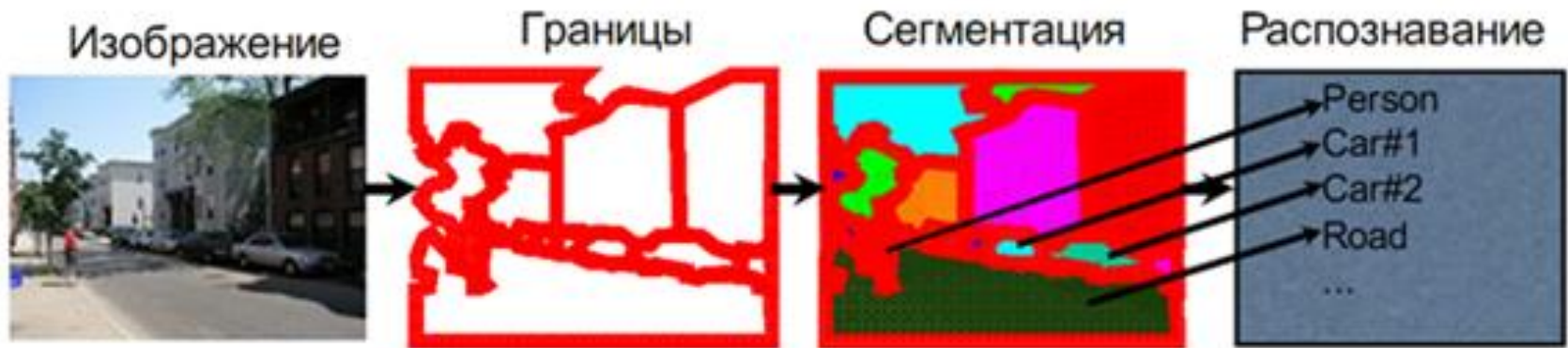
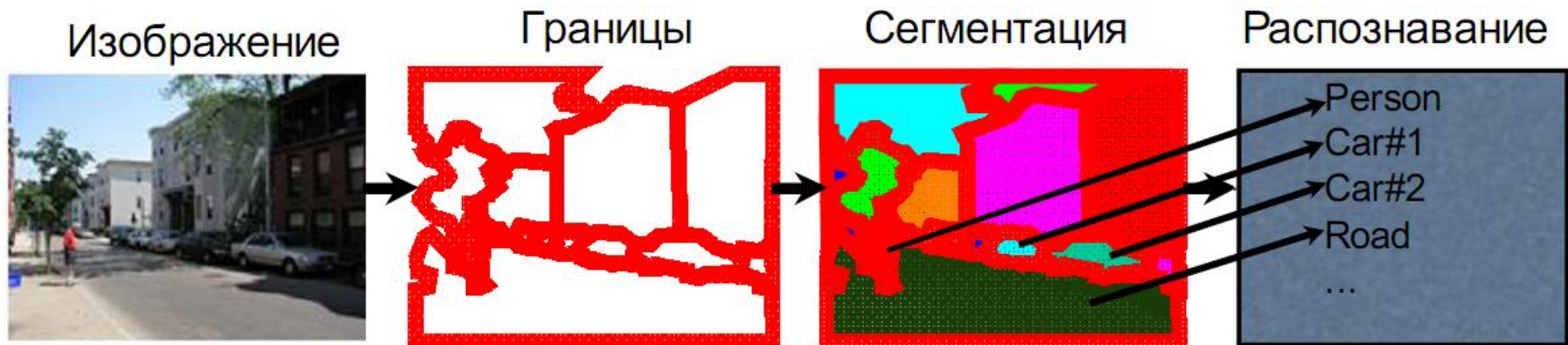


# Интерактивная сегментация изображений

Лекция 7

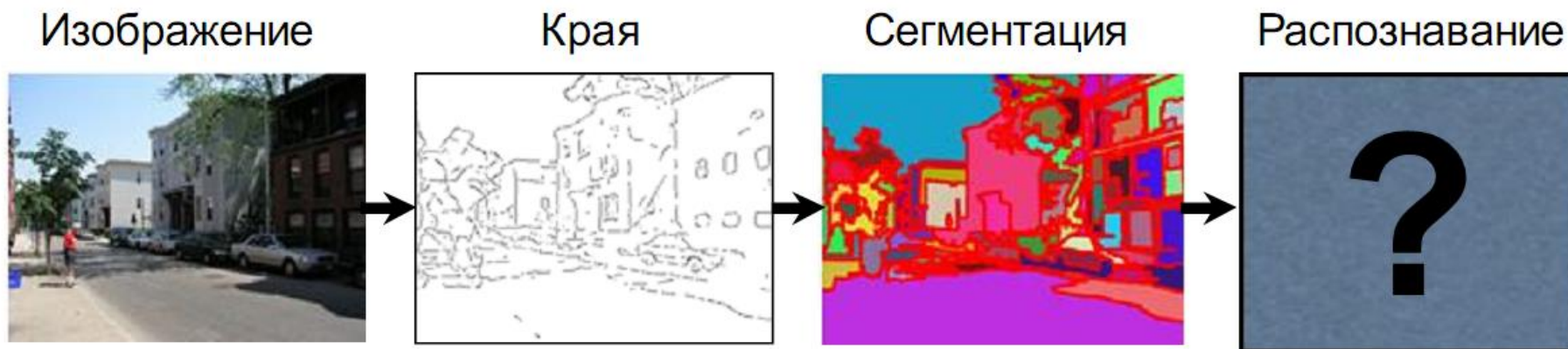
# План





Теория ↑

Практика ↓



# Сколько сегментов и какие?



# Или такие? Грубая сегментация

Небо

Строения

Автомобиль

Автомобиль

# Определение сегментации

- «Жесткая» сегментация
  - Разбиение изображения на неперекрывающиеся области, однородные по некоторому признаку и покрывающие всё изображение
- Формально:
  - Разбиение изображения на набор областей

$$S = \{S_i\}, i = \overline{1, N}$$

$$I = \bigcup_{i=1..N} S_i$$

$$\forall i, j = \overline{1, N} : i \neq j \quad S_i \cap S_j = \emptyset$$

$$\forall i = \overline{1, N}, P(S_i) = \text{истина}$$

$$\forall i, j = \overline{1, N} : i \neq j \quad P(S_i \cup S_j) = \text{ложь}$$

# Критерии «похожести» пикселей

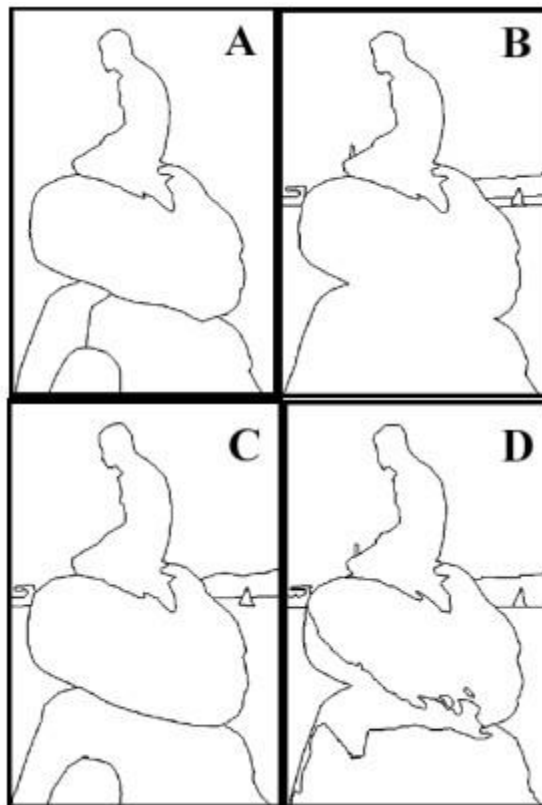
- По расстоянию
- По яркости
- По цвету
- По текстуре

# Сегментация по цвету





# Варианты сегментации изображения. Какой верный?



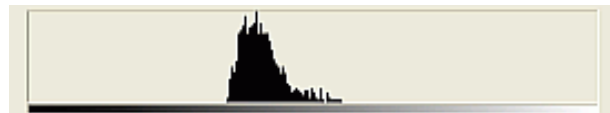
Berkeley Segmentation Dataset

# Оценка качества работы методов сегментации

- Целостность и однородность по некоторому признаку
- Отличие признака для смежных областей
- Отсутствие мелких отверстий внутри
- Гладкие границы

# Критерий однородности

- Гистограмма содержит не больше 1 значительного пика



- Отклонение любого пикселя от средней яркости  $< T_{avg}$

$$\forall p \in S \quad \left| I(p) - \frac{1}{N} \sum_{q \in S} I(q) \right| < T_{avg}$$

- Разница между соседними пикселями  $< T_{diff}$

$$\forall p \in S, \forall q \in N(p) \quad |I(p) - I(q)| < T_{diff}$$

- «Слабая» граница между регионами (только для слияния)

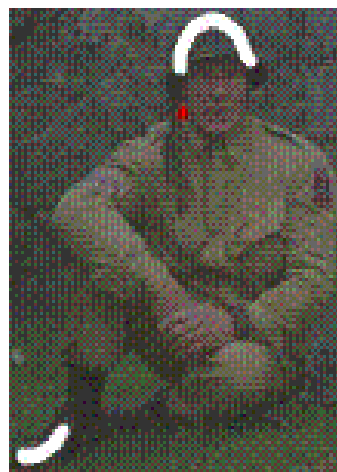
# Два типа сегментации

- Автоматическая
  - Сегментация производимая без взаимодействия с пользователем
    - Картинка на входе, регионы на выходе
- Интерактивная
  - Сегментация, управляемая пользователем, допускающая и/или требующая ввода дополнительной информации
    - Пример – «волшебная палочка» в Photoshop

# Интерактивная сегментация – постановка задачи

- Рассматривается задача разбиения на 2 области - объект и фон
- На вход алгоритм получает:
  - исходное изображение
  - дополнительную информацию от пользователя:
    - ограничение на то, что некоторые конкретные пиксели обязательно должны принадлежать объекту (фону)
    - ограничивающий прямоугольник вокруг объекта
    - примерную границу объекта
- В процессе работы алгоритма пользователь может уточнять или дополнять входные данные.

