

# 一枚の線画イラストにおける閉領域の奥行き推定

昆 将太郎<sup>1</sup> 金森 由博<sup>1,a)</sup> 遠藤 結城<sup>1,†1</sup> 三谷 純<sup>1</sup>

**概要:** 与えられた景観画像中の物体の奥行きを推定する研究が、コンピュータビジョンの分野で盛んに行われてきた。自然画像を対象とした奥行き推定の研究は数多く存在するが、アニメやイラストなどの抽象画像の場合、自然画像のような濃淡や模様がなく、手がかりが少ないので奥行きの推定が難しい。本研究の目的は、白地背景に黒い線で描かれた線画の閉領域に対して、互いに隣接する閉領域の間でどちらが手前か奥かを自動で判定することである。本研究では教師あり学習によって線画の領域の局所的な推定を行い、有向グラフを用いることによって大域的な推定を行う。局所的な推定について、イラストによっては境界線の構造が複雑であり、従来の方法では教師あり学習に用いる特徴量の抽出が困難な場合がある。本研究ではそのようなイラストを考慮した特徴量の抽出方法を提案する。さらに、イラストの奥行き推定に適した新たな特徴量を提案し、精度の向上を図る。また、大域的な推定に用いる有向グラフのアルゴリズムについて、サイクルの除去の方法を新たに提案する。

SHOTARO KON<sup>1</sup> YOSHIHIRO KANAMORI<sup>1,a)</sup> YUKI ENDO<sup>1,†1</sup> JUN MITANI<sup>1</sup>

## 1. はじめに

既存のアニメやイラストに奥行き情報を付加して 3D ステレオ視を実現したい、という需要がある。映像制作の現場では手作業で奥行きを付加しているが、自然画像の物体認識やシーン理解などに活用されている奥行き推定処理によって自動化できれば、この作業を大幅に効率化できる。しかし、アニメやイラストなどの抽象画像の奥行き推定は自然画像に比べ困難である。理由としては、自然画像のような濃淡や模様がなく手がかりが少ないこと、アニメ画像の奥行きデータが一般に存在しないことなどが挙げられる。

本研究の目的は、一枚の線画イラストにおいて、各閉領域の奥行きを自動で推定することである。本研究では Jia ら [1] の手法に基づき、教師あり学習を用いて領域の局所的な奥行きを推定し、有向グラフを用いることによって全体の奥行き順序を決定する。局所的な推定については、閉領域の境界線およびそれらの交差点から、交差点での境界線のなす角度、境界の形状などを特徴量として抽出し、学習を行う。線画によっては境界線の離散化の影響で、交差点での境界線のなす角度の抽出がうまくいかない場合がある。本研究では交差点での境界線のなす角において 2 通り

の角度を抽出し、境界線の形状が複雑なイラストに対しても精度のよい結果を出力する。さらに、Jia らの手法では奥行きを決められないケースがあるが、本手法ではそのようなケースにおける対処方法を提案する。線画の奥行き推定に用いる特徴量を抽出したのち、Jia らの手法では構造化サポートベクターマシン (Structured SVM) を用いて、交差点を囲む 3 つの領域間のランキング問題として奥行き関係を推定するが、本手法ではカーネルサポートベクターマシン (Kernel SVM) を用いて領域ペアごとに相対的な前後関係を推定し、精度の向上を図る。大域的な奥行き推定について、有向グラフを用いて局所的な推定結果の矛盾をなくし、最終的な奥行きを決定する。有向グラフを用いて最終的な奥行きを決定する際、有向グラフにサイクルが存在し奥行き順序を決定できない場合がある。本研究では、有向グラフのサイクル消去のアルゴリズムについても考察する。また、本研究のために作成した簡易データセットおよび Liu ら [2] の論文に掲載されている奥行きデータから作成したアニメデータセットを用いて実験を行い、提案手法の有効性を示す。

## 2. 関連研究

自然画像を対象とした奥行き推定について、古くから多くの手法が提案されている。しかしそれらの手法では、陰影やテクスチャなどの手がかりを必要とするため、アニメ

<sup>1</sup> 筑波大学  
<sup>†1</sup> 現在、豊橋技術科学大学  
<sup>a)</sup> kanamori@cs.tsukuba.ac.jp

やイラストなどの抽象画像には使用できない。

Jia ら [1] の手法では、T 字交差点と呼ばれる手掛かりに着目することによって、抽象画像と自然画像の両方を対象に奥行きを推定している。T 字交差点を利用した奥行き推定は他にも、Dimiccoli ら [3], Palou ら [4] が研究を行っている。彼らの手法では閉領域の色やテクスチャも考慮して学習を行うが、本研究では白黒の線画のみを対象としているため、色やテクスチャについては考慮しない。

Hoiem ら [5] や Liu ら [6] のように、各閉領域に意味的ラベル (例: 空、地面) を割り当てて学習を行い、奥行き推定を行う研究もある。Hoiem らの手法の推定結果では、上の領域は空、下の領域は地面といったように領域ごとに意味的ラベルを推定し、それをもとに全体の奥行きを推定している。例えば、空となる領域は他のすべての領域より奥となり、地面となる領域は下の部分になるほど手前となる。この手法は限られた自然画像においては良い推定結果が得られるが、空や地面が存在しない画像や、抽象画像や線画イラストなどの推定は難しい。また、線画イラストの場合閉領域がすべて白なので、推定に必要な色やテクスチャなどの情報が得られない。

