for exact or approximate energy minimisation. The basic technique is to construct a specialized graph for the energy function to be minimized such that the minimum cut on the graph also minimizes the energy (either globally or locally). The minimum cut, in turn, can be computed very efficiently by max flow algorithms. These methods have been successfully used for a wide variety of vision problems, including image restoration [4, 5, 6, 7], stereo and motion [9, 10, 11, 12, 13, 14, 15], image segmentation [16], multi-camera scene reconstruction [17], and medical imaging [18, 19, 20, 21]. Although network flows have been traditionally used to solve problems in physical networks, often a large variety of problems which have no inherent physical network can also be modelled using a network and a notion of flow in such a network.

1. The Pixel Labelling Problem

The classical use of energy minimization is to solve the pixel labelling problem, which is itself a generalization of such problems as stereo, motion, and image restoration. In the pixel labelling problem, the variables represent individual pixels and the possible values for an individual variable represent, e.g. its possible displacements or intensities. The input is a set of pixels P and a set of labels L. The goal is to find a labelling $f(P) \mapsto L$. Figure 1, which is found in [1] provides an illustration of the problem.

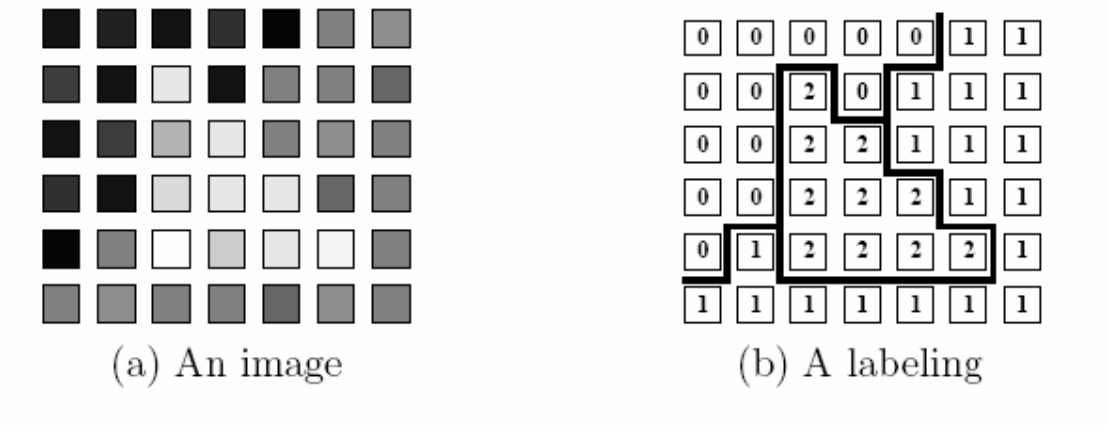

Figure 1: An example of image labelling. An image in (a) is a set of pixels P with observed intensities I_p for each $p \in P$. A labelling L shown in (b) assigns some label $L_p \in \{0,1,2\}$ to each pixel. Such labels can represent depth (in stereo), object index (in segmentation), original intensity (in image restoration), or other pixel properties. Normally, graph-based methods assume that a set of feasible labels at each pixel is finite. Thick lines in (b) show labelling discontinuities between neighbouring pixels.

Typically, certain visual constraints are used to constrain these labels. In vision and image processing, these labels often tend to vary smoothly within the image, except at some kind of

region boundaries where discontinuities are allowed. These constraints are reflected as defining the pixel labelling problem in terms of minimising some energy function. A standard form of the energy function is

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{p,q \in N} V_{p,q}(f_p, f_q) \quad (1)$$

Where $N \subset P \times P$ is a neighbourhood system on pixels, $D_p(f_p)$ is a function derived from the observed data that measures the cost of assigning the labels f_p, f_q to the adjacent pixels, p, q and is used to impose spatial smoothness.

