

平滑化処理の繰り返しによる 画像セグメンテーションのためのグラフカット

永橋 知行[†] 藤吉 弘亘[†] 金出 武雄^{††}

[†] 中部大学大学院工学研究科 〒 487-8501 愛知県春日井市松本町 1200
^{††} カーネギーメロン大学 〒 15213 米国ペンシルバニア州ピッツバーグ市
 E-mail: [†]{kida,hf}@vision.cs.chubu.ac.jp, ^{††}tk@cs.cmu.edu

あらまし 本稿では、平滑化度合いを変化させ、繰り返し処理による Graph Cuts を用いた高精度な画像セグメンテーション法を提案する。ガウシアンフィルタの平滑化度合いを変化させた画像に対し、平滑化度合いが大きなものから Graph Cuts を行い、そのセグメンテーション結果からグラフの t-link に反映させ、次の平滑化度合いのセグメンテーションに利用する。これらを繰り返し処理することにより、大域的なセグメンテーションから段階的に局所的なセグメンテーションを行うことができるため、画像に複雑なエッジが存在する場合でも頑健なセグメンテーションが可能となる。評価実験より、提案手法は従来の Interactive Graph Cuts と比較し、4.7% セグメンテーション精度を向上させることができた。

キーワード 物体抽出, 画像セグメンテーション, グラフカット

Iterated Graph Cuts by Multi-level Smoothing for Image Segmentation

Tomoyuki NAGAHASHI[†], Hironobu FUJIYOSHI[†], and Takeo KANADE^{††}

[†] Dept. of Computer Science, Chubu Univ. Matsumoto 1200, Kasugai, Aichi, 487-8501 Japan
^{††} The Robotics Institute, Carnegie Mellon Univ. Pittsburgh Pennsylvania, 15213 USA
 E-mail: [†]{kida,hf}@vision.cs.chubu.ac.jp, ^{††}tk@cs.cmu.edu

Abstract We present a novel approach to image segmentation using iterated Graph Cuts based on multi-level smoothing. We compute the prior probability obtained by the likelihood from a color histogram and a distance transform using the segmentation results from graph cuts in the previous process, and set the probability as the t-link of the graph for the next process. The proposed method can segment the regions of an object with a stepwise process from global to local segmentation by iterating the graph-cuts process with Gaussian smoothing using different values for the standard deviation. We demonstrate that we can obtain 4.7% better segmentation than that with the conventional approach.

Key words Foreground Extraction, Image Segmentation, Graph Cuts

1. はじめに

画像処理における重要な問題の一つに一枚の画像から対象となる領域を抽出するセグメンテーションの問題がある。セグメンテーションは、一般物体認識、合成画像の生成などの前処理として利用されるため重要な問題といえる。

近年、セグメンテーション問題をエネルギー最小化問題として解く手法が提案されている。そのような手法として、Snake などの動的輪郭モデル [1] や Level Sets [2], Graph Cuts [3]-[8] などが挙げられる。Snake や Level Sets は境界線に対してのエネルギー関数を作成し、エネルギー関数が小さくなるように境

界線を変化させる手法である。そのため、これらの手法では局所解しか求めることができないというデメリットがある。それに対し、Graph Cuts では各領域からエネルギー関数を定義し、それらの大域解を求めることが可能である。

Graph Cuts を用いたセグメンテーション手法として、Boykov らにより Ineractive Graph Cuts [4] [5] が提案されている。Interactive Graph Cuts では、ユーザが与えた正解ラベルと画像からグラフを作成し、minmum cut/maximum flow algorithm を用いることで、エネルギー関数の最小化を行う。この Interactive Graph Cuts を拡張した手法として、高速化のためにスーパーピクセルをノードとしてグラフへ拡張した Lazy

Snapping [7] やセグメンテーション精度を向上させるために繰り返し処理により前景と背景の色分布をセグメンテーション結果から再学習し、繰り返しセグメンテーションをする Grab Cut [8] などが提案されている。これらの Graph Cuts による画像セグメンテーションでは、複雑なエッジ情報を含んだ画像に対しては正確な物体領域を抽出することが困難という問題点がある。これは、ピクセル間の輝度値から計算される n-link の影響により、局所的なエッジを乗り越えることができないためである。

そこで、本研究では複数の平滑化画像を用いて繰り返し処理によるグラフカットセグメンテーション法を提案する。本手法では、ガウシアンフィルタの平滑化度合いを変化させた画像に対し、平滑化度合いが大きなものから Graph Cuts を行い、そのセグメンテーション結果からグラフの t-link に反映させ、次の平滑化度合いのセグメンテーションに利用する。これらを繰り返し処理することにより、大域的なセグメンテーションから段階的に局所的なセグメンテーションを行う。

2. Graph Cuts

本章では、従来法である Interactive Graph Cuts [4] [5] について説明する。

2.1 Graph Cuts による画像セグメンテーション

画像 P に対する各ピクセルを $p \in P$ としたとき、ラベルを $L = \{L_1, L_2, \dots, L_p, \dots, L_{|P|}\}$ とし、各 L_p には物体 (“obj”) が背景 (“bkg”) のラベルが与えられる。また、 p の近傍ピクセルを $q \in N$ とする。Graph Cuts では、エネルギー関数を式 (1) のように定義する。

$$E(L) = \lambda \cdot R(L) + B(L) \quad (1)$$

λ は、 $R(L)$ と $B(L)$ の比率のパラメータである。 $R(L)$ は領域に対するペナルティ関数、 $B(L)$ は物体と背景の境界に対するペナルティ関数であり以下に示すように定義する。

$$R(L) = \sum_{p \in P} R_p(L_p) \quad (2)$$

$$B(L) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(L_p, L_q) \quad (3)$$

$$\delta(L_p, L_q) = \begin{cases} 1 & \text{if } L_p \neq L_q \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$R_p(L_p)$ は、ピクセル p がラベル L_p である確率が高ければ値が小さくなるような関数として定義する。 $B_{\{p,q\}}$ は、 p と q の輝度値が似ていれば大きな値を出力する関数として定義する。 $R(L)$ と $B(L)$ により定義したエネルギー関数 $E(L)$ を最小とするようなラベル L を Graph Cuts Algorithm [3] を用いて計算することでセグメンテーションを行う。