奥行き推定の応用例として、次のようなタスクが挙げられる。Liu ら [2] の研究では、2D セルアニメーションを入力として、半自動で各フレームの奥行き順序を推定し、立体視のアニメーションを生成する。また、Yeh ら [7] の研究では、グラフカットにより髪の毛のセグメンテーションを行い、奥行き推定を自動で行うことで、レイヤ構造に分解して、髪の毛のアニメーションが可能になる。

### 3. Jia らの手法

本研究では Jia ら [1] の手法をベースとしているため、本節で詳しく説明する。なお、本研究では線画イラストにおける奥行き推定のみを行うため、同じく Jia らが提案した、画像の遮蔽境界の検出方法については本節では説明しない。

Jia らの手法について、まず交差点ごとに局所的奥行き推定のための特徴量を抽出する。次に、サポートベクターマシン (SVM) を用いて領域の手前らしさを表す尤度を計算する。最後に有向グラフを用いて全体の奥行きを決定する。

#### 3.1 局所的奥行き推定

局所的奥行き推定における前処理として、線画から領域と境界線および、それらが交差する交差点を抽出する。そして、交差点ごとに特徴量を抽出し、経験則に基づいた局所的な奥行き推定を行う。本節では、Jia らの提案した T 字交差点における角度特徴と境界線における境界特徴について説明する。

また、学習に使用するラベルを次のように設定する。領域  $R_1$  が領域  $R_2$  に対して手前であるということを  $R_1 \rightarrow R_2$

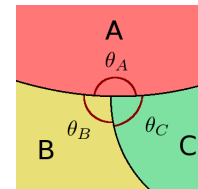


図 1 T 字交差点の例。 $\theta_A$  がおよそ  $180^\circ$  であり、 $\theta_B, \theta_C$  がそれぞれおよそ  $90^\circ$  であるため、領域 A は領域 B, C よりも手前であると判断する。

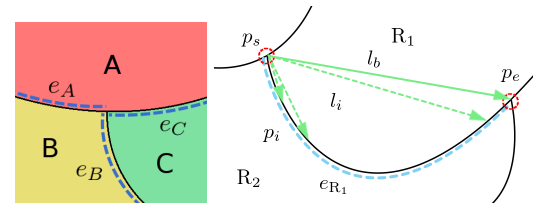


図 2 境界特徴の例 (左) と境界特徴量の抽出方法 (右)。

と表すことにする。このとき特徴量  $f(R_1 \rightarrow R_2)$  に対するラベル  $y$  を、 $R_1 \rightarrow R_2$  という関係が正しいならば  $y = 1$ 、そうでなければ  $y = -1$  と割り当てる。

##### 3.1.1 角度特徴

T 字交差点とは、図 1 に描かれるような 3 つの境界線が交差する点である。経験則により、T 字交差点においては、ある 1 つの領域に接する 2 つの境界線のなす角 (図 1 の領域 A の角度  $\theta_A$ ) が  $180^\circ$  に近く、他の 2 つの領域にそれぞれ接する 2 つの境界線のなす角 (領域 B, C の角度  $\theta_B, \theta_C$ ) が  $90^\circ$  に近ければ、 $180^\circ$  に近い角度を有する領域 (領域 A) が、他の 2 つの領域 (領域 B, C) よりも手前であると判断する。

##### 3.1.2 境界特徴

境界線の形状を奥行き推定の手掛かりとして利用する。例えば図 2 左では、領域 A に属する境界線  $e_A$  は領域 B に対して凸状であるため、領域 A は領域 B を遮蔽していると考え、領域 A は領域 B よりも手前であると判断する。また、領域 B に属する境界線  $e_B$  は領域 C に対して凹状であるため、領域 B は領域 C よりも奥であると判断する。

境界の形状を判断するための特徴量の抽出方法について、まず図 2 右のように始点となる交差点  $p_s$  から終点となる交差点  $p_e$  に向かうベクトル  $l_b$  を生成する。そして  $p_s$  から境界線上の各点  $p_i$  に向かうベクトル  $l_i$  を生成し、各ベクトル  $l_i$  と  $l_b$  のなす角度  $\theta_i \in [-\pi, \pi]$  を求める。すべての  $\theta_i$  を取得したあと、 $[-\pi, \pi]$  を 36 個の区間で区切ったヒストグラムに  $\theta_i$  を割り当て、ヒストグラムを正規化する。これらの 36 次元の特徴量を境界特徴  $f_c(R_1 \rightarrow R_2)$  とする。

#### 3.2 大域的奥行き推定

各 T 字交差点における領域の奥行き推定は経験則に基づく判断なので誤りがある場合があり、最終的な領域の奥行き順序を決めるときに矛盾が生じる可能性がある。例え

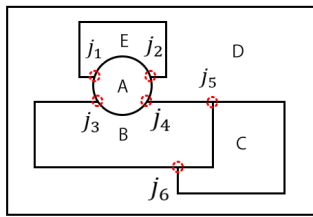


図 3 局所的奥行き推定に矛盾が生じる線画の例。

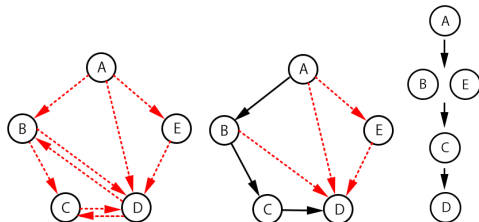


図 4 図 3 の局所的奥行き推定の結果から生成した有向グラフ。

ば図 3 では、交差点  $j_5$  において領域 D と領域 B, C が T 字交差点を形成している。各境界線が直線であるため境界特徴では有益な情報が得られないが、角度特徴では領域 D が領域 B, C より手前であると判定することになる。しかし、この例では直感的には領域 D が領域 B, C よりも奥である方が自然だと思われ、もし他の交差点での推定結果がそのような判定であれば、交差点  $j_5$  での判断と矛盾が生じる。

