Vol.2019-CG-176 No.21
Vol.2019-DCC-23 No.21
Vol.2019-CVIM-219 No.21
2019/11/8

人物3Dモデルの対話的な意味的領域分割

金森 由博[1,a)]

概要：映像制作や広告用途で，実在の人物を対象とした 3D スキャンの需要が高まっている．撮影で得られる人物 3D モデルは，髪，肌，衣服などの区別がなくつながっており，後工程での利用のためには手作業での切り分けが必要となる．この自動化には，教師あり学習による意味的領域分割が有効だと考えられるが，現状では十分な数の教師データは手に入らない．そこで本研究では，対話的かつ効率的に意味的領域分割を行えるアノテーションツールを提案する．ユーザがブラシを使って意味ラベルを 3D メッシュ上で粗く指定すると，マルコフ確率場の定式化に基づいて，システムが未割り当て部分に意味ラベルを自動的に伝搬させる．提案ツールは指定領域の塗り潰し機能やマスク機能なども備えている．本稿では，現在開発中のアノテーションツールを使った利用事例について紹介する．

Interactive Semantic Segmentation for Human 3D Models

1. はじめに

世界的に映像制作の現場で，役者を 3DCG モデル化したデジタルダブルが長年用いられてきた．最近では国内でも映像制作や広告用途でも需要が高まっており，実在の人物を対象とした 3D スキャンが盛んに行われている．3D スキャンで得られる人物 3D モデルは，ノイズや欠損を含むため，手作業での修正が必須である．さらに，髪，肌，衣服などの領域の区別がなくつながっており，実際の映像制作の際には手作業での切り分けや，領域ごとに別途手作業でのモデリングが必要になる．髪，肌，衣服などの領域に自動で分割できれば，これらの手作業の効率化につながるほか，領域ごとのテクスチャやメッシュの加工に有用だと考えられる．この自動化には，人物を対象とした意味的領域分割が有効だと考えられる．人物を対象とした意味的領域分割は，従来は 2D 画像を対象として研究が行われてきたが，最近では人物 3D モデルを対象とした研究もある．しかし，人物 3D モデルを対象とした既存手法はいずれも教師あり学習を前提としており，正解の意味ラベルを付与した教師データを用意するのは手間がかかる．

本研究では，人物 3D モデルを対象に，対話的かつ効率的に意味的領域分割を行えるアノテーションツールを提案する．ユーザがブラシを使って意味ラベルを 3D メッシュ上で粗く指定すると，マルコフ確率場の定式化に基づいて，システムが未割り当て部分に意味ラベルを自動的に伝搬させる．自動伝播のアルゴリズムとしては現在のところ，次の 3 つを実装している．

(1) テクスチャの色についてのガウス混合モデル (GMM) に基づいたグラフカット
(2) 高解像度メッシュ向けに三角形をクラスタリングしたのち上記と同様にグラフカットを適用
(3) 色・法線・座標を特徴量とした三角形クラスタ単位の多層パーセプトロン (MLP) による多クラス分類

提案ツールは，指定領域の塗り潰し機能や，すでに意味ラベルを指定済みの領域を保護するマスク機能なども備えている．本稿では，現在開発中のアノテーションツールを使った利用事例について紹介する．実験の結果，提案ツールを用いると 1 体あたり平均 46 分程度でアノテーションできることがわかった．

2. 関連研究

人物の意味的領域分割 [1], [2] は，元々は人物画像を対象とした意味的領域分割の一種として研究が始まった．その後，一般の 3D 物体に対する意味的領域分割の研究 [3], [4] が行われるようになると，人物 3D モデルを対象とした意味的領域分割も行われるようになった．

人物 3D モデルを対象とした従来手法 [5], [6] では，人物

[1] 筑波大学
University of Tsukuba, Tennoudai 1-1-1, Tsukuba, Ibaraki, 305–8573, Japan
[a)] kanamori@cs.tsukuba.ac.jp

ⓒ 2019 Information Processing Society of Japan

Vol.2019-CG-176 No.21
Vol.2019-DCC-23 No.21
Vol.2019-CVIM-219 No.21
2019/11/8

3D モデルを投影して得られる 2D 画像を，既存の 2D 画像用のニューラルネットワークに入力して 2D の意味ラベルマップを得，それを 3D メッシュ上に逆投影することで，人物 3D モデルの意味的領域分割を実現する．従来手法のほとんどは，ニューラルネットワークへの入力として色情報しか用いていなかったが，我々の研究 [6] で，色情報のみならず法線や座標といった形状情報も入力することで精度が上がることを確認した．しかしいずれの手法も教師あり学習を前提としているため，教師データを作るためのアノテーション作業が煩雑である．また，2D 画像向けの学習済みネットワークを用いて自動でラベリングを行うにしても，出力される意味ラベルマップの精度が低い上に低解像度であるため，3D メッシュ上に逆投影しても誤りばかりになり，むしろ手作業で修正する手間が増えてしまう．これは特に高解像度メッシュの場合に顕著である．

