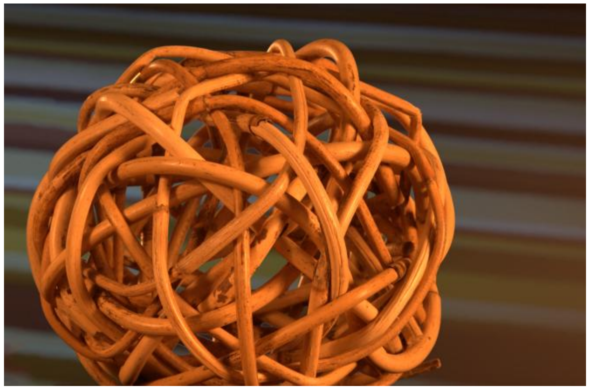
Алгоритмы для задачи матирования

Юрий Гитман

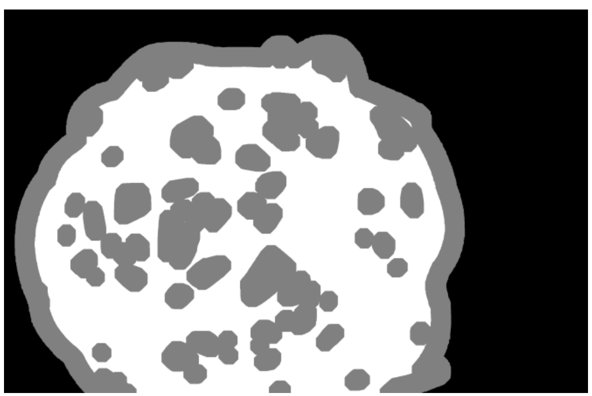
Video Group
CS MSU Graphics & Media Lab

Table of content

Matting problem
Common statement



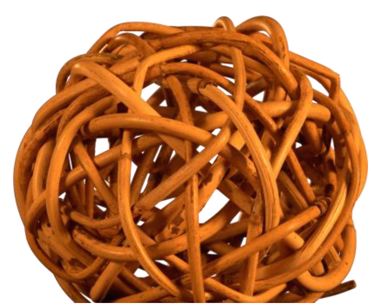
Исходное изображение



Грубая разметка (Trimap)

Фон

Карта Прозрачности



Объект

alphamatting.com/eval_25.php

State of the art
Image Matting

Sum of Absolute Differences	Overall rank	Average (small trimap)	Average (large trimap)	Average (user-defined trimap)
SVR Matting (Support Vector Regression)	5.2	6.3	4.8	4.5
Weighted Color and Texture Matting	5.5	4.5	6.5	5.4
Shared Matting	6.1	6.0	7.5	4.9
Global Sampling Matting	7.3	5.5	8.8	7.8
Segmentation-based Matting	7.7	8.0	7.3	7.9
Fast Automatic Matting	7.8	7.1	8.1	8.1
Improved Color Matting	8.2	7.9	7.8	9.0
LSR Matting (Local Spline Regression)	9.0	10.4	6.9	9.6
Global Sampling Matting (filter version)	9.1	8.4	9.8	9.3
KNN Matting (K-Nearest Neighbor)	9.7	11.1	10.5	7.4
Learning-based Matting	10.1	10.3	9.4	10.6
LMSPIR Matting	10.2	9.4	10.9	10.3
Shared Matting (real-time)	10.3	10.4	10.4	10.3
Closed-form Matting	10.5	10.1	9.1	12.4

alphamatting.com/eval_25.php

Problem of papers on Video matting

« Да, статей по matting’у во времени не мало, но многие из них опираются на существование идеального оптического потока»