### 3.2.1 有向グラフの作成

局所的奥行き推定での矛盾をなくすために、領域間の奥行き順序を表す有向グラフを用い、大域的に整合の取れた奥行き順序を推定する。例えば図 3 では、まず図 4 左のように有向グラフを作成する。エッジの方向は領域間の奥行き順序を表し、例えば「 $A \rightarrow B$ 」は、領域 A が領域 B より手前であることを示している。また、エッジの重みとして、各交差点で求めた領域の手前らしさを表す尤度を用いる。領域 B と領域 D のように双方向のエッジが発生するときには、2つのエッジの重みを比べ、小さい方のエッジを削除する。

### 3.2.2 最長パスを基にした最終的な奥行き順序の決定

3.2.1 節のように更新した有向グラフ (図 4 中央) から最長パスを探し、最長パスに含まれる領域の奥行き順序を決定する。最長パスに含まれない、奥行きが未割り当ての領域については、エッジを逆に辿って、その領域より手前の領域をすべて列挙する。もし列挙した領域がすべて奥行きを割り当て済みなら、その中で最も奥の領域より 1 つ奥になるよう奥行きを割り当て、そうでなければ保留する。この処理を未割り当て領域がなくなるまで繰り返す。例えば図 4 中央では、領域 E は領域 A より奥にあるので、領域 A よりも 1 つ奥に配置する。最終的に、奥行き推定の結果は図 4 右のようになる。

また、本研究では隣接する 2 つの領域の奥行き関係にの

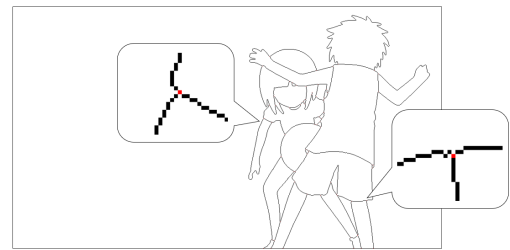


図 5 線画イラストにおける境界線と交差点 (赤のピクセルは交差点)。

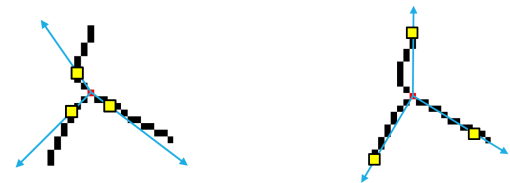


図 6 交差点から遠くのピクセルを利用してベクトルを生成したときの問題点。左: 角度特徴の抽出に使用する理想的なベクトル。右: 交差点から遠くのピクセルを利用して生成したベクトル。



図 7 交差点の近くのピクセルを利用してベクトルを生成したときの問題点。左: 角度特徴の抽出に使用する理想的なベクトル。右: 交差点の近くのピクセルを利用して生成したベクトル。

み着目する。そのため、領域 B と領域 E のような、互いにエッジのない領域 (つまり、隣接していない領域) の奥行き関係については考慮しない。

## 4. 提案手法

### 4.1 局所的奥行き推定

#### 4.1.1 複雑な境界線を考慮した角度特徴

交差点での角度は、3本の境界線を局所的に線分で近似して計算する。各線分の端点は、一方を交差点、他方を交差点から逆方向に数ピクセル辿ったピクセルとする。しかし、何ピクセル辿るかによって線分の向きが変わり、角度が変動する場合がある。例えば、Liu ら [6] の奥行き画像から生成した線画イラストである、図 5 の左側の T 字交差点では、図 6 右のようにベクトルを決定するピクセルを遠くに設定していると、想定とは違う角度を抽出してしまう。また、図 5 の右側の T 字交差点のように境界線がぶれていることもあるため、図 7 右のようにベクトルを決定するピクセルを近くに設定していると角度の抽出がうまくいかない場合がある。

そこで本研究では、これらの 2 通りの角度を考慮できるように、交差点の近くのピクセルおよび遠くのピクセル両方を使って角度特徴を計算する。本研究では近くのピクセルについては 3 ピクセル先を、遠くのピクセルについては 10 ピクセル先をベクトルの決定に使用する。2 つの角度特

