

# Energy Minimization with Linear Smoothness Term: A Proof of Uniqueness of Solution for an Undirected Graph Formulation

Martin Trudeau<sup>1</sup> and Sébastien Roy<sup>2</sup>

<sup>1</sup> Natural Microsystems, 4200 De la Périère, St-Hubert, Québec, J3Y 9G3  
martin\_trudeau@nmss.com

<sup>2</sup> Université de Montréal, Département d'informatique et de recherche opérationnelle,  
C.P. 6128, Succ. Centre-Ville, Montréal, Québec, H3C 3J7  
roys@iro.umontreal.ca

**Abstract.** Recently many graph theoretic formulations were proposed for solving specific energy functions minimization problems. The specific case where a linear penalty term is used to enforce smoothness is of particular interest since it makes it possible to find the global solution using an efficient flow algorithm in a graph. However, a proof was needed to establish equivalence between a minimum s-t cut of a simple undirected graph and the original minimization problem. This paper presents such a proof for very general graphs, thereby making unnecessary the workarounds used until now, while providing some performance enhancement. Extending on previous one-dimensional proof, it is general enough to allow arbitrary graphs of any dimension as well as arbitrary neighborhood structure.

## 1 Introduction

One of the fundamental problem of computer vision is the *labeling* problem, where one must assign a label to each pixel in an image. This problem can take many forms, such as disparity estimation in stereo, gray level intensity in image restoration, or pixel displacement in optical flow estimation.

The contribution of this paper concerns the case where the problem is cast as a minimization of an energy function  $E$  over the space of all possible labelings of the image pixels. The function is designed so that its global minimum represent the best labeling for the particular problem to solve.

One kind of energy function is of particular interest.

$$E(f) = E_D + E_S$$

It separates the energy into two terms: a label term  $E_D$  and a smoothness term  $E_S$ . The label term expresses the cost of assigning a specific label to a particular

pixel. The smoothness term enforces smoothness between neighboring pixels over the image.

The key property of this formulation is that if the smoothness term is linear, then it is possible to find efficiently the global minimum of the function. This is accomplished by transforming the problem of minimizing into one of finding the minimum cost cut of a carefully designed flow graph.

The space of all possible cuts of the flow graph is clearly larger than the space of solutions of the original problem. Indeed, some cuts correspond to associating more than one label to a single pixel, which is not allowed. It was conjectured that these *illegal* cuts can never be minimum [1] so the space of all minimum cuts essentially corresponds to the space of optimal labelings in the original problem.

While a partial proof was provided [2], the full proof turned out to be elusive and some workarounds were proposed where the graph is modified to explicitly remove undesired solutions [1, 3–6]. These modifications result in a more complex graph, for instance with infinite edge capacities.

This paper provides a proof that no *illegal* cut can ever be minimum, for graph topologies general enough to include the very common 2D grid topology and much more, and some variation of the linear smoothness penalty term. Modifications of the original graph formulation is thus unnecessary.

The following sections summarize the transformation from a pixel labeling problem with a linear smoothness term into finding the minimum cut in a flow graph.

### 1.1 Pixel labeling as an energy minimization problem

Suppose you want to find a mapping  $f : \mathbf{x} \rightarrow \delta$  from the set  $\mathbf{x}$  of all pixels in an image to a label  $\delta \in \{1, \dots, n\}$  such that it minimizes the following energy function

$$E(f) = E_D + E_S \tag{1}$$

where  $E_D$  is an energy term containing the matching costs of assigning a label to a pixel, and where  $E_S$  is a smoothness term penalizing the change of labels between two neighboring pixels.

The matching cost term  $E_D$  has the form

$$E_D = \sum_{\mathbf{x}} g(x, f)$$

where  $g(x, f)$  is the cost of attributing the label  $f(x)$  to the pixel  $x$ . Typically, the labels represent disparities in a stereo correspondence problem but could represent another parameter that must be assigned to each pixels in an image.