Михаил Ерофеев

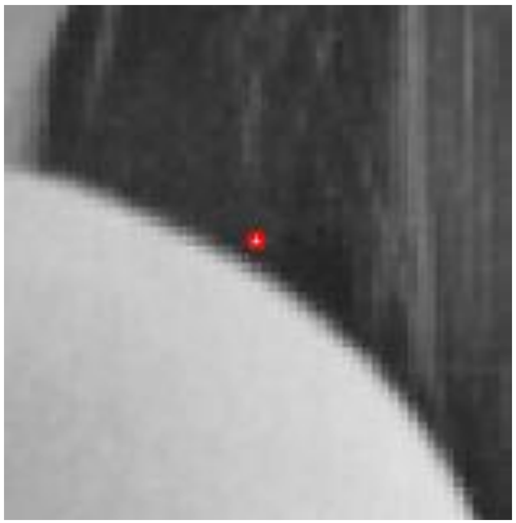
В предыдущих сериях

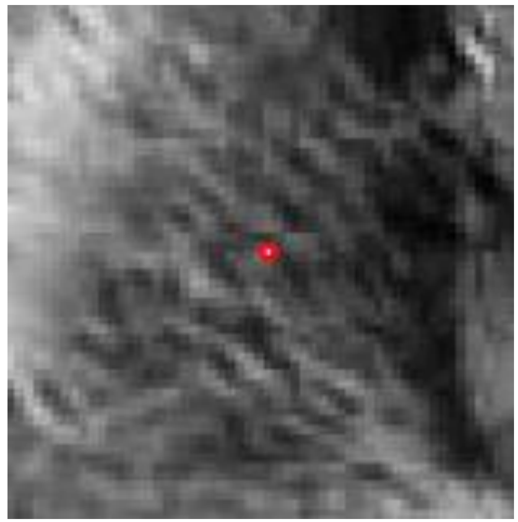
X. Bai, J. Wang, D. Simons, "Towards Temporally-coherent Video Matting," in *IEEE Mirage*, 2011

Содержание

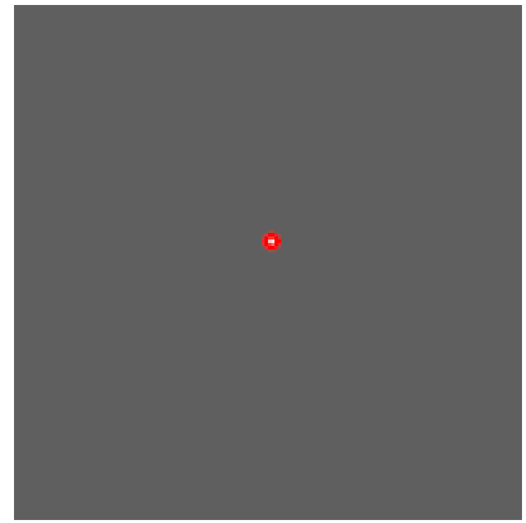
- Introduction
- **Guided Filter**
- PatchMatch
- Closed-form Matting
- Alpha Flow
- Conclusion

New edge-preserving filter



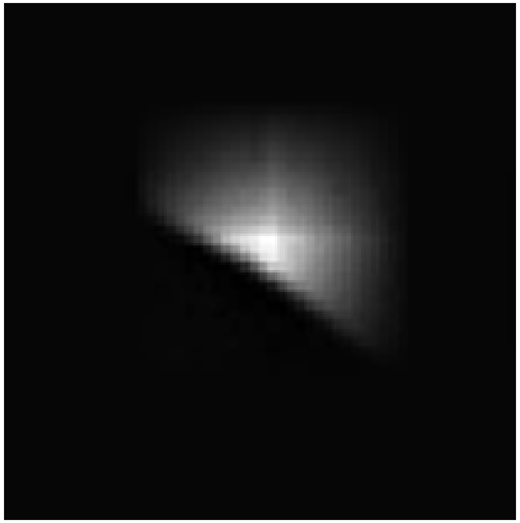


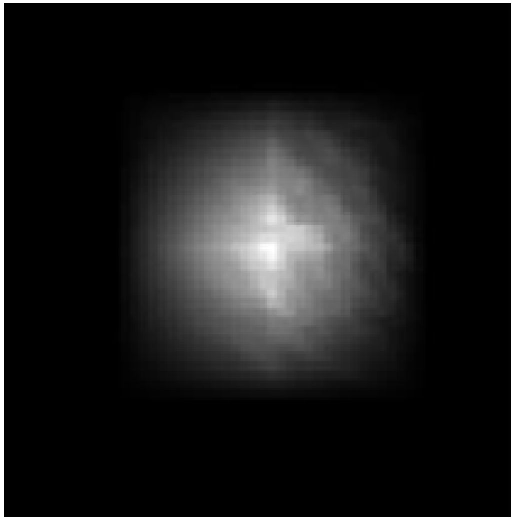
Guidance



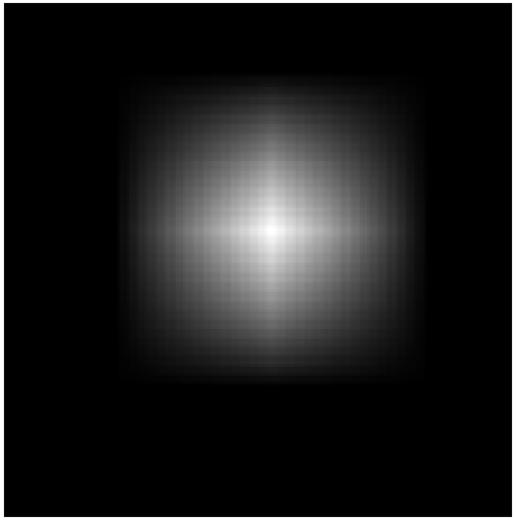
K. He, J. Sun, X. Tang, "Guided Image Filtering," in *ECCV*, 2010

