

# 物体のドラッグによるインタラクティブな画像合成

## Interactive Image Composition through Draggable Objects

山口 祐一郎<sup>†</sup> 齋藤 拓也<sup>†</sup> 坂東 洋介<sup>‡</sup> 陳 炳宇<sup>\*</sup> 西田 友是<sup>†</sup>  
Yuichiro YAMAGUCHI<sup>†</sup>, Takuya SAITO<sup>†</sup>, Yosuke BANDO<sup>‡</sup>, Bing-Yu CHEN<sup>\*</sup>, and Tomoyuki NISHITA<sup>†</sup>

<sup>†</sup> 東京大学            <sup>†</sup> The University of Tokyo

<sup>‡</sup> (株) 東芝            <sup>‡</sup> TOSHIBA Corporation

<sup>\*</sup> 国立台湾大学    <sup>\*</sup> National Taiwan University

E-mail: <sup>†</sup>{yuichi, saito, ybando, nis}@nis-lab.is.s.u-tokyo.ac.jp <sup>\*</sup>robin@ntu.edu.tw

## 1 背景と目的

近年、デジタルな画像処理はテレビ番組、映画、医療などの分野で広く扱われている。また、ネットワーク環境の発達やデジタルカメラの普及により、一般の人々のなかにも画像処理、特に画像合成の需要は高まってきている。しかしながら、大部分の多機能画像処理ソフトウェアは物体抽出やレイヤー分割などの初心者にとっては難しい処理を持ち、自分の望みどおりに画像を合成できない原因となっている。その一方で、従来の単純な画像合成法では、切り出した物体を別の画像の上に貼り付けることしかできず、別の画像の中で物体の後ろに隠れるような合成を行うときには切り出した物体の重なる部分を逐一削る必要がある。

我々はこの問題点を解決するために、マウสดラッグを用いたユーザインターフェースと自動的な前景物体抽出を提案する。マウสดラッグを用いることで提案法ではユーザの指示に応じてインタラクティブに画像が合成できる。また、物体を別の物体の背景に引きずり込むという直感的なインターフェースを提供できる。ユーザがまず切り出した物体（ソース物体）をターゲット画像中に配置し、それを貼り付けようとする場所にドラッグする間に、我々の手法はいずれソース物体によって覆われる部分から色分布を見積もり自動的に前景物体を検出する。この処理によって前計算なしにソース物体を自動的に検出した物体の後ろに合成することが可能になる。

## 2 関連研究

画像処理や画像合成には様々な手法が存在する。以下で、我々の研究に関係するいくつかの論文について紹介する。まず、画像の分割、マット抽出について、次に二枚の画像を継ぎ目なく自然に合成する研究について、最後にインタラクティブな画像処理に関する関連研究についてそれぞれ述べる。

### 2.1 画像分割

画像の分割は画像処理に際して非常に重要な処理であり、様々な手法が提案されている。Boykov らによって提案された Graph Cut による画像の領域分割法では画素値を元に作成した無向グラフの最小カットを求めることで前景背景を判断する [1]。この手法は Rother らの GrabCut でさらに改良され、物体を長方形で囲むだけというより少ないユーザアクションでの画像分割が可能になった [10]。一方で前景背景の混合率まで計算するマット抽出はさらに様々な手段によって行われている。Chuang らによる Bayesian Matting ではベイズ理論を用いてマット抽出を行っている [2]。Sun らの Poisson Matting ではポアソン方程式を用いることでマット抽出を行っている [12]。Levin らによって提案された Closed form matting では疎行列線形方程式によりコスト関数を最小化することで、Wang らの Robust matting ではランダムウォーク理論を用いたエネルギー関数の最小化でそれぞれ実現している [4][14]。これらの手法は画像からの前景物体の抽出が目的であり、抽出した素材を使用した合成法については言及していない。我々の手法ではこれらのアルゴリズムを前景部分と背景部分の自動的な分割のために使用し、事前にターゲット画像を領域分割することなく前景物体の後ろに回りこんだ合成画像を作成する手法を提案する。