The smoothness term  $E_S$  has the form

$$E_S = \sum_{x \in \mathbf{x}, y \in \mathcal{N}(x)} v(x, y, f)$$

where  $\mathcal{N}(x)$  is the set of neighbor pixels of  $x$ , and  $v(\cdot)$  is the penalty cost of assigning labels  $f(x)$  and  $f(y)$  to pixels  $x$  and  $y$ .

When  $v(x, y, f)$  is of the form

$$v(x, y, f) = k(x, y)|f(x) - f(y)| \quad (2)$$

then there is a polynomial time algorithm for finding the global solution of the minimization [1] through the reformulation of the problem as a minimum cost cut in a flow graph. For most other forms of  $v(\cdot)$ , such as the Potts model or any other *robust* penalty function, there is no polynomial algorithm and one must resort to approximations [2, 4].

For the linear penalty functions, the ordering of the labels is constrained to be one-dimensional, such as disparities in stereo or gray level intensity in image restoration. For two-dimensional labels, such as optical flow displacements, a linear penalty is not directly usable [7].

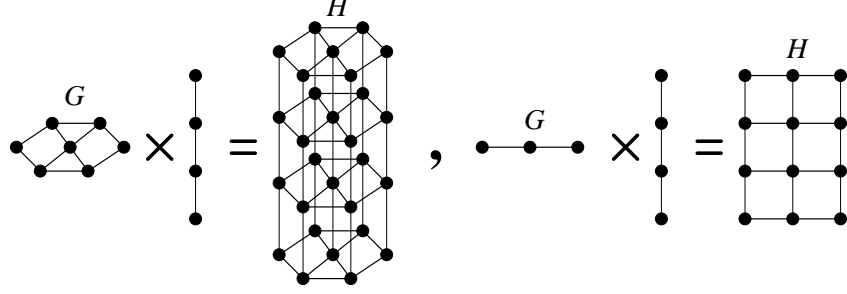
## 1.2 The flow graph formulation

From the energy minimization with linear penalty function (Eq. 2), we can construct a flow graph such that the minimum cut through the graph will represent the assignment of labels to the pixels minimizing the original energy function of Eq. 1 [references].

Let  $G = (V, E, k)$  be a finite capacitated undirected graph with vertex set  $V$ , edge set  $E$  and non-negative edge capacities  $k : E \rightarrow \mathbb{Z}^{\geq 0}$ ; we will denote an undirected edge between vertices  $x$  and  $y$  by  $\{x, y\}$ , so its capacity is  $k(\{x, y\})$ , or  $k(x, y)$  for short. In fact, it corresponds to the term  $k(x, y)$  of Eq. 2. The nodes of  $G$  represent the pixels to label (typically a 2D grid) and the edges represent the neighborhood structure (4-connected is typical). The topology of  $G$  can be arbitrary, thus allowing different dimensionality of pixels (3D voxels for example, or temporal sequence), and different neighborhood relationships.

Suppose  $H = (V_H, E_H, c)$  is a graph whose vertex set  $V_H$  is

$$\{s, t\} \cup V \times \{1, \dots, n\},$$



**Fig. 1.** Typical graph topology.  $G$  depicts pixels and their neighborhood structure.  $H$  is the complete graph after product with label nodes  $\{1, 2, 3, 4\}$ . The source  $s$  and sink  $t$  in  $H$  are not illustrated.

$n$  being an integer greater than 1 representing  $n - 1$  possible labels to assign, as depicted in Fig. 1, whose edge set  $E_H$  is

$$\begin{aligned}
& \{ \{s, (x, 1)\} \mid x \in V \} \\
& \cup \{ \{ (x, n), t \} \mid x \in V \} \\
& \cup \{ \{ (x, i), (y, i) \} \mid \{x, y\} \in E, 1 \leq i \leq n \} \\
& \cup \{ \{ (x, i), (x, i + 1) \} \mid x \in V, 1 \leq i < n \}
\end{aligned}$$