New edge-preserving filter



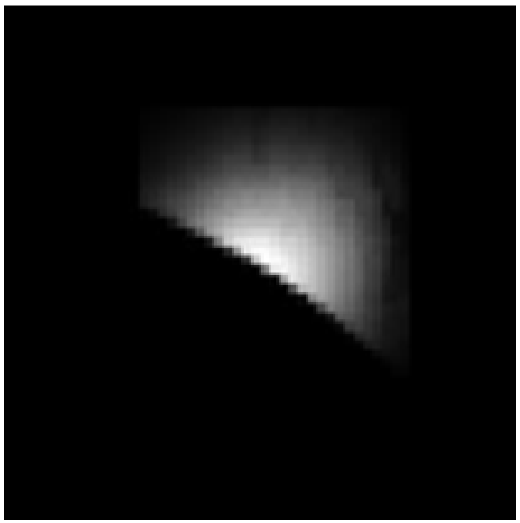


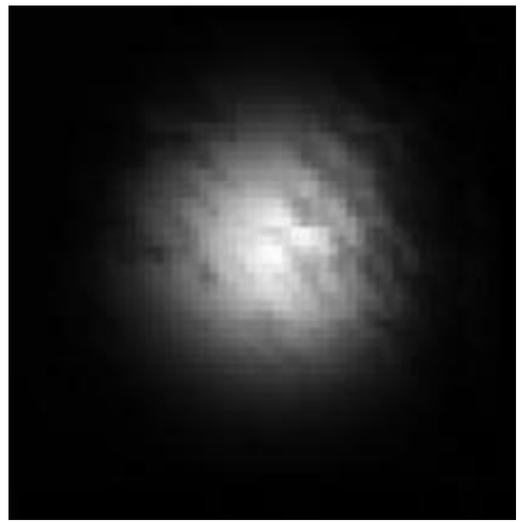
Guided Filter

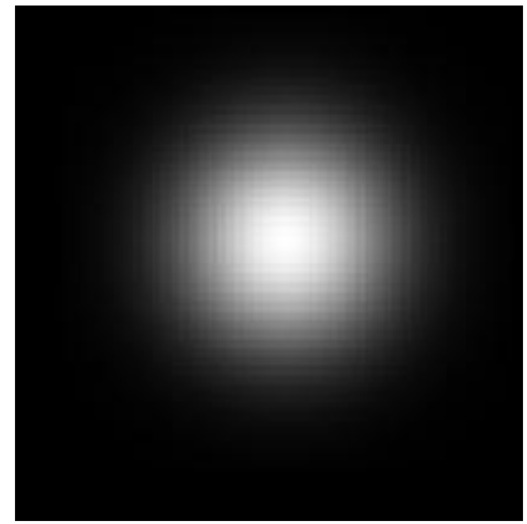


K. He, J. Sun, X. Tang, "Guided Image Filtering," in *ECCV*, 2010

New edge-preserving filter



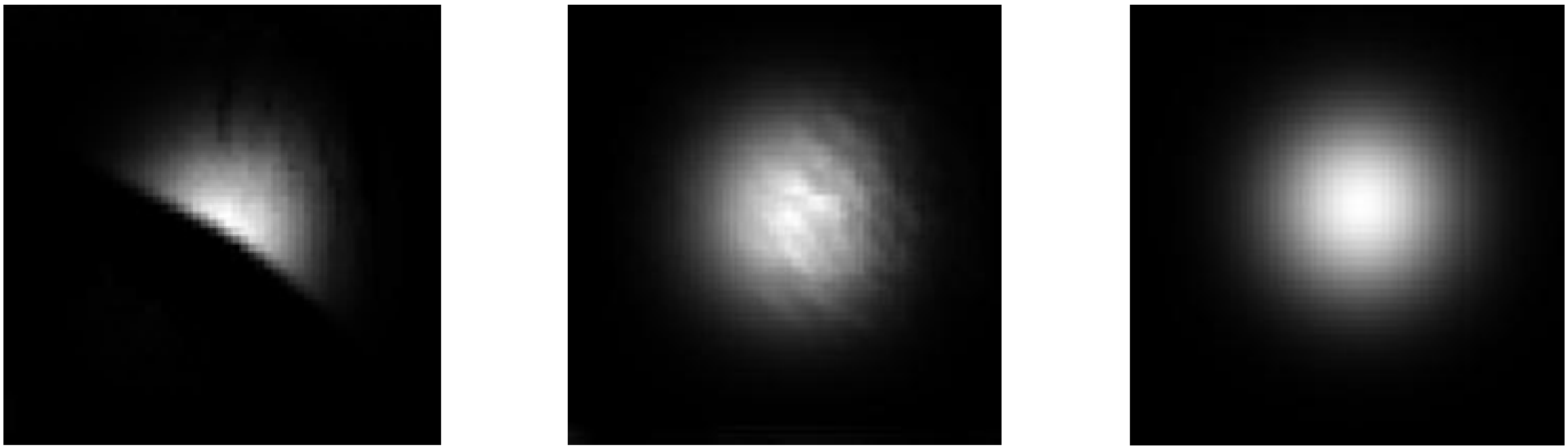




Bilateral Filter

K. He, J. Sun, X. Tang, "Guided Image Filtering," in *ECCV*, 2010

New edge-preserving filter



Gaussian Guided Filter

K. He, J. Sun, X. Tang, "Guided Image Filtering," in *ECCV*, 2010

Idea of Filtering

Для каждого окна p_k:

$$p_{i,j}^{k} = a_k I_{i,j} + b_k$$

Коэффициенты a_k и b_k, определяются так, чтобы наилучшим образом соответствовать исходному изображению

