

## 改良領域拡張法による高速画像切り抜き手法の提案と評価

清野 達也<sup>†1,\*1</sup> 林 貴宏<sup>†2,\*2</sup> 尾内 理紀夫<sup>†1,†2</sup>  
三條 正裕<sup>†3</sup> 森 正弥<sup>†3</sup>

領域拡張法 (Seeded Region Growing) を改良した高速な画像切り抜き手法を開発し、評価した。本手法は Seeded Region Growing を出発点としており、ユーザの引いた手書き線によって指示される初期前景と初期背景を隣接ピクセルへと伝播させていくことで、画像全体を前景領域、背景領域に分割する。飛び地を含む画像に対する切り抜き精度向上のために、Seeded Region Growing を改良し、閾値を用いた伝播条件を導入している。さらにテクスチャパターンを含む画像に対する切り抜き精度向上のために、Seeded Region Growing を改良し、ユーザの引いた線から前景で利用される色と背景で利用される色を推定し、推定色を用いて各隣接ピクセルに対する伝播の優先度を決定することとした。また、新たなデータ構造を本手法に導入し、高速な処理を行うことを可能にした。本手法を実装したシステムを開発し、評価実験を行い、既存手法と比較し、本手法が処理速度を低下させることなく、手書き線のような初期前景領域、初期背景領域の面積が小さい入力での切り抜き精度の優位性を持つことを確認した。

### Proposal and Evaluation of Fast Image Cutout Based on Improved Seeded Region Growing

TATSUYA KIYONO,<sup>†1,\*1</sup> TAKAHIRO HAYASHI,<sup>†2,\*2</sup>  
RIKIO ONAI,<sup>†1,†2</sup> MASAHIRO SANJO<sup>†3</sup> and MASAYA MORI<sup>†3</sup>

In this paper, we propose and evaluate a method for Fast Image Cutout using improved Seeded Region Growing. The starting point for the method is Seeded Region Growing, which divides an image into foregrounds and backgrounds by growing the initial foregrounds and backgrounds represented by user drawing lines to the neighbor pixels. To improve the precision of Seeded Region Growing for images including enclaves, the method adopts a threshold condition. In addition, to improve the precision of Seeded Region Growing for images including texture patterns, the method estimates foreground colors and background colors from initial foregrounds and backgrounds and decides growing-priorities for each neighbor pixel depending on the estimated colors. We propose a new

data structure for the method, and could achieve speed up. The experimental results have shown the method has the same processing speed as traditional methods and has better precision than traditional methods, when we input small area of initial foregrounds and backgrounds such as user drawing lines.

#### 1. はじめに

画像から任意の領域を抽出する処理は、画像や映像の編集において多くの場面で用いられている<sup>1)</sup>。古典的な画像の切り抜き手法として、マウスなどのポインティングデバイスを用いて領域の境界線をなぞることで抽出する手法があげられる。この手法はユーザに高度な技能を要求し、作業負担も大きいため、これに対処すべく画像切り抜きの効率化研究が数多く行われてきた。初期の研究では Intelligent Scissors<sup>2)</sup> や Image Snapping<sup>3)</sup> など、ユーザが前景領域と背景領域の境界線の一部をマーキングすることで最適な境界線を推定する手法が開発された。これらの研究によって領域の境界線全体をなぞるというユーザの負担は軽減されたが、領域が複雑な形状の場合にはいまだ負担は大きく長時間の作業となってしまう。

そのため近年では、Lazy Snapping<sup>4)</sup>、Random Walks<sup>5)</sup>、SIOX<sup>6)</sup> など、ユーザが前景領域の一部と背景領域の一部にそれぞれマーキングすることで画像を切り抜き手法が開発されている。これらの手法はユーザがマーキングした領域を初期前景領域、初期背景領域とし、マーキングされていない未知領域が前景であるか背景であるかを推定する。よってユーザは領域の一部を指定するだけで、複雑な形状の領域に対しても容易に画像を切り抜くことが可能となる。

マーキングの入力には、図 1 左に表されるような手書き線を用いた入力法が一般的に用いられる。手書き線によるマーキングでは、手書き線が通るピクセルを初期前景領域、初期背景領域とする。手書き線のマーキングを用いる手法はインタラクティブな入力を可能とす

†1 電気通信大学大学院電気通信学研究科

Graduate School of Electro-Communications, The University of Electro-Communications

†2 電気通信大学電気通信学部

Faculty of Electro-Communications, The University of Electro-Communications

†3 楽天株式会社楽天技術研究所

Rakuten Institute of Technology, Rakuten Inc.

\*1 現在、グーグル株式会社

Presently with Google Japan Inc.

\*2 現在、新潟大学工学部

Presently with Faculty of Engineering, Niigata University



図 1 左：対象画像とユーザが入力した手書き線．実線が前景領域，点線が背景領域を表す．右：出力のマスク画像．白が前景領域，黒が背景領域を表す

Fig.1 Left: Example of a target image and lines. Solid lines mean foreground. Dotted lines mean background. Right: Example of an output mask image. White region means foreground. Black region means background.

る研究が多く，切り抜き結果をユーザに表示し，切り抜きに失敗した場合にユーザが手書き線を追加入力することで修正を容易にしている．しかし既存研究は大きい画像サイズの切り抜きにおいて低速であり，追加入力までの待ち時間が大きいという問題が存在する．

そこで本論文は，手書き線のマーキングを用いて高速に画像を切り抜き手法を提案する．本手法は Seeded Region Growing (領域拡張法<sup>7)</sup>) を出発点とする．しかし Seeded Region Growing には，飛び地を含む画像の切り抜きに失敗するという欠点と，テクスチャパターンを含む画像の切り抜きに失敗するという欠点が存在する．そこで本手法は Seeded Region Growing を改良し，これらの欠点の解決を図った．さらに高速化のための新たなデータ構造を設計，導入した．以下では，2章で関連研究について述べる．3章で Seeded Region Growing の改良を含む本手法について説明し，4章で高速化のための新たなデータ構造を提案する．5章で本手法を実装したシステムによる精度評価実験，6章で速度評価実験について述べ，7章でまとめる．

## 2. 関連研究

本手法で用いる手書き線によるマーキングの入力方法以外にも，GrabCut<sup>8)</sup> で用いられるような，初めに矩形で注目領域を指定することで矩形外の領域を初期背景領域とし，その後手書き線を用いるマーキングの入力方法や，後述の Alpha Matting で用いられるような，領域の境界線付近のみを未知領域とするマーキングの入力方法が存在する．これらのマーキングは一般的に Trimap で表現される．Trimap とは図 2 で示されるような，対象画



図 2 左：対象画像．右：対象画像の Trimap．白が初期前景領域，黒が初期背景領域，灰色が未知領域を表す  
Fig.2 Left: Example of a target image. Right: Example of Trimap. White region means initial foreground. Black region means initial background. Grey region means unknown region.

像を初期前景領域，初期背景領域，未知領域の 3 つに分割した画像である．

一方，本手法は画像の切り抜きを前景領域，背景領域の 2 つの領域で表現する．このように前景と背景の 2 値で切り抜きを表現する手法は Hard Segmentation と呼ばれる．ただし画像には透明な物体や，アンチエイリアスによる輪郭のボケのように前景と背景の色が混合した領域が存在する場合があります，この場合は Hard Segmentation のような 2 値での切り抜きでは不十分である<sup>9)</sup>．そのため Alpha Matting と呼ばれる手法では，前景と背景の混合率を推定することで前景と背景が混合した領域に対応する．

Alpha Matting の手法には Bayesian Matting<sup>10)</sup>，Poisson Matting<sup>11)</sup>，Robust Matting<sup>12)</sup>，Spectral Matting<sup>13)</sup> などがある．Alpha Matting は前景と背景の色が混合した領域の混合率を推定するため，Hard Segmentation よりも情報量の大きな画像の切り抜きが可能であるが，未知領域の面積が大きい場合には精度が大きく低下し，処理速度も遅くなる傾向がある．そのため Alpha Matting では，広面積の初期前景領域や初期背景領域を指定しなければならず，ユーザに負担を強いことになる．

この問題を解決する研究も行われており，Hard Segmentation で生成した前景領域，背景領域の境界周辺を未知領域と再定義することで Trimap を生成し，その Trimap を Alpha Matting への入力とする枠組みが提案されている<sup>14)</sup>．この枠組みを用いることで，Hard Segmentation のインタラクティブ性と容易な入力を維持したまま Alpha Matting の適用が可能となる．本手法は切り抜き結果の前景領域，背景領域の境界周辺から未知領域を再定義することで，Alpha Matting のための前処理としても利用可能である．

## 3. 提案手法

出発点とする Seeded Region Growing に基づく画像切り抜きと，その欠点解消のための

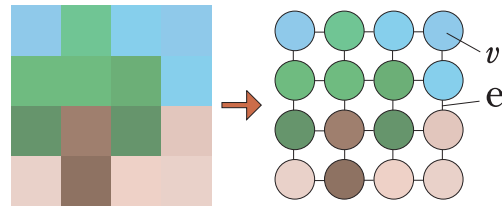


図 3 画像のグラフ表現

Fig. 3 Expression of an image by a graph.

改良法について述べる。なお、本論文ではこの提案手法を本手法と呼称する。

### 3.1 本手法の入力と出力

本手法における入力、出力の例を図 1 に示す。ユーザはマウスなどのポインティングデバイスを用いて、図 1 左に表されるように前景領域、背景領域を手書き線で指定する。本手法はこれらの入力された手書き線を基に画像を切り抜き、その結果をユーザに表示する。ユーザは表示された切り抜き結果が意図した切り抜きと異なる場合に新たな手書き線を追加することで、インタラクティブに画像を切り抜くことが可能となる。最終的な切り抜き結果は、図 1 右に表されるような前景領域、背景領域を表すマスク画像である。

### 3.2 Seeded Region Growing に基づく画像の切り抜き

Seeded Region Growing は画像のセグメンテーション手法であり、ユーザの注目する領域が、隣接した領域を取り込み、拡大を行うことでセグメント化を行う。画像のセグメンテーションとは画像を複数の領域に分割する処理で、画像認識の前処理として用いられる。

ここで図 3 に示すように、画像をグラフ  $G = \langle V, E \rangle$  で表す。  $V$  はノードの集合であり、  $E$  はノードをつなぐ辺の集合である。このときノード  $v \in V$  は画像のピクセルに対応し、辺  $e = (s, t) \in E (s, t \in V)$  は上下左右 4 方向の隣接ピクセル間に対応する。

