

基単調領域の非交差和領域の最適イメージ切り出しアルゴリズム

Algorithms for computing the optimal image segmentation of nonintersecting union of base monotone regions

全 眞嬉* 葛西 亮生* コルマン マティアス* 徳山 豪*
 Jinhee Chun Ryosei Kasai Matias Korman Takeshi Tokuyama

1 はじめに

$n \times n$ のピクセルグリッド平面 G からのイメージ切り出し問題は、画像処理における最重要問題の一つである。様々な実用的なアプローチが知られているが、Asano ら [1] はイメージ切り出し問題を組合せ最適化問題として定式化し、パラメトリック最適化の技法を利用して解決する枠組みを提案し、先端的なアルゴリズム理論の応用として注目された。画像はピクセルグリッド G における輝度分布として表される。具体的には、各ピクセル $p = (i, j)$ に対して、実数値 $b(p) = b_{i,j}$ を与える。色画像の場合は、適当な色ベクトルへの射影を考えて輝度を計算し、その後で色ベクトルについての最適化を行うことができる。 G のピクセル集合の適当な族 \mathcal{F} を考え、 \mathcal{F} の要素 S をイメージ領域と呼ぶ。最適化問題として定式化するときには、イメージ領域 S に対してある目的関数を与え、その最小化もしくは最大化を行う。具体的に Asano らの利用したものはクラス内分散と呼ばれるものであり、 S 内の輝度の分散和と $G \setminus S$ 内の輝度の分散和の和を最小化する。パラメトリック最適化の手法を用いると、次のような問題を解くことが鍵になる。

最大重み領域問題：グリッド G の各ピクセル (i, j) に実数値の重み $w_{i,j}$ を与える。このとき、 $S \in \mathcal{F}$ で、重み和

$$w(S) = \sum_{(i,j) \in S} w_{i,j}$$

を最大化するものを求めよ。

パラメトリック最適化の理論から、適当なパラメタ値 θ が存在し、 $w_{i,j} = b_{i,j} - \theta$ に対する最大重み領域がクラス内分散を最小化することが示される。さらに、このパラメタ値は、最大重み領域問題を最悪 $O(n^2)$ 回 (実用的には $O(\log n)$ 回) 解くことで求める事ができる。

したがって、最大重み領域問題を効率よく解ければ、クラス内分散の最小化が達成される。しかしながら、最大重み領域問題は領域族 \mathcal{F} の性質に計算複雑度が大きく依存し、たとえば \mathcal{F} として連結なグリッド領域全てからなる族を取ると、NP 完全となる [1]。一方、最大重み領域問題が多項式時間で解ける場合も多くあり、過去にはそのような領域族が利用されてきた。代表的なものとして、基単調領域、 x 単調領域、連結 x 単調領域があり、それらにおいては最大重み領域はグリッドサイズ $N = n^2$ に対して線形時間で計算することができる。

ここで、各列 j に対して、 $0 \leq f(j) \leq n$ を対応させる関数 f を考える。このような関数はグリッド上の曲線を定義し、そのような曲線の下側領域で表されるピクセル集合を (x 軸、あるいはグリッドの底辺を基とした) 基単調領域 (base monotone region) と呼ぶ。具体的には $S = \{(i, j) : i \leq f(j)\}$ として表