and with capacities such that

$$\begin{aligned}
c[s, (x, 1)] &= \infty, & x \in V \\
c[(x, n), t] &= \infty, & x \in V \\
c[(x, i), (y, i)] &= \lambda_i k(x, y), & \lambda_i > 0, \{x, y\} \in E, 1 \leq i \leq n \\
c[(x, i), (x, i + 1)] &= g(x, i), & g(x, i) > 0, x \in V, 1 \leq i < n
\end{aligned} \tag{3}$$

where  $g(x, i)$  is the cost of assigning label  $i$  to pixel  $x$ . The infinite capacities represent the source  $s$  and sink  $t$  connected respectively to the front  $(x, 1)$  and back  $(x, n)$  of the graph. We refer to edges of the form  $(x, i), (x, i + 1)$  as *label edges*, and edges of the form  $(x, i), (y, i)$  as *penalty edges*. Since  $H$  is an undirected graph, the *reverse* capacities  $c[(y, i), (x, i)]$  and  $c[(x, i + 1), (x, i)]$  are not explicitly defined.

The minimum s-t cut (minimum cost cut separating the source  $s$  and sink  $t$ ) of graph  $H$  yields the labeling of minimum cost according to Eq. 1.

### 1.3 Other graph capacities

In order to remove the possibility of having a minimum cut that would *fold* on itself, yielding multiple solutions for a pixel, several modifications of the original undirected graph were proposed.

In [6], the following change of the cost structure of Eq. 3 was proposed

$$\begin{aligned} c[(x, i), (y, i)] &= k(x, y, i), & \{x, y\} \in E, 1 \leq i \leq n \\ c[(x, i), (x, i + 1)] &= g(x, i), & g(x, i) > 0, x \in V, 1 \leq i < n \\ c[(x, i + 1), (x, i)] &= \infty, & x \in V, 1 \leq i < n \end{aligned} \quad (4)$$

making  $H$  a directed graph by making the *reverse* capacity of the label edges infinite. Since it is easy to show that a fold in a cut always include a *reverse* label edge with infinite capacity, it follows directly that no such cut can be minimum. Also, this change has the advantage of allowing more freedom in assigning capacities to the penalty term, since  $k(x, y, i)$  is more general than the original  $\lambda_i k(x, y)$  formulation. This potentially allows a wider range of convex functions to be used as a smoothness term.

Another formulation [4] suggests the following change from Eq. 3

$$\begin{aligned} c[(x, i), (y, i)] &= k(x, y), & \{x, y\} \in E, 1 \leq i \leq n \\ c[(x, i), (x, i + 1)] &= g(x, i) + K(x), & g(x, i) > 0, x \in V, 1 \leq i < n \end{aligned}$$

where  $K(x)$  is large enough. This undirected graph is in fact equivalent to the directed graph

$$\begin{aligned} c[(x, i), (y, i)] &= k(x, y), & \{x, y\} \in E, 1 \leq i \leq n \\ c[(x, i), (x, i + 1)] &= g(x, i), & g(x, i) > 0, x \in V, 1 \leq i < n \\ c[(x, i + 1), (x, i)] &= g(x, i) + 2K(x), & x \in V, 1 \leq i < n \end{aligned}$$

which corresponds to lowering the infinite capacity of Eq. 4 to the large finite value  $g(x, i) + 2K(x)$ .

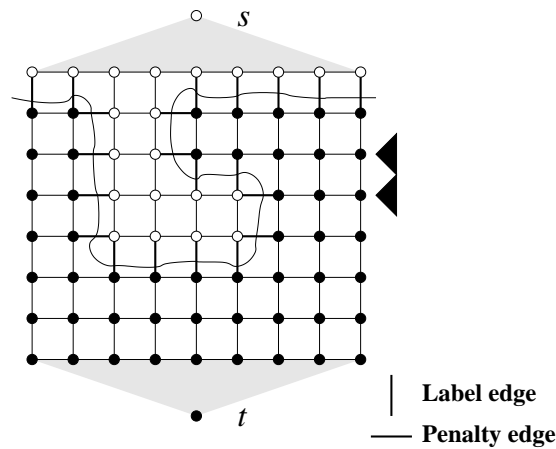
As a consequence of the following proof, these modifications are not needed anymore to guarantee a valid labeling.