Graph Cuts Algorithm では、画像から図 1 のようにグラフを作成し min-cut/max-flow algorithm を用いてグラフの分割を実現している。グラフ G は、画像の各ピクセルに対応したノードと source と sink と呼ばれるターミナルからなる。グラフの作成に各ピクセルをノードとしてグラフを構成する。

各ノード間を接続するエッジを n-link と呼び、各ノードから source(S) と sink(T) のターミナルを接続するエッジを t-link と呼ぶ。n-link と t-link のエッジコストは、表 1 に従い設定する。

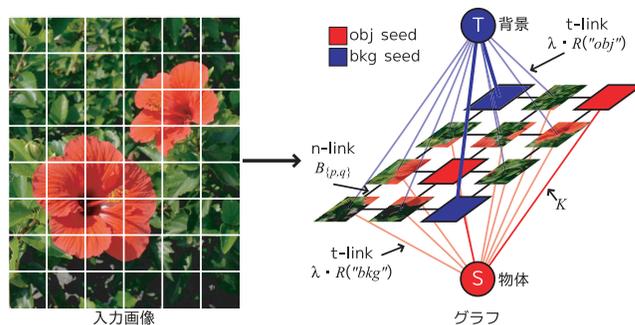


図 1 グラフの作成

表 1 エッジに与える重み

edge		cost	for
n-link	$\{p, q\}$	$B_{\{p, q\}}$	$\{p, q\} \in N$
t-link	$\{p, S\}$	$\lambda \cdot R_p(\text{"bkg"})$	$p \in P, p \notin \mathcal{O} \cup \mathcal{B}$
		K	$p \in \mathcal{O}$
	$\{p, T\}$	$\lambda \cdot R_p(\text{"obj"})$	$p \in P, p \notin \mathcal{O} \cup \mathcal{B}$
		K	$p \in \mathcal{B}$

このとき、

$$R_p(\text{"obj"}) = -\ln \Pr(I_p | \mathcal{O}) \quad (5)$$

$$R_p(\text{"bkg"}) = -\ln \Pr(I_p | \mathcal{B}) \quad (6)$$

$$B_{\{p,q\}} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{\text{dist}(p, q)} \quad (7)$$

$$K = 1 + \max_{p \in P} \sum_{q: \{p,q\} \in N} B_{\{p,q\}} \quad (8)$$

となる。 \mathcal{O} は物体、 \mathcal{B} は背景を意味し、また、 I_p はピクセル p の輝度値である。ユーザは一部のピクセルに対して \mathcal{O}, \mathcal{B} を入力する。このとき入力された \mathcal{O}, \mathcal{B} を seed と呼ぶ。 $\Pr(I_p | \mathcal{O}), \Pr(I_p | \mathcal{B})$ は seed 以外のピクセルの t-link に設定する物体と背景の尤度である。 $\text{dist}(p, q)$ はピクセル p, q のユークリッド距離を用いる。作成したグラフに対して、min-cut/max-flow algorithm を用いることで、物体と背景にグラフを分割する。このような処理により、インタラクティブな画像セグメンテーションを実現する。

2.2 従来法の問題点

従来法である Interactive Graph Cuts [4], [5] では、seed が与えられていない t-link のエッジコストには、seed の色分布から得られる尤度を利用して計算するか 0 とする。尤度の計算を行う場合、n-link に比べ t-link が大きくなると、色分布による影響が強くなり、突発的な誤検出が多くなる場合がある。そのため、このような誤検出を抑制するためには、 λ によって n-link の影響を強くする必要があり、n-link の影響が強いと、

画像中のエッジ情報に対しての依存性が強くなる．そのため，図 2 に示すように Interactive Graph Cuts では画像に複雑なエッジが存在する場合，局所的なエッジを乗り越えてセグメンテーションすることが困難となる．

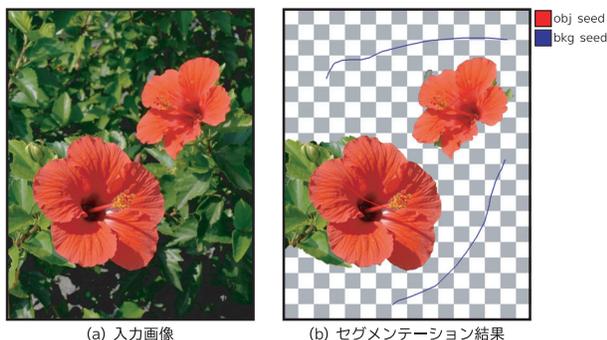


図 2 複雑なエッジを含む画像での Graph Cuts

3. 平滑化処理の繰り返しによる Graph Cuts

本研究では平滑化度合いの異なる複数の画像を使用し，平滑化度合いの大きなものから Graph Cuts を繰り返し行うことにより，局所的なエッジ情報に頑健なセグメンテーション手法を提案する．