K. He, J. Sun, X. Tang, "Guided Image Filtering," in *ECCV*, 2010

Steps of the Algorithm

Algorithm 1 Guided Filter.

Input: filtering input image p, guidance image I, radius r, regularization ϵ

Output: filtering output q.

1: $\text{mean}_I = f_{\text{mean}}(I)$
   $\text{mean}_p = f_{\text{mean}}(p)$
   $\text{corr}_I = f_{\text{mean}}(I. * I)$
   $\text{corr}_{Ip} = f_{\text{mean}}(I. * p)$

2: $\text{var}_I = \text{corr}_I - \text{mean}_I. * \text{mean}_I$
   $\text{cov}_{Ip} = \text{corr}_{Ip} - \text{mean}_I. * \text{mean}_p$

3: $a = \text{cov}_{Ip} \; ./ \; (\text{var}_I + \epsilon)$
   $b = \text{mean}_p - a. * \text{mean}_I$

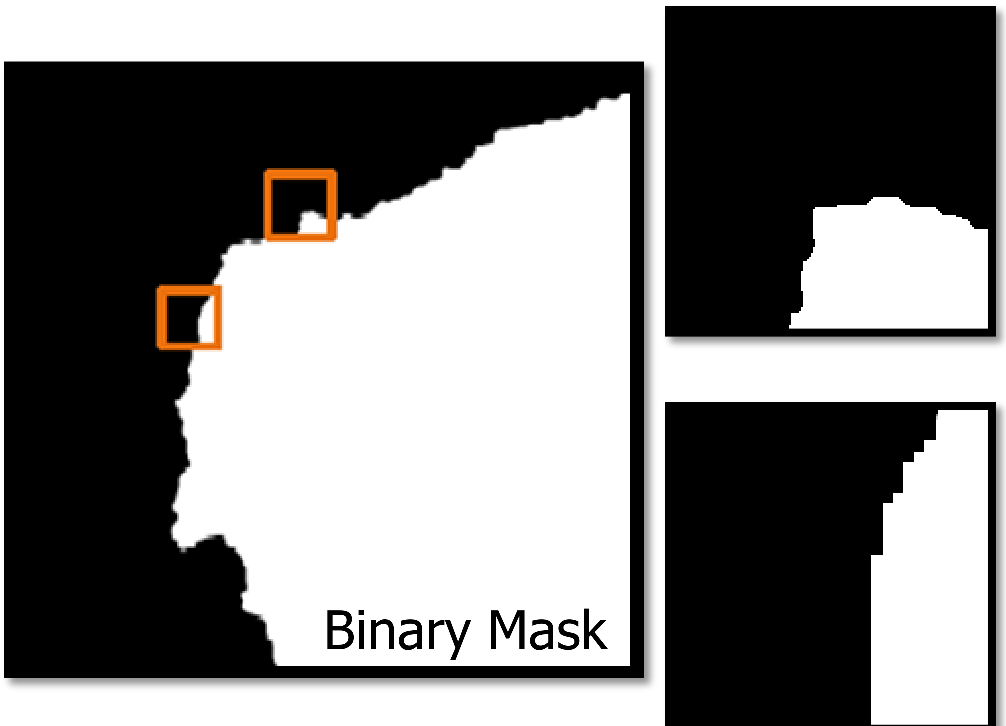
4: $\text{mean}_a = f_{\text{mean}}(a)$
   $\text{mean}_b = f_{\text{mean}}(b)$