In practice, a performance degradation of 10% was observed when the additional flow capacity introduced by these modifications of Eq. 3 is added to the graph [2]. Our own preliminary experiments did not produce any conclusive evidence of a significant change in performance. A degradation of performance would be consistent with recent results showing that worst case running time for flow algorithms such as preflow-push depends on maximum flow capacity as well as the number of vertices and edges [8, 9].

### 1.4 Brief description of the proof

The proof proceeds by reducing an arbitrary cut through a sequence of cuts, the last one of which features only single solutions and is guaranteed to cost less than all the preceding ones. This sequence is finite since it relies on a height function that takes positive integral values and decreases from one step to the next.

Consider the graph depicted in Figure 2. A cut with single solutions has only white nodes on top of black nodes. Each step of the proof consists of selecting two consecutive rows (indicated by two arrows) where the node order is reversed (black on top of white) somewhere inside them; one then flips black nodes to white or white to black to restore the order everywhere it is required in those two rows.



**Fig. 2.** Typical graph with a cut  $A$  ( $\circ \in A, \bullet \notin A$ ).

The following section contains the complete proof.

## 2 Proof

Let  $G = (V, E, k)$  and  $H = (V_H, E_H, c)$  be the graphs defined in Sec. 1.2.

We now consider subsets of  $V_H$  containing  $s$  but not  $t$ , i.e.  $s$ - $t$  cuts; we want to show that in a minimal  $s$ - $t$  cut  $A$  of  $H$ , i.e. one that minimizes the cost function

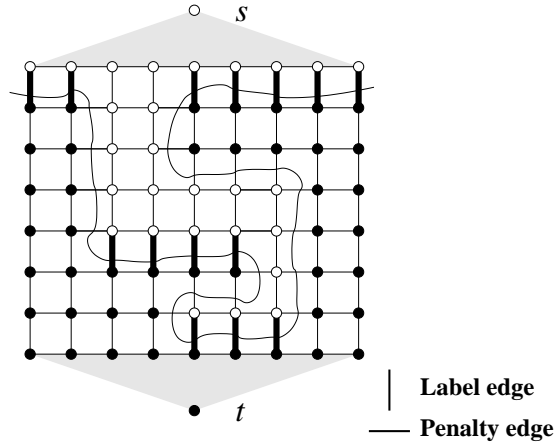
$$\sum_{a \in A, b \notin A, \{a, b\} \in E_H} c(a, b),$$

there are no two cut edges with non-zero capacities in the same *column*, that is, for all  $x \in V$  there is an  $1 \leq i \leq n - 1$  such that  $(x, 1), \dots, (x, i) \in A$  and  $(x, i + 1), \dots, (x, n) \notin A$  (from this point on, the term '*s-t* cut' will imply finite cost, which amounts to saying for a cut  $A$  that  $(x, 1) \in A$  and  $(x, n) \notin A$  for all  $x \in V$ ).

To do this, we first define the height  $h$  of an *s-t* cut  $A$ : let

$$\begin{aligned}
 P(A) &= \{ \{(x, i), (x, i + 1)\} \mid (x, i) \in A, (x, i + 1) \notin A \} \\
 &\quad \cup \{ \{(x, i), (x, i + 1)\} \mid (x, i) \notin A, (x, i + 1) \in A \} \\
 &= \text{the set of cut label edges} \\
 P_0(A) &= \{ \{(x, i), (x, i + 1)\} \in P(A) \text{ such that} \\
 &\quad | \{ \{(x, j), (x, j + 1)\} \in P(A) \mid j < i \} | \text{ is even} \} \\
 &= \text{the set of cut edges in an odd-numbered position in a given } \textit{column} \\
 Q(A) &= \{ \{(x, i), (y, i)\} \mid \{x, y\} \in E, (x, i) \in A, (y, i) \notin A \} \\
 &= \text{the set of cut penalty edges} \\
 \|Q(A)\| &= \sum_{\{(x, i), (y, i)\} \in Q(A), x, y \in V, 1 \leq i \leq n} k(x, y) \\
 h(A) &= |P(A)|M^2 + \|Q(A)\|M + |A|, \tag{5}
 \end{aligned}$$

where  $M$  is an integer strictly greater than  $|V_H|$  and the largest value of  $\|Q(A)\|$ . If  $A_1$  and  $A_2$  are *s-t* cuts such that  $|P(A_1)| > |P(A_2)|$ , then



