

# グラフカット

石川 博

名古屋市立大学 大学院システム自然科学研究科  
〒467-8501 名古屋市瑞穂区瑞穂町字山の畑 1  
E-mail: hi@nsc.nagoya-cu.ac.jp

あらまし ビジョンや画像処理の多くの問題を, エネルギー最小化の問題と捉えることができる. そのエネルギー最小化の方法として, 近年, グラフの最小切断 (グラフカット) アルゴリズムが, ますます有用で人気のある手法となり, ビジョンや画像処理に盛んに応用されるようになってきた. 本稿では, グラフカットの離散最適化への応用を概観し, 主要な3つのアルゴリズムを詳述するとともに, 具体的にステレオやセグメンテーション等, ビジョンやグラフィックスへの応用を紹介する.

キーワード グラフカット, 最小切断, 離散最適化, マルコフ確率場

## Graph Cut

Hiroshi Ishikawa

Department of Information and Biological Sciences, Nagoya City University  
E-mail: hi@nsc.nagoya-cu.ac.jp

**Abstract** Many computer vision and image processing problems can be formulated as an energy minimization problem. Among several available optimization schemes, minimum-cut algorithms on graphs have emerged as an increasingly useful and popular tool for performing these optimizations. Here, we overview the graph-cut energy minimization as well as presenting the three main algorithms. We also review some of their applications in vision and graphics.

**Key words** Graph cut, Minimum cut, Discrete optimization, Markov Random Field

### 1. はじめに

近年, グラフの最小切断 (グラフカット) アルゴリズムを使ったエネルギー最小化が, ビジョンや画像処理, またグラフィックスに盛んに応用されるようになってきた. 用途は画像復元 [10], [28], ステレオ [10], [24], [26], [37], [49], [50], セグメンテーション [5]~[7], [11], [27], [30], [34], [38], [41]~[44], [48], [56], 動画画像解析 [51], [55] 等の中

心になっているが, 近年特に SIGGRAPH などグラフィックス関係でも, テクスチャ合成 [39], インタラクティブな前景セグメンテーション [42] [46] やビデオのセグメンテーション [41] [54], またインタラクティブ・フォトモンタージュ [1] など, 急速にポピュラーになってきた.

本稿では, グラフカットの主要な3つのアルゴリズムを詳述するとともに, 上記のような用途に具体的にどのように使われるかを紹介する.

## 2. エネルギー最小化

ビジョンや画像処理の多くの問題を、エネルギー最小化という枠組みで捉えることができる。本稿で紹介するグラフカットは、主に次の形のエネルギーを最小化するための手法である。

$$E(X) = \sum_{v \in V} g_v(X_v) + \sum_{(u,v) \in E} h_{uv}(X_u, X_v) \quad (1)$$

ここで、 $V$ はピクセルなど「場所」(サイトと呼ぶ)を表す集合、 $E \subset V \times V$ は、サイト間の隣接関係を表す集合で、 $(u,v) \in E$ のときサイト $u$ と $v$ は隣接していると考え、したがって、 $(u,v) \in E \Leftrightarrow (v,u) \in E$ である。これは $V$ を頂点の集合とし、 $E$ を辺の集合とする無向グラフと考えることができるが、グラフカットの「グラフ」は、このグラフのことではない。

問題は $V$ の各サイトに、有限個のラベルを1つずつ割り振ることである。ラベルの集合を $L$ として、以下、この割り振り方を $(V,L)$ の配置と呼ぶことにする。つまり、 $(V,L)$ の配置 $X$ とは $V$ の各サイト $v$ にラベル $X_v \in L$ を与える写像である。

$$\begin{aligned} X: V &\rightarrow L \\ v &\mapsto X_v \end{aligned}$$

式(1)は、このような $X$ にエネルギー $E(X)$ と呼ばれる数値を与える関数である。式(1)の最初の和は、 $V$ の各サイトについて、そのサイトとそこに割り当てられたラベルのみに依存する値である $g_v(X_v)$ を足し合わせたものだが、普通、データ項(data term)と呼ばれる。これは、各サイトにどのラベルを割り当てるかについて、画像などのデータから来る直接的な影響が、 $g_v$ を通してこの和に現れるからである。これに対して第2の和は、隣接するサイト間で $X$ に与えられるラベルがどのような関係にあるべきかについての、我々の事前知識を反映する。普通はこれが隣接サイト間のラベルの変化を少なくするものであるため、この項は平滑化項(smoothing term)と呼ばれる。

式(1)のようなエネルギーはマルコフ確率場(Markov Random Field: MRF)の最大事後確率

(MAP)推定のときによく現れる。データ $D$ を与えられたときに、その条件下で事後確率 $P(X|D)$ を最大化する $X$ を求めるのがMAP推定で、それは $X$ を与えた時のデータの尤度 $P(D|X)$ 及び $X,D$ の事前確率 $P(X), P(D)$ から、ベイズの式

$$P(X|D) = \frac{P(D|X)P(X)}{P(D)}$$

で与えられるが、 $D$ を固定しているので $P(D)$ は定数となり、 $P(D|X)P(X)$ を最大化する $X$ を求めればよいことになる。このうち、データ $D$ の尤度 $P(D|X)$ が(1)のデータ項に、 $X$ の事前確率 $P(X)$ が平滑化項に、それぞれ対数値の $-1$ 倍として現れる。

最も一般的には、ピクセルをサイトとし、上下左右の隣接、またはそれに加えて斜めの隣接を使う。しかし、用途によってはより離れたピクセルを隣接するものとして扱ったり[14]、ピクセル以外のものをサイトとすることもある[47]。

### 2.1 例：ノイズ除去

上でグラフカットで使われるエネルギー最小化の枠組みを概観したが、少し例を使って説明する。最も単純な例として、2値画像のノイズ除去を考える。今、0と1の2値の画像にノイズが入っているとす。これについて、エネルギーを以下のように定義する。まず、ラベルの集合 $L$ を $L = \{0, 1\}$ とし、与えられた画像 $Y$ のピクセルの集合を $V$ とする。また上下左右に隣接するピクセル $u, v$ の組 $(u, v)$ 全体からなる集合を $E \subset V \times V$ とする。

次にデータ項を

$$g_v(X_v) = \lambda |Y_v - X_v| \quad (2)$$

で定義する。ここで $Y_v$ は、与えられたノイズ入り画像 $Y$ のピクセル $v$ での値(0か1)である。ここでは $X$ と $Y$ のとり得る値は0か1のみだから、 $g_v(X_v)$ は $X_v = Y_v$ のとき0を、 $X_v \neq Y_v$ のとき定数 $\lambda$ を与える。