3.1 提案手法の流れ

図 3 に，提案手法の流れを示す．はじめに，入力画像に対し

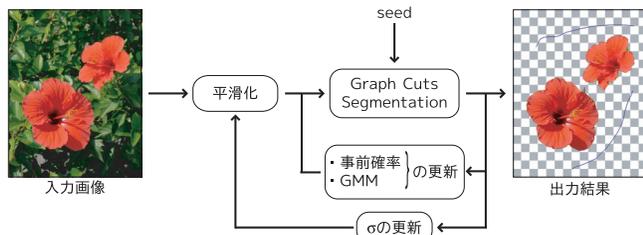


図 3 手法の流れ

てユーザが物体と背景の seed の入力を行う．次に，平滑化度合いを決定する σ の初期値を決定する．これは画像を 1/4 にダウンサンプリングした際の画像の長辺が一定値 1 以下になる画像を平滑化を開始する画像とし，そこから σ の初期値を決定する．決定した σ を用いてガウシアンフィルタにより平滑化画像を作成し，それを入力画像として Graph Cuts を行う．Graph Cuts により，物体領域と背景領域の色分布に対して GMM(Gaussian Mixture Model) の当てはめを行う．また，物体領域と背景領域に対して距離変換を行い各ピクセルごとに物体と背景の事前確率を更新する．GMM の尤度と距離変換による事前確率から各ピクセルごとに物体と背景の事後確率を計算したものをグラフの t-link に設定し，次の Graph Cuts 処理に利用する．この処理を $\sigma < 1$ でセグメンテーション結果に変化がないか， $\sigma = 0$ となるまで繰り返す．この条件を満たさない場合は， $\sigma = \alpha \cdot \sigma$ として更新を行い再度 Graph Cut を行う．このときの $0 < \alpha < 1$ である．

これらの手順を以下に示す．

- Step1. seed の入力
- Step2. σ の初期値計算
- Step3. 画像の平滑化
- Step4. Graph Cuts
- Step5. セグメンテーション結果から事後確率の計算
- Step6. $\sigma < 1$ でセグメンテーション結果が変化しなくなるか， $\sigma = 0$ となったら終了，それ以外ならば $\sigma = \alpha \cdot \sigma (0 < \alpha < 1)$ と更新し Step 3 へ

このような繰り返し処理により，大域的なセグメンテーションから段階的に局所的なセグメンテーションを実現している様子を図 4 に示す．以下に平滑化画像の作成方法と繰り返し処理の詳細について述べる．

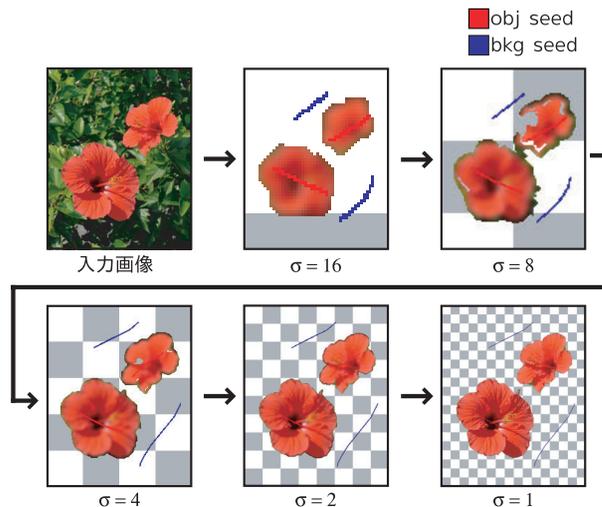


図 4 繰り返し処理によるセグメンテーション例

3.2 平滑化画像の作成

ガウシアンフィルタを用いて，平滑化画像を作成する．画像を I ，ガウス分布を $G(\sigma)$ としたとき，平滑化画像 $L(\sigma)$ は以下の式で計算される．

$$L(\sigma) = G(\sigma) * I \quad (9)$$

大きい σ の場合にはフィルタのウィンドウサイズを大きくする必要があり．ウィンドウサイズが大きくなると，画像の端といった信頼性が低い部分が多くなるという問題点が挙げられる．そこで本手法では，画像のダウンサンプリングにより σ の変化の連続性を保持した平滑化画像を作成する．

はじめに，入力画像 I_1 に対して σ を 1 から 2 へ増加させて平滑化画像 $L_1(\sigma)$ を作成する．次に，画像 I のサイズを 1/4 へダウンサンプリングしたものを I_2 とする．この画像 I_2 に対して， $\sigma = 1$ として平滑化画像 $L_2(\sigma)$ を作成する． $L_1(\sigma)$ と $L_2(\sigma)$ には以下の式が成り立つ．

$$L_1(2\sigma) \doteq L_2(\sigma) \quad (10)$$

この関係を利用することにより，ダウンサンプリングを行ったものに対して， σ を 1 から 2 の間で変化させることで，ウィンドウサイズを変化させることなく元の画像サイズに対して σ を連続的に大きくすることができる．これをダウンサンプリング

後の画像サイズがある程度小さくなるまで繰り返すことにより、通常のガウシアンフィルタでは不可能なサイズの σ に対しても連続的に平滑化画像を作成することが可能となる。図 5 にその流れを示す。

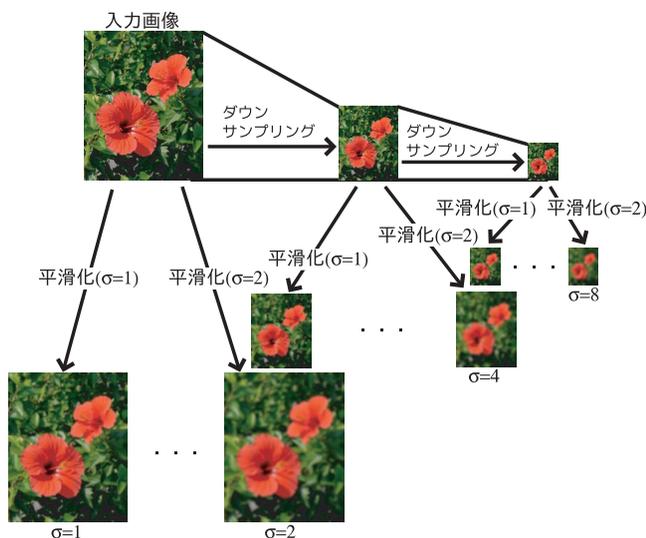


図 5 ダウンサンプリングを用いた平滑化

3.3 繰り返し処理による Graph Cuts

σ が大きな平滑化画像から Graph Cuts を繰り返し行う。従来の Graph Cuts では、t-link は式 (5), (6) の尤度を用いて計算する。これを本手法では 1 回前の Graph Cuts の結果から事後確率を以下の式により計算し、t-link に用いる。

$$R'_p(\text{"obj"}) = -\ln \Pr(\mathcal{O}|I_p) \quad (11)$$

$$R'_p(\text{"bkg"}) = -\ln \Pr(\mathcal{B}|I_p) \quad (12)$$

このとき、 $\Pr(\mathcal{O}|I_p)$ と $\Pr(\mathcal{B}|I_p)$ はベイズの定理から式 (13), (14) により計算する。

$$\Pr(\mathcal{O}|I_p) = \frac{\Pr(\mathcal{O})\Pr(I_p|\mathcal{O})}{\Pr(I_p)} \quad (13)$$

$$\Pr(\mathcal{B}|I_p) = \frac{\Pr(\mathcal{B})\Pr(I_p|\mathcal{B})}{\Pr(I_p)} \quad (14)$$