2. Markov Random Fields

Markov Random Fields is a generative model often used in Image Processing and Computer Vision to solve labelling problems. This section briefly introduces the theory of Markov Random Fields (MRF), and it is commonly used to model the optimisation problem discussed above. A Markov Random Field consists of three sets; a set S of sites, a neighbourhood system N and a set of random variables F. The neighbourhood system $N = \{N_i \mid i \in S\}$, where each N_i is a subset of sites of S which form the neighbourhood of site i. The random field $F = \{F_i \mid i \in S\}$ consists of random variables F_i that take on a value f_i from a set of labels $L = \{l_1, l_2 ...\}$. A particular set of labels, often denoted by f is called a configuration of F. The probability of a particular configuration f, $P(F = f)$ must satisfy the *Markov property* in order for F to be a Markov Random Field.

$$P(f_i \mid f_{S-i}) = P(f_i \mid f_{N-i}), \forall i \in S \quad (2)$$

This means that the state of each random variable depends only on the state of its neighbours. The fact that a particular pixel label depends on the labels of its neighbours allows modelling the optimization problem as a MRF. From a probabilistic perspective, one wishes to estimate the configuration f based on observed data D that maximises the likelihood function, $P(D \mid f)$ Using Bayes Theorem, this likelihood function can be expressed as an energy function $E(f)$ and the maximum a posterior (MAP) estimate of f should maximize this energy function.

3. Energy Minimisation Models

There are currently two main models used to define the energy function, E, which are chosen depending on properties of labelled regions, such as piecewise consistency or discontinuities at boundaries. The first model is Potts Interaction Energy Model, which is given as follows:

$$E(I) = \sum_{p \in P} \left| I_p - I_p^{\circ} \right| + \sum_{(p,q) \in N} K_{(p,q)} . T(I_p \neq I_p) \quad (3)$$

where $I = \{I_p \mid p \in P\}$ are the unknown true labels over the set of pixels P and $I^{\circ} = \{I_p^{\circ} \mid p \in P\}$ are the observed labels corrupted by noise. The Potts interaction are specified by $K(p,q)$, which are the penalties for label discontinuities between adjacent pixels. The function T is an indicator function. Potts model is useful when the labels are likely to be piecewise constant with discontinuities at boundaries. The Pott energy can be optimally solved for binary labelling using max-flow, however the multiple label case is NPO-hard.

The second model is the Linear Interaction Energy Model:

$$E(I) = \sum_{p \in P} \left| I_p - I_p^{\circ} \right| + \sum_{(p,q) \in N} A_{(p,q)} . T\left(I_p \neq I_p \right) \quad (4)$$

The fundamental difference between this and the Pott equation is the inclusion of constants $A_{(p,q)}$, which stores the importance of the interaction between neighbouring pixels p and q .Unlike the Pott Energy Model, the Linear Interaction Energy produces labelings which are piecewise smooth, but with discontinuities across boundaries.

4. Pixel Labelling as a Graph Cut problem

Greig *et al.* [4] were first to discover that powerful min-cut/max-flow algorithms from combinatorial optimization can be used to minimize certain important energy functions in vision. In this section we will review some basic information about graphs and flow networks in the context of energy minimization. A directed weighted graph $G = (E,V)$ consists of a set of nodes V and a set of directed edges E that connect them. Usually the nodes correspond to pixels, voxels, or other features. A graph normally contains some additional special nodes that are called terminals, usually called the source $s \in V$ and the sink $t \in V$. In the context of vision, terminals correspond to the set of labels that can be assigned to pixels. In Figure 2(a) we show a simple example of a two terminal graph (due to Greig et al. [4]).