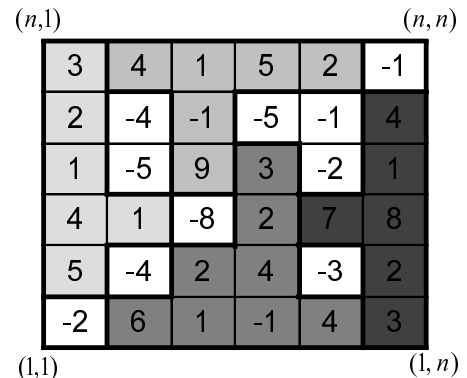


図 1: 問題の例とその最適解

される領域である。同様にグリッド境界の 4 つの辺それぞれを基として基単調領域を考えることができる。二つの関数 f, g を用いて、 $S = \{(i, j) : g(j) \leq i \leq f(j)\}$ と表される領域を x 単調領域と呼び、 x 単調領域であり、かつグリッド位相で連結な領域を連結 x 単調領域と呼ぶ。これらの領域に関する最大重み領域問題はデータマイニングや放射線医療における最適化においても広く利用されている [5, 6, 2, 4]。たとえば、現在の放射線治療においては、放射線の照射は正方形の照射窓を上下から伸びる遮蔽棒で遮り、照射される対象の腫瘍に放射線を当てると共に、重要な臓器には放射線を遮ることができるようになってきている (Interleaf collimator) が、遮蔽棒の位置は上記の二つの関数 f, g で定義され、照射される部分は x 単調領域となる。すなわち、いくつかの x 単調領域での照射を重ねることにより、必要な量を腫瘍に照射し、かつ臓器への副作用を最小化することが求められ、一方照射回数最小化も要求される。この最小化問題は困難な計算問題であり、また、現在の医療機器の性能の関係で x 単調領域にいくつかの制限がつくが、最大重み領域問題を基盤にして実用解法を設計することが可能である。

しかしながら、これらの領域は非常に特殊であり、より複雑なイメージの切り出しを行うためには、これらの領域を複数用いて表示できるような図形を考察することが望ましい。しかしながら、 x 軸を基とした基単調領域と y 軸を基とした基単調領域の和集合として表示できる図形の族を考えると、複雑な交差パターンを持つ図形が生成され、対応する最大重み領域問題は NP 困難問題となってしまう [3]。

本論文では 4 つのグリッド境界辺を基とする基単調領域を考え、それらの非交差和で表示される (すなわち、基単調領域に分割される) 図形の集合を考える (図 1)。たとえば、上下 2 つの辺を基とする基単調領域の非交差和 (この場合は交差和でも同じである) を考えると、その補集合は x 単調領域となり、重

* 東北大学大学院情報科学研究科システム情報科学専攻

表 1: アルゴリズムの計算量

分割数	時間計算量	空間計算量
3 分割以下	$O(N)$	$O(N)$
4 分割	$O(N^2)$	$O(N)$
4 分割	$O(N^{1.5})$	$O(N^{1.5})$

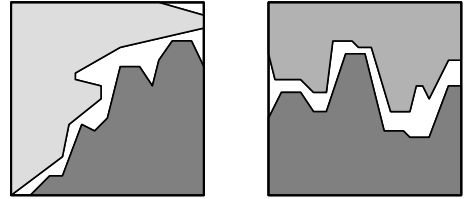


図 2: 2 色塗りの分類

みの符号の逆転を行うことで補集合に関する最大重み領域問題への変換ができることに注意すると、これは既存の x 単調領域に関する最大重み領域問題と等価になる。しかしながら、それ以外の組み合わせに対しては過去に考察されていない。本論文では、基単調領域に分割される数ごとに、表 1 の時間計算量と空間計算量を持つアルゴリズムを紹介する。たとえば 3 分割というと、3 つの基単調領域に分割が可能である領域であり、重みが最大であるものを求める問題を意味する。なお、本文中では分割成分それぞれに色を割り当て、色塗り問題として説明を行う。

2 問題の定義と基本となるアルゴリズム

ここでは、重みを表す $n \times n$ 行列 $W = (w_{i,j}) = (W(i,j))$ を考え、この行列を入力とする。本論文では、各座標に対して重みを与える行列を定義しているが、行列で考えるので、通常の平面座標の (x,y) と縦横が逆であることに注意する必要がある。便宜上、 $W(0,j) = W(n+1,j) = W(i,0) = W(i,n+1) = 0$ とする。行列から任意の個数の成分を選んだ集合 $S = \{(i_1, j_1), \dots, (i_k, j_k)\} (\forall (i,j) \in S \implies 1 \leq i, j \leq n)$ を考えたとき、 S の重み $w(S)$ を

$$w(S) = \sum_{l=1}^k w_{i_l, j_l}$$

と定義する。今、4 つの非交差な成分集合 U, D, L, R を考え、

$$w(U \cup D \cup L \cup R)$$

が最大になるような U, D, L, R を求めたい。ただし、ここで U, D, L, R は 4 つの境界辺に関する基単調領域であり、それぞれが満たす条件を記述すると、以下ようになる。

$$\begin{aligned} (i,j) \in U &\implies ((i+1,j) \in U) \vee (i=n) \\ (i,j) \in D &\implies ((i-1,j) \in D) \vee (i=0) \\ (i,j) \in L &\implies ((i,j-1) \in L) \vee (j=0) \\ (i,j) \in R &\implies ((i,j+1) \in R) \vee (j=n) \end{aligned}$$