分母の $\Pr(I_p)$ は $\Pr(\mathcal{O})$ と $\Pr(\mathcal{B})$ の両方にあるため、これらの大小関係には影響がないので無視することができる。事後確率を計算するときの尤度 $\Pr(I_p|\mathcal{O}), \Pr(I_p|\mathcal{B})$ と、事前確率 $\Pr(\mathcal{O}), \Pr(\mathcal{B})$ は前回のセグメンテーション結果から計算する。図 6 に、尤度と事前確率の更新の流れを示す。

3.3.1 尤度の更新

尤度 $\Pr(I_p|\mathcal{O}), \Pr(I_p|\mathcal{B})$ の計算には、GMM(Gaussian Mixture Model) [9] を用いて色分布を表現する。当てはめる色分布は RGB の 3 次元となるため、以下の式を用いる。

$$\Pr(I_p|\cdot) = \sum_{i=1}^K \alpha_i p_i(I_p|\mu_i, \Sigma_i) \quad (15)$$

$$p(I_p|\mu, \Sigma) = \frac{1}{(2\pi)^{3/2} |\Sigma|^{1/2}} \cdot \exp\left(-\frac{1}{2}(I_p - \mu)^T \Sigma^{-1}(I_p - \mu)\right) \quad (16)$$

GMM を物体と背景の各領域の色分布に対して EM アルゴリズム [10] を用いて当てはめる。これにより、ヒストグラムでは離散的であった値に対して、GMM を用いることにより連続的に取り扱うことができる。当てはめた GMM のパラメータから式 (15) を用いることにより、 I_p の尤度 $\Pr(I_p|\mathcal{O})$ と $\Pr(I_p|\mathcal{B})$ を求める。

3.3.2 事前確率の更新

1 つ前の Graph Cuts のセグメンテーション結果の空間情報を用いて、事前確率 $\Pr(\mathcal{O}), \Pr(\mathcal{B})$ の更新を行う。物体と背景の境界付近では、次のセグメンテーション時にはどちらになるか不明なため、同程度に物体と背景が生起すると考えられる。一方、境界から離れている位置では、次のセグメンテーション時に物体と背景が入れ変わる可能性は低い。このような確率分布を作成するために、物体領域と背景領域に対して距離変換を行う。このとき、境界からの距離 d を 0.5 から 1 に正規化を行う。正規化をした物体の距離 d_{obj} と背景の距離 d_{bkg} を用いて事前確率を以下の式で更新する。

$$\Pr(\mathcal{O}) = \begin{cases} d_{obj} & \text{if } d_{obj} \geq d_{bkg} \\ 1 - d_{bkg} & \text{if } d_{obj} < d_{bkg} \end{cases} \quad (17)$$

$$\Pr(\mathcal{B}) = 1 - \Pr(\mathcal{O}) \quad (18)$$

GMM により得られた $\Pr(I_p|\mathcal{O}), \Pr(I_p|\mathcal{B})$ と距離変換により得られた事前確率 $\Pr(\mathcal{O}), \Pr(\mathcal{B})$ を用いて、式 (11) により物体の事後確率 $\Pr(\mathcal{O}|I_p)$ 、式 (12) により背景の事後確率 $\Pr(\mathcal{B}|I_p)$ を計算する。

物体の事後確率 $\Pr(\mathcal{O}|I_p)$ から式 (11) により $\{p, T\}$ 間の t-link のコストを、背景の事後確率 $\Pr(\mathcal{B}|I_p)$ から式 (12) により $\{p, S\}$ 間の t-link のコストをそれぞれ計算し、これを 3.2 に示したように次の平滑化画像を用いた Graph Cuts 処理に利用する。平滑化度合いを大きな値から小さくしながら以上の処理を繰り返すことにより、大域的なセグメンテーションから段階的に局所的なセグメンテーションを実現することが可能となる。

4. 評価実験

4.1 実験概要

本手法の有効性を示すために評価実験を行う。実験に使用する評価用画像データベースには [11] にて提供されている 50 枚の人・風景・動物等の画像を用いる。これらの画像に対し、同じ seed を与えて提案手法と従来法である Interactive Graph Cuts [4] と GrabCut [8] との比較を行う。評価方法として、物体領域のマスクデータを正解データとし、それに対しての誤検出率により評価する。物体領域を背景と誤検出したものを over segmentation、背景領域を物体と誤検出したものを under segmentation とし、各誤検出率を以下の式で求める。

$$\text{over segmentation} = \frac{\text{物体領域の誤検出ピクセル数}}{\text{全ピクセル数}} \quad (19)$$

$$\text{under segmentation} = \frac{\text{背景領域の誤検出ピクセル数}}{\text{全ピクセル数}} \quad (20)$$