提案ツールは，マルコフ確率場の定式化に基づいた半自動アノテーション機能により，手作業でのアノテーションより格段に効率的に，かつ，視点を変えながら作業することで教師あり学習による自動ラベリングより格段に正確にアノテーションを行える．

3D 物体を対象とした対話的な領域分割の他の手法としては，文献 [7] のように 3D シーンを対象としたものが挙げられる．この手法が対象としているタスクは，意味的領域分割のように予め種類の決まった意味ラベルを割り当てるというものではなく，物体の種類は不定であるとして，物体ごとに領域を分割するというものである．この手法においても，マルコフ確率場の定式化による半自動領域分割が実現されている．一方，この手法における対話的操作である，1 つの領域を 2 つに分割する Split 操作や，逆に 2 つの隣接する領域を 1 つに併合する Merge 操作などは，意味的領域分割の場合，分割・併合される領域の意味ラベルを考慮しなければならないため，そのままでは適用できない．

3. 提案ツールの概要

図 1 に提案ツールの外観を，図 2 に提案ツールを用いた基本的なワークフローを示す．提案ツールの入力となる形状データは，テクスチャ付きの，分割されておらずひとつながりになっている三角形メッシュである．ユーザは意味ラベル選択ウィンドウ (図 1 右) で意味ラベルを選択し，メッシュ表示ウィンドウ (図 1 左) 上でブラシツール (図 1 左の円) を用いて各三角形に意味ラベルを割り当てる．各三角形に割り当てられる意味ラベルは 1 つのみである．ユーザはマウスドラッグで視点を変更しながら，意味ラベルを必要な箇所に粗く塗る．粗く意味ラベルを指定したのち，ユーザが自動ラベリング機能を実行すると，意味ラベルが未割り当ての三角形に対して意味ラベルが自動的に割り当てられる．この自動ラベリング機能によってメッシュ全体を手作業で塗る手間が省ける一方，意味ラベルの割り当てに誤りも含まれるため，ユーザが手作業で適宜修正する．この手作業の修正時に使える他の機能として，誤って意味ラベルを割り当てた部分を消す消しゴム機能，操作をキャンセルして直前の状態に戻す UNDO 機能，孤立した領域を指定した意味ラベルに一括で変更する塗り潰し機能，すでに意味ラベルが割り当てられた領域を保護するマスク機能などを利用できる．これらの機能および自動ラベリング機能を用いて，ユーザは正確な意味ラベルの割り当てが完了するまで作業を繰り返す．

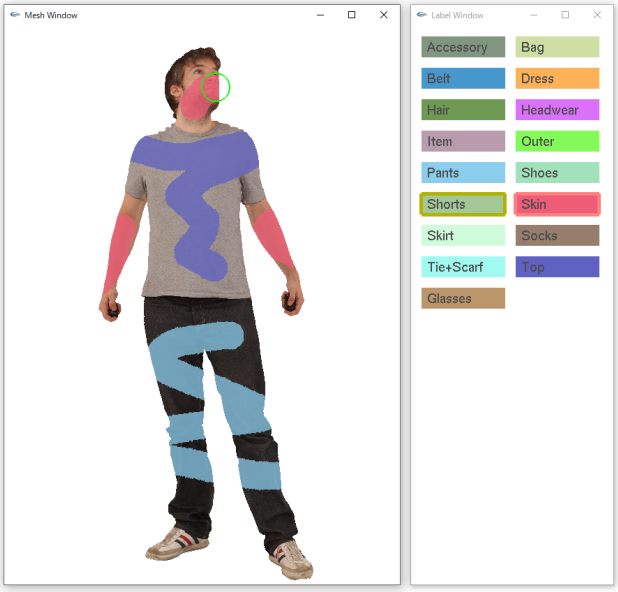


図 1 提案ツールの外観．メッシュ表示ウィンドウ (左) で人物 3D モデルを読み込み，ユーザはマウスドラッグで視点を変えながら，ブラシストロークで意味ラベルを指定する．意味ラベル選択ウィンドウ (右) で意味ラベルを選択できる．