**Fig. 3.** Illustration of  $P_0$ . The elements of  $P_0(A)$  are the thick label edges.

$$\begin{aligned}
& (|P(A_1)| - |P(A_2)|)M^2 + (\|Q(A_1)\| - \|Q(A_2)\|)M + |A_1| - |A_2| \\
& \geq M^2 + (\|Q(A_1)\| - \|Q(A_2)\|)M + |A_1| - |A_2| \geq M^2 - \|Q(A_2)\|M - |A_2| \\
& \geq M^2 - (M-1)M - (M-1) = 1
\end{aligned}$$

so  $h(A_1) > h(A_2)$ ; if instead  $|P(A_1)| = |P(A_2)|$  but  $\|Q(A_1)\| > \|Q(A_2)\|$  then

$$\begin{aligned}
& (|P(A_1)| - |P(A_2)|)M^2 + (\|Q(A_1)\| - \|Q(A_2)\|)M + |A_1| - |A_2| \\
& \geq M - |A_2| \geq M - (M-1) = 1
\end{aligned}$$

so again  $h(A_1) > h(A_2)$ . To sum up,

$$h(A_1) > h(A_2) \text{ iff } \begin{cases} |P(A_1)| > |P(A_2)| \\ \text{or} \\ |P(A_1)| = |P(A_2)| \text{ and } \|Q(A_1)\| > \|Q(A_2)\| \\ \text{or} \\ |P(A_1)| = |P(A_2)|, \|Q(A_1)\| = \|Q(A_2)\| \text{ and } |A_1| > |A_2|. \end{cases}$$

To obtain the result, we will show by induction that for all  $s$ - $t$  cut  $A$ , there is another (call it  $A_0$ ) such that  $|P(A_0)| = |V|$  ( $A_0$  has only one cut edge per column),  $P(A_0) \subseteq P_0(A)$  and  $\sum_{a \in Q(A_0)} c(a) \leq \sum_{a \in Q(A)} c(a)$ :

for  $A$  with  $h(A) < (|V| + 1)M^2$ , this is obviously true, as then  $|P(A)| = |V|$ ;

suppose now that this is true for all  $s$ - $t$  cuts with height  $< N$  and that  $h(A) = N$  ( $N$  some integer  $\geq (|V| + 1)M^2$ ); since  $|P(A)| > |V|$ , we can find an  $1 < i_0 < n$  such that there is an  $x \in V$  with  $(x, i_0) \notin A$  and  $(x, i_0 + 1) \in A$ ; let

$$W_1 = \{x \in V \mid (x, i_0) \notin A, (x, i_0 + 1) \in A\}$$

$$W_2 = \{x \in V \mid (x, i_0), (x, i_0 + 1) \in A\}$$

$$W_3 = \{x \in V \mid (x, i_0), (x, i_0 + 1) \notin A\}$$

$$W_4 = \{x \in V \mid (x, i_0) \in A, (x, i_0 + 1) \notin A\}$$

$$\alpha_1 = \sum_{\substack{x \in W_1, y \in W_2 \\ \{x, y\} \in E}} k(x, y), \quad \alpha_2 = \sum_{\substack{x \in W_1, y \in W_3 \\ \{x, y\} \in E}} k(x, y), \quad \alpha_3 = \sum_{\substack{x \in W_1, y \in W_4 \\ \{x, y\} \in E}} k(x, y);$$