Graph Cuts のパラメータ λ は、Interactive Graph Cuts では

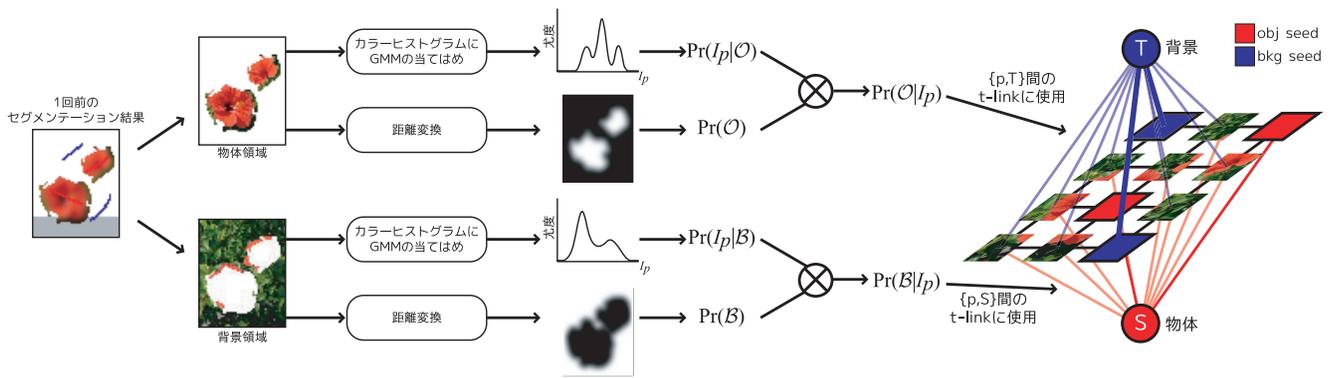


図 6 尤度と事前確率の更新の流れ

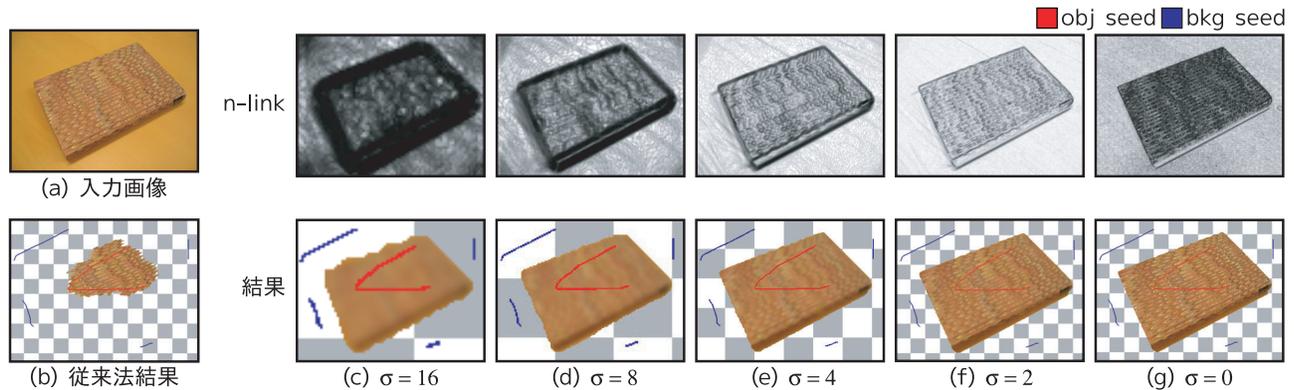


図 7 平滑化画像に対する n-link の変化とセグメンテーション結果

$\lambda = 0.005$ として seed が与えられた位置のみ t-link を計算する。Grabcut と提案手法では、 $\lambda = 0.02$ として実験を行う。

4.2 実験結果

表 2 に、評価用データベース 50 枚の画像に対するセグメンテーション結果を示す。表 2 より、提案手法は従来法に比べ

表 2 誤検出率 [%]

	従来法		提案手法
	Interactive Graph Cuts [4]	GrabCut [8]	
over seg.	1.86	3.33	1.12
under seg.	1.89	1.59	0.49
total	3.75	4.93	1.61

2.14% セグメンテーションの精度を向上させることができた。また実験に用いた画像の中で、従来法で誤検出率が 2% 以下のものを成功画像、2% 以上を失敗画像とした際の実験結果を表 3 に示す。表 3 より、提案手法は従来法で成功する画像に対しては同程度のセグメンテーション精度である。一方、失敗画像に対しては 4.79% セグメンテーション精度を向上させることができています。この理由として、従来の Graph Cuts では n-link の影響が大きいため、図 7(b) のように細かく強いエッジが存在する画像に対して、正確なセグメンテーションを行うことは困難である。それに対し提案手法では、平滑化を行うことにより周波数成分が低い大域的なセグメンテーションから始めることにより大まかなセグメンテーションを行う。そして、徐々に平

表 3 誤検出率 [%]

	従来法			提案手法
	Interactive Graph Cuts [4]	GrabCut [8]		
成功画像 (26 枚)	over seg.	0.29	3.54	0.81
	under seg.	0.43	1.03	0.22
	total	0.72	4.58	1.03
失敗画像 (24 枚)	over seg.	3.56	3.10	1.45
	under seg.	3.47	2.21	0.79
	total	7.04	5.31	2.25

滑化度合いを低くしていくことで、大域的なセグメンテーション情報を保持しながら、より細かいセグメンテーションを行うことが可能となる (図 7)。そのため、提案手法は従来法で失敗した画像に対しても高精度にセグメンテーションを行うことができたと考えられる。図 8 に各手法の画像セグメンテーション例とエラー率 (err) を示す。このエラー率は式 (19) の over segmentation と式 (20) の under segmentation を足した値である。

4.3 考察

4.3.1 λ の変化による影響

式 (1) の t-link と n-link の強さを制御するパラメータ λ は、Interactive Graph Cuts では経験的に決定している。Interactive Graph Cuts [4] と提案手法にて λ を 100 から 0 まで変化させた結果を図 9 に示す。図 9 の $\lambda = 0.005$ のように n-link が強い場合、Interactive Graph Cuts と提案手法ともに大まかな