以下，提案ツールの主な機能である自動ラベリング機能について説明する．

4. 自動ラベリング機能

自動ラベリング機能は，ユーザが指定した意味ラベルを教師データとし，マルコフ確率場の定式化に基づいた半教師あり多クラス分類として実現している．1 節で述べた通り，提案ツールでは 3 種類のアルゴリズムを実装している．以下，各アルゴリズムについて説明する．

4.1 色情報に基づくグラフカット

最初に説明するのは，各三角形が持つテクスチャの色に着目したアルゴリズムである．実行時の時点でユーザが指定済みの意味ラベルの集合 L から意味ラベルを 1 つずつ取り出し，各三角形に対し「注目する意味ラベルであれば前景 ($fore$)，そうでなければ背景 ($back$)」と見なして $\{fore, back\}$ の二値分類を反復する．各反復において，以下に示すエネルギー関数をグラフカットで最小化し，$fore$

ⓒ 2019 Information Processing Society of Japan

Vol.2019-CG-176 No.21
Vol.2019-DCC-23 No.21
Vol.2019-CVIM-219 No.21
2019/11/8

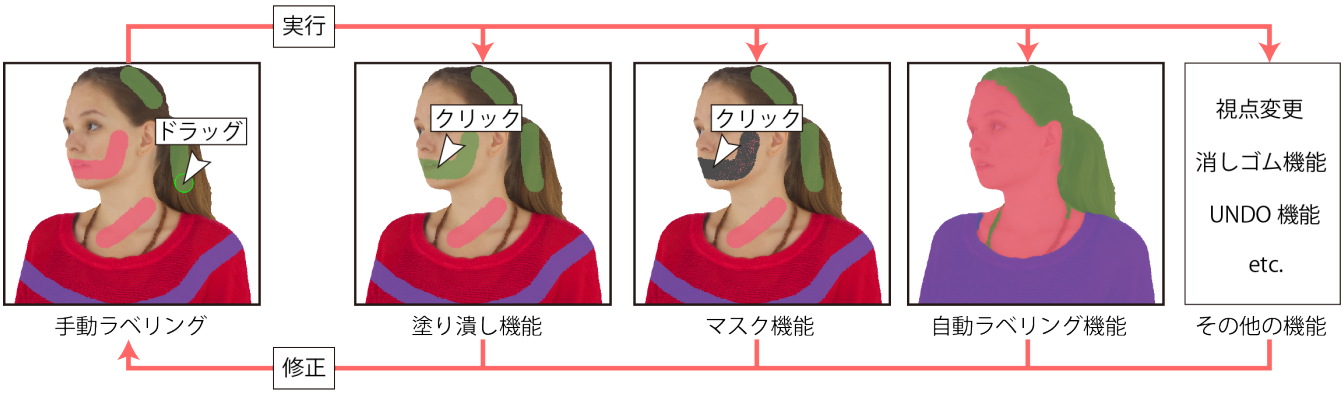


図 2 提案ツールを用いた基本的なワークフロー．ユーザはブラシストロークを用いて手作業で意味ラベルを指定した後，孤立した領域の意味ラベルを変更する塗り潰し機能，領域を保護するマスク機能，未割り当ての三角形に意味ラベルを自動伝搬する自動ラベリング機能，あるいはその他の機能を適用し，必要に応じて再び手作業で調整を行う．この手順を，意味的領域分割が完了するまで繰り返す．

と判定された各三角形に対し，貪欲的に注目する意味ラベルを割り当てる．

$$E = \sum_i \psi_i(l_i) + \lambda \sum_{i,j} [l_i \neq l_j]\, \phi_{ij}(l_i, l_j) \qquad (1)$$

ここで ψ_i は三角形 i のデータ項，ϕ_{ij} は互いに隣接する三角形 i, j についての平滑化項，λ は重み，$l_i, l_j \in \{0, 1\}$ は三角形 i, j が「前景 (*fore*) か否か (つまり注目する意味ラベルか否か)」を表す二値ラベルである．また $[\cdot]$ は指示関数で，引数が真のとき 1，そうでなければ 0 となる．データ項を定義するため，ユーザ指定のブラシストロークに含まれる各三角形内で，ランダムに K 個 (実験では $K = 5$) の点でテクスチャの色をサンプリングし，それらの色を用いてガウス混合モデル (GMM) を初期化した[*1]．三角形 i 内の k 番目のサンプル点での色から計算した尤度を p_{ik} とし，その平均値を $\overline{p}_i = \frac{1}{K}\sum_k p_{ik}$ とする．これを用いて，三角形 i のデータ項は $\psi_i = -\log \overline{p}_i$ とした[*2]．平滑化項 ϕ_{ij} は文献 [3] と同様に $\phi_{ij} = -\log \frac{\theta_{ij}}{\pi} \|\mathbf{g}_i - \mathbf{g}_j\|$ とした (ここで θ_{ij} は隣接する三角形 i, j の二面角を表し，$\mathbf{g}_i, \mathbf{g}_j$ はそれぞれの三角形の重心を表す)．また実験では $\lambda = 1$ とした．