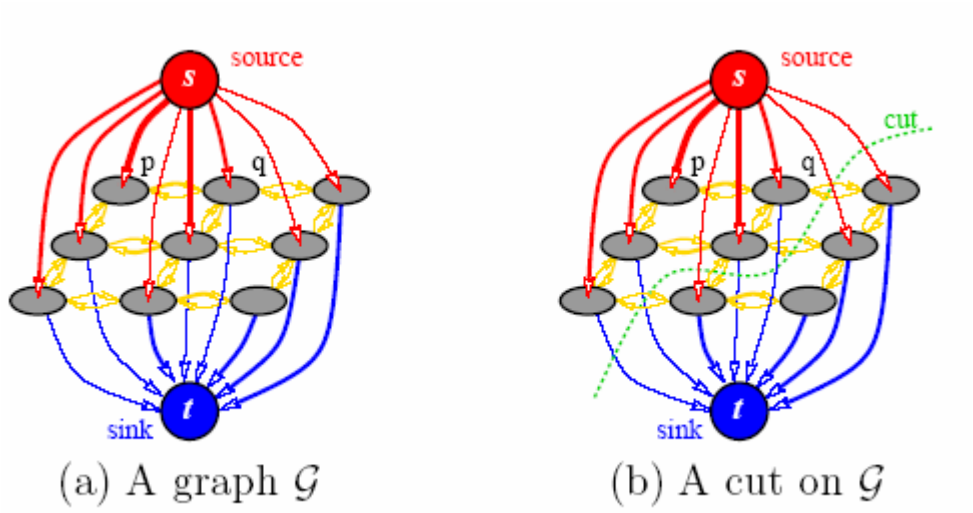

Figure 2: Example of a directed capacitated graph. Edge costs are reflected by their thickness

Normally, there are two types of edges in the graph*: n-links* and *t-links*. n-links connect pairs of neighbouring pixels or voxels. Thus, they represent a neighbourhood system in the image. The cost of n-links corresponds to a penalty for discontinuity between the pixels. These costs are usually derived from the pixel interaction term $V_{p,q}$ in energy equation (1). T-links connect pixels with terminals (labels), and the cost of a t-link connecting a pixel and a terminal corresponds to a penalty for assigning the corresponding label to the pixel. This cost is normally derived from the data term D_p in equation (1).

4.1 Network Flow

The fundamental network flow problem is the minimum cost flow problem; that is, determining a maximum flow at minimum cost from a specified source to a specified sink. A flo*w network* $G(V,E)$ is formally defined as a fully connected directed graph where each edge $(u,v) \in E$ has a positive capacity $c(u,v) \geq 0$. A *flow* in G *is* a real-valued function $f : VXV \rightarrow R$ that satisfies the following three properties:

Capacity Constraint:
For all $u,v \in V$, $f(u,v) \leq c(u,v)$

Skew Symmetry:
For all $u,v \in V$, $f(u,v) = -f(v,u)$

Flow Conservation:
For all $u \in (V - \{s,t\})$, $\sum_{u \in V} f(u,v) = 0$

The value of a flow is defined as $|f| = \sum_{u \in V} f(s,v)$, and interpreted as the total flow out of the source in the flow network G.

4.2 Min Cut

An $s-t$ cut C of flow on a graph with two terminals is a partitioning of the nodes in the graph into two disjoint subsets S and T such that the source s is in S and the sink t is in T. For a given flow f, the net flow across the cut (S,T) is defined as:

$$f(S,T) = \sum_{x \notin S} \sum_{y \in T} f(x,y) \tag{5}$$

The capacity of a cut (S,T) is defined as:

$$c(S,T) = \sum_{x \notin S} \sum_{y \in T} c(x,y) \tag{6}$$

A minimum cur of a flow network is a cut whose capacity is the least over all the $s-t$ cuts of the network. An example of a flow network with a valid flow is shown in figure 3, and a minimum cut in the context of image segmentation is shown in figure 4.

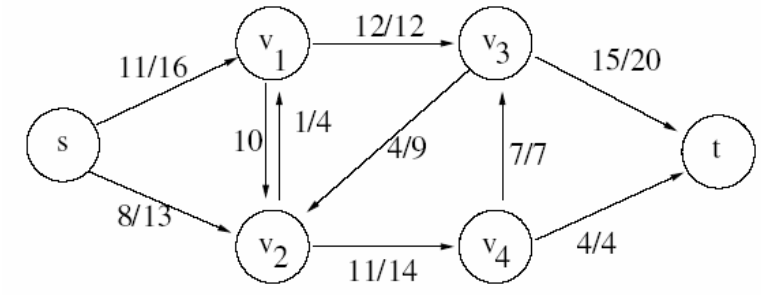

Figure 3: Taken from Cormen et. al. [15], this figure shows a flow network $G(V,E)$ with a valid flow f. The values on the edges are $f(u,v)/c(u,v)$. The current flow has value 19, it is not a maximum (as explained later.)

