

製版用切り抜き WS (CUTART) の開発

— 自動切り抜きシステムの開発 —
Development of Cutout System

木之下 洋* 奥沢 二郎* 米山 努*
Kinoshita, Hiroshi Okuzawa, Jiro Yoneyama, Tsutomu

In the electronic prepress system, automatic cutout system has been required for increasing productivity. Several cutout system have produced. However, in the practical use, little has been used an effective automatic cutout algorithm. A lot of algorithm have needed human assistance. This report describes the new valid cutout algorithm based on a clustering method and the new human interface instructed to extract a contour automatically.

1 はじめに

電子製版において、画像切り抜きシステムが数多く上市されてきた。画像切り抜きシステムには、作業の効率化を図るため自動切り抜き機能が搭載されている。特に濃度差、統計的な判別関数を用いることは広く知られている。

しかし、汎用に利用できる自動輪郭抽出アルゴリズムはいまだ実現されておらず、オペレータに局所領域を指定させ閾値を対話的に調整することによって輪郭を抽出するロジックが広く利用されている¹⁾

本報告は、切り抜きシステムを開発するにあたり、従来の局所的に処理をする対話的なオペレーションのなかで、オペレータの閾値調整の減少、および自動切り抜き精度を向上させることが可能になったのでその実例を示す。

2 輪郭抽出の概要

本システムの基本スキームを Fig. 1 及び Fig. 2 で示す。

2.1 切り抜き領域のサンプリング

輪郭抽出範囲の設定は Fig. 1 に示す 2 点の指定で行う。

自動抽出の第 1 ポイントを切り抜き対象領域内の境界付近に位置するように、第 2 ポイントを背景領域に第 2 サンプル領域が位置するように設定する。

第 1 ポイントを指定した位置で固定し、この状態で輪郭を抽出する範囲を明示する。

第 2 ポイントは所望の切り抜き部位を含むようにオペレータが設定する。輪郭抽出範囲は第 1、第 2 ポイントを中心とする 2 円 (サンプル領域) を結ぶ領域として定義する。

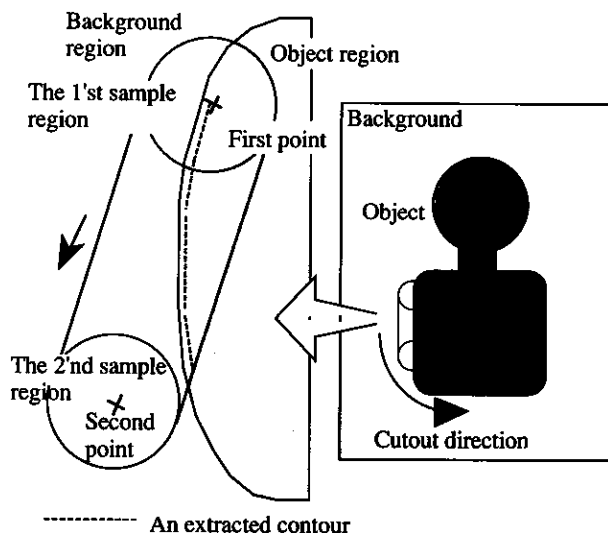


Fig.1 Operation of extracting contour

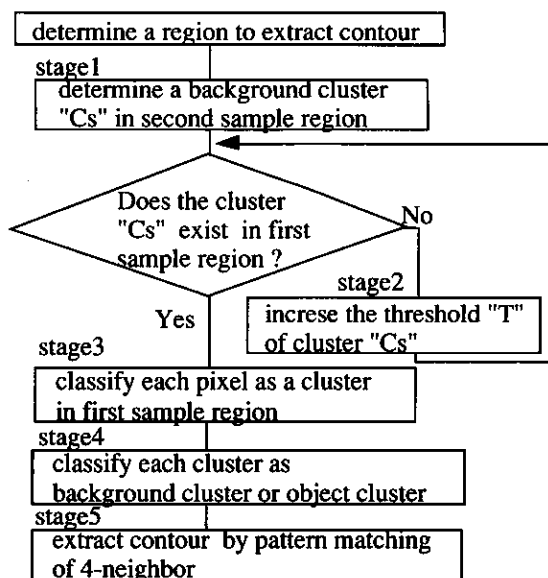


Fig.2 The scheme of extracting contour algorithm

* 画像システム機器事業部 開発部

2.2 輪郭抽出の手法

輪郭抽出の処理手順を以下に示す。

処理1)