上記の各反復において，孤立した面積の小さな意味ラベル領域が発生する．実際の意味的領域は一定以上の面積を持つため，面積に基づいてこれらの小さな意味的領域を削除する．具体的には，閾値より小さな面積を持つ領域を，隣接するより大きな意味的領域にマージする．実験では閾値として，三角形の平均面積の 50 倍を設定した．

4.2 三角形クラスタリングを用いたグラフカット

前述のグラフカットを用いたアルゴリズムの計算時間は，三角形の数が数万程度までは 10 秒前後で済むが，およそ 100 万を超えると対話的な処理として許容できないほどになる．グラフカットはグラフのノード数に応じて計算時間がかかるため，三角形をクラスタリングし，三角形クラスタをグラフのノードとしてグラフカットを適用することで高速化を図る．なお既存の対話的アノテーション手法 [7] においても，三角形のクラスタリングを行った後にグラフカットを適用していた．しかしこの手法では，1) ユーザ入力より先にクラスタリングが行われるため，ユーザは三角形クラスタより細かく意味領域を指定することができない，2) 本来は意味領域の境界となるべき部分をまたいで三角形クラスタが作られてしまうことがある，といった問題がある．これらの問題に対して提案手法では，1) ユーザ入力後にクラスタリングを行い，ブラシストロークが通過した三角形とそれ以外の三角形は必ず別のクラスタに分ける，2) ユーザのブラシストロークで指定された三角形から特徴量を抽出し，その特徴量に基づいてクラスタリングを行うことで，意味ラベル領域の境界となるべき部分にクラスタの境界ができるよう促す，といった改良を行う．

上記のうち 2) の特徴量の計算方法と，それを用いたクラスタリング方法について説明する．特徴量の計算には，4.1 節のグラフカットのデータ項の計算で用いた GMM を流用する．ユーザが指定した各意味ラベル $l \in L$ について，各三角形 i でラベル l に対する尤度の平均値 $\overline{p}_i^l \in [0, 1]$ を計算する．これらを $|L|$ 個の意味ラベルについて連結し，三角形 i の特徴ベクトル $\mathbf{f}_i = \left(\overline{p}_i^1, \overline{p}_i^2, \cdots, \overline{p}_i^l, \cdots\right)^T$ を得る．そして隣接する三角形 i, j に対し，クラスタリングのための類似度 d_{ij} を計算する．4.1 節で平滑化項の計算に用いた二面角 θ_{ij} を使って重み $w_{ij} = \left(1 + \frac{\theta_{ij}}{\pi}\right)^2$ を計算し，これを用いて類似度を $d_{ij} = w_{ij} \|\mathbf{f}_i - \mathbf{f}_j\|^2$ と計算する．そして類似度でソートし，類似度が小さいペア i, j から順にマージしてクラスタリングする．このとき制限を設けなければ巨大なクラスタが発生してしまうため，面積についての閾

[*1] GMM は文献 [8] のように計算が進むにつれて更新される．
[*2] 通常のグラフカットの定式化と同様に，データ項は *fore* および *back* の両方について計算する．

ⓒ 2019 Information Processing Society of Japan

値を設定した．実験ではメッシュの全表面積の 0.1%を閾値とした．以上のクラスタリングによって，例えば 120 万個の三角形が 1,000 個前後のクラスタに分割される．

三角形クラスタをノードとしてグラフカットを適用する場合に，データ項はクラスタ内でランダムに 1 つ選んだ三角形の尤度を用いる．平滑化項には，隣接する 2 つのクラスタの境界で隣接する，三角形ペアの二面角の平均と，2 つのクラスタの重心を用いて同様に計算する．