直感的には、切り出し領域の U, D, L, R の各成分に色を割り当てると、4 つの境界辺から 4 色のペンキを (それぞれ、横、縦の方向に) 塗ってできる行列の色塗り問題と捉えることができる。ペンキの刷毛は途中で持ち上げてはならないとし、境界の辺から開始して色塗りを行うと、一つの色が塗られる領域は基単調領域となる。目的は、色領域の交差を許さない上記のような色塗りを考え、(色に関らず) 塗られた領域の重み和を最大にするのである。本論文で与えるアルゴリズムの説明とその図示に便利であるので、以下、領域切り出し問題という代わりに色塗り問題として説明を行う。

手始めに、1 色のみを利用した色塗りの場合を考える。これは基単調領域に関する最大重み領域問題であり、既存結果である。例として、 $w(D)$ の最大値について考えると、列ごとに分

割して計算することができる。すなわち、第 j 列成分の i 行までのプレフィクス和

$$sum(i,j) = \sum_{k=0}^i W(k,j)$$

を計算し、漸化式

$$D(i,j) = \max \{D(i-1,j), sum(i,j)\}$$

を利用して、第 j 行目のプレフィクス最大値

$$D(i,j) = \max_{0 \leq k \leq i} \{sum(k,j)\}$$

を、最大を達成する行番号と共に計算する。プレフィクス最大値を達成する部分を各列において塗れば、最大領域が計算できる。

同様にして、

$$\begin{aligned} U(i,j) &= \max_{i \leq k \leq n+1} \left\{ \sum_{l=k}^{n+1} W(l,j) \right\} \\ L(i,j) &= \max_{0 \leq k \leq j} \left\{ \sum_{l=0}^k W(i,l) \right\} \\ R(i,j) &= \max_{j \leq k \leq n+1} \left\{ \sum_{l=k}^{n+1} W(i,l) \right\} \end{aligned}$$

も求められる。これらの 4 つのテーブルは以下のより複雑な場合のアルゴリズム設計で利用する。これらのテーブルの構成にかかる計算量は、明らかに時間計算量、空間計算量ともに $O(n^2) = O(N)$ であり、ピクセル数に対して線形である。

2.1 2 色での色塗り

次に、ピクセルグリッド平面の 2 色での最適な色塗りについて考える。2 色の塗り方は大きく以下の 2 つのパターンに分類できる (図 2)。

1. 隣接する位置の境界辺からの色塗り ($D \cup L, U \cup R$ など)
2. 対面する位置の境界辺からの色塗り ($U \cup D, L \cup R$)

これらの 2 つの場合について、1 色での色塗りの結果を用いて計算時間 $O(N)$ で求める。対面位置の場合は簡単であり、既存の x 単調領域に関する最大重み領域問題と等価であるので、省略する。隣接する位置からの塗り分けの例として $w(D \cup L)$ の最大値について考える。ピクセルグリッド平面を (i,j) 成分までに制限したときの最適値 $DL(i,j)$ について考えると、 (i,j) 成分についての状況で、以下のように場合分けができる。

- (1) $(i,j) \in D$ (図 3 左)
- (2) $(i,j) \in L$ (図 3 中央)

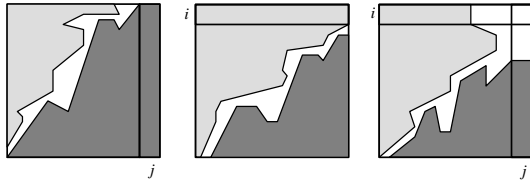


図 3: 隣接位置からの塗り分けの場合分け

(3) $(i, j) \notin (\mathbf{D} \cup \mathbf{L})$ (図 3 右)