第2サンプル領域内の画素 $P_{k,s}$ ($k = [0, n]$, $s = \text{RGB} \mid \text{CMYK}$) の色情報 (RGB | CMYK) の平均値を $\overline{P_s}$ とする (式1) 次に $\overline{P_s}$ から全画素の色情報に対するユークリッド距離を求め、最大となる距離を処理2で行うクラスタリングの初期閾値 T_0 とする (式2、3)。ここでクラスタの重心に $\overline{P_s}$ を持ち、閾値 T_0 の距離をもつクラスタ C_s を作成する。 C_s を背景領域の初期クラスタとする。

$$\overline{P_s} = \frac{1}{n} \sum_{k=1}^n P_{k,s} \quad (1)$$

$$T_i = \left\{ \left\{ \sum_{k=1}^n (\overline{P_k} - P_{k,i})^2 \right\}^{1/2} \right\} \quad (2)$$

$$T_0 = \max_{1 \leq i \leq n} T_i \quad (3)$$

* i は第2サンプル領域内画素

処理2)

クラスタ C_s に属する画素を第1サンプル領域内で抽出し、閾値 T_0 によってその画素を重心とするクラスタとして新規に作成する。このクラスタを背景領域クラスタとする。 C_s に属する画素が見つからない場合は、閾値 T_0 を少なくとも1つの画素が第1サンプル領域内で抽出するまで C_s の閾値を増加する。 C_s に属する画素が抽出できた時点で、 C_s の閾値は T_s とする (式4)。また、抽出された画素を重心とする初期閾値 T_0 を有するクラスタを作成し、背景領域クラスタとする。

$$T_s = \min_{1 \leq i \leq n} T_i \quad (4)$$

* i は第1サンプル領域内画素

処理3)

第1サンプル領域の第1ポイントを切り抜き対象領域のオブジェクトクラスタの重心と仮定し、各画素の色情報とのユークリッド距離を測定し閾値 T_0 内の画素をクラスタ C_f とする。これを切り抜き対象領域の初期クラスタとする。第1サンプル領域において C_s 、 C_f のどちらにも属さない画素が抽出された場合はその画素を重心とし閾値 T_0 で新しいクラスタを作成する。クラスタは当初2クラスタ (背景領域クラスタ、対象領域クラスタ) として $P_{k,s}$ を分類するが、評価基準の大きさに応じて、クラスタの個数を増加させながら、クラスタリングを実行させる。このとき評価基準がクラスタ数を増加させても変化しなければ新クラスタの生成を停止する。

処理4)

処理3においてクラスタ C_s 、 C_f 以外に新しいクラスタが作成された場合は、これらを背景領域に属すクラスタと対象領域に属すクラスタに統合する。この分離方法はクラスタの幾何学的な配置を利用して行う。クラスタの配置のパターンの1例を Fig. 5 に示す。

処理5)

全クラスタを背景領域と対象領域のいずれかに分類した後、4連結の輪郭線を求めるアルゴリズム²⁾を利用して進行方向の2画素のパターンから Fig. 6 の法則に従い輪郭を追跡する。

2.3 各処理における効果

マンマシン設計の工夫により、背景と切り抜き対象領域のサンプリングを同時指定することを可能にした。また、第2サンプリング領域において背景領域に影のような不要なサンプル色が存在する場合はクラスタリングの精度に影響を与えるので、オペレータが容易に影を回避するように第2サンプル領域を再設定し直すことができる (Fig. 3)。

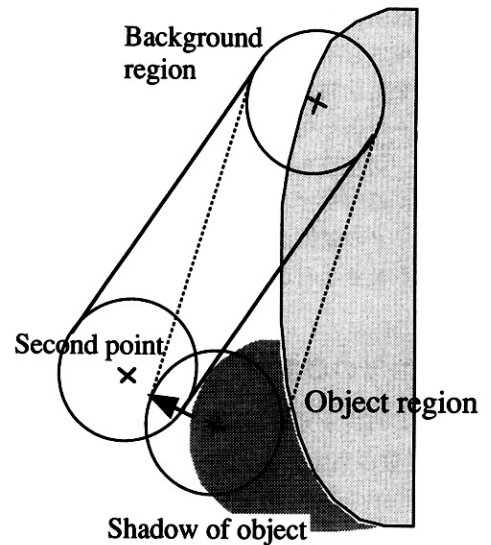


Fig. 3 Operation to avoid shadow

処理2における効果としては、第2サンプル領域から第1サンプル領域にかけて背景領域にグラデーションがかかっている場合に有効となる。

処理3が実行された後、第1サンプル領域において、3つ以上のクラスタが発生した場合は、これらを背景領域と対象領域のどちらに属するかを決定する必要がある。このような複数クラスタを持つのは、複雑な背景画像や対象物を持つ場合や、Fig. 4のような対象領域の境界付近に影やエッジ強調が存在する場合に起こりうる。