4.3 三角形クラスタに基づく多層パーセプトロン

前述の色情報に基づくグラフカットには特徴量選択に問題がある．もし髪，肌，上着，ズボンなどの意味ラベル領域がそれぞれ全く異なった色を持っていれば，色情報のみで正確に分割が可能である．しかし例えばスーツを着た人物の場合，上着とズボンの色は通常同じなので，色情報だけでは領域分割ができない．かといって色情報以外に例えば法線や座標などの情報を追加する場合，複数の情報のうちどれをどのくらい重視するのか，という重み付けに煩雑な試行錯誤が必要となる．

そこで本研究では，各三角形の色，法線，座標を入力として，各三角形の意味ラベルを多層パーセプトロン (MLP) を用いて推定する手法を実装した．グラフカットの場合と同様に，ユーザがブラシストロークを指定した三角形の特徴量を教師データとし，MLP を訓練した後，意味ラベルが未割り当ての三角形の意味ラベルを推定する．この方法では，過去に我々が画像中の画素の多クラス分類を行った [9] のと同様，ユーザがブラシストロークを入力してから学習を開始する．しかし三角形が多い場合，三角形単位では訓練・推定に時間がかかってしまうため，三角形クラスタ単位で学習・訓練を行う．具体的に入力に用いる特徴量は，クラスタ内でランダムに選んだ 1 つの三角形に含まれる，1) K 個の色サンプルの RGB 値，2) 法線ベクトル，3) クラスタの中心座標，からなる $3 \times (K + 2)$ 次元のベクトルである．

本研究で用いた MLP のネットワークアーキテクチャについて説明する．本研究では，全結合層，Batch Normalization [10]，Leaky ReLU [11]，Dropout (棄却率 0.5) を基本ブロックとした．本研究で検討したのは，1) この基本ブロックを複数個連結したもの，2) 基本ブロックの入力と出力をつないで Residual Block [12] として Residual Block を複数個連結したもの，また 3) DenseNet [13] のように，基本ブロックを複数個連結した後に各基本ブロックの入力と出力を密につないだもの，の 3 つである．計算時間との兼ね合いから，我々の実験では基本ブロックを 3 つ用いた DenseNet 方式のアーキテクチャを用いた．

5. 実験結果

提案ツールを C++および OpenGL を用いて実装した．MLP の実装には深層学習のライブラリとして LibTorch を用い，学習には Adam [14] による最適化を行い，学習および推定は GPU を用いて高速化した．実験には，CPU として Intel Core i7-8750H 2.20 GHz，GPU として NVIDIA RTX 2070 を搭載したノート PC を用いた．入力した人物 3D モデルは商用 Web サイトから購入したもので，各メッシュの三角形の数は少ないもので 5 万個程度，多いもので 120 万個超である．

4 節で述べた 3 つの自動ラベリング手法の大まかな計算時間について説明する．図 2 の 5 万三角形からなる人物 3D メッシュモデルに 3 種類の意味ラベルを割り当てた状態で，1) 色情報に基づくグラフカット，2) 三角形クラスタを用いたグラフカット，3) MLP を適用した場合の計算時間は，それぞれ 6.9 秒，2.0 秒 (うちクラスタリングが 1.0 秒でクラスタ数は 1,013 個)，10.6 秒 (うち学習が 8.8 秒) となった．

実際に 12 人のユーザに提案ツールを利用してもらい，フィードバックを得た．ユーザの典型的な利用手順としては以下の通りである．まず色の違いが明確な 2, 3 個の領域 (例: 肌，黒い上着と青いパンツなど) に対してブラシストロークを描いた後，自動ラベリングを実行する．肌の部分は比較的うまく分離されることが多いので，肌がうまく分離されたらマスク機能で保護しておく．マスクされた領域は，手動・自動のラベリングを問わず，影響を受けない．色の違いが明確な領域が他にあれば，同様に自動ラベリングを適用する．ある程度まで自動処理を適用したら，残りは，マウスドラッグで視点を近づけたり回転させたりしながら手作業でラベリングする．ユーザからは，特に領域の境界部分には細かい作業が要求されるため時間がかかる，とのコメントがあった．

実際にかかった作業時間は，テクスチャの色の複雑さやユーザの習熟度によって大きく変動した．例えば上下が同じ色のスーツの場合，自動ラベリングは現状ではあまり機能せず，手作業での指定が主となる．計 340 体にアノテーション作業を行ってもらった結果，1 体あたりの作業時間は，平均値 46 分，中央値 40 分，最小値 8 分，最大値 129 分となった．