(1) の場合,

$$DL(i, j) = DL(i, j - 1) + D(i, j)$$

と表すことができる。同様に, (2) の場合,

$$DL(i, j) = DL(i - 1, j) + L(i, j)$$

と表すことができる。(3) の場合は, (1) と (2) の両方の場合について考えればよいので, 結局 $DL(i, j)$ に対して以下の漸化式が成り立つ。

$$DL(i, j) = \max\{DL(i, j - 1) + D(i, j), \\ DL(i - 1, j) + L(i, j)\}$$

この漸化式を用いて, 動的計画法により任意の (i, j) 成分までに制限した 2 色塗りの重み和の最大値を求めることができる。後に 3 色及び 4 色塗りアルゴリズムでは, 全ての (i, j) について $D(i, j)$ を計算したテーブルを利用するが, このテーブルの要素を一つ新たに計算する計算量は上記の漸化式の計算のみであり, 既に事前に計算されている D 及び L のテーブルを利用すると $O(1)$ であるので, テーブル DL を計算する計算量は時間計算量, 空間計算量ともに $O(N)$ である。定義から, グリッド全体での 2 色塗りの最適値は $w(\mathbf{D} \cup \mathbf{L}) = DL(n, n)$ であり, 動的計画法のバックトラック手法を利用すれば $O(N)$ の時間で $DL(n, n)$ を達成する色塗り (すなわち切り出し領域) を構成することができる。

2.2 3 色での色塗り

一般性を失わず, 上辺, 左辺, 下辺の 3 つの境界辺から色を塗るとしよう。したがって, $w(\mathbf{U} \cup \mathbf{L} \cup \mathbf{D})$ の最大化について考察する。図 4 の上図に示すように, 領域 \mathbf{L} の最も右にある頂点の x 座標 $LeftTop$ を考える。すなわち

$$(i, j) \in \mathbf{L} \implies j \leq LeftTop$$

である。これに注目すると, 3 色の場合の色塗りは, 2 色塗り分けの部分問題のテーブルを参照しながら, 動的計画法を用いて高速に求めることができる。

$UDL(j)$ をピクセルグリッドの第 j 列目までの部分での 3 色塗りの最適領域 (の目的関数値) とする。列番号 j に関する動的計画法により $UDL(j)$ を計算するアルゴリズムを設計する。このとき 2 つの場合, すなわち最適解において $LeftTop < j$ あるいは $LeftTop \geq j$ の場合に分けることができる。最適解の候補領域を第 j 列までに限定したとき, 後者の場合では列 j までの部分に限定した領域では $LeftTop = j$ である。

最適値 $UDL(j)$ の計算をするとき, 第 j 列目までの候補領域において $LeftTop < j$ のとき, 第 j 列の要素が集合 \mathbf{L} に含

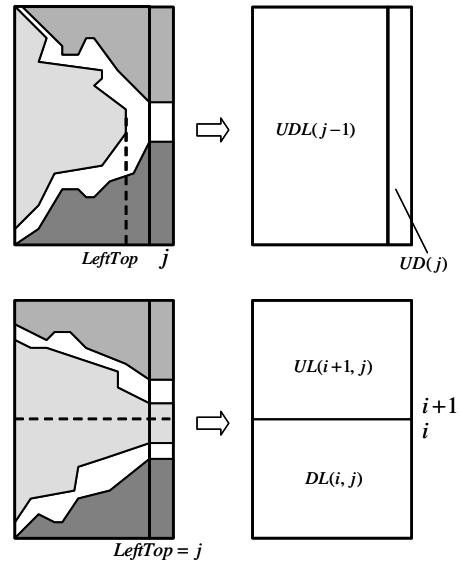


図 4: 3 色塗り問題を解く動的計画法の実行時の分類

まれることはなく, 第 j 列は 2 色の塗り分けとして考えることができるので (図 4 上),

$$UDL(j) = UDL(j - 1) + UD(j)$$

と表すことができる。ここで, $UD(j)$ は第 j 列の上下 2 方向からの塗り分けの最適値, つまり,

$$UD(j) = \max_{0 \leq i \leq n} \{D(i, j) + U(i + 1, j)\}$$

である。

一方, $LeftTop = j$ のときを考えると, 3 色塗り分けはある i 行 ($0 \leq i < n$) を基準として必ず 2 色の塗り分けに分割できるので (図 4 下),

$$UDL(j) = \max_{0 \leq i \leq n} \{DL(i, j) + UL(i + 1, j)\}$$

と表すことができる。

したがって, 結局, $UDL(j)$ に対する漸化式は,

$$UDL(j) = \max\{UDL(j - 1) + UD(j), \\ \max_{0 \leq i \leq n} \{DL(i, j) + UL(i + 1, j)\}\}$$

と表すことができる。配列 UD , DL , UL はあらかじめ線形時間で用意することができ, $\max_{0 \leq i \leq n} \{DL(i, j) + UL(i + 1, j)\}$ の計算は各 j について $O(n)$ で行える。よって, 3 色塗り分けの最適値は時間計算量, 空間計算量ともに $O(n^2) = O(N)$ で求められる。