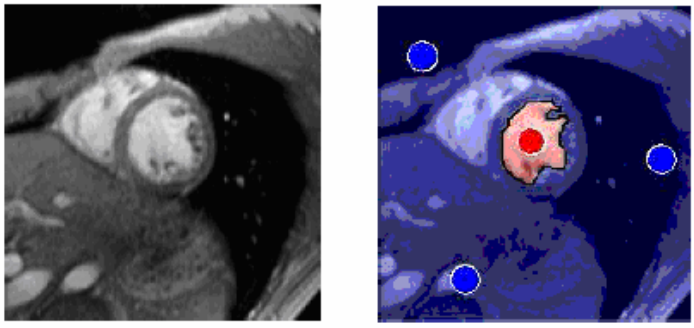

Figure 4: Graph cut/flow example in the context of image segmentation in Section (taken from [1]). Red and blue seeds are "hard-wired" to the source and the sink, respectively. As is common in this approach, the cost of edges between the pixels (graph nodes) is set to low values in places with high intensity contrast. Thus, cuts along object boundaries in the image should be cheaper.

4.3 Determining Min Cut: Max-Flow / Min Cut Correspondence

One of the fundamental results in combinatorial optimization is that the minimum $s-t$ cut problem can be solved by finding a maximum flow from the source s to the sink t. Loosely speaking, maximum flow is the maximum "amount of water" that can be sent from the source to the sink by interpreting graph edges as directed "pipes" with capacities equal to edge weights. The theorem of Ford and Fulkerson states that a maximum flow from s to t saturates a set of edges in the graph dividing the nodes into two disjoint parts $\{S,T\}$, corresponding to a minimum cut. Thus, min-cut and max-flow problems are equivalent.

Theorem 1: The max-flow min-cut theorem. If f is a flow in a flow network $G=(V,E)$ with source s and sink t, then the value of the maximum flow is equal to the capacity of a minimum cut. The proof can be found by referring to Cormen et. al. [15].

4.4 Max-Flow Algorithm

The polynomial algorithms for the single-source single-sink max flow problem can be divided into two classes, algorithms based on the Ford Fulkerson method and those based on the Push-Relabel. method. The two contrasting approaches are briefly described below, with appropriate references for a fuller treatment.

4.4.1 Ford and Fulkerson Algorithm

The algorithm is best explained by segregating it into its two parts, which Ford and Fulkerson called Routine A and Routine B, respectively. The first is a labelling process that searches for a flow augmenting path (.e., a path from s to t t for which $f<c$ along all forward arcs and $f>0$ along all backward arcs. If Routine A finds a flow augmenting path, Routine B changes the flow accordingly. Otherwise, no augmenting path exists, and optimality of the current flow is ensured by their theorem:

Theorem 2:. A flow f has maximum value if, and only if, there is no flow augmenting path with respect to f.

A fuller understanding of the Ford and Fulkerson Algorithm can be found in [21].

4.4.2 The Push-Relabel Method

The generic Push-Relabel algorithm does not construct a flow by constructing paths from s to t. Rather, it starts by pushing the maximum possible flow out from the source s into the neighbours of s, then pushing the excess flow at those vertices into their own neighbours. This is repeated until all vertices of G except s and t have an excess flow of zero (that is, the flow

conservation property is satisfied). Of course this might mean that some flow is pushed back into s. In contrast to the Ford-Fulkerson method where augmenting the flow operates on the complete residual graph, the Push-Relabel algorithms operate locally on a vertex at a time, inspecting only its neighbours. Unlike the Ford- Fulkerson method, the flow conservation property is not satisfied during the algorithm's execution. A more detailed explanation of the Push-Rabel method can be found in [22].