本手法は Seeded Region Growing に基づき、前景領域  $F$ 、背景領域  $B$  に隣接する未知領域のノード  $t \in N$  を取り込むことで前景領域  $F$ 、背景領域  $B$  を拡大することで画像の切り抜きを実現する。前景領域  $F$ 、背景領域  $B$  が未知領域のノード  $t \in N$  を取り込む操作を「伝播」と呼び、前景領域  $F$  の伝播は  $F = F \cup \{t\}$ 、  $N = N - \{t\}$ 、背景領域  $B$  の伝播は  $B = B \cup \{t\}$ 、  $N = N - \{t\}$  という操作で定義される。このときのノード  $t$  を伝播候補と呼び、本手法における領域の伝播は優先度の高い伝播候補  $t$  から順に行われる。優先度はコスト  $C(t)$  によって与え、コスト  $C(t)$  の値が小さい伝播候補ほど伝播の優先度が高い。これらの具体的な手順を以下に示す。

手順 1 ユーザが入力した前景を表す手書き線が通るノードの集合を初期前景領域  $F_0 \subset V$ 、背景を表す手書き線が通るノードの集合を初期背景領域  $B_0 \subset V$  とする。ただし、  $F_0 \cap B_0 = \emptyset$  である。

手順 2 前景領域  $F$ 、背景領域  $B$ 、未知領域  $N$ 、伝播候補  $t$  の集合  $Q$  を次の操作で初期化する。

$$F = F_0$$

$$B = B_0$$

$$N = V - (F \cup B)$$

$$Q = \{t \in N | e(t) \cap (F \cup B) \neq \emptyset\}$$

ただし  $e(t)$  は  $t$  に隣接する伝播元のノードで次式で定義される。

$$e(t) = \{s \in (F \cup B) | (s, t) \in E\} \quad (1)$$

手順 3 次に伝播が行われる伝播候補  $t_m = \operatorname{argmin}_{t \in Q} \{C(t)\}$  を選択する。

手順 4 伝播元のノード  $s \in e(t_m)$  の属する領域  $R(s)$  (つまり  $R(s) = F$  または  $B$ ) を伝播候補  $t_m$  に伝播する。すなわち、  $R(s) = R(s) \cup \{t_m\}$ 、  $N = N - \{t_m\}$  とする。

手順 5 伝播候補  $Q$  を次の操作で更新する。

$$Q = Q - \{t_m\}$$

$$Q = Q \cup \{t \in n(t_m)\}$$

ただし  $n(t)$  は  $t$  に隣接する未知領域のノードで次式で定義される。

$$n(t) = \{u \in N | (u, t) \in E\} \quad (2)$$

手順 6  $Q = \emptyset$  ならば終了。そうでなければ手順 3 に戻り処理を繰り返し行う。

### 3.3 コストの定義

コスト  $C(t)$  は、

$$C(t) = W_1(s, t) \cdot D(s, t) + W_2(s, t) \cdot C_s \quad (3)$$

により定義される。ここで  $s$  は伝播元のノードであり、  $s = e(t)$  である。

$D(s, t)$  はノード  $s$  と  $t$  の RGB 空間における色の二乗距離で、  $s$  と  $t$  の色の類似性に対応する。

$C_s$  は、過去に前景領域  $F$  もしくは背景領域  $B$  がノード  $s$  に伝播したときのコストである。手順 4 において以下の操作が行われているものとする。

$$C_s = C(t_m)$$

ただし、  $s \in F_0$  もしくは  $s \in B_0$  のとき、  $C_s = 0$  とする。  $C_s$  の項によって、過去のコス

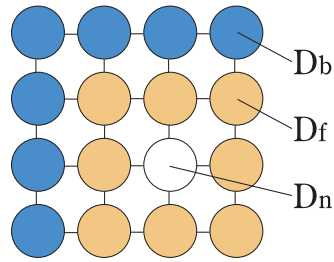


図4 飛び地の例  
Fig.4 Example of enclave.

トが蓄積されることになる．コストを蓄積していく理由は Seeded Region Growing の改良法について述べた 3.4 節で説明する．

$W_1(s, t), W_2(s, t)$  は,  $D(s, t), C_s$  に対する重み付けである．詳細な定義は Seeded Region Growing の改良法について述べた 3.5 節で説明する．

### 3.4 飛び地判定の導入

Seeded Region Growing の欠点として、飛び地の領域伝播ができないことがあげられる．たとえば図 4 で示されるようなドーナツ状の前景領域  $D_f$ ，外側の背景領域  $D_b$ ，内側の未知領域  $D_n$  が与えられたとき， $D_n$  が背景領域になる場合でも  $D_b$  と  $D_n$  は隣接していないことから， $D_b$  と  $D_n$  を同一の領域とすることはできず， $D_n$  は  $D_f$  と隣接していることからコストの値にかかわらず前景領域となってしまうためである．

そこで本手法では、コスト  $C(t)$  が閾値  $T$  を超えた場合に飛び地と判定し、手順 4 において  $R(s) = F$  であっても背景領域  $B$  が  $t$  に伝播するとして飛び地への伝播を実現するという改良を施す．そのため本手法は背景領域の飛び地にのみ対応する．これは画像中にユーザがマーキングした領域と同じ物体が飛び地で存在する場合、これを前景領域とするかどうかの判断をユーザに任せるためである．また飛び地への伝播は、 $C(t)$  の大きさにかかわらず最優先で行われる．閾値  $T$  は初期前景領域  $F_0$ ，初期背景領域  $B_0$  から次式，

$$T = w \cdot \max\{D(f_b, b_d), D(f_d, b_b)\} \tag{4}$$

$$f_b = \operatorname{argmax}_{v \in F_0} \{Y(\mathbf{c}_v)\} \tag{5}$$

$$f_d = \operatorname{argmin}_{v \in F_0} \{Y(\mathbf{c}_v)\} \tag{6}$$

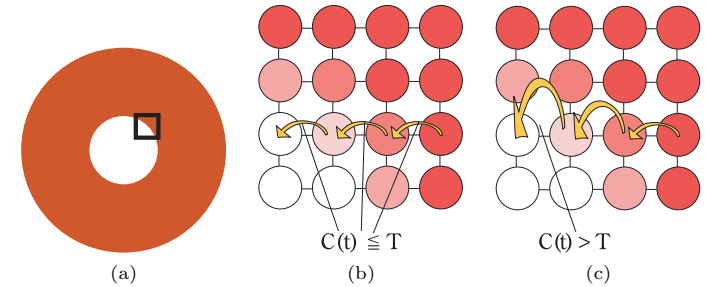


図 5 (a) : 対象画像 . (b) : (a) の矩形箇所の拡大イメージ，コストの蓄積を行っていない場合 . (c) : (a) の矩形箇所の拡大イメージ，コストの蓄積を行っている場合

Fig.5 (a): Target image. (b): The enlarged image of the rectangle in (a), without cost accumulation. (c): The enlarged image of the rectangle in (a), with cost accumulation.

$$b_b = \operatorname{argmax}_{v \in B_0} \{Y(\mathbf{c}_v)\} \tag{7}$$

$$b_d = \operatorname{argmin}_{v \in B_0} \{Y(\mathbf{c}_v)\} \tag{8}$$

$$Y(\mathbf{c}_v) = 0.299r_v + 0.587g_v + 0.114b_v \tag{9}$$

により定義する．

ここで  $w$  は重み係数を示すパラメータであり、 $w$  を変化させることにより閾値  $T$  が変化する．本手法では  $w = 1.5$  としている． $w$  を変化させたときの結果の変化、 $w = 1.5$  とした理由については 5.2 節において述べる．

また、 $\mathbf{c}_v = (r_v, g_v, b_v)$  であり、 $r_v, g_v, b_v$  はそれぞれノード  $v$  に対応するピクセルの R, G, B 値である．関数  $Y(\mathbf{c}_v)$  はノード  $v$  に対する色  $\mathbf{c}_v$  の輝度<sup>15)</sup> を表す．

しかし、ノード  $s$  とその隣接ノード  $t$  との色の距離  $D(s, t)$  が閾値  $T$  を超えることは少ない．これは図 5 (b) のように領域の境界線がアンチエイリアスによってぼやけることで、隣接ノードの色の距離  $D(s, t)$  が小さくなるためである．そこで式 (3) では、 $C_s$  を用いて図 5 (c) のようにコストを蓄積することで、アンチエイリアスによってぼやけた境界線でも飛び地の判定を可能としている．

### 3.5 代表色を用いたテクスチャパターンへの対応

Seeded Region Growing は隣接した領域の色の類似性に注目することから、テクスチャパターンのように色が急激に変化する領域に対応できないという欠点が存在する．そのため本手法ではピクセルに基づくセグメンテーション手法を併用するという改良を施す．

ピクセルに基づくセグメンテーション手法は、まず、初期前景領域および初期背景領域からそれぞれ色特徴をあらかじめ抽出しておく。そして、未知領域内のピクセルの色とあらかじめ抽出しておいた色特徴を比較することで前景または背景へと分類する。テクスチャパターンは利用される色数に限りがあるため、色特徴を用いることでテクスチャパターンへの対応が可能となる。

ピクセルに基づくセグメンテーション手法は既存の画像切り抜き手法においても用いられている。Lazy Snapping では初期前景領域、初期背景領域の色に対して K-means 法を用いたクラスタリングを行い、各クラスタの代表色と各ピクセルとの類似性を評価する。また、SIOX では初期前景領域、初期背景領域の Color Signature に対して kd 木を用いたクラスタリングを行い、各クラスタと各ピクセルとの色の類似性を評価する。

しかしこれらのクラスタリング手法は計算コストが大きいいため、本手法では初期前景領域、初期背景領域から量子化数を下げた色の集合（代表色集合）を色特徴として抽出し、各ノードが代表色集合に属するかどうかを判別する。判別の結果から  $W_1(s, t)$ ,  $W_2(s, t)$  の重み付けを決定し、コストに反映することでテクスチャパターンへの対応を行う。RGB 色空間における色  $c = (r, g, b)$  が量子化数 256 であったとき、量子化数を  $s$  に変更する関数  $q(c)$  は次式で定義される。

$$q(c) = q(r, g, b) = \left( \left\lfloor \frac{r}{256} s \right\rfloor, \left\lfloor \frac{g}{256} s \right\rfloor, \left\lfloor \frac{b}{256} s \right\rfloor \right) \quad (10)$$