定理としてまとめると, 下記のものとなる。

定理 1. 3 つの指定された境界辺からの単調領域に分割できる領域の中で, 重み和を最大化するものを $O(N)$ 時間で $O(N)$ の作業領域を用いて計算することができる。

3 4 色での色塗り

最後に, 4 色の色塗りについて考える。 $LeftTop (= LT)$ と同様に, 頂点 $RightTop (= RT)$, $UpTop (= UT)$, $DownTop (=$

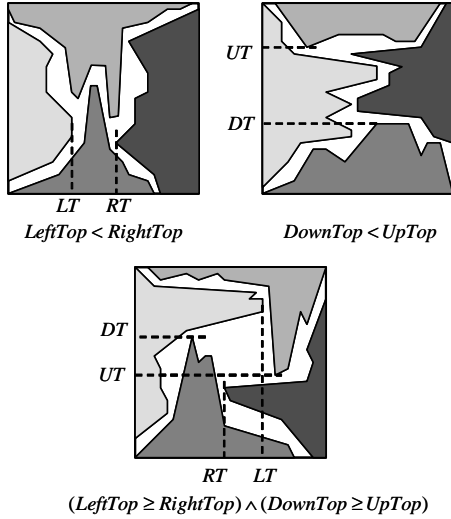


図 5: 4 色塗りのパターンの分類

DT) を,

$$\begin{aligned} (i, j) \in \mathbf{R} &\implies j \geq \text{RightTop} \\ (i, j) \in \mathbf{U} &\implies i \geq \text{UpTop} \\ (i, j) \in \mathbf{D} &\implies i \leq \text{DownTop} \end{aligned}$$

と定義し、これら 4 つの値に注目すると、4 色塗りは以下の 3 つのパターンに分類できる。

- (1) $\text{LeftTop} < \text{RightTop}$ (図 5 左上)
- (2) $\text{DownTop} < \text{UpTop}$ (図 5 右上)
- (3) $\text{LeftTop} \geq \text{RightTop}$ かつ $\text{DownTop} \geq \text{UpTop}$ (図 5 下)

$\text{LeftTop} < \text{RightTop}$ の場合

この場合、ある j 列 ($\text{LeftTop} \leq j < \text{RightTop}$) を基準として 3 色塗りの部分問題に分解して考えることができる。この場合の 4 色塗りの最適値 $UDLR_1$ は、

$$UDLR_1 = \max_{0 \leq j \leq n} \{UDL(j) + UDR(j+1)\}$$

と表すことができる。これは、前節の結果と、そこで求めた配列 UDL, UDR を利用すると、時間計算量、空間計算量ともに $O(N)$ で求められる。

$\text{DownTop} < \text{UpTop}$ の場合

この場合も、ある i 行 ($\text{DownTop} \leq i < \text{UpTop}$) を基準として必ず 3 色塗りの部分問題に分解することができ、塗り分けの最適値 $UDLR_2$ は、

$$UDLR_2 = \max_{0 \leq i \leq n} \{DLR(i) + ULR(i+1)\}$$

と表すことができる。これも時間計算量、空間計算量ともに $O(N)$ で求められる。

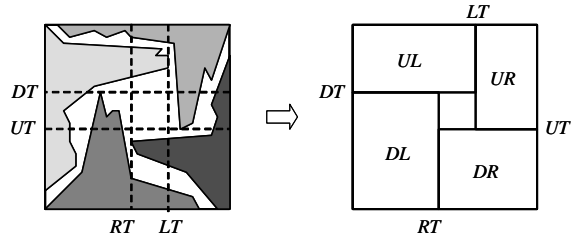


図 6: 4 色塗り分け問題の 2 色塗り分け問題への分割

$\text{LeftTop} \geq \text{RightTop}$ かつ $\text{DownTop} \geq \text{UpTop}$ の場合

この場合、部分問題への分割方法として以下の 2 つの手法を考えることができ、それぞれに従ってアルゴリズムを設計することができる。この二つのアルゴリズムの性能にはトレードオフがあるので、それぞれを独立に紹介する。