# Пример интерактивной сегментации изображения



Одна из основных проблем - отсутствие единой метрики качества  
Субъективное сравнение результатов  
Разный пользовательский ввод (объём)

# Тестирование методов на общей базе изображений

Например, Berkeley Segmentation Dataset насчитывает более 1000 изображений, отсегментированных вручную 30 разными людьми.

<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench>

# Алгоритмы интерактивной сегментации

## Время до 2001

- **Magic wand**
- Intelligent scissors
- Intelligent Paint

## После 2001

- Interactive graph cuts
- Lazy Snapping
- GrabCut
- Progressive Cut
- GrowCut



# Magic Wand (<волшебная палочка>)

- ◆ Идея:

- Пользователь задает пиксель, запоминается его цвет  $C_1$
- Все пиксели, цвет которых  $\|C - C_1\| < \Theta$  - относятся к выделяемому региону



# Эффективность Magic Wand



8 кликов мыши,



более 20 кликов,  
причем некоторые части  
оленя остались  
неотсегментированными

# Алгоритмы интерактивной сегментации

Время до 2001

- Magic wand
- **Intelligent scissors**
- Intelligent Paint

После 2001

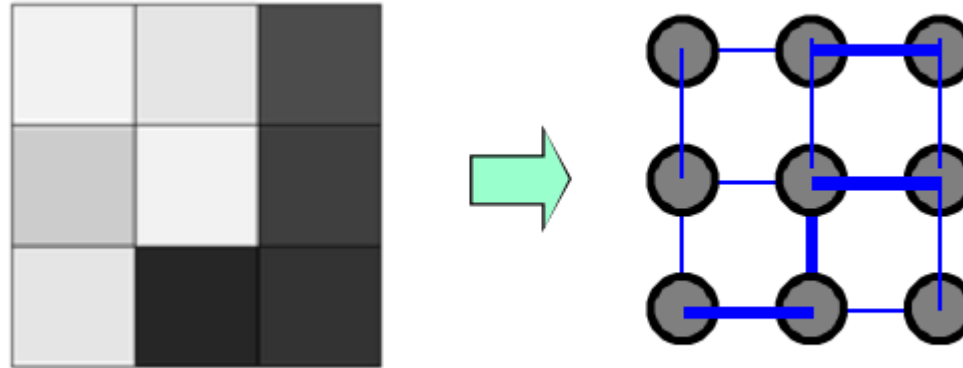
- Interactive graph cuts
- Lazy Snapping
- GrabCut
- Progressive Cut
- GrowCut

# Intelligent Scissors (Умные ножницы) [3] 1996



Eric N. Mortensen, William A. Barrett, Intelligent scissors for image composition, Siggraph 1995

# Принцип действия Intelligent Scissors



1. всё изображение трактуется как взвешенный неориентированный граф, каждая вершина которого соответствует пикселю изображения
2. вершины, соответствующие соседним пикселям связываются ребрами
3. на ребрах данного графа определяется весовая функция
4. значение этой функции мало на ребрах, соответствующих потенциальной границе на изображении.

# Что учитывает Intelligent Scissors

$$l(p, q) = w_z \cdot f_z(q) + w_G \cdot f_G(q) + w_D \cdot f_D(p, q)$$

$f_z(q)$  учитывает локальные максимумы градиента

$f_G(q)$  учитывает силу градиента (чем он больше, тем  $f$  меньше),

$f_D(p, q)$  стимулирует более гладкие границы

# Действия пользователя для Intelligent Scissors



- Пользователь указывает 2 точки на границе объекта.
- Граница объекта находится с помощью алгоритма нахождения пути минимальной стоимости в графе
- Алгоритм Дейкстры на основе динамического программирования

# Плюсы и минусы Intelligent Scissors

- (+) Существуют очень быстрые реализации алгоритма
- (--) Плохо работает в сильно текстурированных областях (существует множество альтернативных минимальных путей) .



# Модификация Intelligent Scissors

- Использование предварительной пересегментации [4]
- Или использование суперпикселей

# Алгоритмы интерактивной сегментации

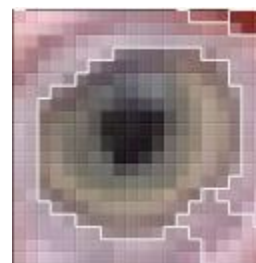
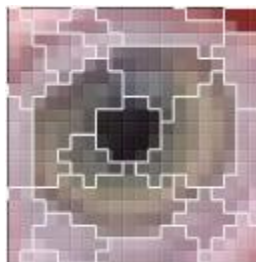
Время до 2001

- Magic wand
- Intelligent scissors
- **Intelligent Paint**

После 2001

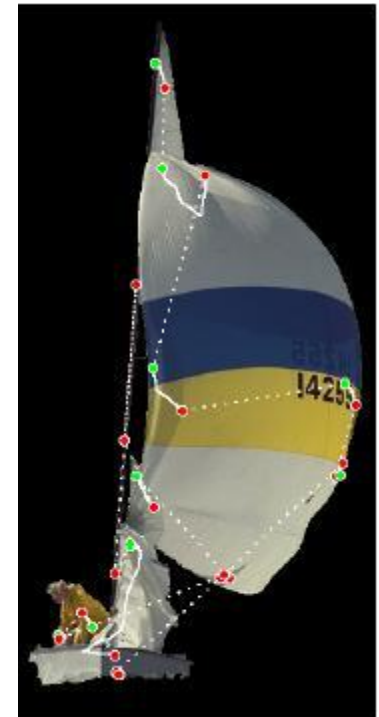
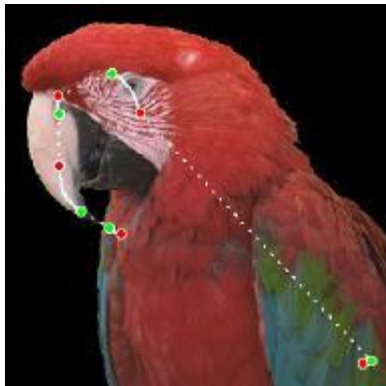
- Interactive graph cuts
- Lazy Snapping
- GrabCut
- Progressive Cut
- GrowCut

# Intelligent Paint [5]



- ◆ Идея:
  - Предварительная иерархическая сегментация изображения
  - Анализ «мазков» пользователя для объединения регионов предварительной сегментации

# Результаты работы Intelligent Paint



# Коммерческие решения (не рассматриваем медицину)

- Magic Wand  
(цветовая статистика)



- Magnetic Lasso  
(Live Wire, Intelligent scissors)



# Алгоритмы интерактивной сегментации

Время до 2001

- Magic wand
- Intelligent scissors
- Intelligent Paint

После 2001

- **Interactive graph cuts**
- Lazy Snapping
- GrabCut
- Progressive Cut
- GrowCut

# Прорыв в данной области

- произошёл в 2000 г.
- с изобретением Юрием Бойковым и Мари-Пьер Джолли алгоритма GraphCut
- GraphCut стал де-факто эталонным алгоритмом интерактивной сегментации

# Действия пользователя GraphCut

- Пользователь указывает жесткие ограничения на сегментацию – пиксели, принадлежащие объекту (O) и пиксели, принадлежащие фону (B)
  - семена объекта и фона



Yuri Boykov and Marie-Pierre Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D images. In International Conference on Computer Vision, (ICCV), vol. I, pp. 105-112, 2001.