$\lfloor x \rfloor$  は実数  $x$  を超えない最大の整数を表す。本手法では  $s = 16$  としている\*1。

前景の代表色集合  $K_F$ 、背景の代表色集合  $K_B$  は、次式で定義される。

$$K_F = \bigcup_{v \in F_0} q(c_v) \quad (11)$$

$$K_B = \bigcup_{v \in B_0} q(c_v) \quad (12)$$

$W_1(s, t)$ ,  $W_2(s, t)$  を伝播元のノード  $s$ 、伝播候補  $t$  の色  $c_t$ 、前景の代表色集合  $K_F$ 、背景の代表色集合  $K_B$  を用いて次式で定義する。

$$W_1(s, t) = \begin{cases} 1 & \begin{cases} R(s) = F, q(c_t) \in K_F - K_B \\ R(s) = B, q(c_t) \in K_B - K_F \end{cases} \\ 4 & \begin{cases} R(s) = F, q(c_t) \in K_B - K_F \\ R(s) = B, q(c_t) \in K_F - K_B \end{cases} \\ 2 & \text{(otherwise)} \end{cases} \quad (13)$$

$$W_2(s, t) = \begin{cases} 0 & \begin{cases} R(s) = F, q(c_t) \in K_F \\ R(s) = B, q(c_t) \in K_B \end{cases} \\ 1 & \text{(otherwise)} \end{cases} \quad (14)$$

$W_1(s, t)$  は色の距離  $D(s, t)$  への重み付けで、1, 4, 2 の 3 値のいずれかの重みとなる。伝播候補  $t$  の色  $c_t$  が領域  $R(s)$  の代表色集合にのみ属する場合すなわち  $K_F$  または  $K_B$  のいずれか一方に属する場合に、 $t$  は  $R(s)$  に属する可能性が高いと判別し 1 を設定する。伝播候補  $t$  の色  $c_t$  が領域  $R(s)$  とは逆 ( $R(s) = F$  のときは  $B$ ,  $R(s) = B$  のときは  $F$ ) の領域の代表色集合にのみ属する場合に、 $t$  は  $R(s)$  に属する可能性が低いと判別し 4 を設定する。また、伝播候補  $t$  の色  $c_t$  が前景の代表色集合  $K_F$ 、背景の代表色集合  $K_B$  のどちらにも属する場合またはどちらにも属さない場合は、 $t$  はどちらの前景領域、背景領域に対しても中立であると判別し 2 を設定する。

$W_2(s, t)$  は蓄積する過去のコスト  $C_s$  への重み付けで、伝播候補  $t$  の色  $c_t$  が領域  $R(s)$  の代表色集合に属する場合は過去のコストを蓄積しない。すなわち 0 を設定する。これは前景領域と背景領域の境界線付近のみコストの蓄積を行わせるためである。

色  $c$  が代表色集合に属するかどうかは、ルックアップテーブルを用いることで高速な判別が可能となる。 $K_F$ ,  $K_B$  に対応するルックアップテーブルは、それぞれ  $F_{i,j,k}$ ,  $B_{i,j,k}$  と表すと次式で定義される。

$$F_{i,j,k} = \begin{cases} 1 & (q(i, j, k) \in K_F) \\ 0 & \text{(otherwise)} \end{cases} \quad (15)$$

$$B_{i,j,k} = \begin{cases} 1 & (q(i, j, k) \in K_B) \\ 0 & \text{(otherwise)} \end{cases} \quad (16)$$

この判別の時間計算量は  $O(1)$  である。

\*1 量子化数の変換  $q(x)$  は、変換前の量子化数を  $q_s$ 、変換後の量子化数を  $q_t$  としたとき、変換式を  $q(x) = q_t/q_s \cdot x$  とする手法と、 $q(x) = (q_t - 1)/(q_s - 1) \cdot x$  とする手法が存在する。本手法は変換後の分布を一樣にするために  $q_t/q_s$  を採用している。

#### 4. 高速化のためのデータ構造

Seeded Region Growing を高速化するための新たなデータ構造について述べる．

##### 4.1 Priority Queue による高速化

伝播候補の集合  $Q$  は  $\min_{t \in Q} \{C(t)\}$  となる伝播候補  $t$  を検索することから，Priority Queue を用いて実現できる．Priority Queue とは優先度を持つ要素を格納し，最も優先度の高い要素への高速な参照が可能なデータ構造である．

本手法では伝播候補  $t$  を Priority Queue に格納する要素，そしてコスト  $C(t)$  を優先度とし，コストが小さいほど優先度の高い伝播候補とする．Priority Queue を  $PQ$  で表したとき，伝播候補  $t$  の追加操作を  $PQ.push(t)$ ，コストが最小の伝播候補への参照操作を  $PQ.top()$ ，コストが最小の伝播候補の削除操作を  $PQ.pop()$  と表すものとする．Priority Queue の実現に Binary Heap を用いた場合， $PQ.push(t)$  の時間計算量は  $O(\log(|PQ|))$ ， $PQ.top()$  の時間計算量は  $O(1)$ ， $PQ.pop()$  の時間計算量は  $O(\log(|PQ|))$  である．ただし  $|PQ|$  は， $PQ$  の要素数を表す．

このとき  $PQ.push(t)$  および  $PQ.pop(t)$  の最大時間計算量は， $O(\log N)$  である． $N$  は全ピクセル数である．本手法では  $PQ.push(t)$  および  $PQ.pop(t)$  を未知領域のノード  $n \in N$  すべてに対して行うことから，本手法の最大時間計算量は  $O(N \log N)$  となる．

##### 4.2 Priority Queue と Queue の併用による高速化

$PQ.push(t)$  および  $PQ.pop(t)$  は未知領域のノード  $n \in N$  すべてに対して行うことから，計算機の処理が集中する操作であることが分かる．そこで Priority Queue の代わりに Priority Queue と Queue を併用したデータ構造を提案し，Priority Queue のみを用いた場合よりも高速な処理を行う．Queue は先入れ先出し (FIFO) のデータ構造である．Queue を  $DQ$  で表したとき，伝播候補  $t$  の追加操作を  $DQ.push(t)$ ，先頭の伝播候補への参照操作を  $DQ.top()$ ，先頭の伝播候補の削除操作を  $DQ.pop()$  と表すものとする．このとき， $DQ.push(t)$ ， $DQ.top()$ ， $DQ.pop()$  の時間計算量は  $O(1)$  である．

Priority Queue と Queue の併用を行うデータ構造を  $HQ$  とし， $PQ$ ， $DQ$  と同様に  $HQ.push(t)$ ， $HQ.top()$ ， $HQ.pop()$  を以下の操作により定義する．

$$HQ.push(t) = \begin{cases} DQ.push(t) (PQ \neq \emptyset, C(t) \leq C(PQ.top())) \\ PQ.push(t) (otherwise) \end{cases}$$

$$HQ.top() = \begin{cases} DQ.top() (DQ \neq \emptyset) \\ PQ.top() (otherwise) \end{cases}$$

$$HQ.pop() = \begin{cases} DQ.pop() (DQ \neq \emptyset) \\ PQ.pop() (otherwise) \end{cases}$$

ただし 3.2 節の手順 2 において  $Q$  を初期化する際の伝播候補は，すべて  $PQ$  に格納されるものとする．

$HQ.top()$ ， $HQ.pop()$  はコストの大小にかかわらず，Queue に格納された先頭の伝播候補から優先して参照される．その結果必ずしも最小コストのノードから順に伝播されるとは限らないが，Queue に格納される条件  $C(t) \leq C(PQ.top())$  を満たす伝播候補は，コストが小さいことから  $PQ.top()$  よりも先に伝播が行われることは保証される．Queue を併用することで Priority Queue を単独で使う場合に比べ， $push$ ， $pop$  の操作が高速化される． $HQ.top()$  は，必ずしもコストが最小の伝播候補を参照しないため，Priority Queue と Queue の併用は，Priority Queue のみを用いる場合と同一の結果になるとは限らない．これは式 (3) における過去のコスト  $C_s$  が最小のコストではないことから， $PQ$  に格納される伝播候補においても伝播順序が変わるためである．

$HQ.push(t)$  から  $PQ.push(t)$  が呼び出されるとき，伝播候補  $t$  は  $C(t) > C(PQ.top())$  であることから， $PQ.top()$  が過去の  $PQ.top()$  よりもコストが大きいうという単調性が保証される．そのため， $PQ$  は Radix Heap<sup>16)</sup> を用いた実現が可能となる．Radix Heap を用いた場合， $PQ.push(t)$ ， $PQ.top()$ ， $PQ.pop()$  の時間計算量は  $O(1)$  である．そのため Priority Queue と Queue を併用し，Priority Queue の実現に Radix Heap を用いた場合，本手法の最大時間計算量は  $O(N)$  となる．

#### 5. 精度評価実験

本手法を実装した画像切り抜きシステム「切絵」を開発し，Vision at MSR Cambridge: Ground truth database<sup>\*1</sup>で提供される 50 枚の自然画像と正解画像を用いて，精度に関して既存手法 (Seeded Region Growing, Lazy Snapping, SIOX, Random Walks) との比

\*1 <http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm>



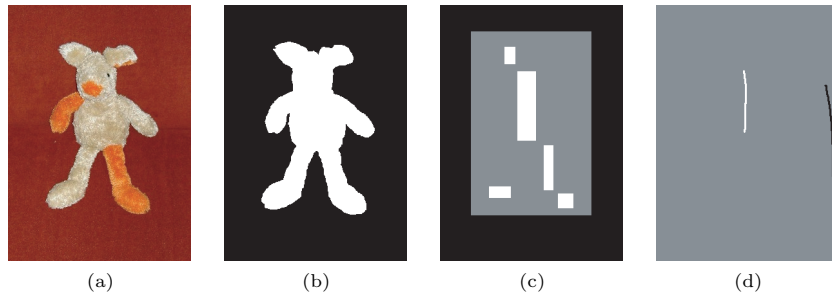


図 6 (a): 対象画像. (b): 正解画像. 白が前景, 黒が背景, 灰色が混合領域を表す. (c): 入力 1. (d): 入力 2  
 Fig. 6 (a): Target image. (b): Correct image. White region means foreground. Black region means background. Grey region means mixed region. (c): Input method 1. (d): Input method 2.

較評価を行った. 以降, 実験方法とパラメータ設定のための予備実験, そして実験結果について述べる.

### 5.1 実験方法

「切絵」システムへの入力として各画像に対し 2 種類の Trimap を用いる. 2 種類の Trimap(入力 1, 入力 2) の例を図 6 に表す. 入力 1 は Benchmarking SIOX<sup>\*1</sup> で提供される Trimap である. 入力 2 は「切絵」システムを用いて, 前景と背景の手書き線を 1 本ずつ入力し生成した Trimap である. これらの入力に対して「切絵」システムが前景と判断した領域を  $F_r$ , 正解画像の前景領域  $F_c$  として,  $F_r, F_c$  から前景領域に対する適合率  $p$  と再現率  $r$ , 適合率と再現率の調和平均である F 尺度  $f$  を求めて評価を行う. ただし正解画像には, 前景領域, 背景領域に判別することが不可能な混合領域  $N_c$  が前景と背景の境界線付近に存在する. そのため  $F_r, F_c$  から混合領域  $N_c$  を除き, 前景領域と判別できる領域のみで評価を行う. このとき適合率  $p$ , 再現率  $r$ , F 尺度  $f$  は次式で定義される.

$$p = \frac{|F_r \cap F_c|}{|F_r|} \quad (17)$$

$$r = \frac{|F_r \cap F_c|}{|F_c|} \quad (18)$$

$$f = \frac{2}{\frac{1}{p} + \frac{1}{r}} = \frac{2|F_r \cap F_c|}{|F_r| + |F_c|} \quad (19)$$

\*1 <http://www.siox.org/details.html>

ここで  $|F|$  は領域  $F$  のピクセル数である.