5: $q = \text{mean}_a. * I + \text{mean}_b$

/* f_{mean} is a mean filter with a wide variety of $O(N)$ time methods. */

K. He, J. Sun, X. Tang, "Guided Image Filtering," in *ECCV*, 2010

Application to Matting

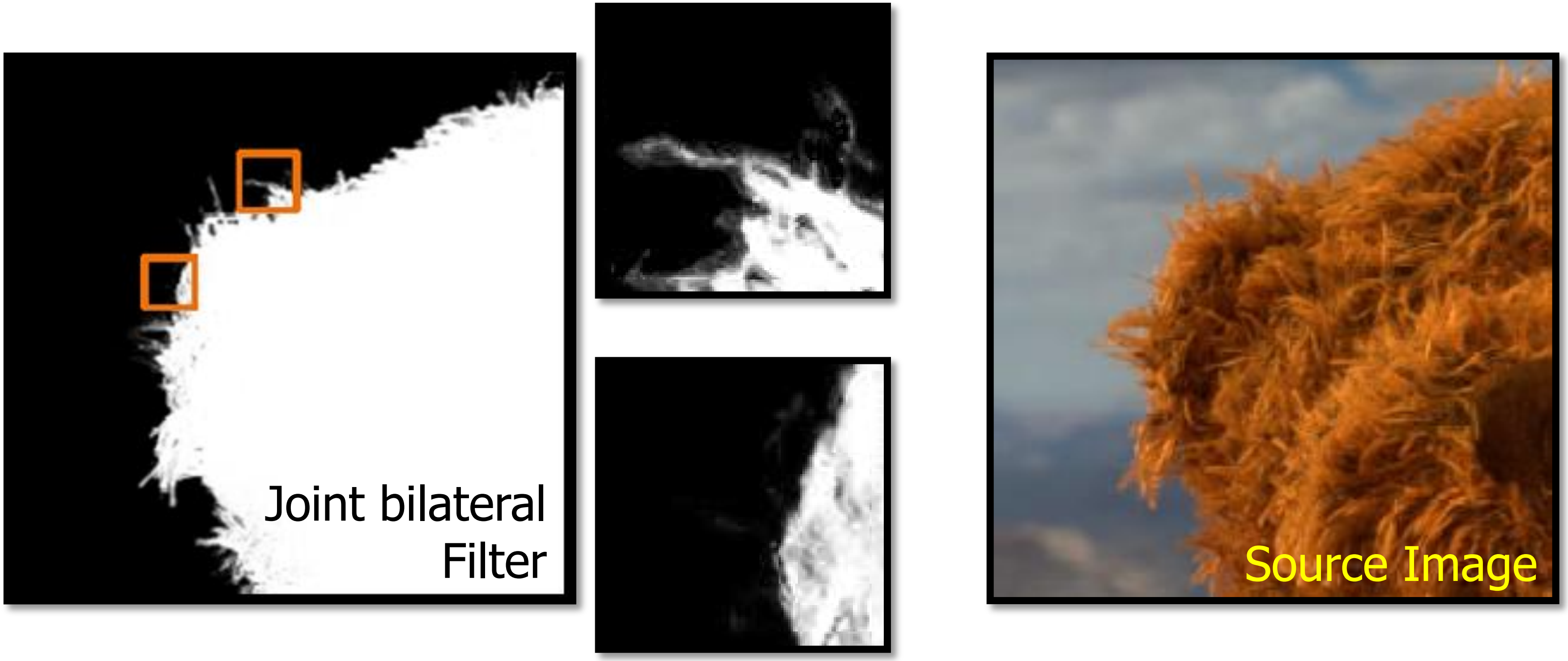


K. He, J. Sun, X. Tang, "Guided Image Filtering," in *ECCV*, 2010

Application to Matting

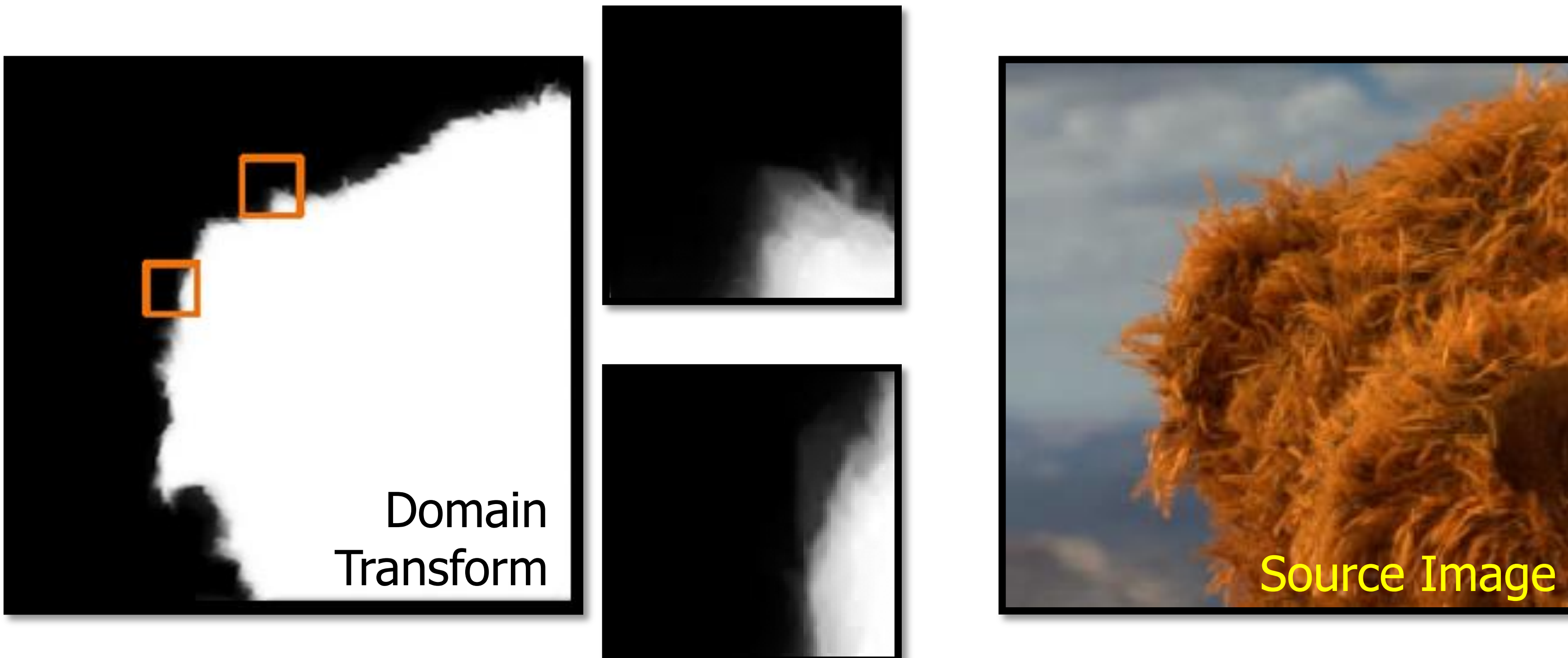
K. He, J. Sun, X. Tang, "Guided Image Filtering," in *ECCV*, 2010