# GraphCut [6]

- всё изображение – граф
- вершины-пиксели + 2 терминальные вершины (исток и сток)
- вес ребра

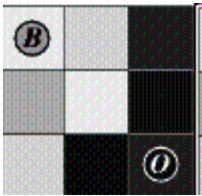
$$B_{p,q} = \exp\left(-\frac{\|C_p - C_q\|^2}{2\sigma^2}\right) * \frac{1}{dist(p,q)}$$

- Формулировка сегментации как минимизации энергии через разрез графа

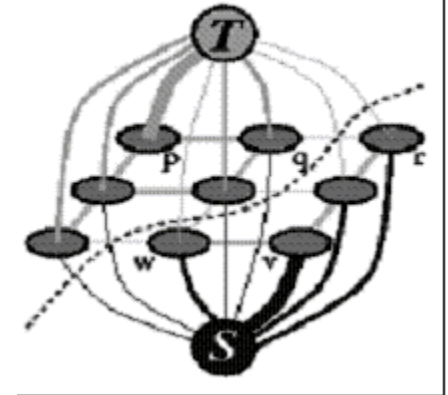
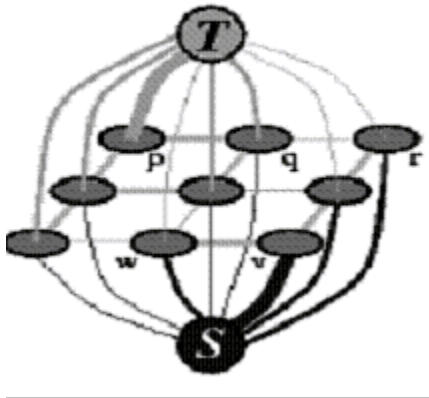
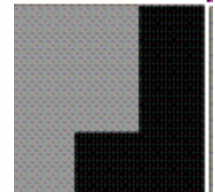
# Иллюстрация алгоритма GraphCut

исходное изображение,  
O, B - семена объекта и фона

финальная сегментация



Для нахождения самого минимального разреза авторы GraphCut-а разработали новый алгоритм, основанный на алгоритме Форда-Фалкерсона [7].



# Улучшение алгоритма

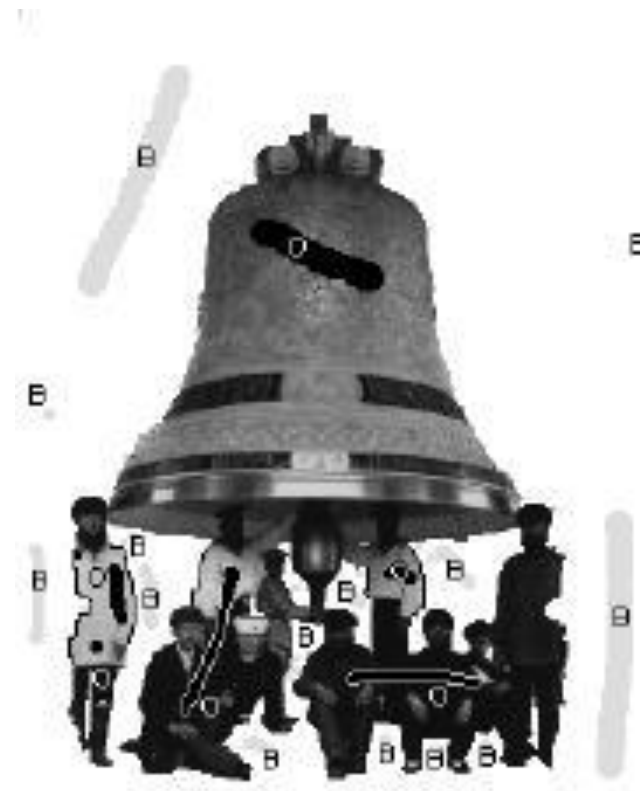
ребро	Вес	Для
$\{p, q\}$	$B_{\{p, q\}}$	$\{p, q\} \in N$
$\{p, S\}$	$\lambda * R_p(\text{" bkg"})$	$p \in P, p \notin O \cup B$
	$\infty$	$p \in O$
	0	$p \in B$
$\{p, T\}$	$\lambda * R_p(\text{" obj"})$	$p \in P, p \notin O \cup B$
	0	$p \in O$
	$\infty$	$p \in B$

цветовая  
статистика  
объекта и фона,  
собранная на  
основе  
отмеченных  
пользователем  
семян объекта и  
фона

# Алгоритм GraphCut. Примеры



# Пример работы алгоритма GraphCut



# Алгоритмы интерактивной сегментации

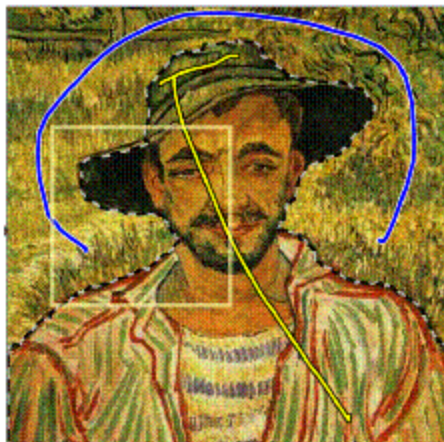
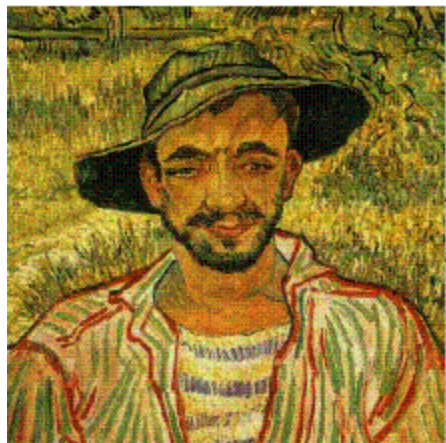
Время до 2001

- Magic wand
- Intelligent scissors
- Intelligent Paint

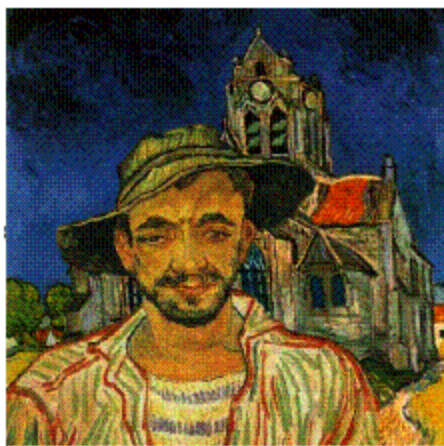
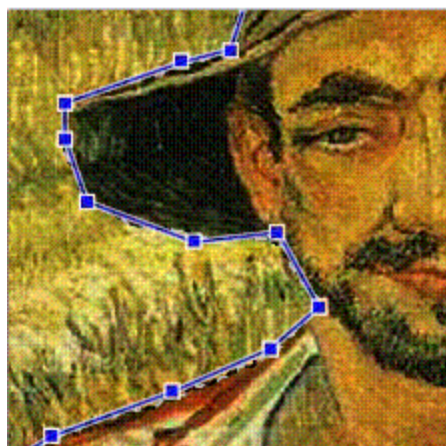
После 2001

- Interactive graph cuts
- **Lazy Snapping**
- GrabCut
- Progressive Cut
- GrowCut

# Lazy Snapping [11]



(а) - исходное изображение;  
(б) - пользователь указывает всего 3 линии (2 внутри объекта, 1 снаружи), все линии рисуются далеко от границы

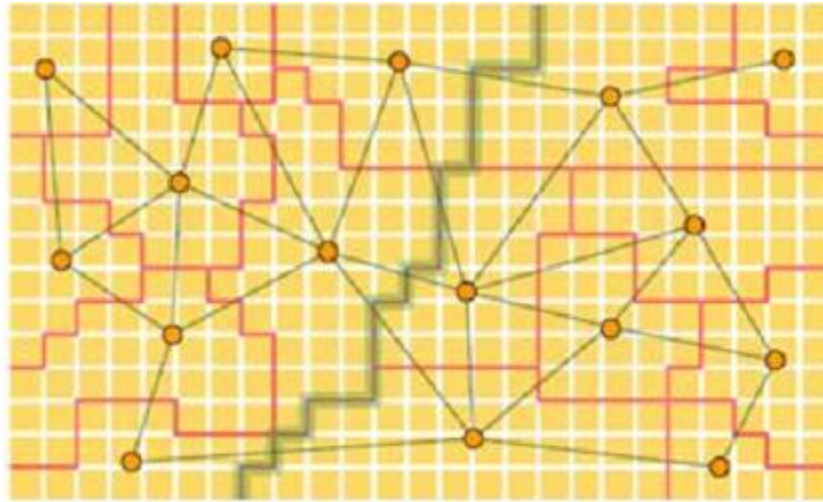


(в) - полигональное редактирование границы объекта;  
(г) - композиция вырезанного изображения и ещё одной картины Ван Гога.