最後に平滑化項を

$$h_{uv}(X_u, X_v) = \kappa |X_u - X_v| \quad (3)$$

で定義する。上と同様に、 $h_{uv}$ は隣接するピクセル $u, v$ での $X$ の値 $X_u, X_v$ が異なる場合に定数 $\kappa$ を与

え、等しいときに0を与える。

式(1)(2)(3)で定義されたエネルギー  $E(X)$  を最小化するような  $X$  は、どのようなものだろうか。各ピクセル  $v$  において、 $X_v$  は  $Y_v$  と等しい値をとったほうが(2)は小さくなる。一方で、 $X_v$  は隣接するピクセル間であまり変化しないほうが(3)は小さくなる。例えば、ノイズ入り画像  $Y$  上で、周りと値が違う孤立したピクセルがあれば、そのピクセルでは  $X_v$  を  $Y_v$  と等しくするより周囲の値と合わせたほうが、総和は小さくなる。このように、 $E(X)$  を最小化することは、「隣接したピクセルの値はあまり激しく変わらない」という予想を表した事前分布に従いながら、 $X$  を元の画像  $Y$  にできるだけ近づけるというトレードオフを表している。そのトレードオフは、 $X$  と  $Y$  が異なることのコスト(2)と隣接したピクセルで値が違うことのコスト(3)によって明確に表されている。

### 3. グラフカット概観

#### 3.1 小 史

式(1)のエネルギーを最小化することは、一般にはNP困難である[10]。そのため、従来、反復条件付き最大化(Iterated Conditional Modes: ICM [4])やアニーリング法(Simulated Annealing [18])といった近似解法が使われてきたが、前者は局所最小解に落ち込みやすく、後者は理論的には大域最小値に収束することが保証されているものの、それには無限の時間を要し、現実にはどれだけ時間をかければよいか判りにくいという問題があった。

グラフカット(GC)は、オペレーションズ・リサーチでは少なくとも40年前から使われていた[22]。画像処理に導入されたのは80年代後半[20][21]で、前節で説明したような2値のノイズ除去において、大域最小化が可能であることが示された。このとき、GCによって得られた大域最小解をアニーリング法による結果と比べた結果、アニーリング法は平滑化しすぎる傾向があることが判明した。これはグラフカットによって大域最小解が得られたから初めて判ったことで、ある意味で画期的であった。90年代後半になって、GCはビジョンに再

導入された[9][26][27][50]。このとき、実数値をラベルに持つ場合に、平滑化項がラベルの差の絶対値について線形、つまり  $h_{uv}(X_u, X_v) = \lambda_{uv}|X_u - X_v|$  ( $\lambda_{uv}$  は  $u, v$  ごとに決まる定数)の形ならば、大域最小解が得られることが示された[27]。また、任意のラベルについて、 $h_{uv}(X_u, X_v)$  が、 $X_u$  と  $X_v$  が等しいとき0、異なるとき  $\lambda_{uv}$  を与えるPottsモデルの場合に、グラフカットを使って近似解を得ることの出来るアルゴリズムが与えられた[9]。これは後に改良され、 $h_{uv}$  が半距離(後述)であるとき、エネルギー(1)を大域最小値のある定数倍以内に最小化することを保証するアルゴリズムが現れた[10]。

その後、実数値をラベルに持つ場合[25]及び2値の場合[37]にGCで大域最小化可能な条件が正確に与えられた。

#### 3.2 現時点でできること

グラフカットのビジョンへの応用は、まだ活発な研究が行われ、発展途上にあるが、現時点でグラフカットに出来ることをエネルギー最小化の言葉でまとめれば、以下ようになる。

まず、ラベルが2つ(ここでは0と1とする)の場合、submodularityと呼ばれる次の条件が満たされるとき、グラフカットによって(1)のエネルギーは大域的に最小化できる[37]:

$$h_{uv}(0,0) + h_{uv}(1,1) \leq h_{uv}(0,1) + h_{uv}(1,0). \quad (4)$$

この結果の別の証明が[17]に与えられている。

次に、ラベルが一行に並んで順序があるとき ( $L = \{l_0, \dots, l_k\}$ ),  $h_{uv}(l_i, l_j) = \tilde{h}_{uv}(i - j)$  と書いて  $\tilde{h}$  が凸であれば、すなわち

$$\tilde{h}_{uv}(i+1) - 2\tilde{h}_{uv}(i) + \tilde{h}_{uv}(i-1) \geq 0$$

が成り立てば、グラフカットによって(1)は大域的に最小化できる[25]。これは、ラベルが等間隔の実数値 ( $L = \{a, a+b, a+2b, \dots, a+kb\}$ ,  $l_i = a+ib$ ) であるときに  $h_{uv}(X_u, X_v)$  が  $X_u - X_v$  の凸関数であればよいということを意味する。

以上がエネルギーを大域最小化できる場合であるが、他に重要な近似アルゴリズムが存在する。次

の3条件が全てのラベル  $\alpha, \beta, \gamma$  について成り立つとき、 $h(\alpha, \beta)$  は距離 (metric) であるという。

$$h(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta, \quad (5)$$

$$h(\alpha, \beta) = h(\beta, \alpha) \geq 0, \quad (6)$$

$$h(\alpha, \beta) \leq h(\alpha, \gamma) + h(\gamma, \beta). \quad (7)$$

もし (5) と (6) のみを満たすならば、 $h(\alpha, \beta)$  は半距離 (semimetric) であるという。エネルギー (1) の  $h_{uv}(X_u, X_v)$  が半距離であるとき、大域最小値の  $2c$  倍以内に最小化することを保証する近似アルゴリズムを [10] は与えた。ここで

$$c = \max_{u, v \in V} \left( \frac{\max_{X_u \neq X_v} h_{uv}(X_u, X_v)}{\min_{X_u \neq X_v} h_{uv}(X_u, X_v)} \right)$$

である。これらのアルゴリズムは  $\alpha$ - $\beta$  交換及び  $\alpha$  拡張と呼ばれる。半距離という条件はアルゴリズム中で使う2値の大域最小化で  $h_{uv}$  が正でなければならないという仮定に由来し、その後それが submodularity に緩和された結果、この条件も少し緩和されている (5.3 参照)。

以上のそれぞれについては、後で詳細に説明する。

実は、これらの結果の多くは、オペレーションズ・リサーチでは知られていた [22] [23]。特に [23] は submodularity 条件を緩和する方法も与えている (これについては近くレビューが出る [36])。また、平滑化項がラベルの差の絶対値について線形の場合も、全く用途は異なるが、マルチプロセッサへのタスクの割り振りアルゴリズムに同様の方法が使われている [40]。ただ [25] の結果は知られていなかったようである。