- (i) 2 色塗り分けへの分割
- (ii) L 字型の 3 色塗り分けへの分割

3.1 2 色塗り分けへの分割を利用したアルゴリズム

$\text{UpTop}, \text{DownTop}, \text{LeftTop}, \text{RightTop}$ の位置と、塗り分けの解の図に注目すると、4 色塗り分け問題は図 6 のように 4 つの 2 色塗り分け問題 (4 畳半型) に分解できる。また、 $\text{UpTop}, \text{DownTop}, \text{LeftTop}, \text{RightTop}$ の位置を固定して考えたとき、図 7 のように、2 色塗り分けのパターンは 2 つ存在する。4 畳半型の中央の領域にはどの色も入り込むことはできない。これは、ある色に注目したとき、必ずブロッカーとして働く色が存在するためである。例えば図 6 では R のピクセルは U の列に阻まれて中央領域に入ることができない。さらに、分割された領域での色塗りの条件は他の領域での色塗りの状況に左右されず、したがって独立に最適化することができる。

したがって、この場合の 4 色塗り分けの最適値 $UDLR_3$ を求める式は以下ようになる。

$$UDLR_3 = \max_{\substack{1 \leq RT \leq LT \leq n \\ 1 \leq UT \leq DT \leq n}} \left\{ \begin{aligned} &DL(DT, RT - 1) \\ &+ DR(UT - 1, RT) + UL(DT + 1, LT) \\ &+ UR(UT, LT + 1), \\ &DL(UT - 1, LT) + DR(DT, LT + 1) \\ &+ UL(UT, RT - 1) + UR(DT + 1, RT) \end{aligned} \right\}$$

この場合の時間計算量は、パラメタ RT, LT, UT, DT について考えなければならないので $O(n^4) = O(N^2)$ である。空間計算量については 2 色塗り分けの結果を記憶しておけばよいので $O(n^2) = O(N)$ となる。 $\text{LeftTop} < \text{RightTop}, \text{DownTop} < \text{UpTop}$ の場合も含めると、結局、4 色塗り分けの最適値 $UDLR$ は、

$$UDLR = \max\{UDLR_1, UDLR_2, UDLR_3\}$$

で与えられる。合計の時間計算量は $O(n^4) = O(N^2)$ 、空間計算量は $O(n^2) = O(N)$ となる。

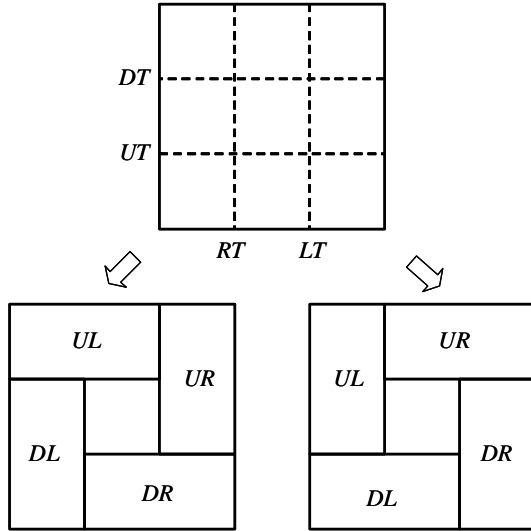


図 7: 2色塗り分け問題への分割のパターン

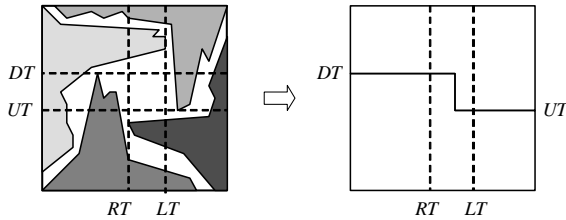


図 8: 4色塗り分け問題の L 字型の 3色塗り分けへの分割

3.2 L 字型の 3 色塗り分けへの分割による高速化アルゴリズム

$LeftTop \geq RightTop$ かつ $DownTop \geq UpTop$ の場合の計算時間の高速化を考える。塗り分けの解に注目すると、図 8 のように 2 つの L 字型の 3 色塗り分けに分割することができる。この性質に注目して高速化アルゴリズムを考える。

まず、L 字型の 3 色塗り分けについて考える。図 9 のように、L 字型は $ULR_R, ULR_L, DLR_L, DLR_R$ の 4 パターンが考えられる。それぞれのパターンに対して、パラメタ s, t, u を図 9 のように定義する。例として DLR_L について述べると、

- s : 左側の高さ
- t : 右側の高さ
- u : 高さが s である、最も右の位置
- $DLR_L(s, t, u)$: 図の L 字型に対する 3 色塗り分けの最適値