То же, что и Interactive GraphCuts, но на карте суперпикселей

# Lazy Snapping - особенности

- Карта суперпикселей строится алгоритмом watershed



- В 10 раз быстрее чем на пикселях при сравнимом визуальном качестве
- Учитывается цвет фона и объектов.
- Выборка из размеченных пользователем областей кластеризуется k-средними на 64 кластера, вероятности считаются как отношение близости к объекту/фону



# Lazy Snapping - редактирование границы

- получившаяся граница объекта преобразуется в полигон
- для редактирования границы объекта можно:
  - переместить, удалить 1 или несколько вершин полигона, добавить новую вершину;
  - использовать Overriding Brush в неправильно отсегментированной области (начало и конец мазка должны лежать на полигоне)



# Результат Lazy Snapping



- Уточнение границ считается как задача поиска оптимального разреза в узкой полосе, отмеченной пользователем

# Примеры сегментации с помощью Lazy Snapping



# Алгоритмы интерактивной сегментации

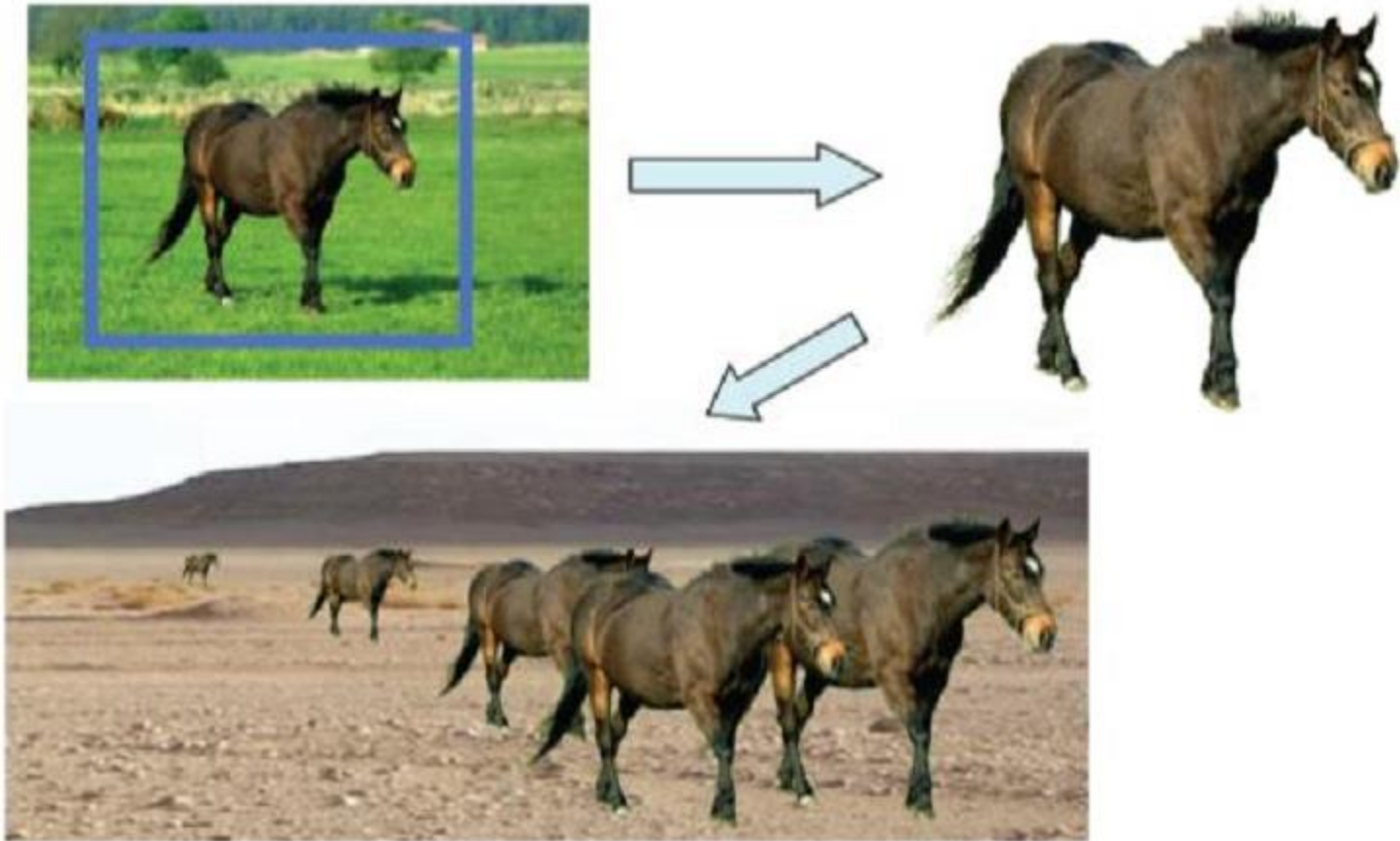
Время до 2001

- Magic wand
- Intelligent scissors
- Intelligent Paint

После 2001

- Interactive graph cuts
- Lazy Snapping
- **GrabCut**
- Progressive Cut
- GrowCut

# GrabCut [9]



C. Rother, V. Kolmogorov, and A. Blake. Grabcut – interactive foreground extraction using iterated graph cuts. Proc. ACM Siggraph, 2004.

# Итеративная схема сегментации GrabCut :

- исходя из цветового распределения внутри и снаружи ограничивающего прямоугольника, строится первая цветовая статистика объекта и фона
- сегментация GraphCut-ом, использующим цветовую статистику
- уточнение цветовой статистики. Граф, в котором ищется минимальный разрез, перевзвешивается





Инициализация пользователя



К-средних для  
обучения статистик  
цветов



Graph cuts для  
сегментации

- Цвет фона и объекта моделируется смесью гауссиан (обычно 5-8 компонент)
- Каждый пиксель объекта/фона сопоставляется компоненте смеси
- Затем уточняются параметры смеси

Slide by C. Rother

# Примеры



Slide by C. Rother



# Сложные примеры

Camouflage &  
Low Contrast

Initial  
Rectangle



Initial  
Result



Fine structure



No telepathy



Slide by C. Rother

# Сравнение

Magic Wand  
(198?)

User  
Input



Result



Regions

Intelligent Scissors  
Mortensen and Barrett (1995)



Boundary

GrabCut



Regions & Boundary

Slide by C. Rother

# Сравнение более поздних

Boykov and Jolly (2001)

User  
Input



Result



**Error Rate: 0.72%**

GrabCut



**Error Rate: 0.72%**

Slide by C. Rother



Magic Wand  
(198?)

Intelligent Scissors  
Mortensen and  
Barrett (1995)

Graph Cuts  
Boykov and  
Jolly (2001)

LazySnapping  
Li et al. (2004)

GrabCut  
Rother et al.  
(2004)

Slide by C. Rother

# GrabCut [9]

- GrabCut запатентован компанией Microsoft
- реализован в одной из рабочих версий графического редактора Microsoft Expression [10] (версия V4.0d)

# Алгоритмы интерактивной сегментации

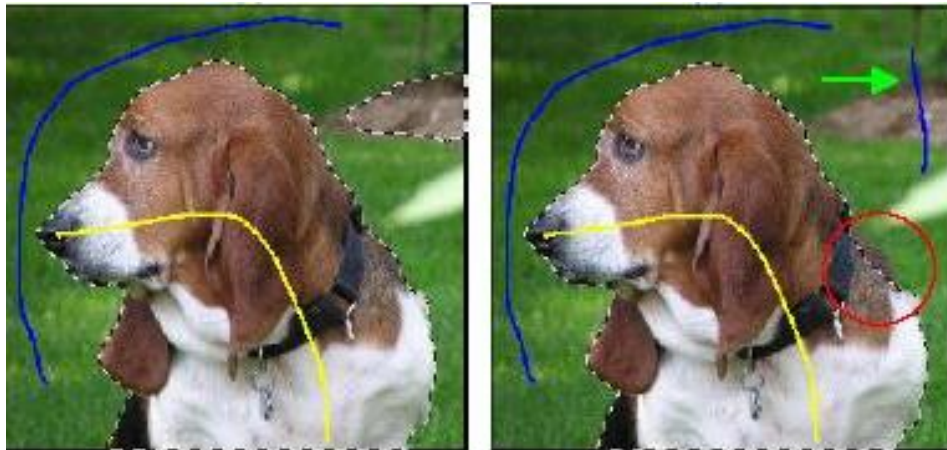
Время до 2001

- Magic wand
- Intelligent scissors
- Intelligent Paint

После 2001

- Interactive graph cuts
- Lazy Snapping
- GrabCut
- **Progressive Cut**
- GrowCut

# Progressive Cut [13]

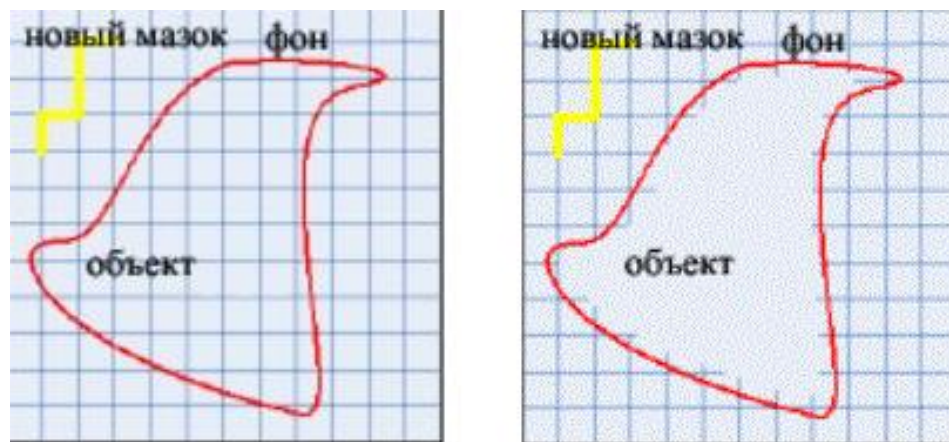


Типичное поведение GraphCut-а

# Progressive Cut – намерение пользователя

Progressive Cut пытается понять намерение пользователя, лежащее под тем или иным дополнительным вводом.