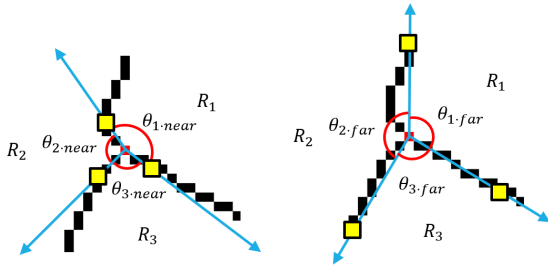


図 8 交差点から境界線上の近く(左)および遠く(右)のピクセルに向かうベクトル群から抽出する角度特徴。

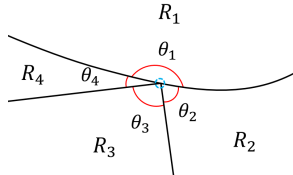


図 9 4つ以上の境界線が交差する点の例。

徴の抽出方法を図 8 に示す。近くのピクセルを使用した角度特徴量を  $f_{a.near}$ 、遠くのピクセルを使用した角度特徴量を  $f_{a.far}$  とし、領域  $R_2$  に対する領域  $R_1$  の隣接関係の角度特徴量をそれぞれ

$$f_{a.near}(R_1 \rightarrow R_2) = (\theta_{1.near}, \theta_{2.near}) \quad (1)$$

$$f_{a.far}(R_1 \rightarrow R_2) = (\theta_{1.far}, \theta_{2.far}) \quad (2)$$

と表す。ただし、 $\theta_{1.near}, \theta_{2.near}, \theta_{1.far}, \theta_{2.far} \in [0, 2\pi]$  である。最終的な角度特徴量は

$$f_a(R_1 \rightarrow R_2) = (\theta_{1.near}, \theta_{2.near}, \theta_{1.far}, \theta_{2.far}) \quad (3)$$

とする。

なお、線画から境界線と交差点を抽出する際、まれに図 9 のように 4 つ以上の境界線が交差する点が発生する可能性があるが、本研究ではそのような交差点においても 3 つの境界線が交差する点と同じ処理を行い、角度特徴量を抽出する。例えば図 9 では、 $f_a(R_1 \rightarrow R_2) = (\theta_{1.near}, \theta_{2.near}, \theta_{1.far}, \theta_{2.far})$  となる。また、境界線が短くベクトルの生成に使用するピクセルが存在しないときは、境界線の終点にあたる交差点をベクトルの生成に使用する。

## 4.2 縁特徴

3 節で述べた Jia らの手法では奥行きを判断できない場合がある。例えば図 10 の領域 C と領域 D では、T 字交差点における角度がどちらもおよそ  $90^\circ$  であり、2 つの領域が挟む境界線の形状も直線であるため、どちらが手前か奥かを判定することができない。そこで、本研究では新たに提案した縁特徴を用いて奥行きを推定する。アニメなどの線画イラストでは、画像の縁に接している領域は背景などの奥側の領域であることが多い、という経験則に基づいて特徴の抽出を行う。実際に図 10 では、領域 D は縁の全体

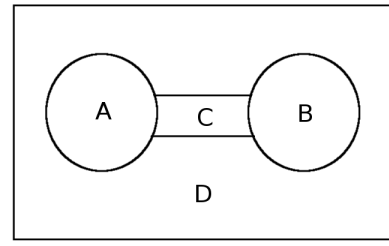


図 10 角度特徴と境界特徴を用いても奥行きを判別できない領域を含む画像の例。Jia らの提案した 2 つの特徴を用いても、領域 C が領域 D より手前であることを推定することができない。

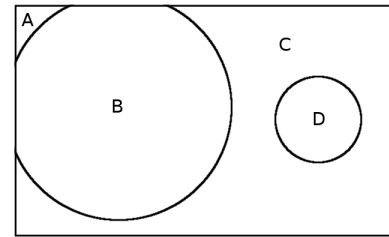


図 11 交差点が存在しない境界線を含む画像の例。領域 A と領域 B、領域 B と領域 C は、2 つの領域が挟んだ境界線の両端が画像の縁に接しているため交差点が存在しない。領域 C と領域 D は、領域 D が領域 C に囲まれており交差点が存在しない。

に接しているため、領域 A, B, C よりも奥である可能性が高い。特徴量の抽出方法について、領域  $R_1$  が接する縁の長さを  $l_1$ 、領域  $R_2$  が接する縁の長さを  $l_2$ 、全体の縁の長さを  $l_{total}$  として、 $f_b(R_1 \rightarrow R_2) = (l_1/l_{total}, l_2/l_{total})$  を縁特徴として用いる。

## 4.3 交差点を持たない領域

Jia らの手法では交差点ごとに特徴量を抽出し、経験則に基づいた局所的な奥行き推定を行う。しかし、交差点が存在しない境界線を挟んだ 2 つの領域については特徴量の抽出ができず、どちらの領域が手前か奥かを推定することができない。例として、図 11 の領域 A と領域 B、領域 B と領域 C については、境界線の両端が画像の縁に接しているため交差点が存在しない。また、領域 C と領域 D についても、領域 D が領域 C に囲まれており交差点が存在しない。そこで、本研究では境界線の両端が画像の縁に接している場合は領域の縁に接する点を交差点と見なし、特徴量を抽出する。領域が囲まれていて交差点が存在しない場合は学習による推定を行わず、囲まれている領域が囲んでいる領域より手前であると強制的に定める。

## 4.4 カーネル SVM

Jia らの手法では、交差点を囲む 3 つの領域間のランキング問題として、構造化 SVM を用いて交差点の周りの奥行きを推定する。このとき、一部の適合度が極端に大きい



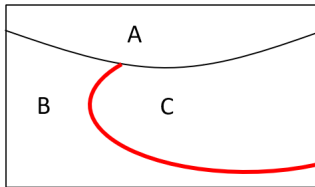


図 12 ランキング問題では正しく推定できない交差点の例。領域 B に対する領域 C の凸状境界の手前らしさが極端に大きいため、推定結果が  $C \rightarrow A \rightarrow B$  となり、領域 A を含む全体の順序に影響を及ぼす。