5. Applications

5.1 Image Restoration

The image restoration problem aims at recovering the original pixel intensities of an image when the observed image is noisy. Here the labels are the image intensities and the most likely labelling is obtained by minimizing an energy function similar to the ones described in earlier sections. The visual constraints exploited here are the fact that image intensities tend to vary smoothly in most images except at boundaries. Both the Pott Energy as well as Linear Interaction Energy model yields reasonable results. The actual choice of $K_{(p,q)}$ and $A_{(p,q)}$ determines the degree of smoothness in the restored images. Figure 5 shows examples from [1].

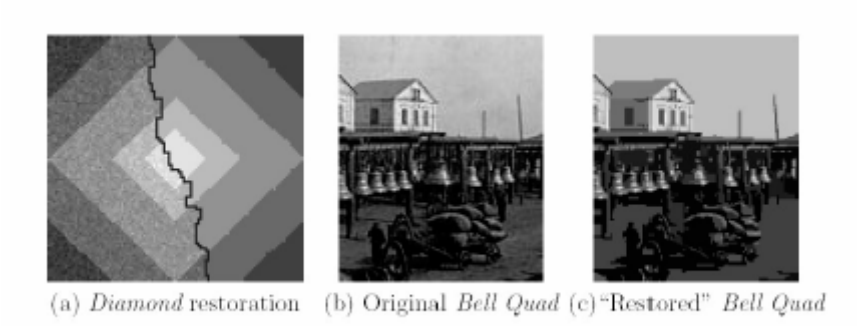

Figure 5: Image Restoration Examples taken from Boykov et. al. [1]. (a) Synthetic data : An example of a noisy image and its restored version. (b) & (c) Real data; An example ofa noisy image and its restored version.

5.2 Stereo

Dense stereo is a popular method for 3D Reconstruction from two calibrated views of a scene. It involves first recovering matching pixels (pixels corresponding to the same 3D feature) in the two views and then recovering the depth of the 3D point by triangulation (intersecting rays back projected from the two matching pixels). Finding accurate matching pairs for all pixels is a difficult problem to solve accurately because often such matching can be ambiguous depending on factors like camera baseline, amount of texture in the scene or the degree of specularity of objects in the scene. In a stereo pair, matching pixels are recovered using disparities. Every pixel $p_1(i, j)$ in the first image has a particular disparity d with respect to the matching pixel $p_2(i, j)$. A method called image rectification can ensure that corresponding pixels are always on identical scan lines in the rectified image pair. The problem of recovering an accurate disparity image can be posed as an energy minimization problem using the same MRF framework we have been studying. The problem of stereo is identical to the image restoration problem except that here the labels are disparity values. Energy models like the Pott Energy, shown in Equation 3 can incorporate such contextual information within the MRF framework.