6. まとめと今後の課題

本研究では，3D スキャンされた人物 3D モデルを対象とした意味的領域分割を，対話的かつ効率的に行えるアノテーションツールを提案した．提案ツールは，指定領域の塗り潰し機能や，すでに意味ラベルを指定済みの領域を保護するマスク機能などを備えている他，ユーザがブラシストロークで粗く指定した意味ラベルを，自動的に伝播する機能を備えている．この自動的なラベリングの実現方法として，現時点では 1) 色情報に基づくグラフカット，2) その三角形クラスタリングによる高速化，3) 色だけでなく法

ⓒ 2019 Information Processing Society of Japan

線や座標を考慮した多層パーセプトロン (MLP) による処理，の 3 通りを実装した．実験の結果，提案ツールを用いると 1 体あたり平均 46 分程度でアノテーションができることがわかった．

現時点での課題として，自動ラベリングの精度向上が挙げられる．色情報のみに基づくグラフカットは，領域ごとに色情報が明確に分かれている場合にのみ，高速かつ高精度に分割ができる．一方 MLP の場合，深層学習の能力の高さから，潜在的には高精度の分割が期待されるが，現時点では多数のブラシストロークを与えなければ適切な分割ができない．この改良には，例えば人物 3D モデルの大域的情報を含むような，より情報量の多い特徴量を考慮することが考えられる．また MLP の学習・推論の計算コストが高いため，事前学習を利用するなどして高速化することも課題として挙げられる．

その他の課題として，一般に三角形は意味ラベル領域の境界で綺麗に分割されているわけではなく，例えば服や肌の境界をまたぐように三角形が配置されていることもある．これは例えば，意味的領域分割を済ませた人物 3D モデルを投影して 2D の画像データセットを構築するなどの用途では目立たないので問題ないが，用途によっては問題になる可能性もある．理想的にはテクスチャなどの境界に沿ったリメッシングも必要になるとも考えられる．

謝辞　提案ツールを用いて多数の人物 3D モデルのアノテーション作業を行い，有益なフィードバックをくれた，筑波大学・計算幾何学とグラフィックス研究室のメンバーに深謝する．

参考文献

[1] Gong, K., Liang, X., Zhang, D., Shen, X. and Lin, L.: Look into Person: Self-Supervised Structure-Sensitive Learning and a New Benchmark for Human Parsing, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, pp. 6757–6765 (2017).

[2] Zheng, S., Yang, F., Kiapour, M. H. and Piramuthu, R.: ModaNet: A Large-scale Street Fashion Dataset with Polygon Annotations, *2018 ACM Multimedia Conference (MM 2018)*, pp. 1670–1678 (2018).

[3] Guo, K., Zou, D. and Chen, X.: 3D Mesh Labeling via Deep Convolutional Neural Networks, *ACM Trans. Graph.*, Vol. 35, No. 1, pp. 3:1–3:12 (2015).

[4] Kalogerakis, E., Averkiou, M., Maji, S. and Chaudhuri, S.: 3D Shape Segmentation with Projective Convolutional Networks, *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 6630–6639 (2017).

[5] Habermann, M., Xu, W., Zollhöfer, M., Pons-Moll, G. and Theobalt, C.: LiveCap: Real-Time Human Performance Capture From Monocular Video, *ACM Trans. Graph.*, Vol. 38, No. 2, pp. 14:1–14:17 (2019).

[6] 金森由博，遠藤結城，山口智史，三谷　純：多視点投影を利用した人物 3D モデルの意味的領域分割，*Visual Computing 2019* (2019).

[7] Nguyen, D. T., Hua, B., Yu, L. and Yeung, S.: A Robust 3D-2D Interactive Tool for Scene Segmentation and Annotation, *IEEE Trans. Vis. Comput. Graph.*, Vol. 24, No. 12, pp. 3005–3018 (2018).

[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] Endo, Y., Iizuka, S., Kanamori, Y. and Mitani, J.: DeepProp: Extracting Deep Features from a Single Image for Edit Propagation, *Comput. Graph. Forum*, Vol. 35, No. 2, pp. 189–201 (2016).

[10] Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456 (2015).

[11] Xu, B., Wang, N., Chen, T. and Li, M.: Empirical Evaluation of Rectified Activations in Convolutional Network, *ArXiv*, Vol. abs/1505.00853 (2015).

[12] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778 (2016).

[13] Huang, G., Liu, Z., v. d. Maaten, L. and Weinberger, K. Q.: Densely Connected Convolutional Networks, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269 (2017).

[14] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization (2014).

ⓒ 2019 Information Processing Society of Japan