(i) if  $\alpha_1 \leq \alpha_2 + \alpha_3$ : the cut  $A_1 = A \setminus \{(x, i_0 + 1) \mid x \in W_1\}$  satisfies  $|P(A_1)| \leq |P(A)|$ , as for  $x \in W_1$  the cut at  $\{(x, i_0), (x, i_0 + 1)\}$  moves to  $\{(x, i_0 + 1), (x, i_0 + 2)\}$  if  $(x, i_0 + 2) \in A$ , or disappears along with the one at  $\{(x, i_0 + 1), (x, i_0 + 2)\}$



if  $(x, i_0 + 2) \notin A$ ; it also satisfies  $\|Q(A_1)\| \leq \|Q(A)\|$ , as

$$\begin{aligned} & \sum_{\substack{\{(x,i),(y,i)\} \in Q(A) \\ x,y \in V, 1 \leq i \leq n}} k(x,y) - \sum_{\substack{\{(x,i),(y,i)\} \in Q(A_1) \\ x,y \in V, 1 \leq i \leq n}} k(x,y) \\ = & \sum_{\substack{\{(x,i_0+1),(y,i_0+1)\} \in Q(A) \\ x,y \in V}} k(x,y) - \sum_{\substack{\{(x,i_0+1),(y,i_0+1)\} \in Q(A_1) \\ x,y \in V}} k(x,y) = \alpha_2 + \alpha_3 - \alpha_1; \end{aligned}$$

clearly  $|A_1| < |A|$ , therefore  $h(A_1) < h(A)$ , so there is an  $A_0$  with  $|P(A_0)| = |V|$ ,  $P(A_0) \subseteq P_0(A_1)$  and  $\sum_{a \in Q(A_0)} c(a) \leq \sum_{a \in Q(A_1)} c(a)$ ; in a *column*, the cuts that move always are in an even-numbered position and the removed cuts come in pairs, leaving the parity of the other cuts unchanged, therefore  $P_0(A_1) \subseteq P_0(A)$ ;  $\sum_{a \in Q(A)} c(a) - \sum_{a \in Q(A_1)} c(a) = k_{i_0+1}(\alpha_2 + \alpha_3 - \alpha_1)$ , hence  $A_0$  does the trick for  $A$ , which thus satisfies the induction hypothesis.

(ii) if  $\alpha_1 > \alpha_2 + \alpha_3$ : in that case,  $\alpha_2 < \alpha_1 - \alpha_3 \leq \alpha_1 + \alpha_3$ ; the cut  $A_1 = A \cup \{(x, i_0) \mid x \in W_1\}$  satisfies  $|P(A_1)| \leq |P(A)|$ , as for  $x \in W_1$  the cut at  $\{(x, i_0), (x, i_0 + 1)\}$  moves to  $\{(x, i_0 - 1), (x, i_0)\}$  if  $(x, i_0 - 1) \notin A$ , or disappears along with the one at  $\{(x, i_0 - 1), (x, i_0)\}$  if  $(x, i_0 - 1) \in A$ ; it satisfies  $\|Q(A_1)\| < \|Q(A)\|$ , as

$$\begin{aligned} & \sum_{\substack{\{(x,i),(y,i)\} \in Q(A) \\ x,y \in V, 1 \leq i \leq n}} k(x,y) - \sum_{\substack{\{(x,i),(y,i)\} \in Q(A_1) \\ x,y \in V, 1 \leq i \leq n}} k(x,y) \\ = & \sum_{\substack{\{(x,i_0),(y,i_0)\} \in Q(A) \\ x,y \in V}} k(x,y) - \sum_{\substack{\{(x,i_0),(y,i_0)\} \in Q(A_1) \\ x,y \in V}} k(x,y) = \alpha_1 + \alpha_3 - \alpha_2, \end{aligned}$$