### 2.2 シームレスな画像合成

複数の画像をつなぎ合わせる際に、その明るさや色味を調整しておくことは自然な画像合成を行うのに必要な処理であり、これを自動的に行う手法が複数提案されている。Poisson Image Editing ではポアソン方程式の解を元に色を変化させ、継ぎ目を目立たなくさせる [9]。Seamless image stitching は継ぎ目に対してコスト関数を用意し、それを最適化することで複数の画像を縫い合わせることができる [5]。Drag-and-Drop Pasting では Poisson Image Editing に新しい目的関数を用意して境界の最適化を図り、より実用的な手法としている [3]。これらの手法では各画素の画素値を元に画像を変化させる。同様の処理によって提案法

でも合成した物体と背景部分との色あわせが可能になると考えられる。

### 2.3 インタラクティブな画像処理

インタラクティブな処理はユーザアクションに対して即座に反応でき、それに応じて処理結果を修正、改善できる。Intelligent Scissors ではユーザのシード点の指示に応じて二次元グラフの探索問題を動的計画法を用いて解くことで画像から物体を正確に抽出できる [8]。Lazy Snapping では前景、背景を表す線をユーザが画像中に引くことで Graph Cut の手法によって画像から物体をインタラクティブに切り出すことが可能である。[6]。Lischinski らの手法は単純なブラシとスライダーだけで、領域の色や明るさをインタラクティブに変化させる手法を提案した [7]。Soft Scissors はユーザが物体の境界線をなぞっていくことで、リアルタイムなマット画像生成を可能にしている [13]。これらの手法は画像からの物体の抽出、マット画像の作成を目的とするものであり、直接画像を合成するものではない。我々の手法では貼り付ける物体を切り出す以外に合成に対する前計算を行わないため、物体をドラッグして別の物体の後ろに隠すという直観的なユーザアクションを用いたインタラクティブな合成を可能にする。

## 3 インタラクティブな画像合成法

### 3.1 概要

図 1 に提案法を用いた画像合成の概要を示す。ソース物体は従来法により既にソース画像から切り出されているものとする。我々は GrabCut による切り出しと、マット画像を用いた切り出しを実装した。ユーザはソース物体をターゲット画像の任意の場所に貼り付け、ドラッグする (図 1 左)。すると、本手法は自動的にドラッグ先にある前景物体を検出し、ソース物体を後ろに回りこんだように合成する (図 1 右)。提案法では前景物体を抽出するために、ユーザがドラッグする先のターゲット画像の色分布を取得し、そこで得た前景物体と背景部分の色サンプルに基づいて、移動したソース物体が新しく覆う領域の各画素が前景物体と背景部分のどちらに属するかを決定する。

以下でその方法について述べる。

### 3.2 色サンプルによる前景物体検出

提案法ではまず、切り出したソース物体をユーザがドラッグする際に、その方向について色分布を調べ、前景物体の位置を推測する。図 2 において、赤い領域がソース物体の位置、黄色い領域がユーザのドラッグにより移動したソ-

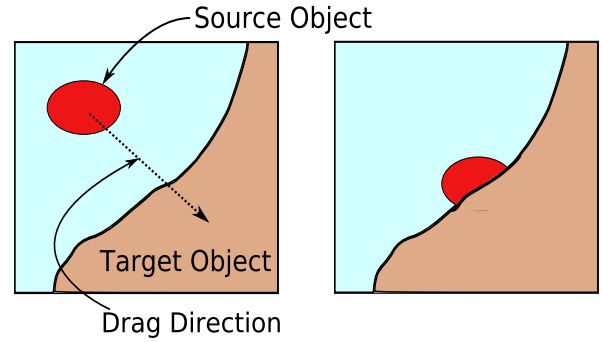


図 1: 提案手法の概要図

ス物体が次の時点で新しく覆う部分 (新規被覆領域と呼ぶ) である。提案法は、ソース物体のドラッグ方向に対し、それに垂直な境界候補線 (図 2 の赤線、ソース物体からの距離を  $t$  とする) と、その両側に色サンプルを取る二つの領域  $M_t, N_t$  を用意する (図 2 の青と緑の線で囲まれた領域)。領域のサイズはドラッグ方向に沿った長さを  $l_t$ 、境界候補線に沿った幅を  $w_t$  とする。具体的な値は後に述べる。