### 3.3 他のエネルギー最小化法との比較

最近になって、信念伝播法 (Belief Propagation: BP [45])、ツリー重み再配分メッセージ伝達法 (Tree-Reweight Message Passing: TRW [53] [33]) 等の新しい近似解法が現れてきた。これらの方法と GC とで (1) の形のエネルギーを最小化する実験的評価が最近行われた [52]。Potts モデルを使ったステレオとセグメンテーションについて GC, BP, TRW を実験的に比較している。全般にこれらの方法は、比較に使われた ICM に比べて非常に改善さ

れており、各ベンチマークについて、(最良の方法はベンチマーク毎に変わるものの) エラー 1 パーセント以下の結果を出している。隣接サイト数が上下左右の4方向であるこの比較では TRW が GC よりよい結果を出しているが、[35] によれば、例えば隠蔽をモデル化したステレオなど、普通よりやや隣接数の多い場合には、最小化されたエネルギーにおいても、また ground truth と比較したエラーにおいても、GC が他の2つよりかなり性能がよい。

## 4. グラフとその最小切断

本節ではグラフカットで使用するグラフと、その最小切断について述べる。最小切断、最大流等についての詳細は [2] [13] を参照されたい。 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  を有向グラフとする。すなわち、 $\mathcal{V}$  は有限個の頂点の集合で、 $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$  は辺の集合である。頂点の組  $(u, v)$  が  $\mathcal{E}$  に含まれるとき、頂点  $u$  から頂点  $v$  への辺があるという。各辺は重みづけをされているとする。つまり、辺  $(u, v) \in \mathcal{E}$  にそれぞれ  $c(u, v)$  という実数値の重みが定義されているとする。以下、便宜上全ての頂点の組  $(u, v)$  に重み  $c(u, v)$  が定義されていて、 $(u, v) \notin \mathcal{E}$  なら  $c(u, v) = 0$  であることとする。このグラフの2頂点  $s, t$  を選んだとき、 $\mathcal{G}$  の  $s, t$  についての切断とは、頂点の集合  $\mathcal{V}$  を2つの部分集合  $S \subset \mathcal{V}$  と  $T = \mathcal{V} \setminus S$  に、 $s \in S, t \in T$  となるように2分割することである。以下この切断を  $(S, T)$  で表す。このとき、この切断  $(S, T)$  のコストとは、 $S$  側から  $T$  側へまたがる辺の重みの総和

$$c(S, T) = \sum_{u \in S, v \in T} c(u, v)$$

である。逆向き、つまり  $T$  側から  $S$  側へまたがる辺の重みは可算しないことに注意する。最小切断問題とは、 $s, t$  についての切断のうち、コストが最小のものを見つける問題である。

最小切断問題は、最大流問題の線形計画法における双対にあたり、最大流最小切断定理 [16] により、最小切断は最大流と一致する。辺の重みが全て非負であるときに最大流問題を多項式時間で解くアルゴリズムは多数知られており、それらがグラフカットの有用性の元となっているが、

それらは大きく分けて2種類に分けられる。一つはFordとFulkersonの“augmenting path”アルゴリズム [16] で、もう一つはGoldbergとTarjanによる“push-relabel”アルゴリズム [12] [19] である。漸近的計算量は、 $n$  を頂点の数、 $m$  を辺の数として、前者が  $O(n^3)$ 、後者が  $O(nm \log(n^2/m))$  であるから、push-relabel アルゴリズムの方が速い。画像関係では普通、辺の数は頂点の数の定数倍だから、後者は  $O(n^2 \log n)$  となる。Boykov ら [8] は、augmenting path の一種であるDinicのアルゴリズム [15]、push-relabel アルゴリズム、そしてこれもまた augmenting path の一種である自らの新アルゴリズムを、画像復元、ステレオ、セグメンテーションにおいて実験的に比較した結果、新アルゴリズムが最も高速であるという結果を得ている。

また、同じエネルギーを少しずつ変えて繰り返し最小化するとき効率的な最小切断アルゴリズムもある。これは、動画のセグメンテーション [31] や、各サイトでラベルを一つ固定したときの全体の最小エネルギーを求め、それをこのサイトで他のラベルを選択したときと比較して、特定の選択に他と比べてどれだけ自信があるかの指標としたものを効率的に求める [32] ためにも使われている。

## 5. グラフカットによるエネルギー最小化

本節では、3.2で説明したグラフカットによる3つのエネルギー最小化法を、詳細に説明する。

### 5.1 2値の大域最小化

まず、最も簡単な2値の場合 ( $L = \{0, 1\}$ ) について述べる。この場合、各サイトに割り当てられるラベルは二つに一つであるわけだが、これがちょうど、最小切断の  $S, T$  のどちら側に属しているかに対応するので解りやすい。

今、(1)のエネルギーを考える。これに対応して、有向グラフ  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  を作る (図1)。まず、 $V$  のサイト一つにつき一つずつ頂点を定義する：

$$\mathcal{V}_1 = \{w_v | v \in V\}.$$

そして  $\mathcal{V} = \mathcal{V}_1 \cup \{s, t\}$  とする。つまり、 $\mathcal{G}$  は各サイ

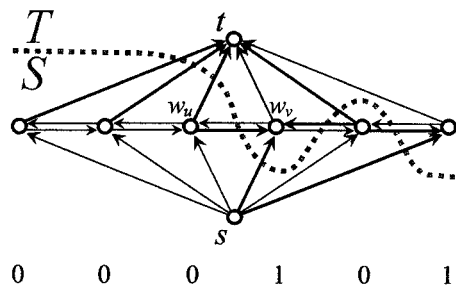


図1 2値MRF最小化のためのグラフ。下部の数はその上の頂点に割り当てられたラベルを表す。太い矢印が切断されている辺である。

ト  $v \in V$  それぞれに対応する頂点  $w_v$  と、さらに2つの頂点  $s, t$  を持つ。辺は、まず  $V$  から作った頂点それぞれについて、 $s$  からと  $t$  へそれぞれ一本ずつ辺を作る：

$$\mathcal{E}_1 = \{(s, w_v) | w_v \in \mathcal{V}_1\} \cup \{(w_v, t) | w_v \in \mathcal{V}_1\}.$$

そして隣接するサイトに対応する頂点間にも辺を作る：

$$\mathcal{E}_2 = \{(w_u, w_v) | (u, v) \in E\}.$$

これらをあわせて  $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$  とする。

さて、このグラフ  $\mathcal{G}$  の  $s, t$  についての任意の切断  $(S, T)$  に、 $(V, L)$  の配置  $X$  を次のように対応させる：

$$X_v = \begin{cases} 0 & (w_v \in S \text{ のとき}), \\ 1 & (w_v \in T \text{ のとき}). \end{cases} \quad (8)$$