1) анализируется, какой тип изменения ожидает пользователь



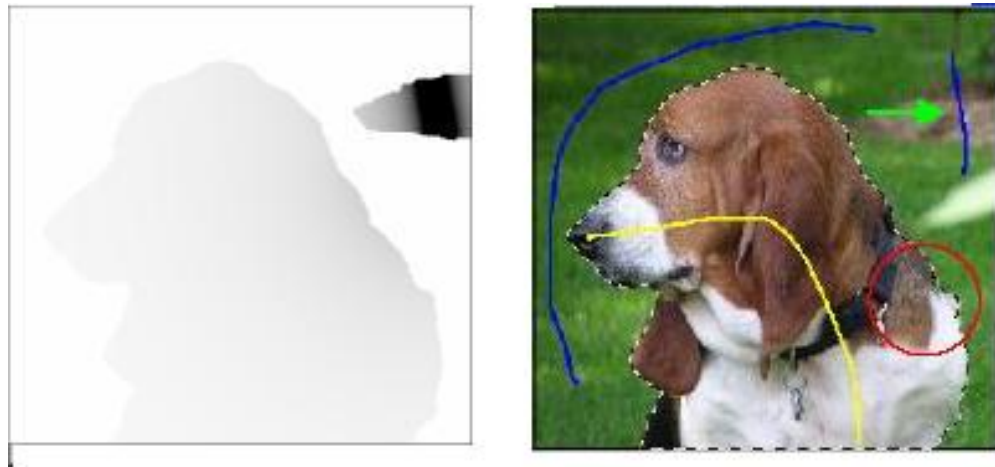
Построение уменьшенного графа на стадии редактирования изначальной сегментации



# Progressive Cut - энергия намерения (intention energy)

2) пользователь обычно ожидает изменения в относительно небольшой области.

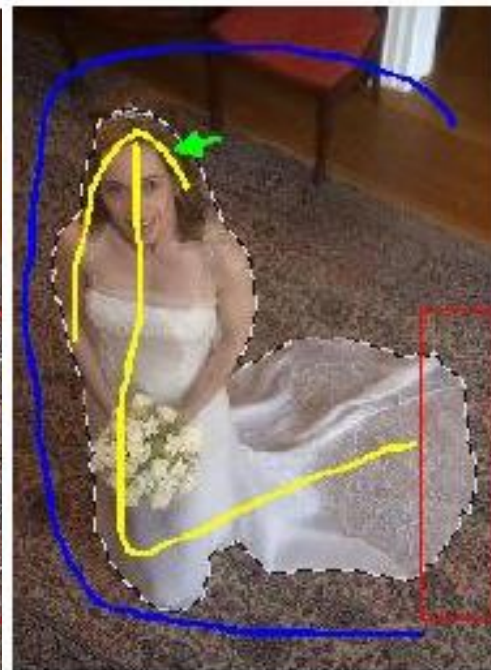
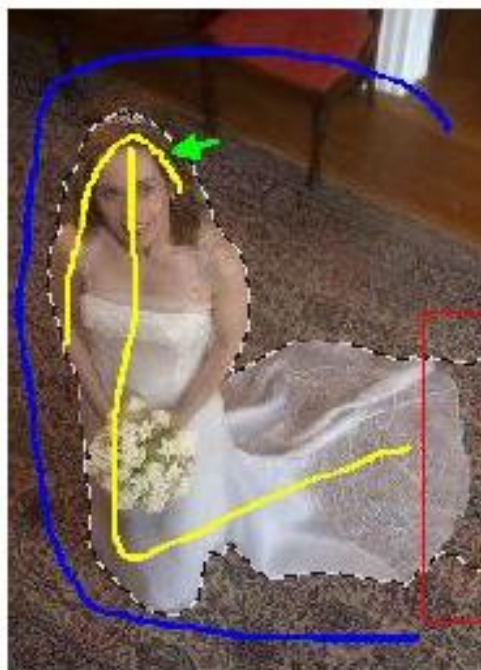
Это записывается в виде дополнительной энергии намерения (intention energy) , которая обратно пропорциональна расстоянию от данного пикселя до новых семян



# Сравнение редактирования сегментации с помощью GraphCut и Progressive Cut



# Сравнение редактирования сегментации с помощью GraphCut и Progressive Cut



# Алгоритмы интерактивной сегментации

Время до 2001

- Magic wand
- Intelligent scissors
- Intelligent Paint

После 2001

- Interactive graph cuts
- Lazy Snapping
- GrabCut
- Progressive Cut
- **GrowCut**

# Параметры GrowCut [17]

- основывается на клеточных автоматах
- состояние каждой клетки -  $(L, \Theta, C)$ ,  
где  $L$  - метка ('объект', 'фон', 'неизвестно'),  
 $\Theta \in [0,1]$  - сила клетки,  
 $C$  - вектор признаков клетки (RGB цвет)

# Алгоритм GrowCut

- $g(x)$  – монотонно убывающая функция, изменяющаяся в диапазоне  $[0, 1]$ , которая описывает близость цветов двух пикселей ( $x$  обычно является нормой разницы цветов пикселей)
- на каждой итерации каждую клетку  $P$  'атакуют' все её соседи  $Q$
- если  $\theta_q * g(\|C_p - C_q\|) > \theta_p$ , клетка  $Q$  захватывает  $P$

# GrowCut. Процесс эволюции клеточного автомата

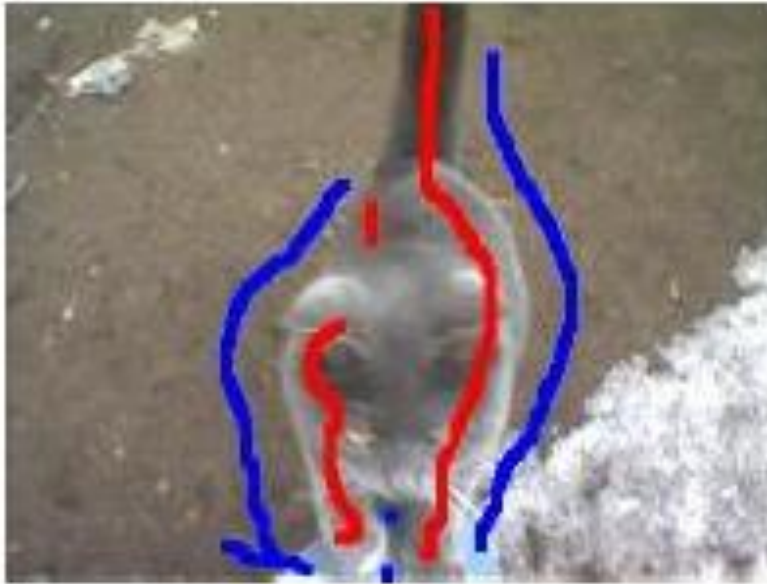


# GrowCut

- Данный алгоритм реализован в виде одноименного плагина к Adobe Photoshop [19]
- легко реализуется на GPU [18]
- Официальный сайт алгоритма  
[www.growcut.com](http://www.growcut.com)



# Результат GrowCut



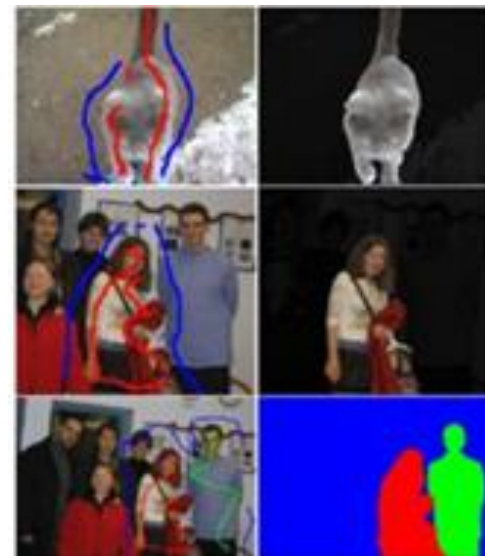
# Достоинства и недостатки GrowCut

## Плюсы

- прост
- данный алгоритм можно обобщить на случай более 2 сегментов
- время реакции алгоритма близко к нулю
- коррекцию можно выполнить по ходу алгоритма

## Минусы

- недостаточно быстр для больших фото
- более рваная граница, чем у Graph Cuts



# Развитие GrowCut

- В качестве развития данного алгоритма были предложены иерархическая версия, значительно ускоряющая базовый алгоритм, а также версия, налагающая на границы итоговых сегментов дополнительные ограничения на гладкость.