本手法は Priority Queue と Queue の併用を行う場合と, Priority Queue のみを用いた場合において結果に変化がでることから, 併用を行う場合を「本手法 (Hybrid Queue)」, Priority Queue のみの場合を「本手法 (Priority Queue)」と表記し比較を行う.

なお本手法 (Hybrid Queue), 本手法 (Priority Queue) は, とともに飛び地判定の導入, および代表色を用いたテクスチャパターンへの対応が含まれている.

飛び地判定の導入とテクスチャパターンへの対応の有無に関する効果を確認するために, 本手法 (Hybrid Queue) において, 飛び地判定の導入を行わなかった場合を「本手法 (飛び地対応なし)」, テクスチャパターンへの対応を行わなかった場合を「本手法 (テクスチャ対応なし)」と表記しこれらの手法に対しても精度の比較を行う. また, 既存手法として Seeded Region Growing, Lazy Snapping, SIOX, Random Walks を選択し, 本手法との精度の比較評価を行う.

### 5.2 予備実験

実験にあたっては, 既存手法, 本手法とも適切なパラメータを設定する必要がある. そのための予備実験を行った. 以降, その方法と結果について述べる.

#### (1) 方法

Lazy Snapping, Random Walks はそれぞれ結果に影響を及ぼすパラメータ  $\lambda, \beta$  が存在するため, 精度評価のためにはこれらのパラメータの適切な設定が必要である. そのためパラメータを一定量ごとに段階的に変化させ, 入力 2 を用いた切り抜き結果における平均 F 尺度の推移から適切なパラメータを選択することとした. Lazy Snapping のパラメータ  $\lambda$  を 0 から 20000 までの間, 2000 ごとに段階的に変化させ, 切り抜きを行った. 同様に, Random Walks のパラメータ  $\beta$  を 0 から 180 までの間, 15 ごとに段階的に変化させ, 切り抜きを行った.

また, 本手法 (Hybrid Queue) においても飛び地判定の閾値  $T$  に含まれる重み係数パラメータ  $w$  による結果の変化を考察するために,  $w$  を 0 から 4 までの間, 0.5 ごとに段階的に変化させ, 切り抜きを行った. そしてその結果を考察し, 適切な  $w$  の値を設定することとした.

#### (2) 結果

Lazy Snapping のパラメータ  $\lambda$  を変化させたときの平均適合率, 平均再現率, 平均 F 尺度の推移を図 7 に示す. 図 7 から,  $\lambda$  が 12000 のときに平均 F 尺度が最大となっており,  $\lambda$  が 12000 を超えてからは平均再現率, 平均適合率の推移は単調減少である. このことが

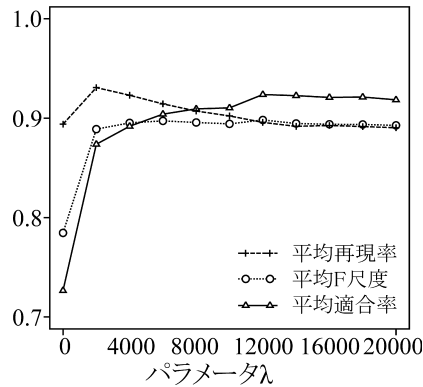


図 7 Lazy Snapping におけるパラメータ  $\lambda$  を変化させたときの精度の推移  
Fig. 7 Accuracy of Lazy Snapping by changing parameter:  $\lambda$ .

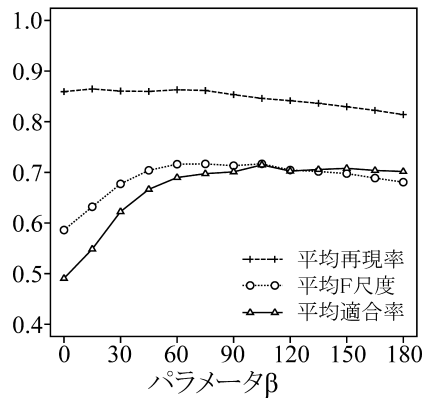


図 8 Random Walks におけるパラメータ  $\beta$  を変化させたときの精度の推移  
Fig. 8 Accuracy of Random Walks by changing parameter:  $\beta$ .

ら Lazy Snapping のパラメータ  $\lambda$  の適切な値を 12000 とした。

Random Walks のパラメータ  $\beta$  を変化させたときの平均適合率, 平均再現率, 平均 F 尺度の推移を図 8 に示す。図 8 から,  $\beta$  が 105 のときに平均 F 尺度が最大であり,  $\beta$  が 105 の前後の平均再現率, 平均適合率の変動は小幅であることから Random Walks のパラメータ  $\beta$  の適切な値を 105 とした。

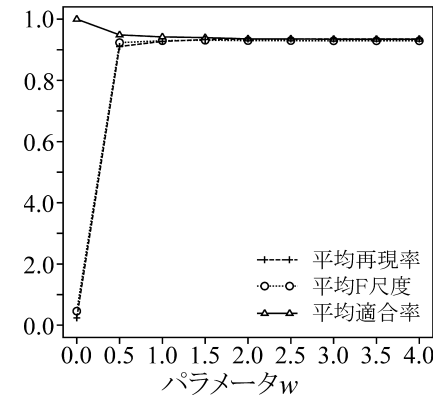


図 9 本手法 (Hybrid Queue) におけるパラメータ  $w$  を変化させたときの精度の推移  
Fig. 9 Accuracy of our method (Hybrid Queue) by changing parameter:  $w$ .

本手法 (Hybrid Queue) のパラメータ  $w$  を変化させたときの平均適合率, 平均再現率, 平均 F 尺度の推移を図 9 に示す。図 9 から,  $w$  が 1.5 のときに平均 F 尺度が最大であり, 1.5 を超えてからの平均再現率, 平均適合率の変動はきわめて小幅であることから, パラメータ  $w$  の適切な値を 1.5 とした。ここで, 図 13 (e) ~ (j) に  $w$  を変化させたときの切り抜き結果の推移を示す。飛び地と判定されるためには, 前景からの伝播コスト  $C(t)$  が閾値  $T$  を超える必要がある。閾値  $T$  に影響を与えるパラメータが  $w$  である (式 (4))。  $w$  が小さいと, 前景からの伝播コストが小さい場合でも飛び地と判定されるため, 飛び地だけでなく, 前景領域内にも背景領域が伝播される可能性が高くなる。その結果, 前景と判定されるべき箇所が背景と判定されるため再現率が低下する。一方,  $w$  が大きいと, 色差が大きくないと飛び地と判定されないため, 飛び地内であっても前景からの色差が小さい領域は背景とは判定されず, 前景がそのまま伝播する可能性が高くなる。その結果, 背景の一部を前景と判定するため適合率が低下する。このことは図 13 (e) ~ (j) に示した例からも確認できる。本手法 (Hybrid Queue) では, 平均 F 尺度が最大となるとき  $w$  の値を適切な値として設定している。本手法 (Hybrid Queue) で設定した  $w = 1.5$  での切り抜き結果は, 図 13 (h) であり,  $w$  を他の値に設定した場合に比べ適切に飛び地を判定できていることが確認できる。

### 5.3 実験結果

予備実験結果に基づくパラメータ設定をし, 画像切り抜きの精度実験を行った結果として得られた適合率, 再現率, F 尺度の平均値を表 1 (入力 1 を利用した場合), 表 2 (入力 2



表 1 入力 1 における画像切り抜き精度

Table 1 Accuracy comparison of Input method 1.

	平均適合率	平均再現率	平均 F 尺度
本手法 (Hybrid Queue)	0.9546	0.9108	0.9286
本手法 (Priority Queue)	0.9547	0.9142	0.9306
本手法 (テクスチャ対応なし)	0.8729	0.8901	0.8694
本手法 (飛び地対応なし)	0.9514	0.9053	0.9233
Seeded Region Growing	0.8875	0.8644	0.8676
Lazy Snapping	0.9617	0.9240	0.9383
SIOX	0.9530	0.9469	0.9485
Random Walks	0.9001	0.8919	0.8856

表 2 入力 2 における画像切り抜き精度

Table 2 Accuracy comparison of Input method 2.

	平均適合率	平均再現率	平均 F 尺度
本手法 (Hybrid Queue)	0.9397	0.9334	0.9311
本手法 (Priority Queue)	0.9343	0.9351	0.9292
本手法 (テクスチャ対応なし)	0.6080	0.8703	0.6584
本手法 (飛び地対応なし)	0.8819	0.9173	0.8856
Seeded Region Growing	0.7330	0.8211	0.7429
Lazy Snapping	0.9238	0.8958	0.8982
SIOX	0.8985	0.8348	0.8582
Random Walks	0.7147	0.8460	0.7168

を利用した場合) に示す。

また適合率, 再現率の平均値による考察だけでなく, 画像の切り抜き精度に関して, 各手法における, 適合率や再現率の入力データごとのばらつき程度, 切り抜き結果の安定性などを可視化して考察するため, 図 10, 図 11 に示すような適合率と再現率の分布図を導入した。分布図を用いれば, 図の右上に入力データが集中している手法ほど精度の良い切り抜きが安定して行われていることが読み取れる。また, 図の左や下に向かって入力データが広がりを見せている手法は, 切り抜きの結果が安定しない, すなわち, 精度の良い切り抜きが行われないことが多くなるという傾向を読み取ることができる。図 10 (入力 1 を利用した場合), 図 11 (入力 2 を利用した場合) はそれぞれ 50 枚の画像のそれぞれの切り抜き結果の適合率と再現率の分布図である。図 11 における横軸は適合率  $p$ , 縦軸は再現率  $r$  である。このようにばらつき程度, 安定性などを可視化した分布図と表 1, 表 2 を用いて, 各手法に関する考察を以降において行う。

図 10 (a), (b), 図 11 (a), (b) から, 本手法 (Hybrid Queue) と本手法 (Priority Queue)