つまり、サイト  $v$  に、それに対応する  $\mathcal{G}$  の頂点  $w_v$  が切断の  $s$  側にあるならば0、 $t$  側にあるならば1のラベルを割り当てる。このように、 $\mathcal{G}$  の  $s, t$  についての切断と  $(V, L)$  の配置は1対1に対応する。

最小切断は、前節で述べたように、最大流アルゴリズムを使って効率よく大域最小化できる。したがって、エネルギー (1) を最小化する  $(V, L)$  の配置を求めるためには、例えば配置のエネルギーとそれに対応する切断のコストが等しいか、あるいは差が定数であるというように、最小エネルギーを与える配置と最小切断が対応していればよい。

これを2.1の例について考えると、次のように

定義すればよいことがわかる:

$$c(s, w_v) = g_v(1) \quad (9)$$

$$c(w_v, t) = g_v(0) \quad (10)$$

$$c(w_u, w_v) = h_{uv}(0, 1) \quad (11)$$

このとき, ある切断  $(S, T)$  に対応する配置  $X$  を考える.

まず, 一つのサイトのみに依存する項 (データ項) について見ると, まず  $w_v \in S$  ならば (8) により  $X_v = 0$  だが, このとき辺  $(w_v, t)$  が切断されているから (10) により  $g_v(X_v) = g_v(0)$  が切断のコスト  $c(S, T)$  に加えられることがわかる. また  $w_v \in T$  ならば  $X_v = 1$  だが, このとき辺  $(s, w_v)$  が切断されているから, (9) により  $g_v(X_v) = g_v(1)$  が  $c(S, T)$  に加えられる.

次に, 隣接する二つのサイトに依存する項 (平滑化項) を考える. 隣接するサイト  $u, v$  について,  $w_u \in S, w_v \in T$  ならば辺  $(w_u, w_v)$  が切断されているから, (11) により  $h_{uv}(0, 1) = \kappa$  が  $c(S, T)$  に加えられる (辺  $(w_v, w_u)$  は  $T$  から  $S$  に向かっているから, その辺の重みは加えない). 同様に  $w_u \in T, w_v \in S$  のときも,  $h_{vu}(0, 1) = \kappa$  が  $c(S, T)$  に加えられる. これら二つの場合は (8) によりそれぞれ  $(X_u, X_v) = (0, 1)$  と  $(X_u, X_v) = (1, 0)$  に対応するが, それぞれの場合に  $c(S, T)$  に加えられる値は, ちょうどエネルギー (1) の平滑化項に等しいことがわかる. また  $w_u, w_v$  の両方が  $S, T$  のどちらかに属する場合は,  $h_{uv}(0, 0) = h_{uv}(1, 1) = 0$  に対応して,  $c(S, T)$  には何も加えられない.

このように, 配置  $X$  のエネルギー (1) とそれに対応する切断のコストが正確に一致することがわかる. 以上のような構成は [21] による.

上のようにグラフを構成すれば, 最大流アルゴリズムを使って最小切断を見つけることで, エネルギー (1) の大域最小値と, そのときの配置が求められることがわかるが, そのためには辺の重みが全て非負でなければならない. 従って, 上の構成そのままでは,  $g_v, h_{uv}$  が正であるという条件が (1) に課せられることになる. 他にも, 上の構成は  $h_{uv}(0, 0) = h_{uv}(1, 1) = 0$  を仮定している.

しかし, 辺の重みの決め方にはもう少し自由度がある. 例えば,  $w_v$  は  $S$  か  $T$  のどちらかに必ず属するのだから,  $(s, w_v), (w_v, t)$  のどちらかは常に切断される. 従って,  $c(s, w_v), c(w_v, t)$  に同じ値を加えても, どちらを選ぶかという選択には影響しない. このため,  $g_v$  は任意の実数でよいことがわかる. つまり, (9)(10) に従って  $c(s, w_v), c(w_v, t)$  を定義した後,  $-\min\{c(s, w_v), c(w_v, t)\}$  を  $c(s, w_v), c(w_v, t)$  に加えれば, 両方とも非負になるからである.

このような変形をすることによって, グラフで最小化できるためにどのような条件が必要十分であるかを決めたのが [37] であり, その条件である submodularity は (4) に挙げた.

今,  $h_{uv}$  が全て (4) を満たすとして, 辺の重みを次のように定義する:

$$c(s, w_v) = g_v(1) + \sum_{(u,v) \in E} \{h_{vu}(1, 0) - h_{vu}(0, 0)\} \quad (12)$$

$$c(w_v, t) = g_v(0) + \sum_{(u,v) \in E} \{h_{uv}(1, 0) - h_{uv}(1, 1)\} \quad (13)$$

$$c(w_u, w_v) = h_{uv}(0, 1) + h_{uv}(1, 0) - h_{uv}(0, 0) - h_{uv}(1, 1) \quad (14)$$

その後,  $c(s, w_v)$  または  $c(w_v, t)$  が負である  $w_v$  があれば, 上記のように両者に定数を加えて非負にする.

このようにすれば, submodularity より  $c(w_u, w_v)$  は非負だから, 全ての重みを非負にできる. また, このようにして決めたグラフの最小切断が対応 (8) によりエネルギー (1) を最小化する  $(V, L)$  の配置に対応する. 詳細は [37] を参照されたい.

## 5.2 多値の大域最小化

次に, 大域最小化が可能なもう一つの場合である, ラベルが一行に並んで順序がある ( $L = \{l_0, \dots, l_k\}$ ) 場合について説明する.

前と同様に, 特別な頂点  $s, t$  を持つグラフを定義する (図2). まず, 各サイト  $v \in V$  について一連の頂点  $\{w_v^1, \dots, w_v^k\}$  を定義する. 表記の簡単のため,



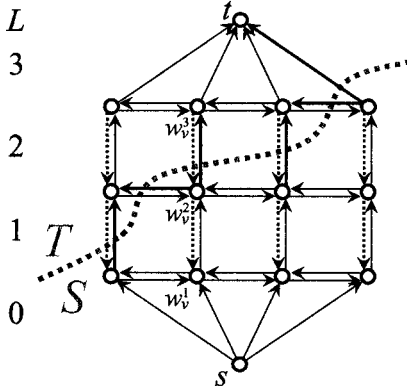


図2 多値MRF最小化のためのグラフ

以下では  $w_v^0$  は  $s$  の別名,  $w_v^{k+1}$  は  $t$  の別名とする.