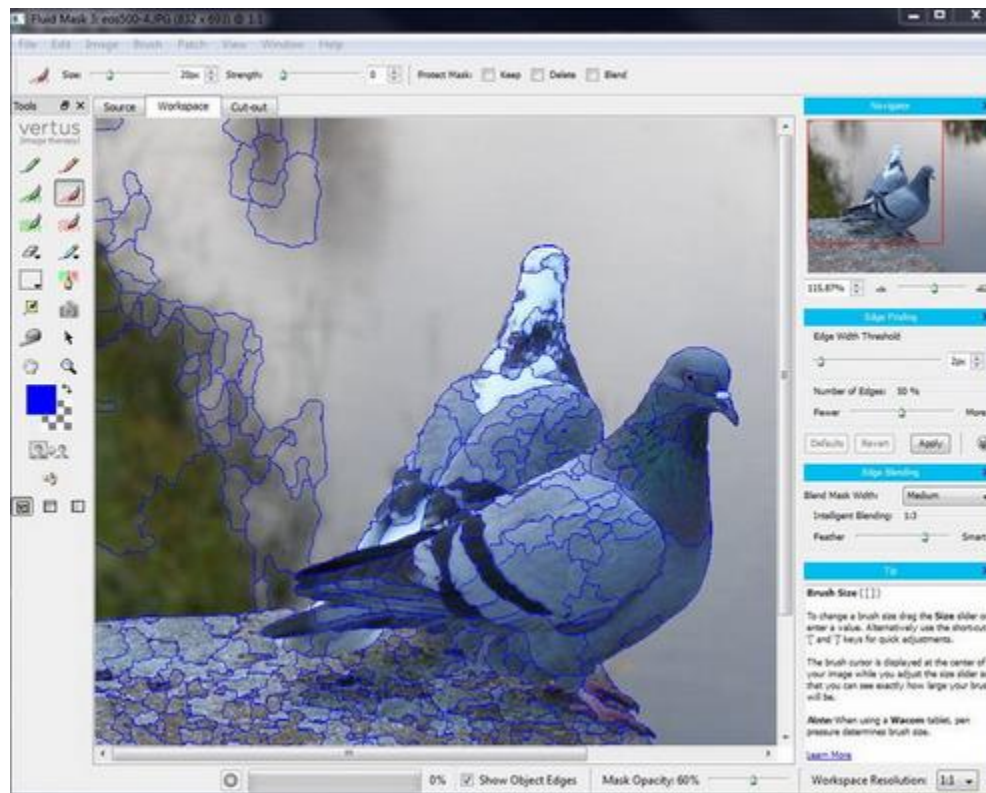
# Заключение

- Magic Wand является самым примитивным инструментом сегментации, прост в реализации
- Intelligent Scissors: опорные точки в самой близи от границы объекта, высокая скорость алгоритма
- Intelligent Paint: придуман относительно давно, не реализован в коммерческих программах
- GrowCut: простота реализации, хороший алгоритм, очень маленькое время реакции на дополнительный ввод и высокая интерактивность, некоторая рваность границ, требователен к оперативной памяти

# Заключение

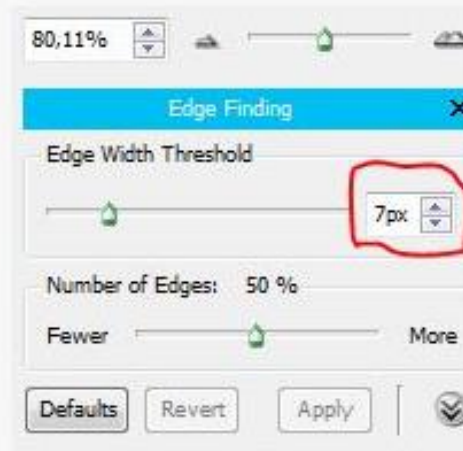
- GraphCut: упрощает действия пользователя, значительное время реакции, требует много оперативной памяти, запатентован
- Lazy Snapping, сохраняя достоинства GraphCut-а, значительно уменьшает время реакции на дополнительный ввод.
- GrabCut: самый удобный интерфейс (ограничивающий прямоугольник), в случаях сильного перекрытия цветовых распределений фона и объекта, выдаёт неправильный результат, и изображение приходится сегментировать также, как и в обычном GraphCut-е, сложнее для реализации, запатентован.
- Progressive Cut: упрощает процесс редактирования сегментации, изменение исходного графа может увеличить время реакции алгоритма

# Коммерческий Vertus Fluid mask



# Анализ цветowych переходов

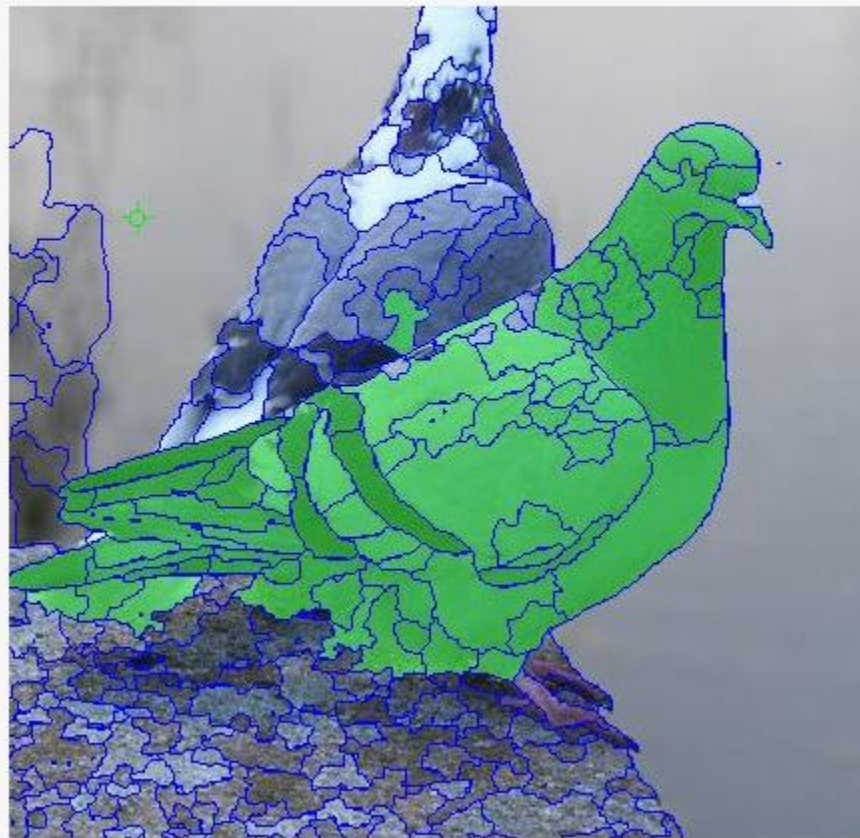
Смысл работы плагина в том, что он анализирует все цветowych переходы и показывает их. Плагин достаточно мощный, и можно исходя из изображения, настроить порог анализа. Если вырезаемый объект сливается с фоном, то лучше поставить минимальный порог, что бы плагин выделил как можно больше переходов(3).



У наших голубей, есть места, которые сливаются и я выбрал маленький порог 2px.