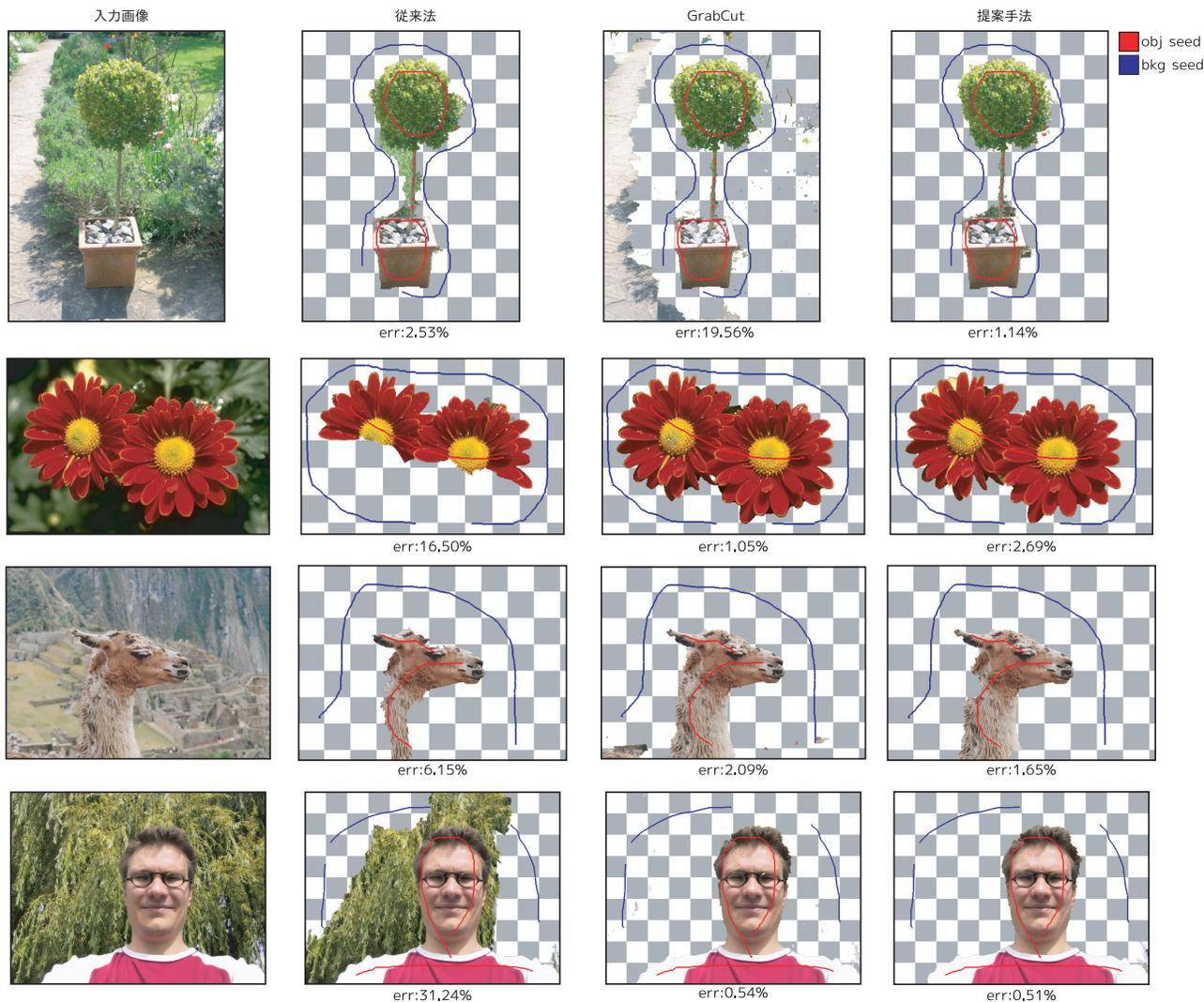


図 8 セグメンテーション例と誤検出率

物体領域の抽出はできているが、細部に誤検出がある。一方、 λ の値を大きくして t-link の影響を強くした場合、Interactive Graph Cuts では誤検出領域が多いが、提案手法では安定して物体領域を抽出できていることがわかる。これは、従来法では t-link に色情報のみが使われているため、物体領域の色に似ている色の背景領域が誤検出される。提案手法では繰り返し処理における 1 つ前の Graph Cuts の結果から、大まかな物体領域と背景領域の形を捉えた t-link が得られるため、 λ を大きくした場合でも突発的な誤検出を抑制することができる。このことから、提案手法は λ を変化させても、安定したセグメンテーション結果を得ることができる。

4.3.2 処理時間について

従来法と提案手法の処理時間の計測を行う。処理時間計測には画像サイズを 150x113, 300x225, 600x450 の 3 種類を用いる。使用した PC は Intel(R) Xeon 2.66GHz \times 8, メモリ 4.0GB である。表 4 に各手法での処理時間と繰り返し回数を示す。表 4 より、提案手法は繰り返し処理が入るため、処理時間が大幅に増加する。解決策として、繰り返し回数の軽減や Lazy Snapping [7] のようにスーパーピクセルからグラフを作成することによるグラフカットの高速化が考えられる。

表 4 各手法の処理時間 [s] (繰り返し回数)

	Interactive Graph Cuts [4]	GrabCut [8]	提案手法
150x113	0.38	2.91 (4 回)	9.81 (8 回)
300x225	0.94	8.15 (3 回)	37.59 (10 回)
600x450	3.57	98.04 (12 回)	154.70 (12 回)

4.4 動画への応用

本手法は、動画への適応が可能である。動画からグリッド上に n-link を作成する。t-link は、画像と同様にすべてのピクセルに対して作成する。図 10 に動画に対応したグラフを示す。このとき、seed は 1 フレーム目のみに与える。対象とする動画は画像サイズ 320x240, フレーム数 40 である。3.3 で示すように、各フレームごとに距離変換を行い事前確率を更新する。尤度は、すべてのフレームのセグメンテーション結果から物体と背景の GMM を更新する。図 11 に人と車のシーケンスに対するセグメンテーション結果を示す。図 11 より、本手法を用いて動画のセグメンテーションが可能であることがわかる。これは、フレーム間で物体領域が重なるピクセルが存在するため、時間軸方向の n-link より物体ラベルを伝播させる