次に  $i=0, 1, \dots, k$  について辺  $(w_v^i, w_v^{i+1})$ , また逆方向に,  $i=1, \dots, k-1$  について辺  $(w_v^{i+1}, w_v^i)$  (図の点線矢印) を作り, 後者の重みは無限大に設定する:

$$c(w_v^{i+1}, w_v^i) = \infty. \quad (i=1, \dots, k-1)$$

これによつて  $s = w_v^1 \rightarrow w_v^2 \rightarrow \dots \rightarrow w_v^{k-1} \rightarrow w_v^k = t$  の一列の辺のうち, 一つだけが切断されることが保証される. (なぜなら,  $s = w_v^1, w_v^2, \dots, w_v^k, w_v^{k+1} = t$  の最初の  $s$  は  $S$  に属し最後の  $t$  は  $T$  に属するから, この並びのうち一度は  $S$  から  $T$  に変わらなければならないが,  $T$  から  $S$  へ変わると無限大の辺を切断することになるから, ちょうど一度だけしか切断できないからである.)

このため, 前と同様に, このグラフの切断と  $(V, L)$  の配置との間に次のような1対1対応がつく:

$$X_v = l_i \quad (w_v^i \in S, w_v^{i+1} \in T \text{ のとき}). \quad (15)$$

だから,

$$c(w_v^i, w_v^{i+1}) = g_v(l_i) \quad (16)$$

と定義することによって, 2値のときと同様にデータ項は切断のコストにそのまま反映される. もし負の重みがあれば, 同じ  $v$  の辺  $c(w_v^i, w_v^{i+1})$  全てに同じ値を加えることによって非負にできるので,  $g_v$  に制限はない.

さらに, 平滑化項  $h_{uv}(X_u, X_v)$  を実現するために

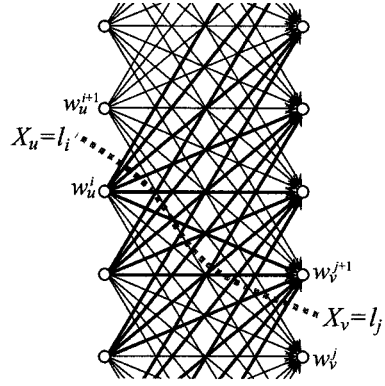


図3 多値MRF最小化のためのグラフの平滑化辺

辺を作るのだが, 図2は水平方向(同じ  $i$  の  $w_u^i$  と  $w_v^i$  の間)の辺のみの場合である. この場合は, これらの辺に全て同じ正数  $\kappa$  を重みとして与えれば,  $h_{uv}(l_i, l_j) = \kappa|i-j|$  を実現していることがわかる.

前述のように, 一般には  $h_{uv}(l_i, l_j) = \tilde{h}_{uv}(i-j)$  と書けて  $\tilde{h}$  が凸であれば, すなわち

$$\tilde{h}_{uv}(i+1) - 2\tilde{h}_{uv}(i) + \tilde{h}_{uv}(i-1) \geq 0 \quad (17)$$

が成り立てば, グラフカットによつて(1)のエネルギーは大域的に最小化できる[25].

そのためには, 水平方向だけでなく, 斜め方向の辺も必要とする(図3). つまり, 隣接するサイト  $u, v$  について, 一般には  $(w_u^i, w_v^j)$  の形の辺がすべて存在する. そしてこれらの辺には

$$c(w_u^i, w_v^j) = \quad (18)$$

$$\frac{1}{2} \{ \tilde{h}_{uv}(i-j+1) - 2\tilde{h}_{uv}(i-j) + \tilde{h}_{uv}(i-j-1) \}$$

という重みを与える. 凸条件より, これは非負となる.

さて, 今, 切断  $(S, T)$  が隣接するサイト  $u, v$  上で辺  $(w_u^i, w_u^{i+1})$  と  $(w_v^j, w_v^{j+1})$  をそれぞれ切断している, つまり  $w_u^i, w_v^j \in S, w_u^{i+1}, w_v^{j+1} \in T$  であるとする. すると,  $\{(w_u^l, w_v^m) | 0 \leq l \leq i, j+1 \leq m \leq k+1\}$  及び  $\{(w_v^l, w_u^m) | 0 \leq l \leq j, i+1 \leq m \leq k+1\}$  の辺は全て切断される. これらの辺の重みの総和を18に従って計算すると, ほとんどの項は打ち消しあつて,

$$\tilde{h}_{uv}(i-j) + \tilde{h}_{uv}(k+2) + r_{uv}(i) + r_{uv}(j)$$

が残る。ここで

$$r_{uv}(i) = -\frac{1}{2} \{ \tilde{h}_{uv}(k+1-i) + \tilde{h}_{uv}(i+1) \}$$

である。 $\tilde{h}_{uv}(k+2)$ は定数なので無視できる。また、 $r_{uv}(i), r_{uv}(j)$ はあらかじめデータ項から引いておけばよい。従って、(16)の代わりに

$$c(w_v^i, w_v^{i+1}) = g_v(l_i) - \sum_{(u,v) \in E} r_{uv}(i). \quad (19)$$

と定義することによって、上の総和はちょうど $\tilde{h}_{uv}(i-j)$ すなわち $h_{uv}(l_i, l_j)$ を表すことになる。以上より、切断のコストは対応する配置のエネルギー（+定数）に等しくなり、最小切断は大域的な最小配置を与える。

### 5.3 多値の近似最小化

上述の多値大域最小化には、平滑化項が凸であるという条件があるが、多くのビジョンの問題では、これが満たされない。そのため、近似解ではあるが、以下に述べる $\alpha$ - $\beta$ 交換及び $\alpha$ 拡張の手法が実際にはよく使われる。

$\alpha$ - $\beta$ 交換も、 $\alpha$ 拡張も、ある配置から別の配置への「移動」を定義する。以下、 $\alpha, \beta \in L$ とする。

まず、配置 $X$ からの $\alpha$ - $\beta$ 交換とは、 $X$ が $\alpha$ または $\beta$ を割り当てているサイトのみで $\alpha$ と $\beta$ を入れ替えることだけを許す移動である。つまり、配置 $X$ から $X'$ への移動は

$$X_u \neq X'_u \Rightarrow X_u, X'_u \in \{\alpha, \beta\}$$

を満たすとき $\alpha$ - $\beta$ 交換であるという。

次に、配置 $X$ からの $\alpha$ 拡張とは、 $X$ が $\alpha$ を割り当てているサイトを増やすことだけを許す移動である。つまり、配置 $X$ から $X'$ への移動は

$$X_u \neq X'_u \Rightarrow X'_u = \alpha$$

を満たすとき $\alpha$ 拡張であるという。

ある配置から始めて、その配置から最もエネルギーを減少させる $\alpha$ - $\beta$ 交換または $\alpha$ 拡張を2値グラフカットで見つけ、それを繰り返すというのが、