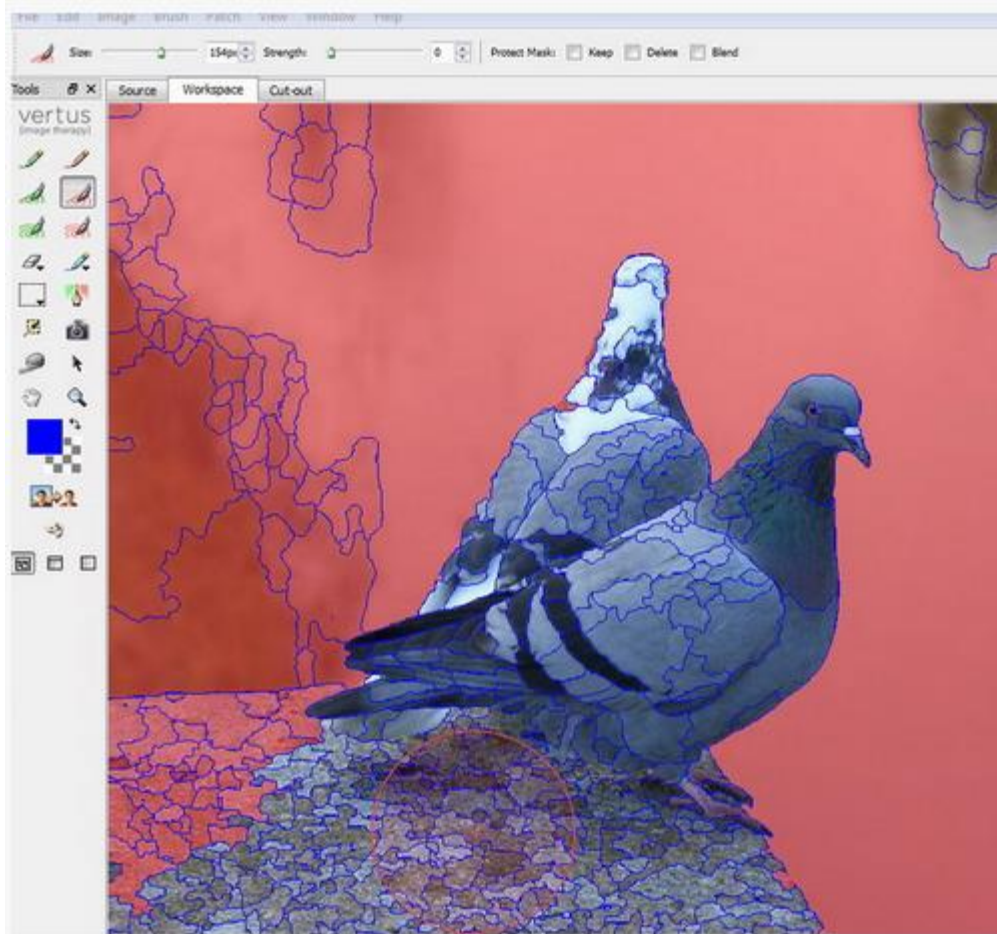
# Зелёное – то что нужно

Теперь берём красную кисточку, и закрашиваем всё, что нам не нужно. Можно пойти от обратного и закрасить зелёным, что нужно оставить(4).

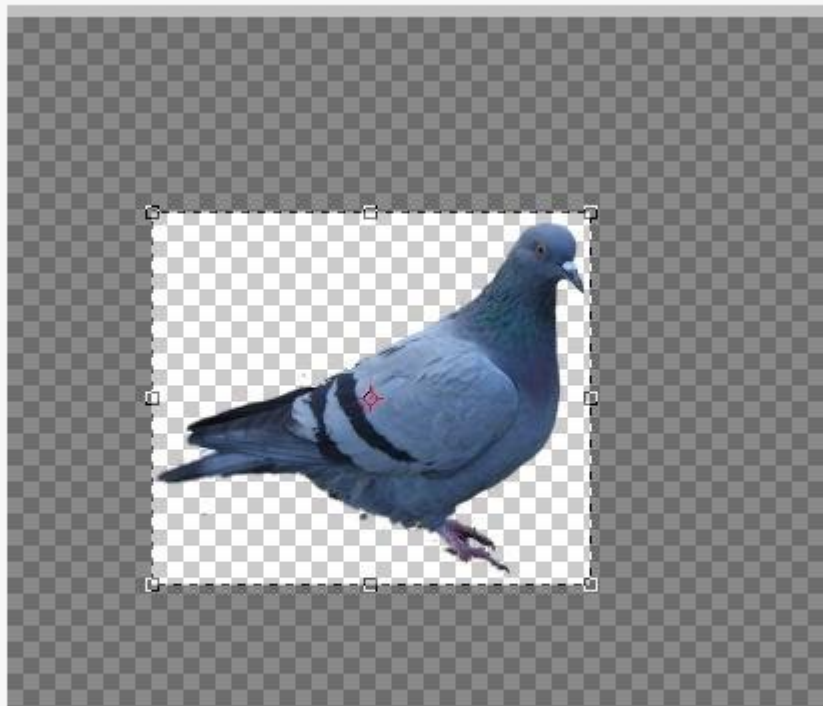




Или красным, то что не нужно



# Результат Vertus Fluid mask

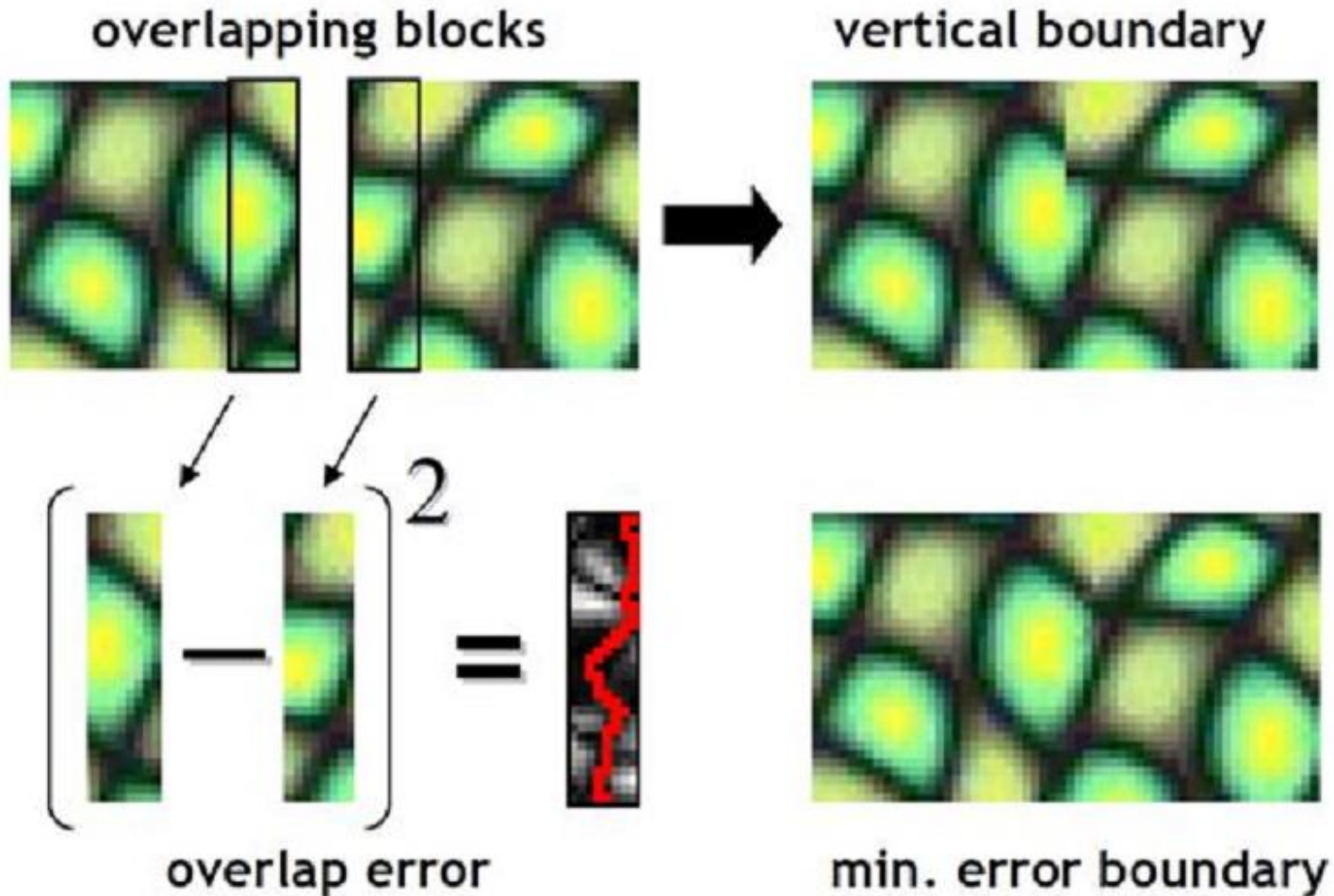


Теперь "ластиком" убираем остатки и "размытием" проходимся по краям(11).

# Применение

- Композиция изображений
- Сшивка
- Создание текстур

# GraphCut для сшивки



KWATRA, V., SCHODL, A., ESSA, I., TURK, G., AND BOBICK, A. Graphcut textures: image and video synthesis using graph cuts. SIGGRAPH, 2003

# GraphCut для сшивки



# Создание текстур



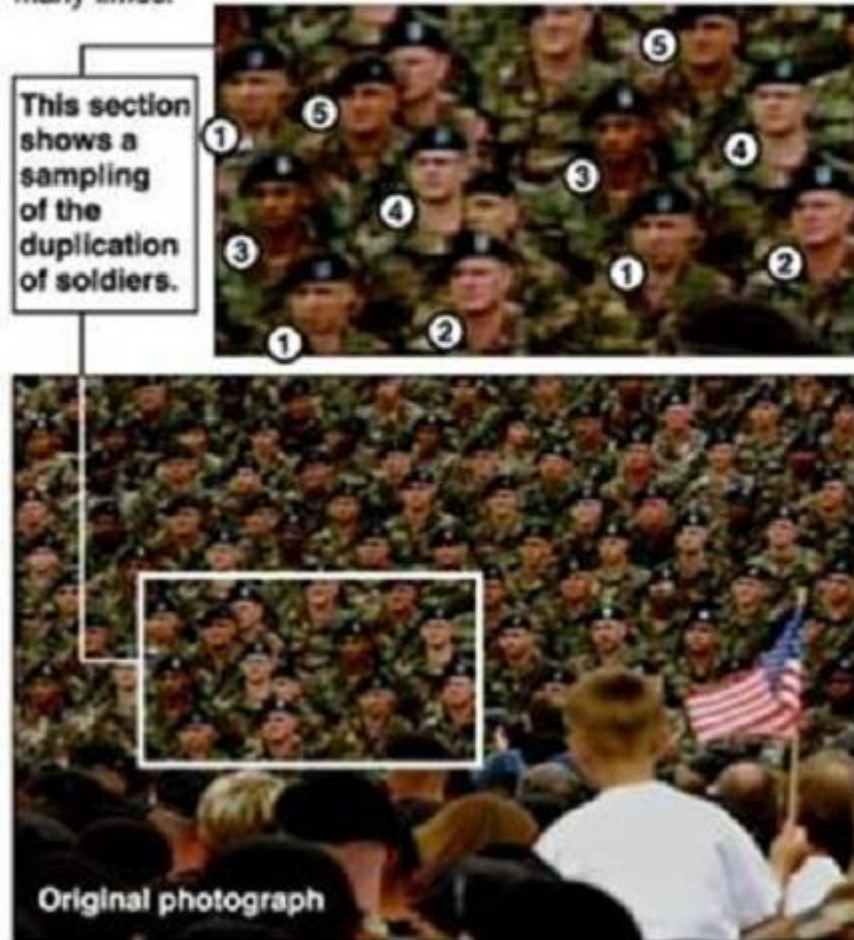
- «Накидываем» исходные экземпляры на изображение и сшиваем их с помощью разрезов графов.