Application to Matting



K. He, J. Sun, X. Tang, "Guided Image Filtering," in *ECCV*, 2010

Application to Matting

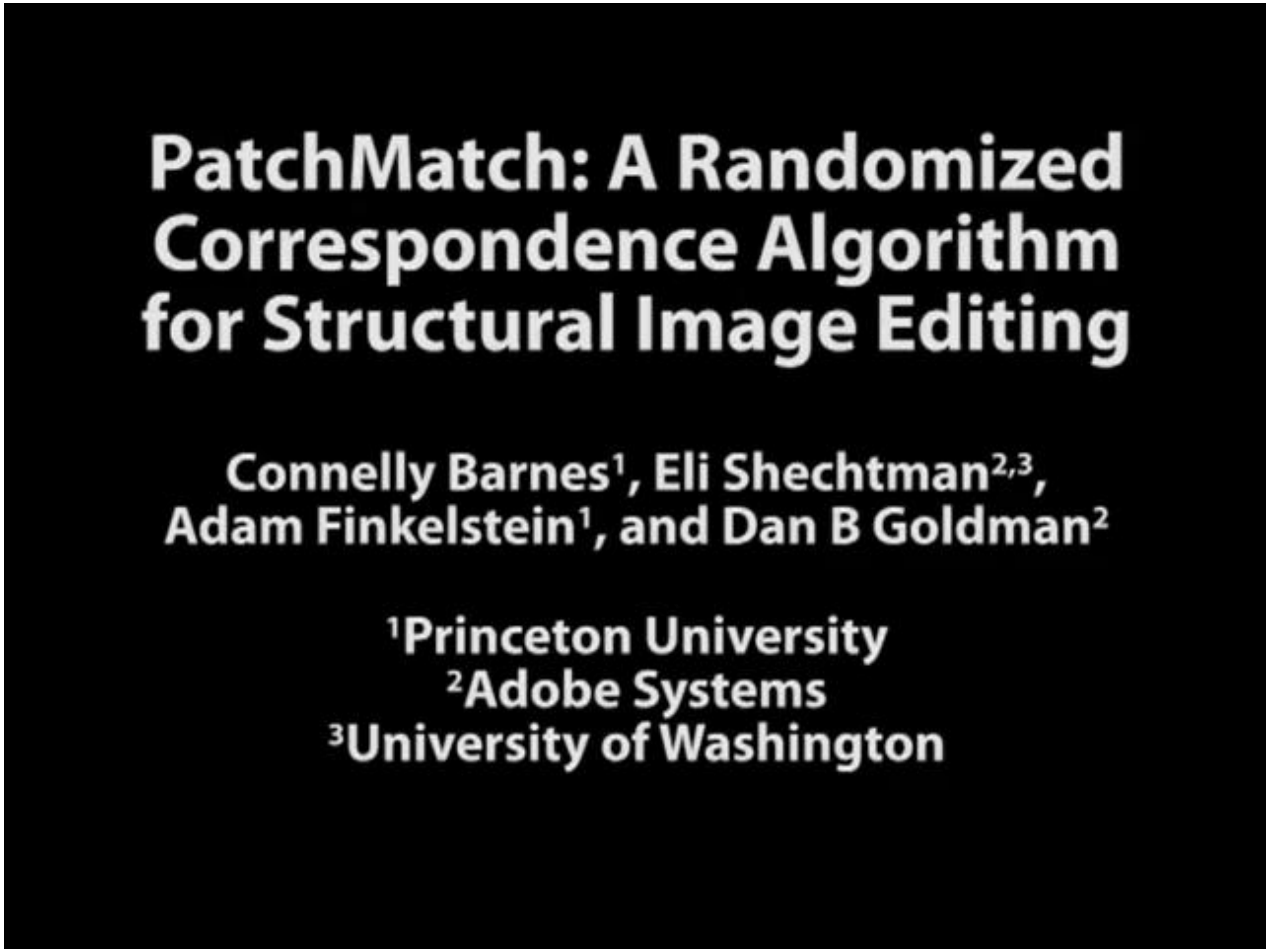


K. He, J. Sun, X. Tang, "Guided Image Filtering," in *ECCV*, 2010

Содержание

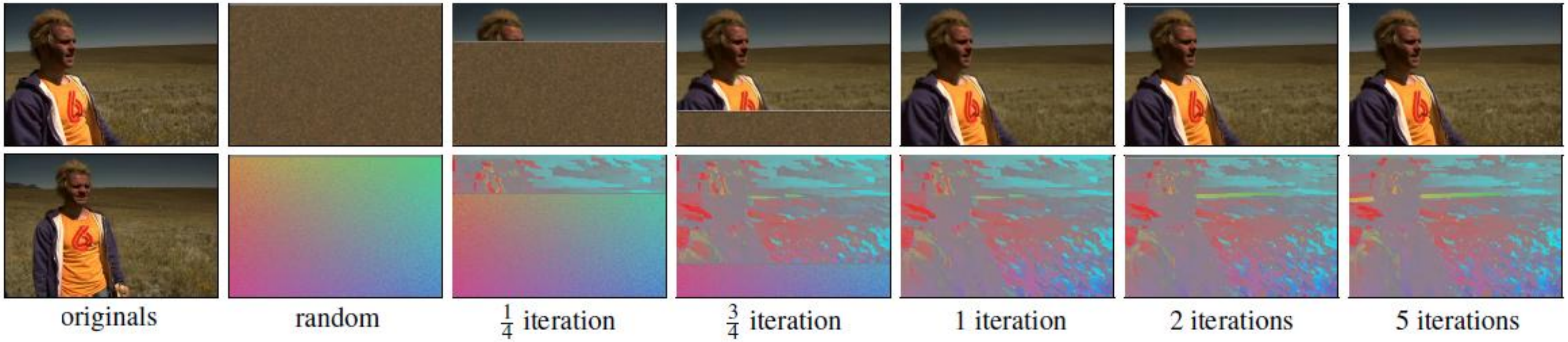
- Introduction
- Guided Filter
- **PatchMatch**
- Closed-form Matting
- Nonlocal Matting
- Alpha Flow
- Conclusion

Fast k nearest neighbors search (KNN)



C. Barnes, E. Shechtman, A. Finkestein, D. Goldman, "PatchMatch. A Randomized Correspondence Algorithm for Structural Image Editing," in *ACM Transactions on Graphics (TOG)*, 2009

Пример работы алгоритма



Поиск потока из нижнего изображения в верхнее.
Hue компонента соответствует углу
Magnitude компонента длине вектора

C. Barnes, E. Shechtman, A. Finkestein, D. Goldman, “PatchMatch. A Randomized Correspondence Algorithm for Structural Image Editing,” in *ACM Transactions on Graphics (TOG)*, 2009