場合、奥行き順序を誤って推定することがある。例えば図 12 では、領域 B に対する領域 C の凸状境界の手前らしさが極端に大きくなる。そのため、正解順序が  $A \rightarrow C \rightarrow B$  であるのに対し、推定結果が  $C \rightarrow A \rightarrow B$  となっており、領域 A を含む全体の順序に影響を及ぼしている。

そこで、本研究では局所的奥行き推定において、領域ペアごとに奥行きを推定する。例えば、図 12 における境界特徴を用いた奥行き推定について、領域 AB 間または領域 AC 間の奥行き推定には、それぞれの領域が挟む境界線の形状のみを特徴量として抽出するため、領域 BC 間の境界線の形状は領域 AB 間または領域 AC 間の奥行き推定に影響を及ぼさない。

本研究では特徴量の学習に、RBF (radial basis function) カーネルによるカーネルサポートベクターマシン (SVM) を用いる。カーネル SVM とは、線形分類器の一種である SVM に対してカーネル関数を使い、非線形分類に対応した分類器である。本研究ではカーネル関数として分散が 1 のガウス関数を用いる。カーネル SVM を用いて、交差点ごとに隣接する領域間の特徴量を抽出し、学習およびテストを行い領域の手前らしさを表す尤度を求める。

#### 4.5 大域的奥行き推定のための有向グラフにおけるサイクルの削除

最長パスを探索するにあたり、有向グラフは有向非巡回グラフ (Directed acyclic graph; DAG) である必要がある。しかし、実際には矛盾するエッジを取り除いた有向グラフにおいて、サイクルが存在する場合がある。例えば図 13 の場合、 $A \rightarrow C \rightarrow E \rightarrow B \rightarrow A$ 、または  $A \rightarrow D \rightarrow E \rightarrow B \rightarrow A$  のサイクルが存在しているため、最長パスを探索することができない。そこで、本研究では以下のような手法を提案する。

まずサイクルを検出し、サイクルを形成するエッジの中で最も重みの小さいエッジを削除する。これを有向グラフからサイクルがなくなるまで繰り返す。サイクルを除去した有向グラフの例を図 14 に示す。図の赤い点線で記されたエッジは削除されたエッジを示す。サイクルがなくなったら逆方向のパスを辿る。逆方向のパスが存在したらもとのエッジと逆方向のエッジを配置し、存在しなければもとの

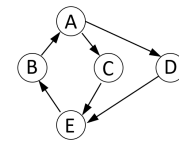


図 13 サイクルが存在する有向グラフの例。

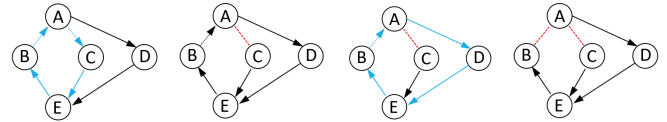


図 14 ループの検出およびエッジの削除。

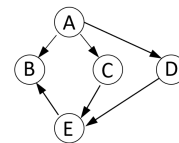


図 15 最終的な有向グラフ。

のエッジをそのまま配置する。最終的な有向グラフは図 15 のようになる。

## 5. 実験

### 5.1 データセット

本研究のテストデータにはアニメなどの線画とその奥行きデータを用いるのが好ましいが、アニメ画像の奥行き画像は一般には存在せず、また作成にも手間がかかる。そこで、簡単な図形の線画および奥行きデータを半自動で生成するプログラムを作成した。本実験ではこのプログラムを用いて作成したデータを用いて実験を行った。このデータセットを、以下では簡易データセットと呼ぶ。また、Liu ら [2] の実験で生成された奥行きデータをもとに線画を生成し、データセットを作成して実験を行った。このデータセットを、以下ではアニメデータセットと呼ぶ。

#### 5.1.1 簡易データセット

本研究では、アニメなどの手描きで描かれた線画イラストが奥行き推定の対象となる。そこで、手描きで図形を作成し、重なり順を順列で並べ替えつつ、回転、拡大縮小、平行移動をランダムに加えて線画のバリエーションを増やし、さらにそれらの奥行きデータを自動で作成するプログラムを作成した。プログラムの作成には C++ を用いた。簡易データセットにおける 1 枚の画像サイズは  $500 \times 500$  ピクセルであり、一枚当たりの交差点の数の平均は 13.1 個である。線画のバリエーションについて、各図形の回転角度が  $[0^\circ, 360^\circ]$ 、図形の面積が  $[0.5, 1.5]$  倍、図形の重心となる座標の  $x$  成分が  $[0, 500]$ 、 $y$  成分が  $[0, 500]$  の範囲になるようにして 100 枚の線画と奥行きデータを作成した。

#### 5.1.2 アニメデータセット

本研究では簡易データセットの他に、Liu ら [2] の論文に掲載されている奥行きデータのうち、23 枚の奥行きデータ

表 1 本研究で使用する特徴量。

	特徴量	次元数
<b>Near</b>	$(f_{a\cdot near}, f_c)$	38
<b>Far</b>	$(f_{a\cdot far}, f_c)$	38
<b>Com</b>	$(f_{a\cdot near}, f_{a\cdot far}, f_c)$	40
<b>Ours</b>	$(f_{a\cdot near}, f_{a\cdot far}, f_c, f_b)$	42

をもとにアニメデータセットを作成した。この奥行きデータから各領域の輪郭線を辿って線画を生成した。アニメデータセットにおける1枚の画像サイズは1104×622ピクセルから1920×1080ピクセルであり、一枚当たりの交差点の数の平均は123.4個である。