ここでいう多値の近似最小化アルゴリズムである。

従って、 $\alpha$ - $\beta$ 交換アルゴリズムでは、配置 $X$ について、 $X_v = \alpha$ または $X_v = \beta$ となっているサイト $v$ だけに注目し、その上で $\alpha$ と $\beta$ を交換する・しないという2値についてエネルギーを定義する。その上で、その問題を上記2値の場合のグラフカットを使って最小化する。これを全ての $\alpha, \beta \in L$ の組について繰り返し、最良の移動を見つけ、移動先を新しい $X$ として繰り返す。

また、 $\alpha$ 拡張アルゴリズムも同様に、配置 $X$ について $\alpha$ にラベルを変える・変えないという2値についてエネルギーを定義して、2値グラフカットを使って最小化し、これを全ての $\alpha \in L$ について繰り返し、最良の移動を見つけ、移動先を新しい $X$ として繰り返す。

これらのアルゴリズムは、[37]によれば、 $h_{uv}$ が

$$h_{uv}(\alpha, \alpha) + h_{uv}(\beta, \beta) \leq h_{uv}(\alpha, \beta) + h_{uv}(\beta, \alpha).$$

を全ての $\alpha, \beta \in L$ について満たすとき、 $\alpha$ - $\beta$ 交換アルゴリズムが使え、

$$h_{uv}(\alpha, \alpha) + h_{uv}(\beta, \gamma) \leq h_{uv}(\alpha, \gamma) + h_{uv}(\beta, \alpha).$$

を全ての $\alpha, \beta, \gamma \in L$ について満たすとき、 $\alpha$ 拡張アルゴリズムが使える。

## 6. グラフカットの応用

本節では、グラフカットを使ったエネルギー最小化の応用のいくつかを紹介する。

### 6.1 ステレオ

まず、ステレオへの応用を紹介する。密な2眼ステレオにおいては、典型的にはサイトは左画像のピクセルと1対1に対応し、ラベルは各左ピクセルと対応する右ピクセルとの間の視差（ディスパリティ）を表す。

今、左右の画像 $I_L(i, j), I_R(i, j)$ が与えられていて、左右画像の同じ $j$ 座標のピクセルが対応するエッジライン上にあるとする。つまり、左画像のピクセル $(i, j)$ と右画像の $(i+d, j)$ が3次元内の同じ点の投影であるときに、視差が $d$ である。



この場合、左画像の各ピクセル  $(i, j)$  毎にサイト  $v(i, j) \in V$  を一つ定義する。サイト間の隣接の定義は、普通、上下左右の4方向隣接か、斜めを入れた8方向隣接を使う。ラベルは一定の範囲の視差の値を表し、ここでは  $L = 0, 1, \dots, k$  とする（もちろん視差は整数値である必要はない）。

エネルギー (1) において、データ項  $g_v(X_v)$  は、

$$g_v(i, j)(d) = D(I_L, I_R, i, j, d)$$

とする。ここで、 $D(I_L, I_R, i, j, d)$  は視差が  $d$  であるときに、与えられたデータ  $I_L, I_R$  と、その視差との整合性を、視差から同じ点の投影であるとされる2ピクセルを比べることによって調べる。ここは自由度が大きく、いろいろ工夫できる部分である。例えば [24] では複数スケールのウェーブレットを特徴として使って、その中から最も自信のある特徴を選択している。また [10] では比較する左右の点を微妙に動かして比べて最小値をとり、さらに上限を定めるということをしている：

$$D(I_L, I_R, i, j, d) = \{\min(D_f(i, d), D_r(i, d), \text{const.})\}^2.$$

ただし

$$D_f(i, d) = \min_{d-\frac{1}{2} \leq x \leq d+\frac{1}{2}} |I_L(i, j) - I_R(i+x, j)|,$$

$$D_r(i, d) = \min_{i-\frac{1}{2} \leq x \leq i+\frac{1}{2}} |I_L(x, j) - I_R(i+d, j)|.$$

である。

次に、(1) の平滑化項  $h_{uv}$  であるが、これは見つけるべき曲面についての事前知識を表す。普通、隣接した点は近い視差を持つという仮定をする。グラフカットを使うためには、ここに条件がつく。前節で述べたように、 $h_{uv}$  の形によって使えるアルゴリズムが変わってくるからである。

視差を等間隔の実数値として 5.2 の多値アルゴリズムを使う [24] [26] [49] [50] のが最も自然であるが、 $h_{uv}$  が凸でなければならないという制限により、深さの突然の変化に対応しきれないで潰してしまうという問題がある。

文献では Potts モデルを使って  $\alpha$ - $\beta$  交換あるい

は  $\alpha$  拡張アルゴリズムを使ったものが多い。Potts モデルは基本的に等しいか等しくないかしか区別しないので、深さが区分的に定数、つまり領域ごとに視線と直交するような平面からなるような場合にはうまくいく。しかし、そうでない場合には、区分的に定数な解を無理に求める傾向がある。この問題を避けるために、[3] はラベルに対応するものを視差そのものではなくて視差を与えるアファイン関数とし、ラベル付けと、各ラベル毎の視差関数の推定を交互に繰り返すということをしている。

## 6.2 セグメンテーション

インタラクティブなセグメンテーションでの成功は、最近になってグラフカットが急速にポピュラーになった大きな理由の一つである。

ラベル割り当てはセグメンテーションそのものであるから、基本的に原理は単純である。スネークなど境界を見つけるセグメンテーションと違って、次元に左右されないから、医療画像などの高次元画像 [6] [7] やビデオ [41] [54] のセグメンテーションなどに、近年よく使われるようになった。グラフカットを使ったセグメンテーションには最近特に多くのバリエーションが発表されているが [5] ~ [7], [11], [30], [34], [38], [41] ~ [44], [48], これらについては、最近の論文 [6] を参照されたい。

以下では [7] に従って、グラフを使った典型的なセグメンテーション法を紹介する。

グレイスケール画像  $I(i, j)$  が与えられたとする。サイトの定義はステレオのときと同様に、各ピクセル  $p$  毎にサイト  $v_p \in V$  を一つ定義し、隣接の定義は、4方向か8方向の隣接を使う。（当然、高次元の場合は隣接数が増える。）ラベルは2値  $L = \{0, 1\}$  として、0 が背景、1 が前景を表すこととする。

データ項は、与えられた画像の明るさが、背景らしいか前景らしいかを表す数値を返すわけだが、[7] では、まずユーザーに、マウスで太い線を引く要領で、前景と背景の典型的と言える部分を大雑把に指定させる。これから前景と背景の指定部分の明るさのヒストグラム  $\theta(c, l)$ ,  $l = 0, 1$  を得る。ここで  $\theta(c, l)$  は、背景 ( $l = 0$ )、前景 ( $l = 1$ ) それぞ

れに指定された部分の中での明るさ $c$ の相対頻度 ( $\sum_c \theta(c, l) = 1$ ) を表す. これを使って, データ項は

$$g_{v(p)}(l) = -\log \theta(I(p), l)$$

とする. 右辺は, 例えば  $l=0$  なら背景に指定された部分で明るさ  $I(p)$  がたくさん使われていればいるほど小さくなるので, 背景らしい明るさほどエネルギー最小化で好まれるということになる.