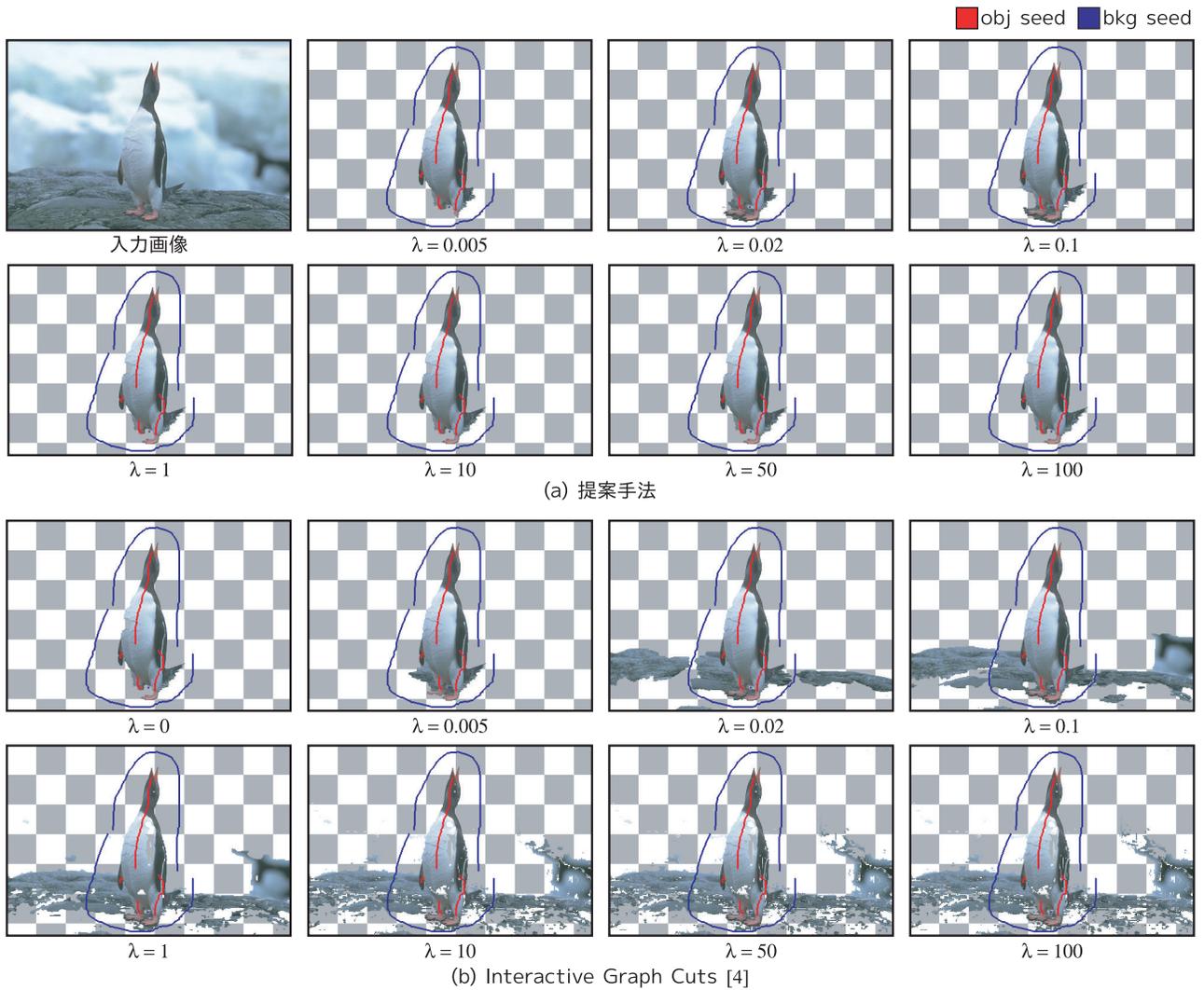


図 9 λ の変化によるセグメンテーション結果の違い

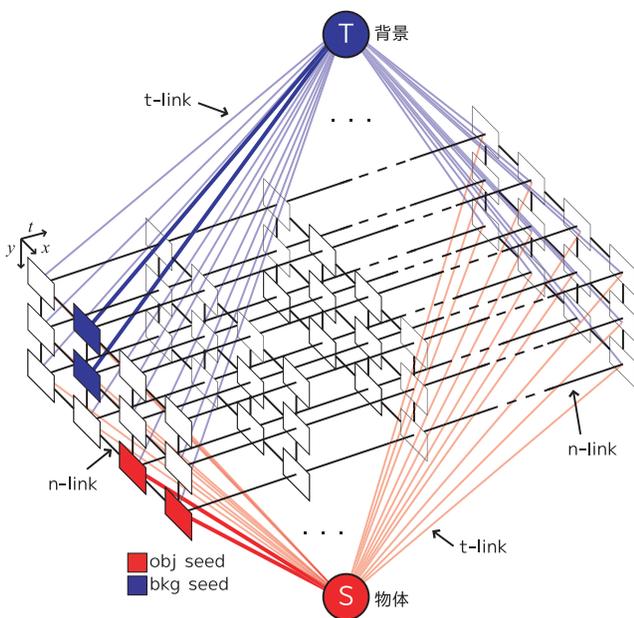


図 10 動画画像からグラフの作成

ことができるため、セグメンテーションが可能である。

5. おわりに

本稿では、ガウシアンフィルタの平滑化度合いを変化させた画像に対し、平滑化度合いが大きなものから Graph Cuts を繰り返し行う手法を提案した。本手法では、セグメンテーション結果から色分布と各ピクセルの事前確率を次のセグメンテーションの入力とすることにより、大域的なセグメンテーションから局所的なセグメンテーションと変化させることにより、従来の Graph Cuts と比較し 4.7% 向上させることができた。また、通常の Graph Cuts で成功する画像に対しても、本手法は同程度の精度でセグメンテーション可能であることを確認した。今後は、スーパーピクセルを用いた高速化と動画画像に対しての高精度なセグメンテーション手法を検討する予定である。