## 5.2 実験環境

実験を進めるにあたり、細線化処理を行った線画からハーフエッジデータ構造を構築し、特徴量の抽出および推定結果の精度の算出を行うプログラムをC++により実装した。また、カーネルSVMを用いた局所的奥行き推定および有向グラフを用いた大域的奥行き推定を行うプログラムをPythonで実装した。ライブラリとして、カーネルSVMの計算にはscikit-learnを、有向グラフの処理にはNetworkXを用いた。実験にはCPUとしてIntel Core i7-7700K 4.20GHz、メモリを32GB搭載したPCを用いた。

## 5.3 実験方法

実験の方法について、各データセットでk-分割交差検証を用いて学習、テストおよび精度の測定を行う。簡易データセットでは100枚の画像を10分割、アニメデータセットでは23枚の画像を23分割して実験を行う。精度の測定方法についてはJiaら[1]の手法と同様、推定結果と正解画像を比べて、隣接する2つの領域の奥行き関係が正しくラベル付けされているかを判断する。本研究では隣接する2つの領域の奥行き関係にのみ着目するので、隣接していない離れた領域間の奥行き関係の推定結果が正解画像と比べて間違っていたとしても、精度には影響しない。また、精度の評価方法について、マクロ平均とマイクロ平均の両方を算出する。マクロ平均は各画像の隣接する2領域のペアの数を無視して平等に扱う。一方、マイクロ平均は各画像の隣接する2領域のペアの数を考慮し、ペアの数が少ない画像の影響は小さくなる。

本研究で使用する特徴量を表1のように定義する。**Near**は交差点の3ピクセル先のピクセルを用いた角度特徴と境界特徴の組み合わせであり、**Far**は交差点の10ピクセル先のピクセルを用いた角度特徴と境界特徴の組み合わせである。また、**Com**は2つの角度特徴と境界特徴と組み合わせであり、**Ours**は**Com**と縁特徴の組み合わせである。

## 5.4 結果

簡易データセットについての定性的な結果を図16に、

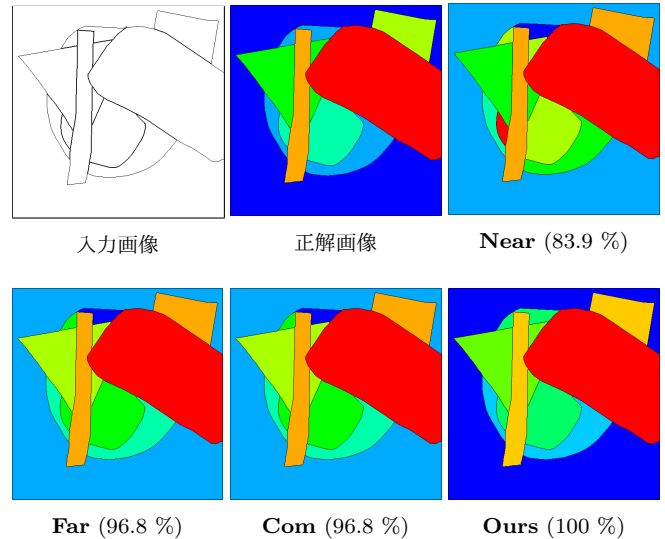


図 16 簡易データセットによる奥行き推定の結果。正解画像および推定結果では、領域が手前にあるほど赤く、奥にあるほど青く色づけされる。推定結果において、**Ours**のみ背景の奥行きを正しく推定している。

表 2 簡易データセットによる奥行き推定の精度の平均 (%)。

	<b>Near</b>	<b>Far</b>	<b>Com</b>	<b>Ours</b>
マクロ平均	87.6	91.7	91.4	<b>94.6</b>
マイクロ平均	87.1	91.4	91.3	<b>94.3</b>

定量的な結果を表2に示す。各手法において、1枚あたりの特徴量抽出および奥行き推定における計算時間は約0.12秒であった。図16の正解画像および推定結果について、領域が奥にあるほど青く、手前にあるほど赤く色づけされる。定性的な結果について、**Ours**以外の手法では背景の奥行きの推定に失敗しているが、**Ours**では正解画像と同じ奥行きの隣接関係を推定することができた。これは、角度特徴と境界特徴だけでは推定できなかった領域同士の奥行き関係が、縁特徴を用いることによって推定できるようになったからであると考えられる。定量的な結果についても、**Ours**の精度はマクロ平均とマイクロ平均の両方において、**Com**と比べて3%ポイントの向上が確認できる。また、**Com**については**Near**よりは精度が向上したが、**Far**より精度が若干下がっていた。簡易データセットにはアニメデータセットのような複雑な境界線が少ないため、 $f_{a\cdot near}$ の利点が少なく、 $f_{a\cdot far}$ のほうが優れているケースが多い。そのため、 $f_{a\cdot near}$ と $f_{a\cdot far}$ を組み合わせることで $f_{a\cdot far}$ 単体よりも精度が下がり、**Com**の精度が下がったと考えられる。

次に、アニメデータセットについての定性的な結果を図17～図19に、定量的な結果を表3に示す。各手法において、一枚あたりの特徴量抽出および奥行き推定における計算時間は約0.44秒であった。図17について、**Near**、**Far**および**Com**では腕や脚、キャラクターの周りの背景を正しく推定できなかったが、**Ours**では縁特徴を用いること