となる (ただし、 $s \geq t$)。 $DLR_L(s, t, u)$ を求める漸化式を考える。 $DownTop$ に注目すると、 $0 \leq DownTop \leq s-1$ の場合と $DownTop = s$ の場合が考えられる。 $0 \leq DownTop \leq s-1$ の場合、

$$DLR_L(s, t, u) = DLR_L(s-1, t, u) + L(s, u)$$

が成り立つ。 $DownTop = s$ の場合は、集合 R に対するブロッカーに注目し、2 色塗り分け問題に分割して考える (図 10)。集合 R に対するブロッカー b とは、集合 D の高さが t 以上となる、最も右の列 j であり、

$$b = \max \{j | ((i, j) \in D) \wedge (i \geq t)\}$$

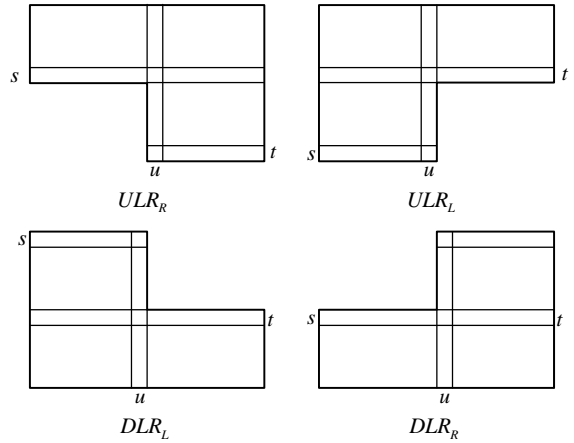


図 9: L 字型のパターンとパラメタ

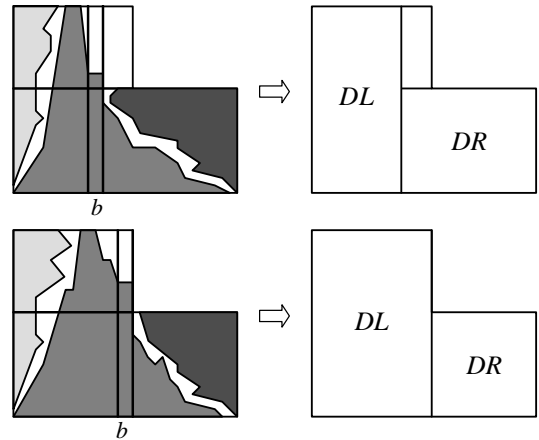


図 10: ブロッカーに注目した L 字型の分割

で表され、 $j \leq b$ である (i, j) 成分は集合 R の要素となることはできない。また、常に $b \leq u$ が成り立つ。ブロッカー b に注目すると、

$$DLR_L(s, t, u) = \max_{0 \leq b \leq u} \{DL(s, b) + DR(t, b+1)\}$$

が成り立つ (図 10)。さらに、 $b < u$ の場合について、図 10 上の空白領域に注目すると、

$$\begin{aligned} DLR_L(s, t, u) &= \max_{0 \leq b \leq u} \{DL(s, b) + DR(t, b+1)\} \\ &= \max \{DLR_L(s, t, u-1), \\ &\quad DL(s, u) + DR(t, u+1)\} \end{aligned}$$

が成り立つ。2 つの場合をまとめると、 $DLR_L(s, t, u)$ を求める漸化式は $s > t$ に対して、

$$\begin{aligned} DLR_L(s, t, u) &= \max \{DLR_L(s-1, t, u) + L(s, u), \\ &\quad DLR_L(s, t, u-1), \\ &\quad DL(s, u) + DR(t, u+1)\} \end{aligned}$$

で与えることができる。ここで、初期条件は $s = t$ の場合であり、

$$DLR_L(t, t, u) = DLR(t)$$

で与えられる。これは、パラメタ s, t, u について計算、記憶し、前もって求めておいた 2 色塗りのテーブルを利用すると、

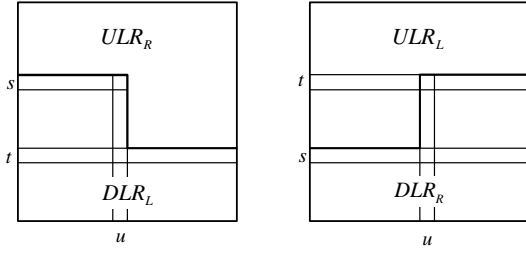


図 11: L 字型の組み合わせ

テーブル DLR_L の要素あたり $O(1)$ で計算を行えるので、時間計算量、空間計算量ともに $O(n^3) = O(N^{1.5})$ である。

同様にテーブル $DLR_L, DLR_R, ULR_L, ULR_R$ を計算することができる。これらを利用し、L 字型分割を組み合わせさせて $UDLR_3$ を求める。L 字型の組み合わせパターン 2 つについて考えると (図 11), $UDLR_3$ は以下の式で与えられる。