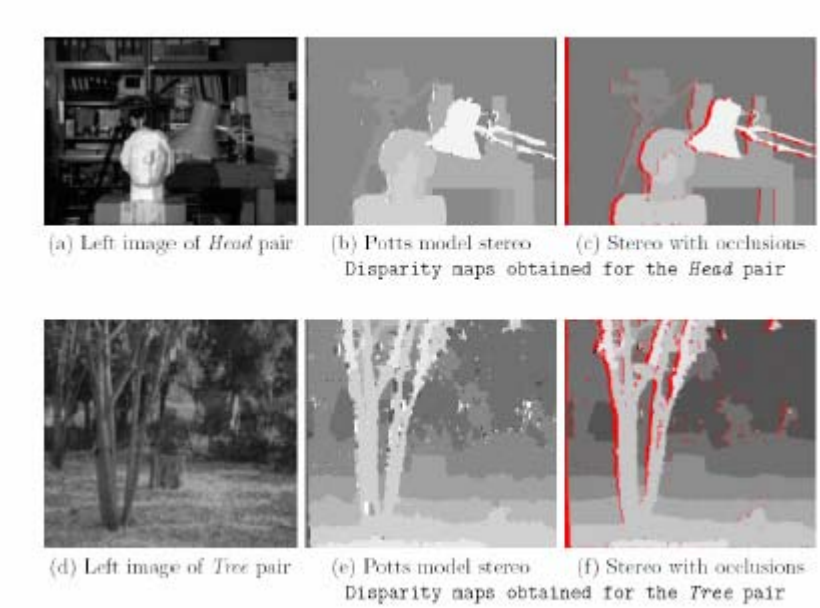

Figure 6: Stereo examples taken from Boykov et. al. [1]. One of the images in a stereo pair along with the computed disparity image. Top Row: Tsukuba dataset. Bottom Row: Tree dataset.

The stereo problem is harder compared to image restoration because of the presence of occlusions. Occlusion occur when 3D points are visible in only one of the stereo image pairs and are typically found near depth-discontinuities in the scene. Occlusions which make the visual correspondence problem harder, can be explicitly modelled in the Energy Minimization framework by a modified labelling problem of the following type. Sites in the MRF for this modified problem do not represent image pixels but pair of pixels which can potentially correspond. The set of labels is $\{0,1\}$ where 0 indicates either of the pixels are occluded and 1 indicates that the pair of pixels are matching. The new energy function is then defined to be:

$$E(f) = E_{data}(f) + E_{occ}(f) + E_s(f) \tag{7}$$

where

$$E_{data} = \sum_{l(p,q)=1} D_{(p,q)}$$

is the term which imposes a penalty based on intensity differences of matching pixels p and q,

$$E_s = \sum_{\{(p,q),(p',q')\}\in N} K_{\{(p,q),(p',q')\}} \cdot T\left(l_{(p,q)} \neq l_{(p',q')}\right)$$

is the smoothness term which forces adjacent pixels to have the same or relatively close disparities, and

$$E_{occ} = \sum_{p\in P_1 \bigcup P2} C_p \cdot T(p \,is\, occluded)$$

is the new occlusion penalty term which imposes a penalty for making a particular pixel p in the stereo image pair P_1 or P_2 occluded. $T(\cdot)$ is the indicator function in the above formulation. Minimizing the energy function is still NP-Hard but an approximate algorithm based on expansion computes a local minimum within a constant factor of the global minimum.

5.3 Segmentation

The Pott Energy function, (see Equation 3) comes up again in the context of image segmentation where the goal is to group image pixels into logical groups or segments which may represent

objects in the scene. In 3D Segmentation, the grouping is done on voxels in volumetric data as is typically encountered in medical imaging. Segmentation is typically posed as a binary labelling problem where f*oreground* and *background* constitutes the set of labels typically assigned to pixels or voxels. The binary labelling problem for the Pott Energy function as mentioned earlier can be optimally solved by a single execution of max-flow. The graph construction and max-flow formulation is quite identical to the one for the image restoration problem To get accurate segmentations, user input is provided into the labelling problem by allowing the user to pre-label (these labels are not allowed to change) some pixels as foreground and some as background. This is illustrated in Figure 7.

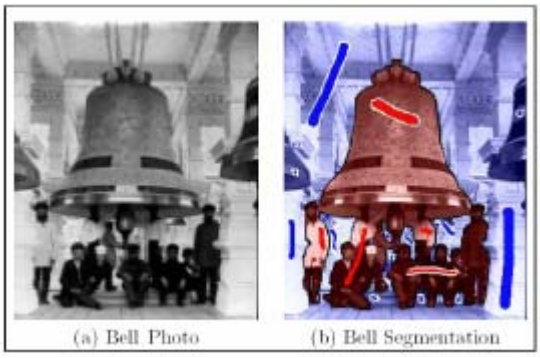

Figure 7: Image Segmentation examples taken from Boykov et. al. [1]. Interactive user input is used to guide the segmentation.