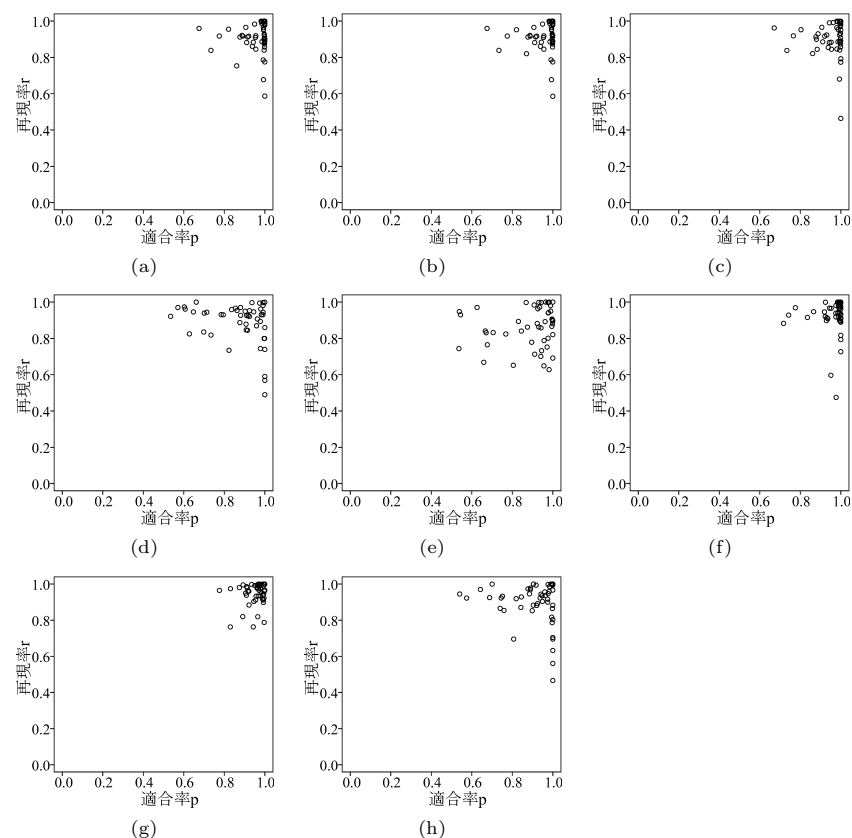


図 10 入力 1 に対する実験結果。(a): 本手法 (Hybrid Queue). (b): 本手法 (Priority Queue). (c): 本手法 (飛び地対応なし). (d): 本手法 (テクスチャ対応なし). (e): Seeded Region Growing. (f): Lazy Snapping. (g): SIOX. (h): Random Walks

Fig. 10 Result of Input method 1. (a): our method (Hybrid Queue). (b): our method (Priority Queue). (c): our method (without dealing with enclave). (d): our method (without dealing with texture). (e): Seeded Region Growing. (f): Lazy Snapping. (g): SIOX. (h): Random Walks.

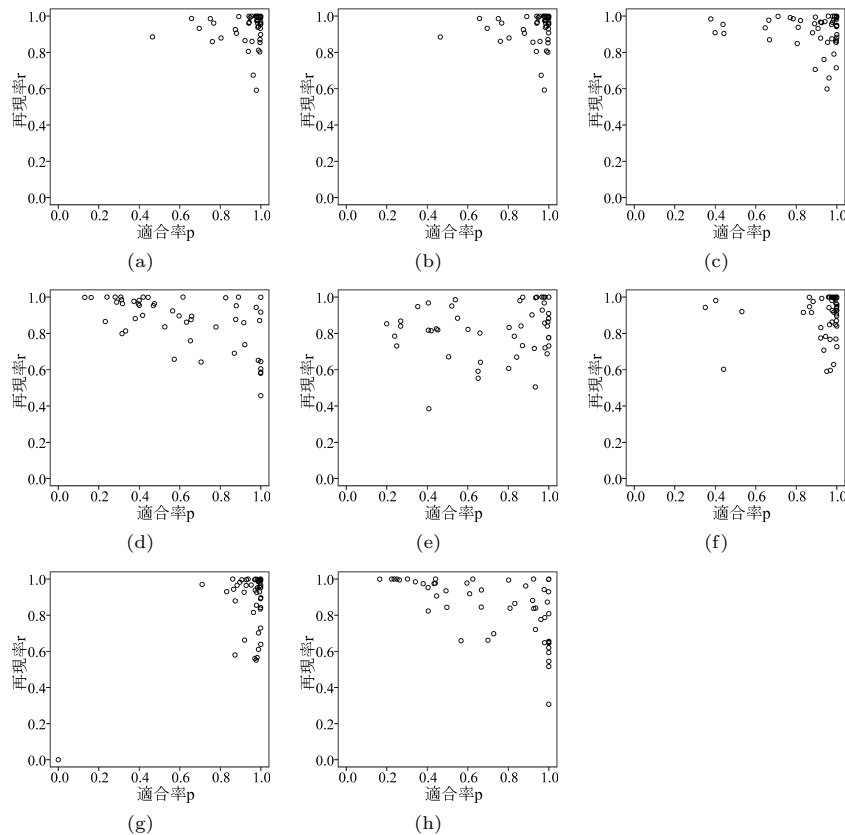


図 11 入力 2 に対する実験結果 . (a) : 本手法 (Hybrid Queue) . (b) : 本手法 (Priority Queue) . (c) : 本手法 (飛び地対応なし) . (d) : 本手法 (テクスチャ対応なし) . (e) : Seeded Region Growing . (f) : Lazy Snapping . (g) : SIOX . (h) : Random Walks

Fig. 11 Result of Input method 2. (a): our method (Hybrid Queue). (b): our method (Priority Queue). (c): our method (without dealing with texture). (d): our method (without dealing with texture). (e): Seeded Region Growing. (f): Lazy Snapping. (g): SIOX. (h): Random Walks.

の精度の分布に大きな変化は見られなかった．本手法 (Hybrid Queue) と本手法 (Priority Queue) の切り抜き結果の差異は，前景領域と背景領域の境界線付近に見られた．これは過去のコスト  $C_s$  を蓄積する領域が要因であると考えられる．Priority Queue と Queue の併用をする場合と Priority Queue のみを用いる場合との切り抜き結果の差異は，過去のコスト  $C_s$  を蓄積する領域に現れる．式 (14) に表される  $W_2(s, t)$  の定義から，コストの蓄積を行う領域は，伝播候補  $t$  の色が領域  $R(s)$  の代表色に属さない領域，すなわち前景領域と背景領域の境界線付近である．その結果，切り抜き結果に差異が生じた領域は前景領域と背景領域の境界線付近になったと考えられる．

表 1, 表 2 において入力 1, 入力 2 に対する切り抜き結果を比較すると，既存手法では入力 1 の方が入力 2 よりも平均 F 尺度が大きいたことが確認できる．

さらに，分布を示す図 10 (e) ~ (h) と図 11 (e) ~ (h) を比較すると，図 11 (e) ~ (h) の方がばらつきが大きく，安定した切り抜きが行われていないことが分かる．これは入力 1 が入力 2 と比較して未知領域の面積が小さい Trimap であることから，それぞれの画像に対して一定の適合率および再現率が保証され，多くの画像に対して安定した切り抜きができるためである．

一方，本手法 (Hybrid Queue) および本手法 (Priority Queue) では平均 F 尺度の大きな差は見られなかったが，図 10 (a), (b), 図 11 (a), (b) を見ると，入力 1 の結果が入力 2 の結果よりも大きく適合率が低くなる画像が存在している．この入力 1 の結果が入力 2 の結果よりも大きく適合率が低い例を図 12 に示す．これは入力 1 が入力 2 と比較して初期背景領域の面積が大きすぎることが要因である．本手法はテクスチャパターンへの対応のために，コスト  $C(t)$  の計算において式 (13) に表される代表色を用いた重み付けを行う．代表色は初期前景領域，初期背景領域により決定されるため，初期前景領域および初期背景領域の面積が大きいほどこれらの領域に共通して含まれる色は多くなる傾向がある．すなわち，前景のテクスチャパターンで用いられる色が背景の代表色にも属する可能性が高くなり，その結果， $W_1(s, t)$  の定義から前景領域，背景領域に対して中立であると判別され，前景領域のテクスチャパターンに対して適切な重み付けがされず，前景領域が取得できなくなる．これらのことから，本手法は，手書き線のような初期前景領域，初期背景領域の面積が小さい入力に対して有効な手法といえる．

表 2 において，本手法 (飛び地対応なし) の平均適合率は本手法 (Hybrid Queue) の平均適合率に比較し低いことが分かる．そして図 11 (c) における本手法 (飛び地対応なし) の切り抜き結果から，図 11 (a) における本手法 (Hybrid Queue) と比較し，適合率が低下して

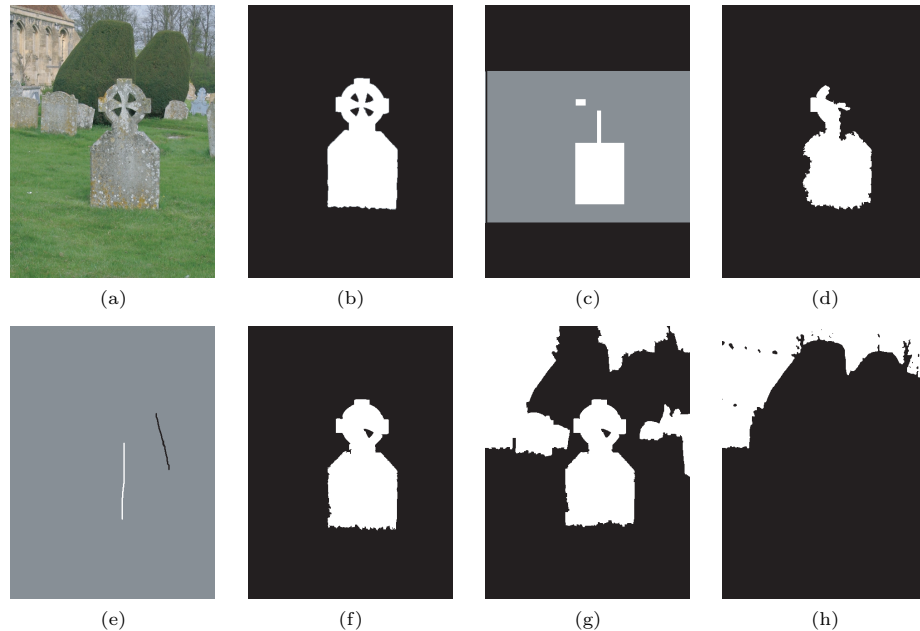


図 12 (a) : 対象画像 . (b) : 正解画像 . (c) : 入力 1 . (d) : 入力 1 における本手法 (Hybrid Queue) の切り抜き結果 . (e) : 入力 2 . (f) : 入力 2 における本手法 (Hybrid Queue) の切り抜き結果 . (g) : Lazy Snapping . (h) : Sixx  
 Fig. 12 (a) : Target image. (b) : Correct image. (c) : Input method 1. (d) : Result of Input method 1. (e) : Input method 2. (f) : Result of Input method 2. (g) : Result by Lazy Snapping. (h) : Result by Sixx.