従来の濃度差切り抜きにおいては、濃度のプロファイルが 1) 背景領域、2) 切り抜き対象領域、3) 背景と対象領域の間のエッジ部分の順になっている場合、エッジ部

分を背景領域に含まれるように濃度閾値を上げていくと対象領域まで含まれ適切に領域を分離するのは困難であった。そこで、本手法の処理4においてクラスタの幾何学的配置をデータベース化し、パターンをルール化 (Fig. 5) する手法を適用しこの問題を回避した。

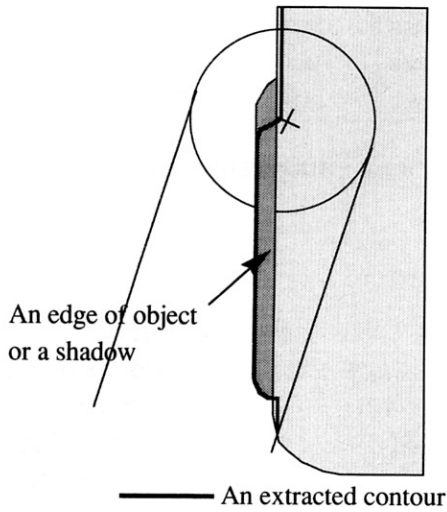


Fig. 4 Mis-extraction of the contour

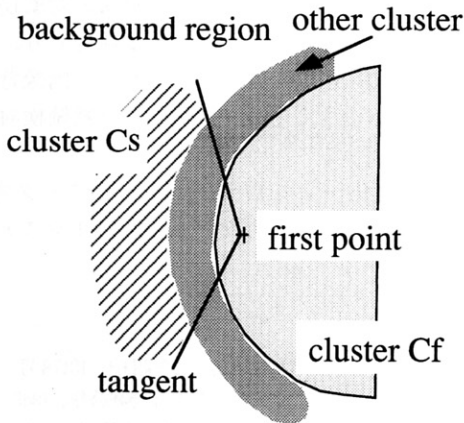


Fig. 5 The method to determine the background cluster

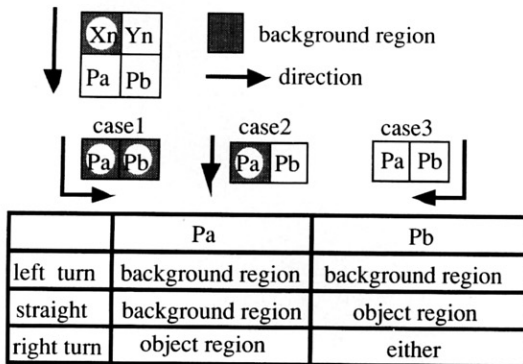


Fig. 6 Condition for extracted contour

3 評価

本手法の評価を以下の3項目について行った。

- 1) 閾値調整回数の低減
- 2) 自動切り抜き速度評価
- 3) 切り抜き精度

3.1 閾値調整回数の低減

本手法を実際に製版現場で使用したカラー画像の輪郭抽出部位 175 点に対し、各処理の効果を確認した。Fig. 7 は各処理の実行において所望の輪郭線が得られた個数を円グラフにしてある。処理1の実行のみで適切な輪郭を得たのが40点に対し、処理2を実行した場合に適切な輪郭を得たのが70点である。これは、背景領域のグラデーションに対し処理2が効果をあげていることを示している。また、処理4によって画像のエッジ部分を対象領域に誤認識することが回避できることが確認できた。これにより、通常オペレーターによる閾値調整が処理2と処理4により110回削減できたことになる。ただし、すべての処理を実行した結果、効果が現れずに抽出輪郭が迷走する場合がある。これは画像の背景領域に雑音成分が多く、第2サンプル領域のクラスタリングが良好に行われないのが原因となっている。

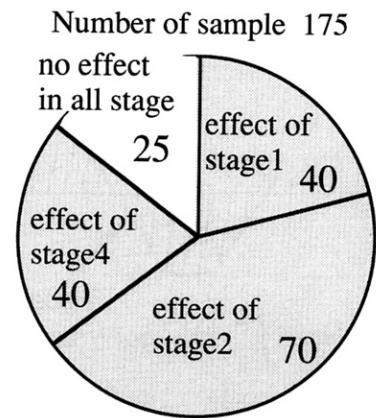


Fig. 7 Result of effect of each stage

Table 1 自動切り抜き速度評価 (実測値)

	データ容量	マニュアル トレース	オート トレース
画像 A	8.4 MB	9分37秒	6分7秒
画像 B	3.0 MB	3分44秒	2分54秒
画像 C	2.5 MB	3分43秒	1分54秒
画像 D	1.8 MB	1分56秒	1分12秒