- Перспективное изменение (масштабирование) по необходимости

# Синтез текстур в жизни

## Bush campaign digitally altered TV ad

President Bush's campaign acknowledged Thursday that it had digitally altered a photo that appeared in a national cable television commercial. In the photo, a handful of soldiers were multiplied many times.



# Задача фотомонтажа

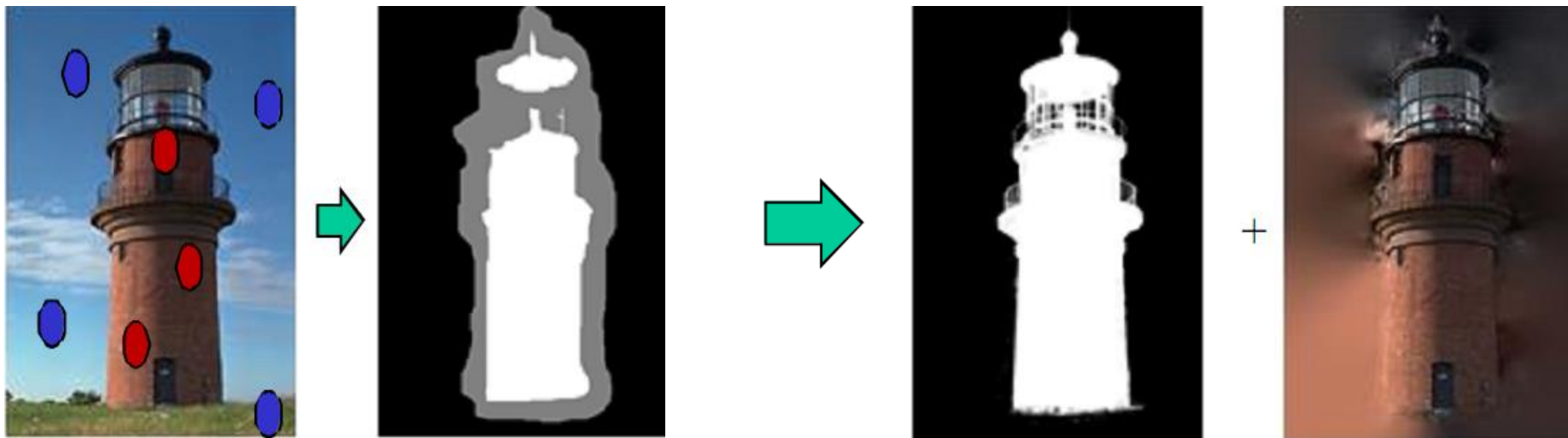
- Манипулирование отдельными объектами сцены с высоким визуальным качеством
  - Добавление нового объекта
  - Удаление объекта
  - Восстановление фона
- Композиция одного изображения из нескольких





# Подход к решению

- Для изображений, полученных в обычных условиях, задача раскладывается в несколько более простых задач
  - Интерактивная «жесткая» сегментация объекта
    - Пользовательский ввод  $\alpha$  из  $\{0,1\}$
  - Мягкая сегментация объекта (собственно *маттирование*)
    - Результат жесткой сегментации используется как начальное приближение



# Формирование изображения

- Мы представляем изображение как композицию объекта и фона, причём какие-то пиксели изображения есть результат смешения объекта и фона (следствие дискретизации)

$$I = \alpha * F + (1 - \alpha) * B$$



- Карта непрозрачности (opacity map)  $\alpha$  из  $[0, 1]$ , где 0 - фон, 1 – непрозрачный объект
- Она же альфа-канал

# Выделение объекта

- Задача маттирования (matting) объекта
  - Выделить интересующий пользователя объект на изображении
- Формально:
  - Для каждого пикселя изображения определить значения прозрачности  $\alpha$  из  $[0, 1]$ , где 0 - фон, 1 – непрозрачный объект
  - Для всех пикселей с  $\alpha > 0$  определить  $f$  – цвет пикселя объекта



# Подход к решению

- С очевидностью, задача -- недоопределенная
  - Сцена состоит из множества объектов, какой из них нужен пользователю, неизвестно
- Поэтому в общем виде автоматически она решается пока только съемкой в специальных условиях

# Chroma keying & Luma keying

- Chroma keying
  - Объект снимается на однородном фоне определенного цвета
  - Широко применяется в кино и на телевидении
  - Работает в real-time
  - Возможно извлечение теней, бликов, прозрачных элементов
- 
- Luma keying
  - Альфа-канал получается путем применения фильтров яркости и контрастности к изображению, преобразованному к градациям серого

# Chroma keying – пример

2003



# Chroma keying

- фильмы, которые были полностью сняты на Chroma keying («Жизнь Пи», например)
- на 90% снятые на Chroma keying («Город грехов», «Аватар», «Алиса в стране чудес» и др)

# Luma keying - пример





# Luma keying - пример



# Литература 1

2. Martin D., Fowlkes C., Tal D., Malik J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *Proceedings of 8th International Conference on Computer Vision*. 2001. 2. N. 1. P. 416-423.
3. Mortensen, E., and Barrett, W. 1995. Intelligent scissors for image composition. *Proc. ACM Siggraph*, 191-198.
4. Mortensen, E., and Barrett, W. 1999. Tobogan-based intelligent scissors with a four parameter edge model. In *Proc. IEEE Conf. Computer Vision and Pattern Recog.*, vol. 2, 452-458.
5. Reese, L. 1999 Intelligent Paint: Region-Based Interactive Image Segmentation. *Master's thesis*, Department of Computer Science, Brigham Young University, Provo, UT.
6. Y. Boykov and M.-P. Jolly. Interactive organ segmentation using graph cuts. In *Medical Image Computing and Computer-Assisted Intervention*, pages 276-286, 2000.
7. Boykov, Y., and Kolmogorov, V. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 9, 1124-1137.
8. Boykov, Y., and Jolly, M.-P. 2001. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *Proc. Of the International Conference on Computer Vision*, vol. 1, 105-112.
9. Rother, C., Kolmogorov, V., and Blake, A. 2004. Grabcut - interactive foreground extraction using iterated graph cuts. *Proc. ACM Siggraph*.

# Литература 2

10. [http://www.microsoft.com/products/expression/en/graphic\\_designer/default.mspix](http://www.microsoft.com/products/expression/en/graphic_designer/default.mspix)
11. Li, Y., Sun, J., Tang, C.-K., and Shum, H.-Y. 2004. Lazy Snapping, *Proc. ACM Siggraph*
12. Vincent, L., and Soille, P. 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-13*, 6 (June), 583-598.
13. Wang, C., Yang, Q., Chen, M., Tang, X., Ye, Z.F. 2006. Progressive Cut. *Proc ACM SIG Multimedia*
14. Grady, L., and Funka-Lea, G. 2004. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In *ECCV Workshops CVAMIA and MMBIA*, 230-245.
15. Grady, L., Schiwietz, T., Aharon, S., and Westermann, R. 2005. Random walks for interactive organ segmentation in two and three dimensions: Implementation and validation. In *Proceedings of MICCAI 2005*, Springer, Palm Springs, CA, J. Duncan and G. Gerig, Eds., no. 2 in LNCS 3750, MICCAI Society, 773-780.
16. Grady, L. 2005. Multilabel random walker image segmentation using prior models. In *CVPR (1)*, 763-770.
17. Vezhnevets, V., and Konouchine, V., 2005. "GrowCut"- interactive multi-label N-D image segmentation by cellular automata, *Proc Graphicon*.
18. <http://www.sonycsl.co.jp/person/nielsen/PT/SHADERX5/shaderx5.html>
19. <http://www.growcut.com>
20. Talbot, J.F., and Xu, X., Implementing Grabcut, <http://students.cs.byu.edu/~jtalbot/research/Grabcut.pdf>