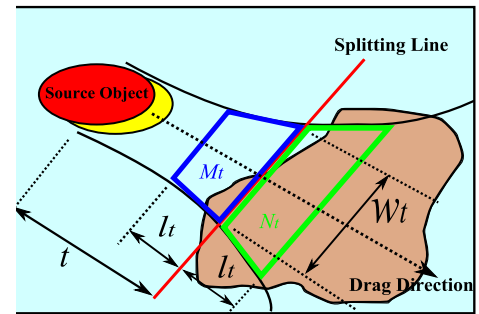


図 2: 色サンプルを取る領域

境界候補線をソース物体の進行方向にそって移動させながら (すなわち  $t$  を変化させながら) 領域  $M_t, N_t$  の中から色サンプル  $P_t, Q_t$  をそれぞれ取り、次の式 (1) によってサンプル間の色の距離を推定する。

$$Dist(P_t, Q_t) = \lambda_t \frac{1}{|P_t||Q_t|} \sum_{m \in P_t} \sum_{n \in Q_t} (Z_m - Z_n)^2 \quad (1)$$

ただし  $\lambda_t$  は境界線の位置に応じた色の距離に対する重み、 $Z_m, Z_n$  は画素  $m, n$  の画素値とする。画素値については RGB 色空間、CIEL\*a\*b\* 色空間それぞれで実験したが、目立った違いが見られなかったため、我々は RGB 空間での画素値を使用する。距離  $Dist(P_t, Q_t)$  は領域の色分布が異なるほど大きくなるので、計算した距離が最大になる際の境界候補線の位置  $t$  が前景部分と背景部分の境界となり、ソース物体の手前側のサンプル  $P_t$  を背景部分の色サンプル、奥側のサンプル領域  $Q_t$  を前景部分の色サンプルと見なすことができる。また、距離が最大となる時の境界候補

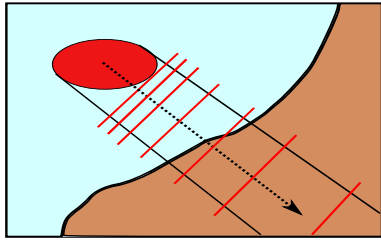


図 3: 計算する境界線の間隔の変更

線の位置  $t$  から，前景物体がソース物体の位置からどれぐ  
らいの距離にあるのかを推定できる．提案法ではこのよう  
にして得た前景物体，背景の色サンプルをもとに Graph  
Cut の手法を用いて新規被覆領域を分割する．

提案法では重み  $\lambda_t$ ，領域のサイズ  $l_t, w_t$  を以下のように  
設定した．

$$\lambda_t = \exp(-c_\lambda t) \quad (2)$$

$$l_t = L \exp(c_l t) \quad (3)$$

$$w_t = W \exp(c_w t) \quad (4)$$

ここで  $L, W$  はソース物体のドラッグ方向に沿った長さ  
とそれに直交する幅， $c_\lambda, c_l, c_w$  はユーザ指定の定数とする．  
我々は実験的にそれぞれ  $c_\lambda = 0.1, c_l = 0.01, c_w = 0.01$  と  
した．式 (2) はソース物体から離れるほど色分布の信頼性  
が低いとして色分布間の距離を減衰させることを意味する．  
最も色が異なっている部分を境界線として検出したい場合  
には，重み  $\lambda_t$  を  $1 (c_\lambda = 0)$  として境界線の位置に関係なく  
色の距離が最大となる場所を検出することができるが，一  
般に，境界線として検出される可能性のある部分が複数並  
ぶようなターゲット画像の場合には， $c_\lambda > 0$  とすることで  
手前にある境界線から順に検出することが可能になる．

式 (3),(4) はソース物体から離れるほどドラッグの先読  
みの信頼性が低いとして領域のサイズを拡大することを意  
味する．ユーザのドラッグの方向が急に変わった場合にも  
妥当な色サンプルを取得するために，境界線の位置が遠く  
なるほどサンプルを広い部分から取ることができる．

さらに，図 3 に示すように，ソース物体からの距離が遠  
い部分に関しては信頼性に乏しいとして色サンプルの距離  
を計算する間隔を距離に応じて広げ，計算時間を減少させ  
ることもできる．

### 3.3 色サンプルの取得方法の変更

3.2 節ではドラッグされているソース物体が前景物体に  
近づくまでのサンプル取得方法について述べた．ソース物  
体が前景物体に近づくまでは，ドラッグの方向について色  
分布を調べ，そこで得た色サンプルを元に新規被覆領域を  
Graph Cut の手法を用いて前景部分と背景部分に分割す