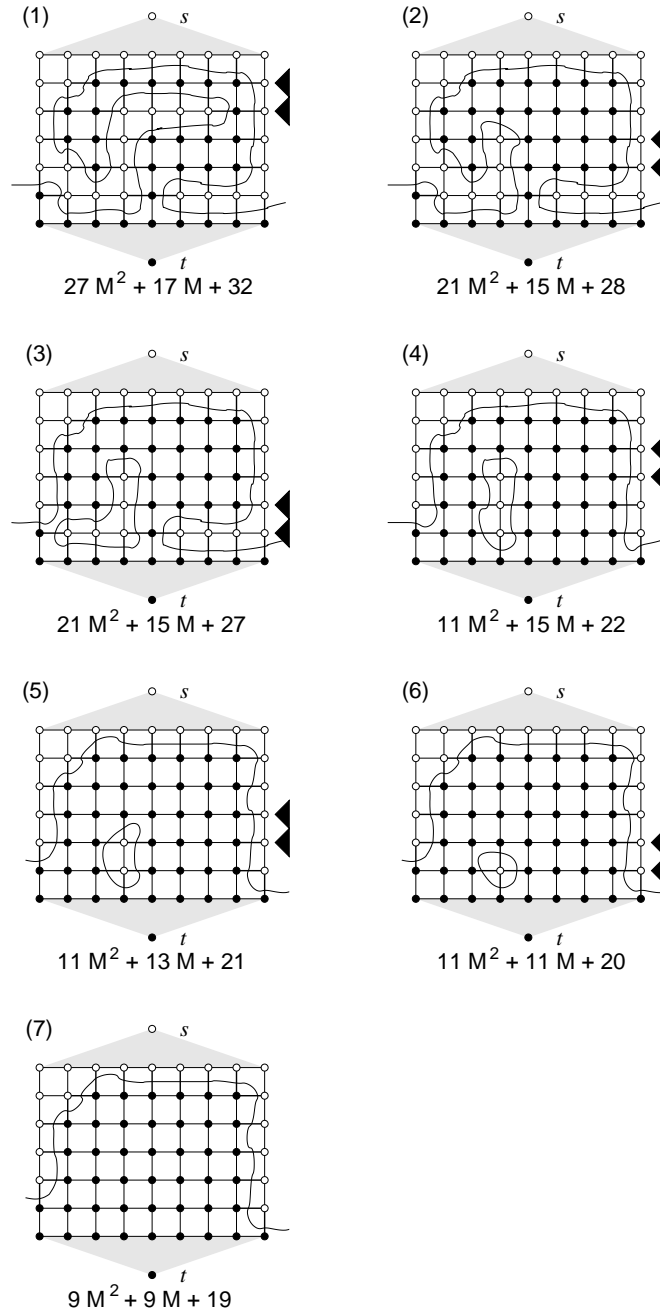
therefore  $h(A_1) < h(A)$ , so there is an  $A_0$  with  $|P(A_0)| = |V|$ ,  $P(A_0) \subseteq P_0(A_1)$  and  $\sum_{a \in Q(A_0)} c(a) \leq \sum_{a \in Q(A_1)} c(a)$ ; as in (i),  $A_0$  does the trick for  $A$ , which again satisfies the induction hypothesis.

We conclude that the hypothesis is true for all  $s$ - $t$  cuts, thus completing the proof.