Bibliography

Summaries of Graph Cut algorithms in Computer Vision:

[1] Boykov, Y., Kolmogorov, V., An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision, *In IEEE Transactions on PAMI, Vol. 26, No. 9, pp. 1124-1137*, 2004

[2] Kolmogorov, K., Zabih, Z., What Energy Functions can be Minimized via Graph Cuts?, *IEEE Transactions on Pattern Analysis and Machine Intellegence, vol. 26, No. 2, pp. 147 - 159,* 2004

[3] Sinha, S. Graph Cut Algorithms in Vision, Graphics and Machine Learning, *Integrative Paper, November, 2004, UNC Chapel Hil,* 2004

Graph Cut algorithms used in image reconstruction

[4] Greig, D., Porteous, B.Seheult, A., Exact Maximum A Posteriori Estimation for Binary Images, J. *Royal Statistical Soc., Series B, vol. 51, no. 2, pp. 271-279*, 1989

[5] Boykov, Y., Veksler, O.Zabih, R., Fast Approximate Energy Minimization via Graph Cuts, *Proc. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222-123*, 2001

[6] Boykov, Y., Veksler, O.Zabih, R., Markov Random Fields with Efficient Approximations, *Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 648-655,* 1998

[7] Ishikawa, H., Geiger, D., Segmentation by Grouping Junctions, *Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 125-131*, 1998

Graph Cut algorithms used in stereo vision

[8] Birchfield, S., Tomasi, C., Multiway Cut for Stereo and Motion with Slanted Surfaces, Proc. *Int'l Conf. Computer Vision, pp. 489-495*, 1999

[9] Ishikawa, H., Geiger, D.Zabih, R., Occlusions, Discontinuities, and Epipolar Lines in Stereo, Proc*. Int'l Conf. Computer Vision, pp. 1033-1040*, 2003

[10] Kim, J., Kolmogorov, V., Visual Correspondence Using Energy Minimization and Mutual Information, *Proc. Int'l Conf. Computer Vision, pp. 508-515*, 2001

[11] Kolmogorov, V., Zabih, R., *Visual Correspondence with Occlusions Using Graph Cuts, PhD thesis, Stanford Univ.*, Dec. 2002

[12] Lin, M.H., Surfaces with Occlusions from Layered Stereo, *Int'l J. Computer Vision, vol. 1, no. 2, pp. 1-15*, 1999

Graph Cut algorithms used in image segmentation

[14] Boykov , Y., Kolmogorov, V., Computing Geodesics and Minimal Surfaces via Graph Cuts, Proc. *European Conf. Computer Vision, pp. 232-248*, 1998

Graph Cut algorithms used in multi-camera scene reconstruction

[15] Roy, S., Cox, I., A Maximum-Flow Formulation of the n-Camera Stereo Correspondence Problem, *Proc. Int'l Conf. Computer Vision, pp. 26-33*, 2003

[16] Kolmogorov, V., Zabih, R., Multi-Camera Scene Reconstruction via Graph Cuts, Proc*. European Conf. Computer Vision*, vol. 3, pp. 82-96, 2002

Graph Cut algorithms used in medical imaging

[17] Boykov, Y., Jolly, M.-P., Interactive Organ Segmentation Using Graph Cuts, *Proc. Medical Image Computing and Computer-Assisted Intervention, pp. 276-286*, 2000

[18] Boykov, Y., Jolly, M.-P., Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images, *Proc. Int'l Conf. Computer Vision, pp. 105-112*, 2001

[19] Kim, J., Zabih, R., Automatic Segmentation of Contrast-Enhanced Image Sequence, *Proc. Int'l Conf. Computer Vision, pp. 502-509,* 2003

[20] Kim, J., Fisher, J.Tsai, A., Wible, C., Incorporating Spatial Priors into an Information Theoretic Approach for FMRI Data Analysis, *Proc. Medical Image Computing and Computer-Assisted Intervention, pp. 62-71*, 2000

Max Flow Algorithms

[21] Ford, L., Fulkerson, D.,Flow in Networks*., Princeton University Press*, 1962.

[22] Goldberg, A.V., Tarjan, R. E., A new approach to the maximum flow problem, *Journal of the Association for Computing Machinery, vol 35, no. 4, pp 921-940,* Oct 1988.