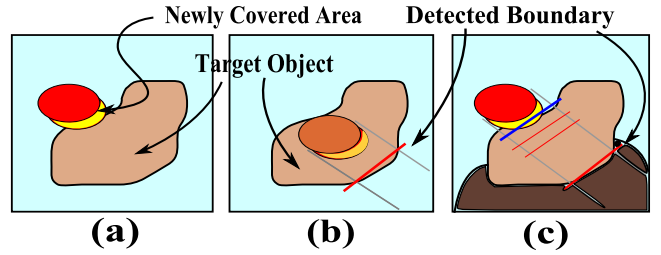


図 4: 色サンプルの取得方法の変更

る．ところがこの方法では既にソース物体が前景物体と背  
景の境界線にさしかかっている場合や，既にソース物体が  
前景物体の背後に回りこんでいる場合に対応できない．そ  
こでここではソース物体が十分に前景物体に近づいた場合  
や，既に重なり始めた場合の，色サンプルの取得方法の変  
更について述べる．

3.2 節で説明したとおり，検出した境界線の位置からそ  
の時点でのソース物体と前景物体の距離が推測できる．そ  
こで十分にソース物体と前景物体に近い場合は，ドラッグ  
方向への色調査を一度取りやめ，新規被覆領域全てを前景  
サンプル，ソース物体が重なっていた領域を背景サンプル  
として領域分割を行う (図 4(a))．ここで赤い領域が背景  
サンプルを取得する領域，黄色の領域が前景サンプルを取  
得する領域である．この変更によって，既に境界線にさし  
かかっている場合に対応できる．次に，ソース物体が完全  
に前景物体の背後に回りこんでいる場合には，次にソー  
ス物体が現れ始める境界線を検出するために，3.2 節と同様  
にドラッグの方向について色分布を調べていき，そこで得  
られた色サンプルについて，手前側を前景物体の色サンプ  
ル，逆に奥側を背景物体の色サンプルとして新規被覆領域  
を分割する (図 4(b))．最後に，検出した境界線に対して  
ユーザが納得しなかった場合には，提案法はユーザの指示  
に応じて検出していた境界線を破棄し，その境界線より先  
の領域についてのみ改めて色分布の調査を行う (図 4(c))．  
ここでソース物体に近い太線が検出していた境界線であり，  
この検出がユーザの意図と反していた場合には指示に  
応じて改めてその先の太線を境界線として検出しなおして  
いる．

### 3.4 検出物体の変更

提案法では前景物体の検出を誤った場合やユーザが前景  
物体を変更した場合に，それを指示することによって結果  
画像をインタラクティブに変更することが可能である．こ  
れについて説明する．ユーザが境界の検出結果に満足い  
かなかった場合，それを指示する．例えば部分的に前景物  
体の後ろに隠れているソース物体に対し，現在手前に来て  
いる物体をソース物体の後ろにまわしたい場合には，ユーザ

はボタンによってそれを指示し，それに応じて提案法ではソース物体が重なっている領域の分割結果を訂正する．これによって，検出している物体に対する分割結果を修正し，新しく別の前景物体を検出することができる．

### 3.5 動画への応用

提案法を用いて，動画の中で動いている物体に対しても同様にその背景に合成することが可能になる．

提案法では，ユーザによるドラッグの履歴を保存，再現する機能によって自動的に動画中の動く物体の背後にソース物体を合成することを実現する．ただし，動画にはシーンの変更は含まれないものとし，固定または緩やかに移動する視点の下で前景物体が動く動画を対象とする．ユーザはそのような動画のうちの一枚のフレームに対してソース物体をドラッグし，画像を合成する．システムはユーザのドラッグ操作の履歴を記録しておき，これを用いて全てのフレームの画像との合成を自動的に行う．このとき提案法はドラッグの軌跡にある前景物体を自動的に検出してその背後にソース物体を回りこませるので，各画像で物体の位置を指定することなく自動的にその時の前景物体の位置に応じてその背後に回りこむ合成が可能となる．また，各画像に対して履歴の全てを用いて合成した場合には最終的なソース物体の位置は共通であり，動いている前景物体の後ろに動かないソース物体を合成できる．また，履歴の再現をフレーム番号に比例して途中で止めることによって，ソース物体がユーザのドラッグの軌跡に沿って移動するような動画を合成することも可能である．