平滑化項は, [7] では次のものを使っている:

$$h_{v(p)v(q)}(l, l') = \begin{cases} 0 & (l = l') \\ \frac{\lambda e^{-\kappa\{I(p)-I(q)\}^2}}{\text{dist}(p, q)} & (l \neq l') \end{cases}$$

ただし,  $\lambda, \kappa$  は定数,  $\text{dist}(p, q)$  はピクセル  $p, q$  間の距離である. 指数部分は, 明るさが近いピクセル間は切りにくくし, コントラストの高い境界で前景と背景を切る効果がある.

以上で定義したエネルギーを使って, グラフカットにより2値大域最小化をすることで, セグメンテーションが得られる. 平滑化項は Potts だから, 大域最小化が可能である.

以上が [7] で, これを次のように改良したのが [46] である. まず, グレイスケールではなくカラー画像で同じことをするために, 背景と前景の色の分布のモデルとして, ヒストグラムではなくてガウス混合モデルを使う. また, グラフカットを一度だけ使うのではなくて, グラフカットでセグメンテーションをしたからそれからまた背景と前景のモデルを学習し, 再びセグメンテーションをするということを繰り返すことによって, 精度を上げている. また, そのおかげで最初にユーザーが指定するのは背景だけでよくなっている.

ビデオのセグメンテーション [54] も, ほぼ同様に行うことができる.

### 6.3 フォトモンタージュ

最後に, インタラクティブ・フォトモンタージュ [1] を紹介する. これは複数の写真を合成して一つの写真を作るのだが, その継ぎ目ができるだけ目立たないようにグラフカットを使う.

ユーザーはどのような目的で写真を合成するかを指定する. 例として, ここでは集合写真を連続して何枚か ( $I_1, \dots, I_k$ ) とったとして, ユーザーはマウスで絵を描くように顔を指定して, その顔について使いたい (例えば目をつむっていない) 写真を指定するとする. その顔に属するピクセルは, 使いたい写真, 即ちラベルを指定されたことになる.

サイトの定義はセグメンテーションの時と同じである. ラベルは写真の中から1枚を指定する ( $L = \{1, \dots, k\}$ ).

データ項は, ピクセル  $p$  が写真  $l'$  から来るべきであると指定されていれば,

$$g_{v(p)}(l) = \begin{cases} 0 & (l = l') \\ M & (l \neq l') \end{cases}$$

で定める. ここで  $M$  は比較的大きな正の定数である. ピクセル  $p$  にユーザーからの指定がなければ,  $g_{v(p)}(l)$  は0にする.

平滑化項は,

$$h_{v(p)v(q)}^A(l, l') = \text{dist}(I_l(p), I_{l'}(p)) + \text{dist}(I_l(q), I_{l'}(q))$$

とする. ここで  $\text{dist}$  は色空間での距離である.

このように定義された  $h_{v(p)v(q)}^A$  は距離 (3.2 参照) であるから, [1] はこのエネルギーを  $\alpha$  拡張アルゴリズムを使って最小化する.

これは単純に色が近いところで継ぎ合わせているだけだが, [1] は他にも, 色の勾配の近いところで継ぎ合わせる平滑化項:

$$h_{v(p)v(q)}^B(l, l') = \text{dist}(\nabla I_l(p), \nabla I_{l'}(p)) + \text{dist}(\nabla I_l(q), \nabla I_{l'}(q))$$

や両者の和, さらにエッジの近くを好むように, これらをエッジの強度で割ったものなどを提案している. (こちらの  $\text{dist}$  は6次元の色勾配空間での距離.)

## 7. む す び

グラフカットは, 大域的最小解を得ることができ, 従って数値的にロバストである. また, 現代の

コンピュータを使えば速度も実用的なものになってきている [8]。そのため、グラフカットは、ここ 5 年程で急速に使われ始め、大変ポピュラーになってきている。本稿では、そのグラフカットの主要な 3 つの構成と、それらのステレオやセグメンテーション等、ビジョンやグラフィックスへの応用を紹介した。

最後に、グラフカットの実装例は、V. Kolmogorov (有名な Kolmogorov とは関係ないそうである) のホームページ [57] にいくつかある。また、N. R. Howe のページ [58] にも MatLab 用のコードがある。