いる画像が複数枚存在することが分かる . この例を図 13 に示す . 図 13(d) は図 13(a) を対象画像 , 図 13(c) を入力としたときの本手法 (飛び地対応なし) の切り抜き結果である . 図 13(d) から , 背景領域の飛び地が前景領域と判別されてしまい , 飛び地がとれていないことが分かる . 一方 , 本手法 (Hybrid Queue) が採用した重み係数パラメータ  $w = 1.5$  における切り抜き結果である図 13(h) では飛び地をとることができており , 本手法 (Hybrid Queue) の適合率が本手法 (飛び地対応なし) の適合率より高いことが分かる . これらのことから , 背景領域の飛び地を含む画像に対して , 本手法における飛び地への対応は有効であるといえる . なお , 表 1 において , 本手法 (飛び地対応なし) の平均適合率と本手法 (Hybrid Queue) の平均適合率との差が少なく , 図 10(c) と図 11(c) のばらつきの程度にあまり差が

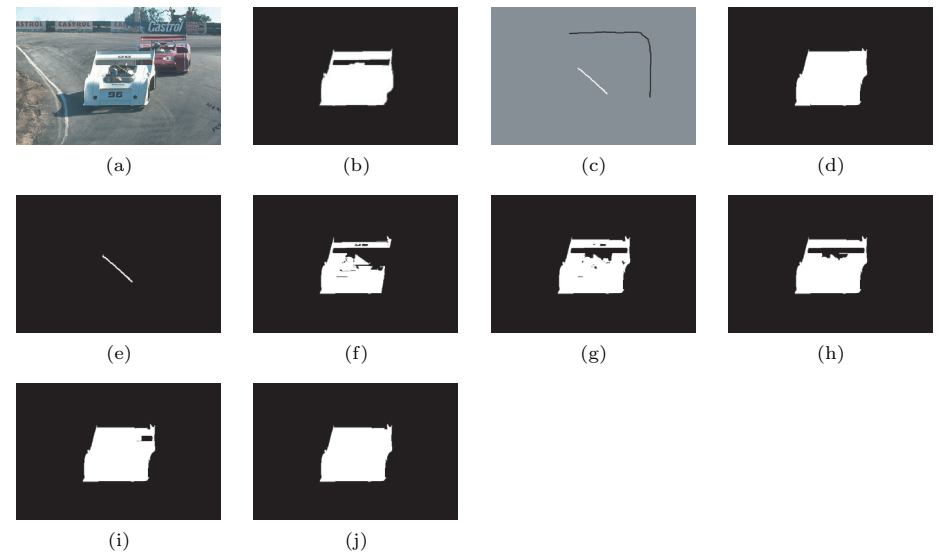


図 13 本手法 (Hybrid Queue) における飛び地対応の例 . (a) : 対象画像 . (b) : 正解画像 . (c) : Trimap . (d) : 本手法 (飛び地対応なし) . (e) : 本手法 (Hybrid Queue)  $w = 0$  . (f) : 本手法 (Hybrid Queue)  $w = 0.5$  . (g) : 本手法 (Hybrid Queue)  $w = 1.0$  . (h) : 本手法 (Hybrid Queue)  $w = 1.5$  . (i) : 本手法 (Hybrid Queue)  $w = 2.0$  . (j) : 本手法 (Hybrid Queue)  $w = 3.0$   
 Fig. 13 Exapmle of dealing with enclave. (a) : Target image. (b) : Corecct image. (c) : Trimap. (d) : our method (without dealing with enclave). (e) : our method (Hybrid Queue)  $w = 0$  . (f) : our method(Hybrid Queue)  $w = 0.5$  . (g) : our method (Hybrid Queue)  $w = 1.0$  . (h) : our method (Hybrid Queue)  $w = 1.5$  . (i) : our method (Hybrid Queue)  $w = 2.0$  . (j) : our method (Hybrid Queue)  $w = 3.0$  .

ないのは , 実験に使用した 50 枚の画像データセットに含まれる飛び地の面積が前景領域の面積と比較すると十分小さいため , 入力 1 のような Trimap の場合 , 飛び地の切り抜きに成功しても失敗しても , それが適合率の数値に反映されにくいためである .

表 1 , 表 2 より , 本手法 (テキスト対応なし) の精度 (平均適合率 , 平均再現率 , 平均 F 尺度) は本手法 (Hybrid Queue) の精度 (平均適合率 , 平均再現率 , 平均 F 尺度) に比較して低いことが分かる . なおかつ , 図 10(d) , 図 11(d) における本手法 (テキスト対応なし) の切り抜き結果は , 図 10(a) , 図 11(a) における本手法 (Hybrid Queue) と比べてばらつきが大きく , 精度の低い切り抜き結果が多いことが分かる . 図 14 にその 1 つの例を示す . 図 14(d) は図 14(a) を対象画像 , 図 14(c) を入力としたときの本手法 (テキストチャ対

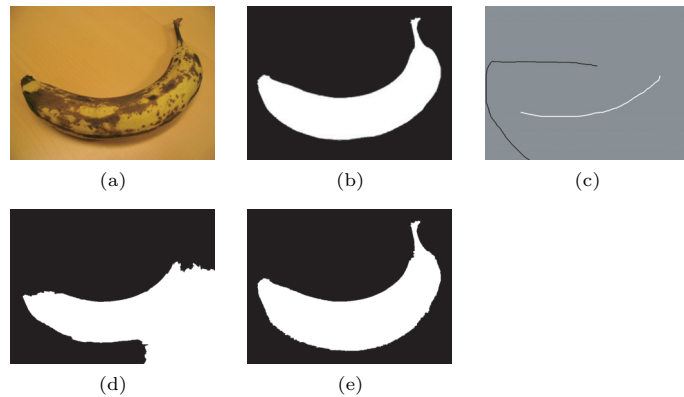


図 14 本手法 (Hybrid Queue) におけるテクスチャへの対応の例 . (a) : 対象画像 . (b) : 正解画像 . (c) : Trimap . (d) : 本手法 (テクスチャ対応なし) . (e) : 本手法 (Hybrid Queue)

Fig. 14 Example of dealing with texture. (a): Target image. (b): Correct image. (c): Trimap. (d): our method (without dealing with texture). (e): our method (Hybrid Queue).

応なし)の切り抜き結果である。図 14 (d) から、前景領域が背景領域と判別され、背景領域が前景領域と判別されている箇所があることが分かる。この精度低下の要因は、前景領域はバナナのテクスチャ領域であるため、前景領域では黄色と黒がまばらに配置されていることから、黄色と黒の境界で前景領域の伝播が行われず、さらに前景領域の黄色と背景領域の木の色差が小さいことから、前景領域に背景領域の伝播が行われ、背景領域に前景領域の伝播が行われてしまうためである。一方、本手法 (Hybrid Queue) の切り抜き結果である図 14 (e) から、テクスチャパターンの対応を行うことで、精度良く切り抜きが行われていることが分かる。これらのことから、本手法におけるテクスチャパターンへの対応は有効であるといえる。

表 1、表 2 から Seeded Region Growing より本手法のほうが平均切り抜き精度が高いことが分かる。また、図 10 (a), (b), (e) と図 11 (a), (b), (e) から、本手法より Seeded Region Growing のほうが精度のばらつきが大きく、精度の低い切り抜き結果が多いことが分かる。この例を図 15 に示す。

図 15 (d) は図 15 (a) を対象画像、図 15 (c) を入力としたときの Seeded Region Growing の切り抜き結果である。図 15 (a) から Seeded Region Growing の切り抜き結果は、初期前景領域と隣接したピクセルが背景領域となってしまうことが分かる。この精度低下の

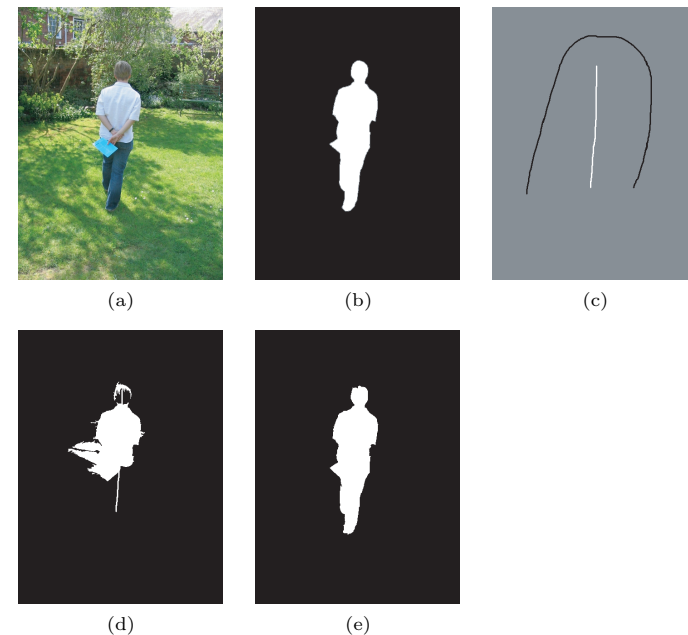


図 15 Seeded Region Growing の切り抜き失敗例 . (a) : 対象画像 . (b) : 正解画像 . (c) : Trimap . (d) : Seeded Region Growing . (e) : 本手法 (Hybrid Queue)

Fig. 15 Example of failed Seeded Region Growing. (a): Target image. (b): Correct image. (c): Trimap. (d): Seeded Region Growing. (e): our method (Hybrid Queue).