## 4 結果

### 4.1 従来法との比較

まず，従来法 [11] と結果画像を比較する．



図 5: 従来法 (左) と提案法 (右) による合成結果

ソース物体として飛行機を，前景物体として桜を用意している．従来法 [11] ではユーザの動きに応じた前景物体の変更や前景物体を通過した後の別の物体の検出を行っていないため，図 5 のように入り組んだ画像の中でソース物体

をドラッグしていくと，前景物体を変更できずに前景背景のサンプルを取り違える可能性がある．一方，提案法ではこういった処理を行っているため，入り組んだ画像の中をドラッグしても前景物体を変更しながらその背後に合成することが可能となっている．

### 4.2 結果画像

次に，提案法による合成結果を示す．我々はソース画像として鳥を，ターゲット画像として夕焼けの画像を用意した (図 6) ．

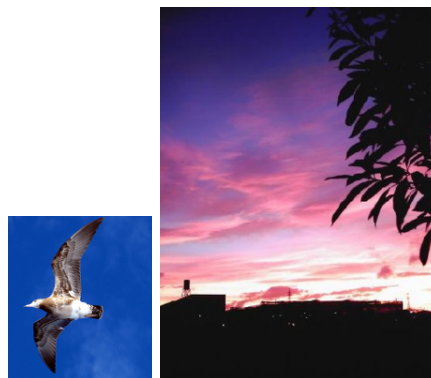


図 6: ソース画像 (左) とターゲット画像 (右)

ソース画像から切り出した鳥をターゲット画像の中心に置き，そこから木の陰に向かってドラッグする (図 7 白矢印) ．

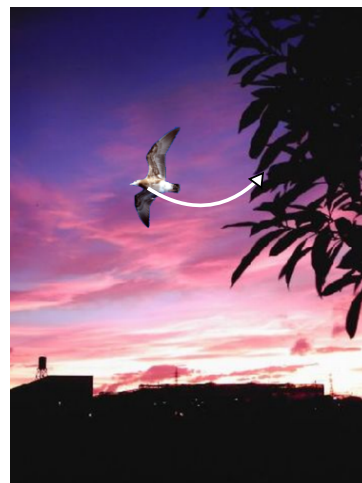


図 7: ソース物体のドラッグ軌跡

このとき提案法では自動的に木の陰が前景物体として抽出され，ソース物体が自動的にその後ろにまわるように合成される (図 8) ．

さらに，合成して得られた結果画像をターゲット画像として合成を繰り返すことで，木の陰に隠れる鳥だけでなく，



図 8: 合成結果

互いに重なり合う鳥についても同様に合成が可能になる (図 9) .



図 9: 複数の鳥の合成結果

次に、熱帯魚をサンゴ礁と別の熱帯魚の画像の中に合成する (図 10) .

まず、左端に置いたソース物体を右に向けてドラッグすることで左下のサンゴ礁を前景物体として検出させ、その背後に合成した結果を図 11 に示す .

この時点で背景のサンプルとして海の青が用意され、サンゴ礁の赤褐色と熱帯魚の黄色がともに前景物体の色として検出されている . ここでユーザからサンゴ礁をソース物体の後ろに移動させるように指示を出し、改めてドラッグすると、熱帯魚のみを前景物体と抽出して、図 12 で示すように熱帯魚とサンゴ礁の間に挟む形の合成も可能になる .

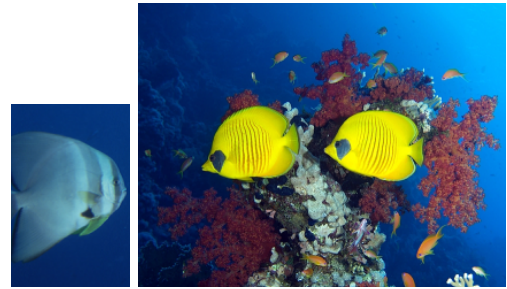


図 10: ソース画像 (左) とターゲット画像 (右)



図 11: サンゴ礁の後ろに合成した結果

#### 4.3 動画への適用例

動画に対して提案法を用いて合成を行った例から、いくつかのフレームを例として以下に示す . 鯉のぼりがはためく動画をフレーム毎に分割し、それぞれに対して履歴を用いて自動的に鳥をその背後に合成した . 動画を分割した際のフレームの通し番号に応じて再現する履歴の量を変化させることで、鯉のぼりの陰に隠れながら左下に鳥が動いている動画が合成できる .