### 参考文献

- [1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive Digital Photomontage. *ACM Transactions on Graphics (Proc. SIGGRAPH2004)*, 23(3):294–302, 2004.
- [2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, NJ, U.S.A. 1993.
- [3] S. Birchfield and C. Tomasi. Multiway Cut for Stereo and Motion with Slanted Surfaces. In *ICCV'99*, I:489–495, 1999.
- [4] J. Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society, Series B*, 48:259–302, 1986.
- [5] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive Image Segmentation Using an Adaptive GMMRF Model. In *ECCV2004*, 428–441, 2004.
- [6] Y. Boykov and G. Funka-Lea. Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.
- [7] Y. Boykov and M-P. Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. In *ICCV2001*, I:105–112, 2001.
- [8] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.
- [9] Y. Boykov, O. Veksler, and R. Zabih. Markov Random Fields with Efficient Approximations. In *CVPR'98*, 648–655, 1998.
- [10] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.
- [11] M. Bray, P. Kohli, and P. H. A. Torr. Posecut: Simultaneous Segmentation and 3D Pose Estimation of Humans Using Dynamic Graph-cuts. In *ECCV2006*, II:642–655, 2006.
- [12] B. V. Cherkassky and A. V. Goldberg. On implementing push-relabel method for the maximum flow problem. In *Proc. 4th International Programming and Combinatorial Optimization Conference*, 157–171, 1995.
- [13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (Second Edition)*. McGraw-Hill, New York. 2001.
- [14] D. Cremers and L. Grady. Statistical Priors for Efficient Combinatorial Optimization Via Graph Cuts. In *ECCV2006*, III:263–274, 2006.
- [15] E. A. Dinic. Algorithm for Solution of a Problem of Maximum Flow in Networks with Power Estimation. *Soviet Math. Dokl.*, 11, 1277–1280, 1970.
- [16] L. R. Ford and D. R. Fulkerson. Maximal Flow Through A Network. *Canadian Journal of Mathematics* 8:399–404, 1956.
- [17] D. Freedman and P. Drineas. Energy Minimization via Graph Cuts: Settling What is Possible. In *CVPR2005*, II:939–946, 2005.
- [18] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
- [19] A. V. Goldberg and R. E. Tarjan. A New Approach to the Maximum-Flow Problem. *Journal of the ACM*, 35:921–940, 1988.
- [20] D. M. Greig, B. T. Porteous, and A. H. Seheult. Discussion of: On the Statistical Analysis of Dirty Pictures (by J. E. Besag.). *Journal of the Royal Statistical Society, Series B*, 48:282–284, 1986.
- [21] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact Maximum A Posteriori Estimation for Binary Images. *Journal of the Royal Statistical Society, Series B*, 51:271–279, 1989.
- [22] P. L. Hammer. Some Network Flow Problems Solved With Pseudo-Boolean Programming. *Operations Research*, 13:388–399, 1965.
- [23] P. L. Hammer, P. Hansen, and B. Simeone. Roof Duality, Complementation and Persistency in Quadratic 0-1 Optimization. *Mathematical Programming*, 28:121–155, 1984.
- [24] H. Ishikawa. Multi-scale Feature Selection in Stereo. In *CVPR'99*, I:1132–1137, 1999.
- [25] H. Ishikawa. Exact Optimization for Markov Random Fields with Convex Priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, 2003.
- [26] H. Ishikawa and D. Geiger. Occlusions, Discontinuities, and Epipolar Lines in Stereo. In *ECCV'98*, 232–248, 1998.

- [27] H. Ishikawa and D. Geiger. Segmentation by Grouping Junctions. In *CVPR'98*, 125–131, 1998.
- [28] H. Ishikawa and D. Geiger. Mapping Image Restoration to a Graph Problem. In *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Antalya, Turkey, 1999.
- [29] H. Ishikawa and D. Geiger. Higher-dimensional Segmentation by Minimum-cut Algorithm. *Ninth IAPR Conference on Machine Vision Applications (MVA 2005)*, 488–491, 2005.
- [30] O. Juan and Y. Boykov. Active Graph Cuts. In *CVPR2006*, I:1023–1029, 2006.
- [31] P. Kohli and P. H. S. Torr. Efficiently Solving Dynamic Markov Random Fields Using Graph Cuts. In *ICCV2005*, II:922–929, 2005.
- [32] P. Kohli and P. H. S. Torr. Measuring Uncertainty in Graph Cut Solutions—Efficiently Computing Min-marginal Energies Using Dynamic Graph Cuts. In *ECCV2006*, II:20–43, 2006.
- [33] V. Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.
- [34] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Probabilistic Fusion of Stereo with Color and Contrast for Bilayer Segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(9):1480–1492, 2006.
- [35] V. Kolmogorov and C. Rother. Comparison of Energy Minimization Algorithms for Highly Connected Graphs. In *ECCV2006*, II:1–15, 2006.
- [36] V. Kolmogorov and C. Rother. Minimizing Non-submodular Functions with Graph Cuts - A Review. *IEEE Trans. Pattern Analysis and Machine Intelligence*, To appear, 2007.
- [37] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized via Graph Cuts? *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.
- [38] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Obj Cut. In *CVPR2005*, I:18–25, 2005.
- [39] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut Textures: Image and Video Synthesis Using Graph. *ACM Transactions on Graphics (Proc. SIGGRAPH2003)*, 22(3):277–286, 2003.
- [40] C.-H. Lee, D. Lee and M. Kim. Optimal Task Assignment in Linear Array Networks. *IEEE Trans. Computers*, 41(7):877–880, 1992.
- [41] Y. Li, J. Sun, and H.-Y. Shum. Video Object Cut and Paste. *ACM Transactions on Graphics (Proc. SIGGRAPH2005)*, 24(3):595–600, 2005.
- [42] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy Snapping. *ACM Transactions on Graphics (Proc. SIGGRAPH2004)*, 23(3):303–308, 2004.
- [43] K. Li, X. Wu, D. Z. Chen, and M. Sonka. Optimal Surface Segmentation in Volumetric Images—A Graph-Theoretic Approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(1):119–134, 2006.
- [44] H. Lombaert, Y. Sun, L. Grady, and C. Xu. A Multilevel Banded Graph Cuts Method for Fast Image Segmentation. In *ICCV2005*, I:259–265, 2005.
- [45] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.
- [46] C. Rother, V. Kolmogorov, and A. Blake. “Grab-Cut”: Interactive Foreground Extraction Using Iterated Graph Cuts. *ACM Transactions on Graphics (Proc. SIGGRAPH2004)*, 23(3):309–314, 2004.
- [47] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital Tapestry. In *CVPR 2005*, I:589–596, 2005.
- [48] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs. In *CVPR 2006*, I:993–1000, 2006.
- [49] S. Roy. Stereo without Epipolar Lines : A Maximum-flow Formulation. *International Journal of Computer Vision*, 34:147–162, 1999.
- [50] S. Roy and I. Cox. A Maximum-flow Formulation of the N-camera Stereo Correspondence Problem. In *ICCV'98*, 492–499, 1998.
- [51] S. Roy and V. Govindu. MRF Solutions for Probabilistic Optical Flow Formulations. In *ICPR2000*, Vol. 3, 7053–7059, 2000.
- [52] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields. In *ECCV2006*, II:16–29, 2006.
- [53] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-Based Reparameterization Framework for Analysis of Sum-Product and Related Algorithms. *IEEE Transactions on Information Theory* 49(5):1120–1146, 2003.
- [54] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen. Interactive Video Cutout. *ACM Transactions on Graphics (Proc. SIGGRAPH2005)*, 24(3):585–594, 2005.
- [55] J. Wills, S. Agarwal, and S. Belongie. What Went Where. In *CVPR2003*, I:37–44, 2003.
- [56] N. Xu, R. Bansal, and N. Ahuja. Object Segmentation Using Graph Cuts Based Active Contours. In *CVPR2003*, II:46–53, 2003.
- [57] <http://www.adastral.ucl.ac.uk/~vldkolm/software.html>
- [58] <http://maven.smith.edu/~nhowe/research/code/#fgseg>