謝 辞

本研究は、文部科学省ハイテク・リサーチ・センター整備事業（平成 16 年度～平成 20 年度）による私学助成を得て行われている。

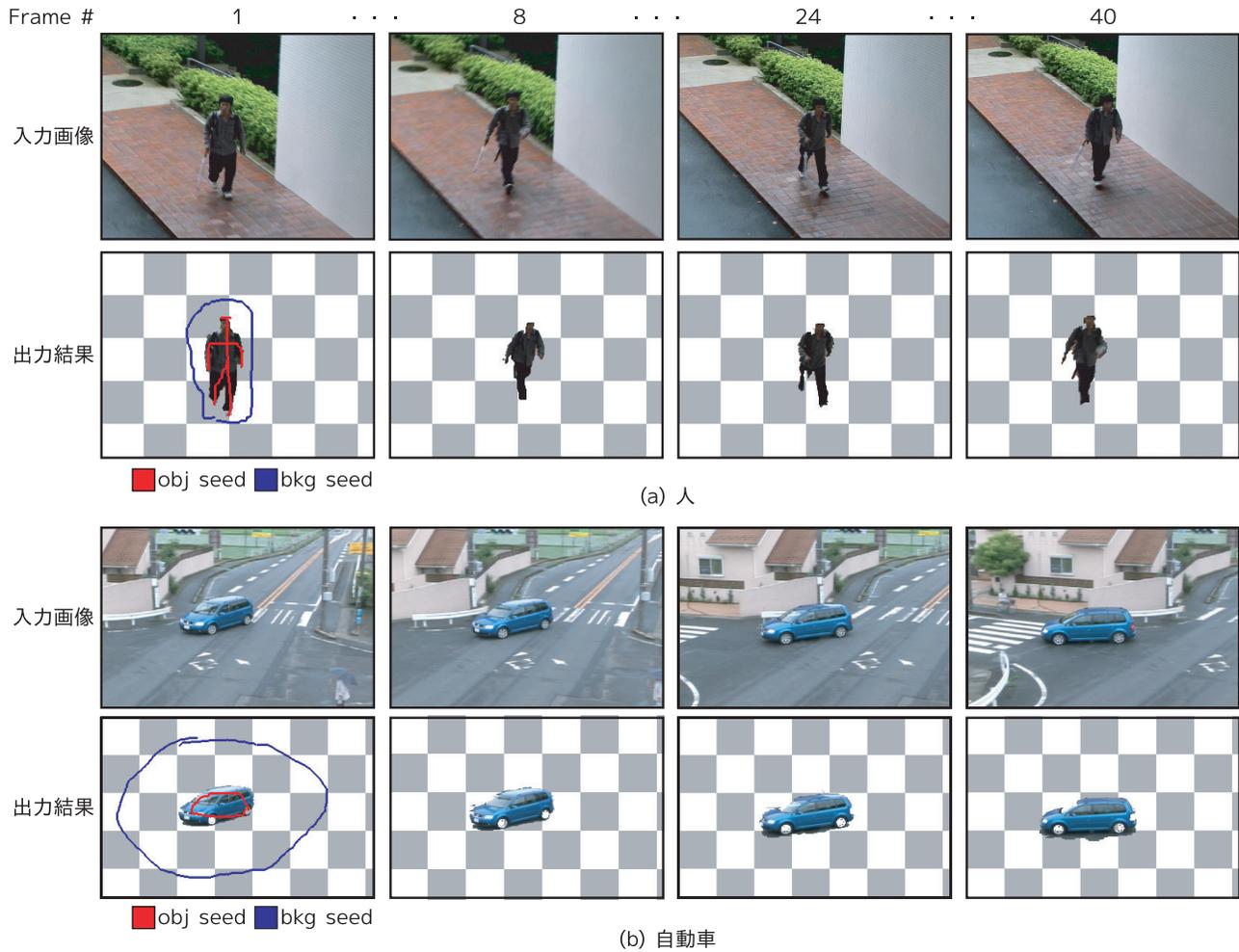


図 11 動画に対するセグメンテーション結果

文 献

- [1] M. Kass, A. Witkin, D. Terzopoulos, "Snakes : Active Contour Models", *Int. J. Comput Vision*, vol.1, no.3, pp.321-331, 1988.
- [2] M. Sussman, P. Smereka, S. Osher, "A Level Set Approach for Computing Solutions to Incompressible Two-phase Flow", *Journal of Computational Physics*, vol.114, p.146-159, 1994.
- [3] Y. Boykov, V. Kolmogorov, "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision", *PAMI*, vol. 26, no. 9, pp. 1124-1137, Sept. 2004.
- [4] Y. Boykov, M-P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images", *ICCV*, vol. I, pp. 105-112, 2001
- [5] Y. Boykov, G. Funka-Lea, "Graph Cuts and Efficient N-D Image Segmentation", *IJCV*, vol. 70, no. 2, pp. 109-131, 2006.
- [6] 石川 博, "グラフカット", 情報処理学会研究報告, 2007-CVIM-158-26, pp. 193-204, 2007.
- [7] Y. Li, J. Sun, C-K. Tang, and H-Y. Shum, "Lazy Snapping", *ACM Trans. Graph.*, Vol. 23, No. 3. (August 2004), pp. 303-308, 2004.
- [8] C. Rother, V. Kolmogorv, A. Blake, "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts, " *ACM Trans. Graphics (SIGGRAPH '04)*, vol.23, no.3, pp.309-314, 2004.
- [9] C. Stauffer and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-time Tracking", *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.246-252, 1999.
- [10] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum-likelihood From Incomplete Data via the EM lgorithm", *Journal of the Royal Statistical Society. Series B*, Vol. 39, No. 1 , pp. 1-38, 1977.
- [11] GrabCut Database: <http://research.microsoft.com/vision/cambridge/i3l/segmentation/GrabCut.htm>