### 5 結論と今後の課題

我々はこの論文で、色分布の先読みと自動的な前景物体検出に基づくインタラクティブな画像合成の手法について述べた . 我々の手法はユーザのドラッグの方向への色分布の取得から前景物体とその境界線を抽出し、そこで得た色サンプルを用いて領域分割を繰り返すことで、従来法では逐一重なる部分を削らなければならなかった重なりあう物体の合成を半自動的に扱うことができる . これにより、ドラッグした物体が別の物体に隠れるという直感的で分かりやすいインターフェースを実現している . 加えて、提案法では前計算を行わずにドラッグの方向のみから重なり合う物体の画像合成を行うため、ユーザはインタラクティブに合成結果や検出された前景物体を変更することができる .



図 12: サンゴ礁と熱帯魚の間に合成した結果



図 13: 鳥を鯉のぼりの奥に合成した例

しかしながら，提案法には依然としていくつかの問題点が残っている．第一に，提案法では色分布の情報のみから前景物体を検出するため，背景と非常に色分布の似通っている物体を検出することができない可能性がある．また，シマウマなど前景物体自身が色分布に非常に大きな断絶を持っている場合，その断絶を境界線として誤検出してしまう可能性がある．また，提案法では領域全体から取得した色サンプルを元に前景物体の位置を推測するため，サンプルを取得する領域の大きさに比べて前景物体が小さく，計算される距離に前景物体からの色サンプルがあまり反映されなくなってしまう場合，前景物体の検出に失敗する．今後は色分布のサンプルを集める方法やその距離を計算する方法について，これらの問題点を解決できるような改善を試みたい．

## 参考文献

- [1] Yuri Y. Boykov, and Marie-Pierre Jolly. “Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images.” In ICCV2001, 105-112, 2001.
- [2] Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski. “A bayesian approach to digital matting.” Proc. CVPR, volume 2, 264-271, 2001.
- [3] Jiaya Jia, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shumand. “Drag-and-drop pasting.” Transactions on Graphics, 25(3):631-637, 2006.
- [4] Anat Levin, Dani Lischinski, and Yair Weiss. “A closed form solution to natural image matting.” Proc. CVPR, 61-68, 2006.
- [5] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. “Seamless Image Stitching in the Gradient Domain.” in European Conf. on Computer Vision, 2004, 377-389, 2004.
- [6] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. “Lazy snapping.” ACM Trans. Graph., 23(3): 303-308, 2004.
- [7] Dani Lischinski, Zeev Farbman, Matt Uyttendaele, and Richard Szeliski. “Interactive local adjustment of tonal values.” In Proceedings of ACM SIGGRAPH, 646-653, 2006.
- [8] Eric N. Mortensen, and William A. Barrett. “Intelligent Scissors for Image Composition.” In Proceedings of ACM SIGGRAPH, 191-198, 1995.
- [9] Patrick Perez, Michel Gangnet, and Andrew Blake. “Poisson image editing.” Transactions on Graphics, 22(3): 313-318, 2003.
- [10] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. “Grabcut: Interactive foreground extraction using iterated graph cuts.” Transactions on Graphics, 23(3): 309-314, 2004.
- [11] Takuya Saito, Yosuke Bando, and Tomoyuki Nishita. “Interactive Composition of Seamless and Color-matched Images of Overlapping Objects,” The Journal of the IIEEJ, Vol.37, No.3, 278-285, 2008.
- [12] Jian Sun, Jiaya Jia, Chi-Keung Tang, and Heung-Yeung Shum. “Poisson matting.” In SIGGRAPH '04: ACM SIGGRAPH 2004 Papers, 315-321, 2004.
- [13] Jue Wang, Maneesh Agrawala, and Michael F. Cohen. “Soft scissors: an interactive tool for realtime high quality matting.” In SIGGRAPH '07: ACM SIGGRAPH 2007 papers, 9, 2007.
- [14] Jue Wang, Michael F. Cohen. “Optimized color sampling for robust matting,” Proc. CVPR, 1-8, 2007.