要因は、Seeded Region Growing が単色の領域を抽出するためのコストを定義しているためである。Seeded Region Growing のコスト  $C'(t)$  は以下の式で定義される。

$$C'(t) = D(\text{mean}(R(s)), t) \quad (20)$$

ここで  $\text{mean}(R(s))$  は領域  $R(s)$  の平均色である。コストに領域の平均色を用いていることから、初期前景領域が複数の色を持つ場合は隣接するノードどうしの色差が小さくても同一の領域と見なされない。一方、本手法では隣接するノードどうしの色差をコストに用い、さらにピクセルに基づくセグメンテーションを用いて重み付けを行っているためこのような精度低下は起こらない (図 15 (e))。

表 1、表 2 より、本手法と比べると Lazy Snapping においては入力 1 より入力 2 のほうが精度が低下する傾向がある。また、精度の分布において図 11 (f) の Lazy Snapping の切

り抜き結果では、図 10 (f) と比べて適合率が著しく低下している画像がいくつか存在することが分かる。この例を図 12 (g) に示す。図 12 (g) は図 12 (a) を対象画像、図 12 (e) を入力としたときの Lazy Snapping の切り抜き結果である。図 12 (g) から Lazy Snapping の切り抜き結果は、初期前景領域と地続きでない飛び地の領域が前景領域となっていることが分かる。この精度低下の要因は、正解の前景領域と同じような色合いを持つ領域が飛び地で存在した場合に前景領域と見なしてしまうことにある。一方、本手法では前景領域と同じような色合いを持つ領域が飛び地で存在する場合には、前景領域としていないためこのような精度低下は起こらない (図 12 (f))。

表 1, 表 2 より、本手法と比べると SIOX においては入力 1 より入力 2 のほうが精度が低下する傾向があることが分かる。また、精度の分布において図 11 (g) の SIOX の切り抜き結果では、適合率、再現率がともに 0 の画像が存在している。この例を図 12 (h) に示す。図 12 (h) は図 12 (a) を対象画像、図 12 (e) を入力としたときの SIOX の切り抜き結果である。図 12 (h) から SIOX の切り抜き結果は、初期前景領域を含まない領域が前景領域と誤判別されていることが分かる。この精度低下の要因は、SIOX がピクセルに基づくセグメンテーション手法のみを用いているためであると考えられる。SIOX は初期前景領域、初期背景領域から Color Signature を用いたクラスタを構築し、このクラスタから各ピクセルが前景か背景かを決定する。すなわち SIOX では、構築されるクラスタによって前景領域、背景領域が決定されるため、必ずしも初期前景領域、初期背景領域はそれぞれ前景領域、背景領域とはならない。一方、本手法では初期前景領域、初期背景領域はそれぞれ前景領域、背景領域として確定するため、ユーザによって指示された初期前景領域、初期背景領域はそれぞれ前景領域、背景領域となる (図 12 (f))。

表 2 より、入力 2 における Random Walks の精度 (平均適合率, 平均再現率, 平均 F 尺度) は本手法 (Hybrid Queue) の精度 (平均適合率, 平均再現率, 平均 F 尺度) に比較し低いことが分かる。

精度の分布において図 11 (h) における Random Walks の切り抜き結果では、図 10 (h) と比べてばらつきが大きく、入力 2 においては精度の低い切り抜き結果が多いことが分かる。この例を図 16 に示す。図 16 (d) は図 16 (a) を対象画像、図 16 (c) を入力としたときの Random Walks の切り抜き結果である。図 16 (d) から Random Walks の切り抜き結果は、初期背景領域より初期前景領域に近い背景領域が前景領域と誤判別されていることが分かる。この精度低下の要因は、Random Walks が初期前景領域、初期背景領域の位置関係によって結果に影響を与える手法であるためと考えられる。Random Walks は図 3 で示さ

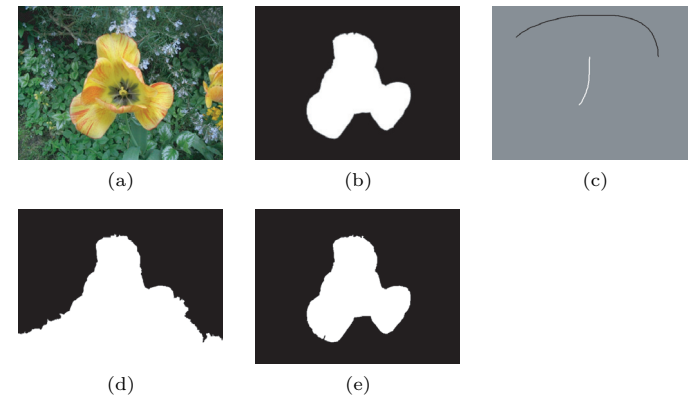


図 16 Random Walks の切り抜き失敗例。(a): 対象画像。(b): 正解画像。(c): Trimap。(d): Random Walks。(e): 本手法 (Hybrid Queue)

Fig. 16 Example of failed Random Walks. (a): Target image. (b): Correct image. (c): Trimap. (d): Random Walks. (e): our method (Hybrid Queue).

れるようなグラフにおいて、初期前景領域に属するノードが注目するノードへとたどり着く確率を推定する手法である。そのため初期前景領域に近いノードほど前景領域となる確率が高く、初期背景領域に近いノードほど前景領域となる確率が低くなり、背景となる領域でも初期前景領域に近い場合は前景と誤判別されてしまう。一方、本手法では類似した色が連続している領域において、初期前景領域、初期背景領域の位置が結果に影響を与えることはない (図 16 (e))。

以上、精度に関して本手法は、手書き線 (入力 2) のような初期前景領域、初期背景領域の面積が小さい入力に対して優位な手法といえる。

## 6. 速度評価実験

本手法を実装した画像切り抜きシステム「切絵」と既存手法 (Seeded Region Growing, Lazy Snapping, SIOX, Random Walks) に関して、図 17 に表される 3 つの画像の切り抜き時間を測定し、比較評価した。以降、実験方法とその結果について述べる。

### 6.1 実験方法

図 17 はそれぞれ画像サイズが  $2048 \times 2048$  ピクセルの画像である。ピクセル数  $N$  に対しての時間計算量を評価するためにそれぞれの画像サイズに対して  $1024 \times 1024$  ピクセル、



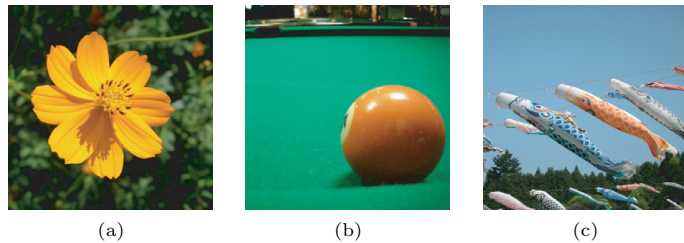


図 17 (a): 対象画像 1. (b): 対象画像 2. (c): 対象画像 3  
 Fig. 17 (a): Target image 1. (b): Target image 2. (c): Target image 3.

512 × 512 ピクセル, 256 × 256 ピクセル, 128 × 128 ピクセルに縮小した画像に対しても切り抜き時間を測定した. 画像の縮小には Cubic 法<sup>17)</sup> を用いた. 入力に精度評価実験における入力 2 と同様に, 対象画像に対して前景と背景の手書き線を 1 本ずつ入力した場合の Trimap を用いる. 画像の切り抜き時間の測定は, それぞれの対象画像に対して切り抜きを連続 10 回行った時間の計測を 1 セットとし, 計測を 10 セット行った. 速度評価実験では本手法 (Hybrid Queue) の実現手法に Radix Heap を用いた.

評価に用いた環境は Dell XPS 730x, CPU: Core i7-940 2.93 GHz で, 本手法, Seeded Region Growing, Lazy Snapping, SIOX は Microsoft Visual C++ .NET 2003 を用いてビルドを行った. また, Random Walks は MATLAB のソースコード<sup>\*1</sup> を利用したため, MATLAB 7.6.0 を用いて実行した.

### 6.2 実験結果

実験結果を図 18, 図 19, 図 20 に示す. 図は横軸が画像サイズ (ピクセル数), 縦軸が処理時間を表す両対数グラフである. 処理時間は, それぞれの画像サイズにおける 1 回の切り抜きに要する平均時間とした.

図 18, 図 19, 図 20 から本手法 (Hybrid Queue) は本手法 (飛び地対応なし), 本手法 (テクスチャ対応なし) と処理時間に関する有意な差は見られなかった. よって, 飛び地への対応, テクスチャへの対応が時間オーバーヘッドとなることはない.

また, 図 18, 図 19, 図 20 から, 本手法 (Hybrid Queue) は既存手法の中で最も高速である Seeded Region Growing とほぼ同等の速度性能を持つことが確認できる.

図 18, 図 19, 図 20 から, Lazy Snapping と SIOX は本手法に比べ, 画像によって処理

\*1 <http://cns-web.bu.edu/~lgrady/>

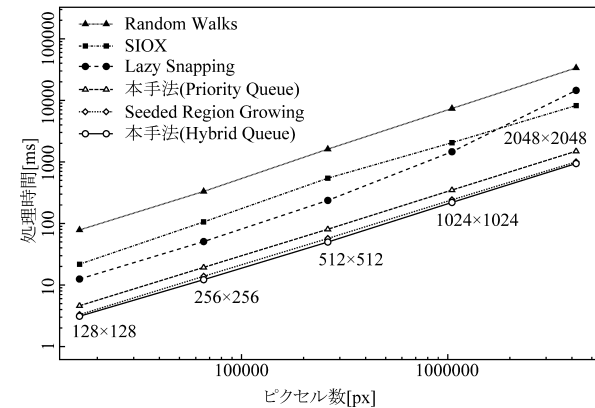
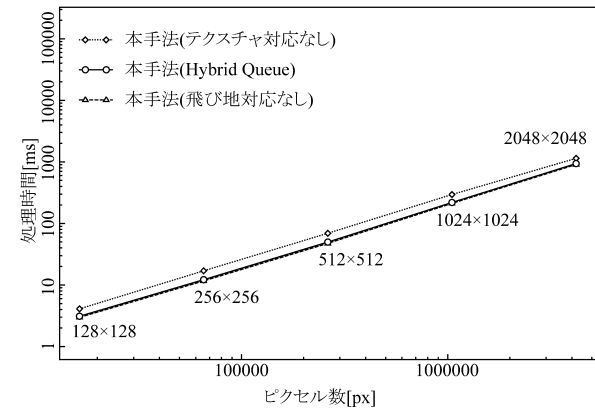


図 18 対象画像 1 の実験結果

Fig. 18 Performance comparison using Target Image 1.

時間にばらつきが生じていることが分かる. これはピクセルに基づくセグメンテーション手法において, 構築するクラスタの大きさが画像によって変化することが処理時間に影響を与えていると考えられる. 一方, 本手法ではピクセルに基づくセグメンテーション手法において, ルックアップテーブルを用いることで時間計算量が  $O(1)$  の判別をしているため処理時間に大きなばらつきは生じていない.

また本手法 (Hybrid Queue) は本手法 (Priority Queue) に対して約 1.5 倍の速度向上が



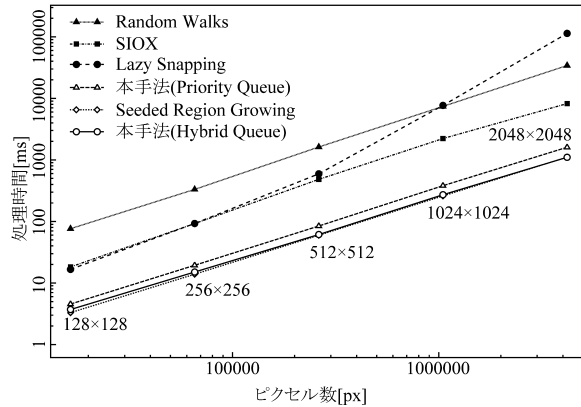
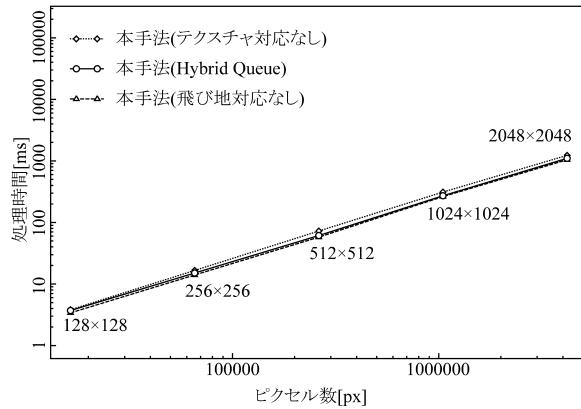


図 19 対象画像 2 の実験結果

Fig. 19 Performance comparison using Target Image 2.