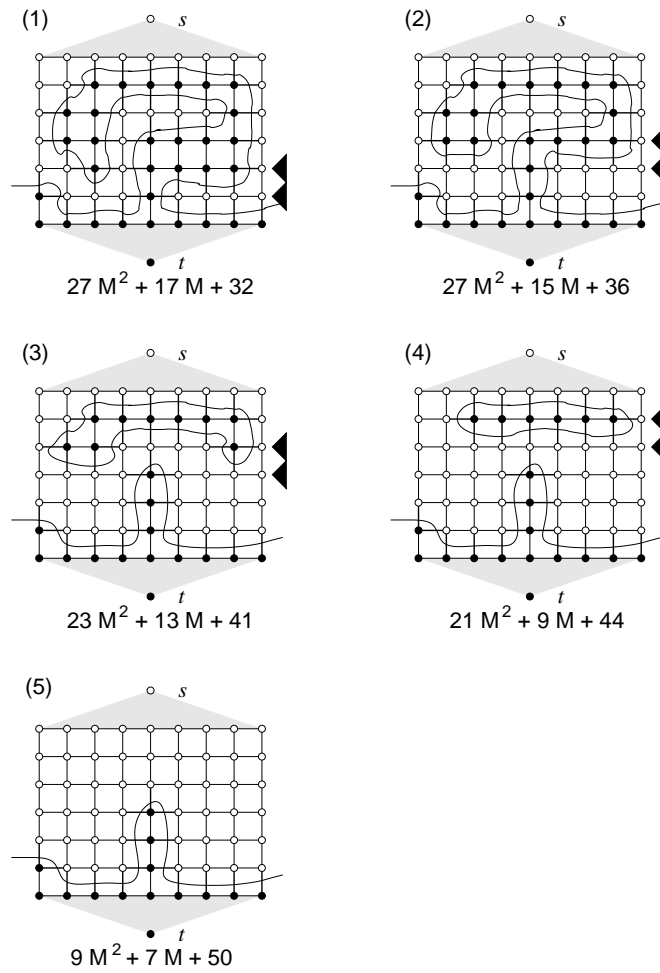
*Note:* We used  $\sum_{\{(x,i),(y,i)\} \in Q(A), x,y \in V, 1 \leq i \leq n} k(x,y)$  instead of the more natural  $\sum_{v \in Q(A)} c(v)$  in the definition of  $\|Q(A)\|$  simply because of the possibility that some of the  $\lambda_i$ 's might be zero.

### 3 Example

Two sample reduction sequences are illustrated in Fig. 4 and 5. Notice that the height function  $h(A)$  always decreases from one step to the next, given the choice



**Fig. 4.** Sample reduction sequence. Penalty edges have capacity  $k(x, y) = 1$ ; label edges have arbitrary capacity. The height function  $h(A)$  from Eq. 5 is provided for each step. The two rows selected for the next reduction are also marked.



**Fig. 5.** Sample reduction sequence with alternate row selection. Penalty edges have capacity  $k(x, y) = 1$ ; label edges have arbitrary capacity. The height function  $h(A)$  from Eq. 5 is provided for each step. The two rows selected for the next reduction are also marked.

of  $M$ . However, the cost of successive cuts does not necessarily decrease but still the final cut has a lower cost than all previous ones. Indeed, all the cut label edges in the final cut were already cut throughout the reduction sequence; moreover, the cost of cut penalty edges is guaranteed never to increase throughout the reduction sequence.

## 4 Conclusion

This paper presented a proof that the minimum cut of an undirected graph formulation of an energy minimization with linear smoothness term always yields a single solution for each pixel to label. This result extends a previous one-dimensional proof [2] to images of arbitrary dimension and neighborhood structure and thus makes proposed workarounds [1, 3–6] unnecessary.

## References

- [1] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proc. Int. Conference on Computer Vision*, pages 492–499, Bombay, India, 1998.
- [2] S. T. Birchfield. *Depth and Motion Discontinuities*. PhD thesis, Stanford, 1999.
- [3] S. Roy. Stereo without epipolar lines : A maximum-flow formulation. *Int. J. Computer Vision*, 34(2/3):147–162, 1999.
- [4] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, june 1998.
- [5] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *Proc. European Conference on Computer Vision*, Freiburg, Germany, juin 1998.
- [6] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, California, june 1998.
- [7] S. Roy and V. Govindu. Mrf solutions for probabilistic optical flow formulations. In *Proc. of Int. Conf. on Pattern Recognition*, Barcelona, Spain, September 2000.
- [8] A. V. Goldberg and S. B. Rao. Length functions for flow computations. Technical Report 97-055, NEC Research Institute, Princeton NJ, 1997.
- [9] A. V. Goldberg. Recent developments in maximum flow algorithms. In *Algorithm Theory - SWAT 98 - Lecture Notes in Computer Science 1432, Proceedings of the 6th Skandinavian Workshop on Algorithm Theory*, pages 1–10. Springer-Verlag, 1998.