測定: 画像読み込み→トレース→マスク保存の作業をマニュアルトレースとオートトレースで比較した

3.2 自動切り抜き速度評価

自動切り抜き機能の速度評価は製版工程において実作業として使用した場合に、作業時間がどれだけ短縮できたかを測定した。測定結果を Table 1 に示す。自動切り抜き機能により、30～40%の時間短縮が達成された。

3.3 切り抜き精度

切り抜きの実作業を行った場合、ポイントの位置や線種によって、切り抜き線の軌跡はある程度ずれが生じる。よって、切り抜きマスクの品質としてはある程度許容範囲を持っていることになる。そこで、オートトレースで切り抜いた軌跡がその許容範囲付近に収まるかを評価した。評価手順を以下に示す。

- ① マニュアルトレースのみで、同一画像から品質として問題のないレベルの複数のマスク画像を作成する。
- ② 2枚のマスク画像を重ねた時、重ならないマスク部分の画素の総和をとり、マスク間の累積誤差を求める。これを軌跡のずれの許容範囲とする。(Fig. 8)
- ③ オートトレースで切り抜きマスク画像を作成し、上記方法でマニュアルトレースのマスク画像と重ならない画素を測定し累積誤差を求める。

本評価において、マニュアルトレースで測定した累積誤差とオートトレースで測定した累積誤差に大きな差がないことを確認した (Table 2)。

Table 2 自動切り抜き精度評価

	周囲長	累積誤差	
		マニュアル	オート
画像A 8.03 MB	3560	5 5 7	5 1 8
画像B 6.24 MB	4258	7 1 7	7 9 7
画像C 3.20 MB	5638	8 3 4	1 4 1 0

単位：ピクセル数

* マニュアルトレースは3回切り抜いた場合の累積誤差の平均値

4 まとめ

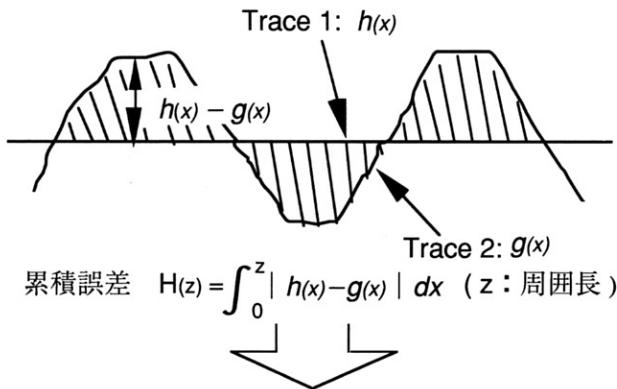
本アルゴリズムの自動切り抜き機能により、切り抜きの作業時間を、30～40%短縮できることを確認した。また、マスク品質においては、マニュアルトレースで切り抜いた、品質の保証されたマスクと同等の品質が得られ、マニュアルの代替として使用できる自動切り抜き機能であることを確認した。

従来の自動切り抜き機能で頻繁に行われていた閾値調整も低減できることを確認した。特に画像の対象物の境界にエッジ強調による白スジや影がある場合でも、オペレータの閾値調整の手間を軽減した。但し、画像背景領域の雑音成分に起因する、誤りに関しては継続検討を実施していく。

なお、本アルゴリズムを搭載した切り抜きワークステーション (CUTART) を Performik-EV の1システムとして上市した。

●参考文献

- 1) 坂本 卓、佐野 鉄雄、児玉英二、"公告昭 63-13178 号 抜きマスク版の作成方法"、日本国特許庁特許公報、1989。
- 2) 安居院猛、長尾智晴 "画像の処理と認識"、昭晃堂、1992。
- 3) 高木幹雄、深澤秀通 "公開平 1-29477 号自動切り抜きシステム"、日本国特許庁特許公報、1989。
- 4) 木之下洋、奥沢二郎、米山努 "第 94 回春期研究発表会" 日本印刷学会、1995。



デジタル画像： 作成した閉曲線を2値画像とし、2つの画像の差分画像の0以外の値を持つ画素の総和を累積誤差とした。

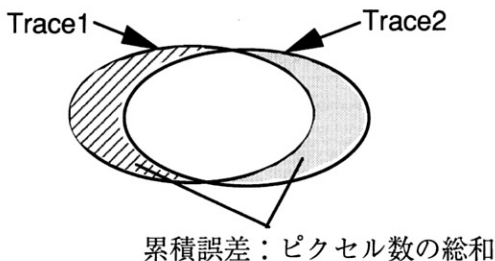


Fig. 8 切り抜き精度評価方法