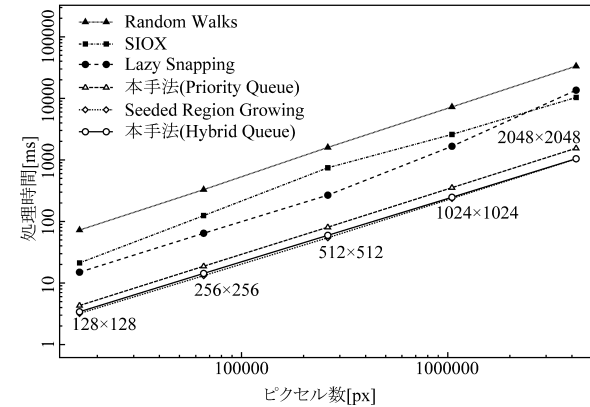
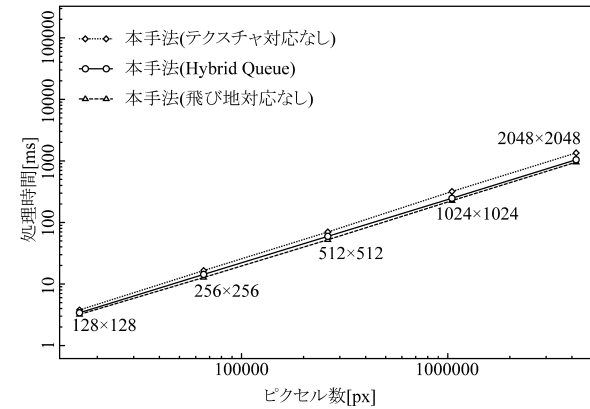


図 20 対象画像 3 の実験結果

Fig. 20 Performance comparison using Target Image 3.

見られた。これは本手法 (Priority Queue) がすべてのノードを Priority Queue に格納する手法であることに対して、本手法 (Hybrid Queue) では一部のノードを Queue に格納することで、定数時間のオーバーヘッドが軽減されたためである。

さらに図 18, 図 19, 図 20 において本手法 (Hybrid Queue) の勾配が本手法 (Priority Queue) よりも小さいことから、時間計算量も向上していることを確認した。これは本手法 (Hybrid Queue) が本手法 (Priority Queue) よりも最大時間計算量の小さい Radix Heap

と Queue を用いたためである。

## 7. おわりに

本論文では Seeded Region Growing を改良した高速画像切り抜き手法を提案し、それを実装したシステム「切絵」を開発し、既存の画像切り抜き手法 (Seeded Region Growing, Lazy Snapping, SIOX, Random Walks) と精度, 処理速度に関する比較実験を行い、評

価した。

Seeded Region Growing には、飛び地の領域伝播ができないことと、テクスチャパターンのように色が急速に変化する領域に対応できないという欠点が存在する。そこで本手法では、飛び地への対応のために、伝播コストを蓄積させ、伝播コストの蓄積がある閾値を超えた場合に前景を背景に反転させるという処理を導入した。また、テクスチャパターンへの対応のために、ピクセルに基づくセグメンテーション処理を導入した。

さらに本手法は、高速化のために新たに Priority Queue と Queue を併用するデータ構造を設計、導入し、ピクセル数  $N$  に対して最大時間計算量が  $O(N)$  である画像切り抜きを可能とした。

精度評価実験により、手書き線のように、ユーザが指定する初期前景領域、初期背景領域の面積が小さい入力の場合において、本手法が既存の画像切り抜き手法 (Seeded Region Growing, Lazy Snapping, SIOX, Random Walks) に比べ、より正確な切り抜きが可能であることを確認した。

さらに本手法で新たに導入した飛び地への対応処理とテクスチャパターンへの対応処理の効果を確認するために、これらの処理を導入した場合と導入しなかった場合の精度比較、処理時間比較を行った。その結果、これらの対応処理により、時間オーバーヘッドなしに、飛び地やテクスチャパターンが適切に処理され、より正確な切り抜きが可能となることを確認した。

また速度評価実験を行い、本手法が既存手法で最も高速である Seeded Region Growing とほぼ同等の処理速度性能を示すことを確認した。

今後は、実応用に向けて、多種多様な画像を対象に、さらなる精度向上、処理速度向上を図りたいと考えている。

謝辞 本研究は楽天技術研究所の支援を受けた。記して深謝する。

## 参 考 文 献

- 1) Wang, J. and Cohen, M.F.: *Image and Video Matting*, Now Publishers Inc., Hanover, MA, USA (2008).
- 2) Mortensen, E.N. and Barrett, W.A.: Interactive Segmentation with Intelligent Scissors, *Graphical Models and Image Processing*, Vol.60, No.5, pp.349–384 (1998).
- 3) Gleicher, M.: Image snapping, *SIGGRAPH '95: Proc. 22nd Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, ACM, pp.183–190 (1995).
- 4) Li, Y., Sun, J., Tang, C.-K. and Shum, H.-Y.: Lazy snapping, *ACM Trans. Graph.*, Vol.23, No.3, pp.303–308 (2004).
- 5) Grady, L.: Random Walks for Image Segmentation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.28, No.11, pp.1768–1783 (2006).
- 6) Friedland, G., Jantz, K. and Rojas, R.: SIOX: Simple Interactive Object Extraction in Still Images, *ISM '05: Proc. 7th IEEE International Symposium on Multimedia*, Washington, DC, USA, IEEE Computer Society, pp.253–260 (2005).
- 7) Adams, R. and Bischof, L.: Seeded Region Growing, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.16, No.6, pp.641–647 (1994).
- 8) Rother, C., Kolmogorov, V. and Blake, A.: “GrabCut”: interactive foreground extraction using iterated graph cuts, *ACM Trans. Graph.*, Vol.23, No.3, pp.309–314 (2004).
- 9) Ruzon, M.A. and Tomasi, C.: Alpha Estimation in Natural Images, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol.1, p.1018 (2000).
- 10) Chuang, Y.-Y., Curless, B., Salesin, D.H. and Szeliski, R.: A Bayesian Approach to Digital Matting, *Proc. IEEE CVPR 2001*, Vol.2, IEEE Computer Society, pp.264–271 (2001).
- 11) Sun, J., Jia, J., Tang, C.-K. and Shum, H.-Y.: Poisson matting, *ACM Trans. Graph.*, Vol.23, No.3 (2004).
- 12) Wang, J. and Cohen, M.F.: Optimized Color Sampling for Robust Matting, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol.0, pp.1–8 (2007).
- 13) Levin, A., Rav-Acha, A. and Lischinski, D.: Spectral Matting, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.30, No.10, pp.1699–1712 (2008).
- 14) Sindeyev, M. and Konushin, V.: A Novel Interactive Image Matting Framework, *Proc. Graphicon 2008*, pp.41–44 (2008).
- 15) International Telecommunications Union: Recommendation ITU-R BT.601, Encoding Parameters of Digital Television for Studios, Geneva (1992).
- 16) Ahuja, R.K., Mehlhorn, K., Orlin, J. and Tarjan, R.E.: Faster algorithms for the shortest path problem, *J. ACM*, Vol.37, No.2, pp.213–223 (1990).
- 17) Keys, R.: Cubic convolution interpolation for digital image processing, *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol.29, No.6, pp.1153–1160 (1981).

(平成 21 年 3 月 27 日受付)

(平成 21 年 9 月 11 日採録)



清野 達也

1984年生．2007年電気通信大学電気通信学部情報工学科卒業．2009年同大学大学院電気通信学研究科情報工学専攻博士前期課程修了．同年グーグル株式会社入社，現在に至る．在学中は画像処理の研究に従事．



林 貴宏（正会員）

1975年生．1998年金沢大学工学部電気・情報工学科卒業．2000年同大学大学院自然科学研究科博士前期課程電子情報システム専攻修了．2003年同研究科博士後期課程数理情報科学専攻修了．博士（工学）．2001年石川工業高等専門学校電子情報工学科助手，2003年電気通信大学電気通信学部情報工学科助手，2007年同学科助教を経て，2009年新潟大学工学部情報工学科准教授，現在に至る．マルチメディア情報検索，インタラクティブシステムの研究に従事．IEEE，電子情報通信学会，日本ソフトウェア科学会，人工知能学会各会員．



尾内理紀夫（正会員）

1950年生．1973年東京大学理学部物理学科卒業．1975年同大学大学院理学系研究科物理学専攻修士課程修了．同年日本電信電話公社（現NTT）に入社．1982年から1985年にICOTプロジェクトに参画，1997年から1998年にRWCプロジェクトに参画．2000年より電気通信大学電気通信学部情報工学科教授．著書に『マルチメディアコンピューティング』（コロナ社），『コンピュータの仕組み』（朝倉書店），編書に『オブジェクト指向コンピューティング』（近代科学社）『インタラクティブシステムとソフトウェア』（近代科学社）等．マルチメディア情報処理，情報検索，セマンティックコンピューティング等に興味を持つ．工学博士（東京大学）．日本ソフトウェア科学会，人工知能学会，ACM各会員．



三條 正裕

1978年生．2003年東京大学工学部卒業．アクセンチュア・テクノロジー・ソリューションズ株式会社を経て，2007年に楽天株式会社に入社．現在，同社楽天技術研究所シニア・テクノロジスト．主にインターネットサービスにおける画像解析技術の研究に従事．



森 正弥（正会員）

1975年生．1998年慶應義塾大学経済学部卒業．アクセンチュア株式会社を経て，2006年に楽天株式会社入社．現在，同社執行役員楽天技術研究所所長．主にインターネットサービスにおける画像解析技術や分散技術の研究に従事．国際高等研究所2008年度・2009年度研究プロジェクト「次世代情報サーチに関する総合的研究」に関与．情報処理学会DBS研究会運営委員．電子情報通信学会データ工学研究専門委員会専門委員．IPA Ruby標準化検討WG委員．総務省スマートクラウド研究会技術WG構成員．Rubyアソシエーション運営委員．著書に『クラウド大全』（日経BP社，共著）．