$$\begin{aligned}
 UDLR_3 &= \max \left\{ \max_{\substack{0 \leq t \leq s \leq n \\ 0 \leq u \leq n}} \left\{ DLR_L(s, t, u) \right. \right. \\
 &\quad \left. \left. + ULR_R(s+1, t+1, u+1) \right\}, \right. \\
 &\quad \left. \max_{\substack{0 \leq s \leq t \leq n \\ 1 \leq u \leq n+1}} \left\{ DLR_R(s, t, u) \right. \right. \\
 &\quad \left. \left. + ULR_L(s+1, t+1, u-1) \right\} \right\} \\
 &= \max_{\substack{0 \leq t \leq s \leq n \\ 0 \leq u \leq n}} \left\{ DLR_L(s, t, u) \right. \\
 &\quad \left. + ULR_R(s+1, t+1, u+1), \right. \\
 &\quad \left. DLR_R(t, s, u+1) + ULR_L(t+1, s+1, u) \right\}
 \end{aligned}$$

したがって、 $UDLR_3$ を時間計算量、空間計算量ともに $O(n^3) = O(N^{1.5})$ で求める事ができた。

以上をまとめると、下記の定理を得る。

定理 2. 4 つの境界辺からの単調領域に分割できる領域の中で、重み和を最大化するものを $O(N^{1.5})$ 時間で $O(N^{1.5})$ の作業領域を用いて計算することができる。また、 $O(N)$ の作業領域を用いて $O(N^2)$ 時間で計算することができる。

4 結論と拡張

本論文では、4 つの方向からの単調領域の非交差和の最適切り出し問題が効率的に解けることを示した。理論的には、交差を許した和領域の最適切り出しの NP 困難性と比較すると、計算量と幾何学的性質の関連の解明として興味深い成果と考えられる。応用面においても、たとえば、放射線治療における interleaf collimator は現在では左右 2 方向のみからの単調領域の非交差和を遮蔽領域としている。医療機器としての interleaf collimator の耐故障性 (leaf の衝突による破損回避) の問題で、現在は左右 2 方向であるが、原理的には 4 方向からの遮蔽を行う機器の開発も可能であり、本論文の成果はそのよう

な機器を用いた場合の最適制御の効率性を保証するものであり、新しい機器の開発に対する一つの動機を与えるものである。

さらに、画像切り出しへの応用を考えると、本論文の手法をグリッド上の複数の星型領域の非交差和の切り出し問題、あるいは直交凸領域の非交差和の切り出し問題へ拡張することは重要な研究テーマである。これについては、いくつかの成果を既に得ているが、それらは別の論文として今後発表する予定である。

参考文献

- [1] T. Asano, D. Z. Chen, N. Katoh, and T. Tokuyama. Efficient algorithms for optimization-based image segmentation. *Int. J. Comput. Geometry Appl.*, 11(2):145–166, 2001.
- [2] D. Z. Chen, X. S. Hu, S. Luan, X. Wu, and C. X. Yu. Optimal terrain construction problems and applications in intensity-modulated radiation therapy. *Algorithmica*, 42(3-4):265–288, 2005.
- [3] J. Chun, M. Korman, M. Nöllenburg, and T. Tokuyama. Consistent digital rays. In M. Teillaud, editor, *Symposium on Computational Geometry*, pages 355–364. ACM, 2008.
- [4] C. Engelbeen and S. Fiorini. Constrained decompositions of integer matrices and their applications to intensity modulated radiation therapy. In *CTW*, pages 177–180. University of Milan, 2008.
- [5] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. In H. V. Jagadish and I. S. Mumick, editors, *SIGMOD Conference*, pages 13–23. ACM Press, 1996.
- [6] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining with optimized two-dimensional association rules. *ACM Trans. Database Syst.*, 26(2):179–213, 2001.