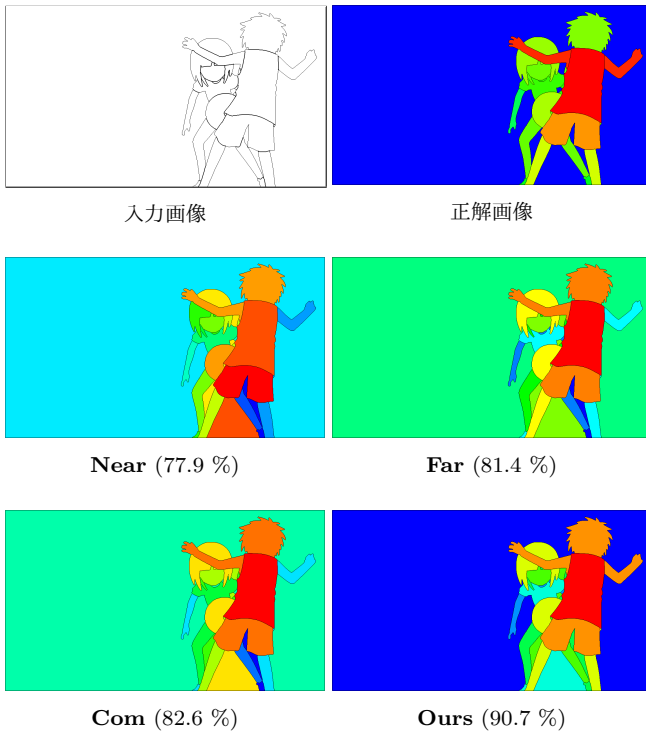


図 17 アニメデータセットによる奥行き推定の結果 (1)。Ours において、腕と背景、脚と背景などの奥行き関係を正しく推定することができた。しかし、身体の間にある背景の奥行きはどれも正しく推定できなかった。

表 3 アニメデータセットによる奥行き推定の精度の平均 (%)。

	Near	Far	Com	Ours
マクロ平均	80.4	81.3	82.7	<b>86.1</b>
マイクロ平均	80.0	81.6	83.1	<b>86.3</b>

によって正しい奥行きを推定することができた。ただし、キャラクター同士の間隙などから見える背景の推定はどの手法でも正しく推定できない場合がある。これは、キャラクター同士の間隙の領域が背景であるということが角度特徴、境界特徴、縁特徴のどれを用いても判断できないためだと考えられる。一方、図 18 では、図 17 のようにキャラクターの間隙に背景が現れないため、他の画像と比べて高い精度が得られた。図 19 ではこれまでのイラストと違い、キャラクター以外にも建物や雲などの領域があり、構造が複雑になっている。そのため、経験則による奥行きの推定が誤りとなる領域が多い。Ours で他の手法よりも精度は向上したが、他のイラストと比べると奥行き推定の精度は低くなっている。

定量的な結果について、Com では Near と比べてマクロ平均でおよそ 2%ポイント、マイクロ平均でおよそ 3%ポイントの精度の向上が、Far と比べてマクロ平均およびマイクロ平均でおよそ 1.5%ポイントの精度の向上が確認できた。また、最終的な提案手法である Ours では Com と比べてマクロ平均およびマイクロ平均でおよそ 3%ポイントの精度の向上が確認できた。

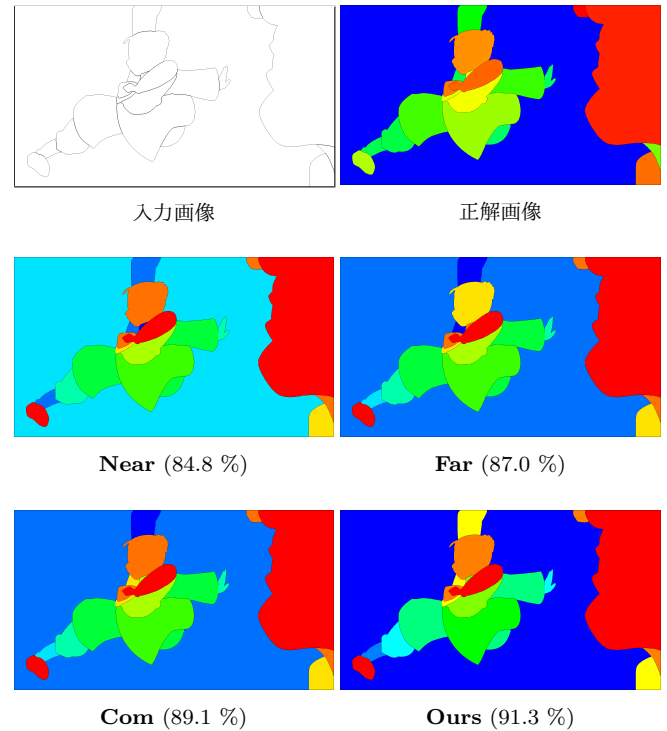


図 18 アニメデータセットによる奥行き推定の結果 (2)。図 17 と同様、Ours では縁特徴を用いることによって背景を正しく推定することができた。

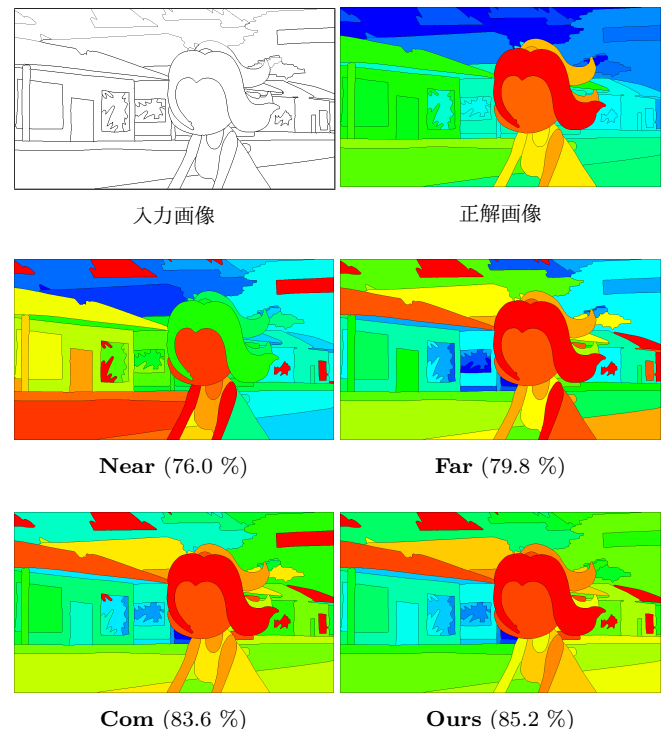


図 19 アニメデータセットによる奥行き推定の結果 (3)。図 17 や図 18 と違い、キャラクター以外にも建物や雲などの領域があり、構造が複雑になっているため、経験則による奥行きの推定が誤りとなる領域が多い。

## 6. まとめ

本稿では、一枚の線画イラスト中の各領域に対して、奥

行きを推定する手法を提案した。Jia らの手法 [1] に対する本研究の新規性は以下の通りである。

- 局所の奥行きを推定する際に使用する特徴量について、交差点の近くのピクセルを用いて抽出した角度特徴  $f_{a-near}$  と交差点の遠くのピクセルを用いて抽出した角度特徴  $f_{a-far}$  の2つの角度特徴を提案した。
- 角度特徴と境界特徴でも推定できない領域の奥行き関係を推定するために、新たに縁特徴  $f_b$  を提案した。
- 内包された領域および境界線の両端が画像の縁に接している領域の奥行きの決定方法を提案した。
- 局所的奥行き推定において、構造化 SVM を用いた3つの領域に対するランキング学習ではなく、カーネル SVM を用いた領域ペアに対する学習を行った。
- 大域的奥行き推定のための有向グラフにおいて、サイクルが発生した際のサイクル除去のアルゴリズムを提案した。

本研究のテストデータにはアニメなどの線画とその奥行きデータを用いるのが好ましいが、アニメ画像の奥行き画像は一般には存在せず、また作成にも手間がかかる。そこで、簡単な図形の線画および奥行きデータを半自動で生成するプログラムを作成し、簡易データセットを作成した。簡易データセットにおける奥行き推定では、 $f_{a-far}$  と  $f_{a-near}$  の2つの角度特徴を用いた場合、 $f_{a-near}$  だけを用いた場合よりも精度が若干下がっていた。これは、簡易データセットでは複雑な境界線が少ないため、 $f_{a-near}$  の利点が少なく、 $f_{a-far}$  のほうが優れているケースが多いからであると考えられる。縁特徴を含めた最終的な提案手法では、他の手法を用いた場合よりも精度が向上しているのを確認できた。アニメデータセットにおける奥行き推定では、簡易データセットと違い複雑な境界線が多いため、2つの角度特徴を用いた場合の精度の向上が確認できた。また、縁特徴を含めた最終的な提案手法についても、他の手法を用いた場合よりも精度が向上しているのを確認できた。しかし、キャラクター同士の隙間に存在する背景など、現在使用している特徴量だけでは推定できない領域も存在する。

今後の課題として、上で記したようなキャラクター同士の隙間に存在する背景など、提案手法だけでは正しく奥行きを推定することができない領域をどのようにして推定するかが挙げられる。改善案として、領域ごとに意味的なラベル付け (例: 腕、脚、背景、建物など) をし、それを学習する方法が挙げられる。

## 参考文献

- [1] Zhaoyin Jia, Andrew Gallagher, Yao-Jen Chang, and Tsuhan Chen. A Learning Based Framework for Depth Ordering. In *Proc. of Computer Vision and Pattern Recognition*, pp. 294-301, June 2012.
- [2] Xueting Liu, Xiangyu Mao, Xuan Yang, Linling Zhang, and Tien-Tsin Wong. Stereoscopizing cel animations.

- [3] Mariella Dimiccoli and Philippe Salembier. Exploiting T-junctions for depth segregation in single images. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*, pp. 1229-1232, April 2009.
- [4] Guillem Palou and Philippe Salembier. Occlusion-based depth ordering on monocular images with binary partition tree. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*, pp. 1093-1096, May 2011.
- [5] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering occlusion boundaries from an image. *International Journal of Computer Vision*, Vol. 91, No. 3, pp. 328-346, February 2011.
- [6] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *Proc. of Computer Vision and Pattern Recognition*, pp. 1253-1260, June 2010.
- [7] Chih-Kuo Yeh, Pradeep Kumar Jayaraman, Xiaopei Liu, Chi-Wing Fu, and Tong-Yee Lee. 2.5D Cartoon Hair Modeling and Manipulation. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 21, No. 3, pp. 304-314, March 2015.