Детали алгоритма (1)

pm_minimal.cpp

```
/* Match image a to image b, returning the nearest neighbor field mapping a => b coords, stored in an RGB 24-bit image as (by<<12)|bx. */
void patchmatch(BITMAP *a, BITMAP *b, BITMAP *&ann, BITMAP *&annd) {
  /* Initialize with random nearest neighbor field (NNF). */
  ann = new BITMAP(a->w, a->h);
  annd = new BITMAP(a->w, a->h);
  int aew = a->w - patch_w+1, aeh = a->h - patch_w + 1;       /* Effective width and height (possible upper left corners of patches). */
  int bew = b->w - patch_w+1, beh = b->h - patch_w + 1;
  memset(ann->data, 0, sizeof(int)*a->w*a->h);
  memset(annd->data, 0, sizeof(int)*a->w*a->h);
  for (int ay = 0; ay < aeh; ay++) {
    for (int ax = 0; ax < aew; ax++) {
      int bx = rand()%bew;
      int by = rand()%beh;
      (*ann)[ay][ax] = XY_TO_INT(bx, by);
      (*annd)[ay][ax] = dist(a, b, ax, ay, bx, by);
    }
  }
  for (int iter = 0; iter < pm_iters; iter++) {
    /* In each iteration, improve the NNF, by looping in scanline or reverse-scanline order. */
    int ystart = 0, yend = aeh, ychange = 1;
    int xstart = 0, xend = aew, xchange = 1;
    if (iter % 2 == 1) {
      xstart = xend-1; xend = -1; xchange = -1;
      ystart = yend-1; yend = -1; ychange = -1;
    }
    for (int ay = ystart; ay != yend; ay += ychange) {
      for (int ax = xstart; ax != xend; ax += xchange) {
        /* Current (best) guess. */
        int v = (*ann)[ay][ax];
        int xbest = INT_TO_X(v), ybest = INT_TO_Y(v);
        int dbest = (*annd)[ay][ax];
```


C. Barnes, E. Shechtman, A. Finkestein, D. Goldman, "PatchMatch. A Randomized Correspondence Algorithm for Structural Image Editing," in *ACM Transactions on Graphics (TOG)*, 2009

Детали алгоритма (2)

pm_minimal.cpp

```
/* Propagation: Improve current guess by trying instead correspondences from left and above (below and right on odd iterations). */
if ((unsigned) (ax - xchange) < (unsigned) aew) {
  int vp = (*ann)[ay][ax-xchange];
  int xp = INT_TO_X(vp) + xchange, yp = INT_TO_Y(vp);
  if ((unsigned) xp < (unsigned) bew) {
    improve_guess(a, b, ax, ay, xbest, ybest, dbest, xp, yp);
  }
}

if ((unsigned) (ay - ychange) < (unsigned) aeh) {
  int vp = (*ann)[ay-ychange][ax];
  int xp = INT_TO_X(vp), yp = INT_TO_Y(vp) + ychange;
  if ((unsigned) yp < (unsigned) beh) {
    improve_guess(a, b, ax, ay, xbest, ybest, dbest, xp, yp);
  }
}

/* Random search: Improve current guess by searching in boxes of exponentially decreasing size around the current best guess. */
int rs_start = rs_max;
if (rs_start > MAX(b->w, b->h)) { rs_start = MAX(b->w, b->h); }
for (int mag = rs_start; mag >= 1; mag /= 2) {
  /* Sampling window */
  int xmin = MAX(xbest-mag, 0), xmax = MIN(xbest+mag+1,bew);
  int ymin = MAX(ybest-mag, 0), ymax = MIN(ybest+mag+1,beh);
  int xp = xmin+rand()%(xmax-xmin);
  int yp = ymin+rand()%(ymax-ymin);
  improve_guess(a, b, ax, ay, xbest, ybest, dbest, xp, yp);
}

(*ann)[ay][ax] = XY_TO_INT(xbest, ybest);
(*annd)[ay][ax] = dbest;
```


C. Barnes, E. Shechtman, A. Finkestein, D. Goldman, "PatchMatch. A Randomized Correspondence Algorithm for Structural Image Editing," in *ACM Transactions on Graphics (TOG)*, 2009

Содержание

- Introduction
- Guided Filter
- PatchMatch
- **Closed-form Matting**
- Nonlocal Matting
- Alpha Flow
- Conclusion

Предположение о природе изображений

Основное уравнение matting'a:

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

$$\rightarrow \alpha = \frac{I}{F-B} - \frac{B}{F-B} = aI - b$$

Предположим, что коэффициенты a, b локально постоянны (окрестности 3×3 в авторской реализации)

A. Levin, D. Lischinski, Y. Weiss, "A Closed-form Solution to Natural Image Matting," in *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, 2008

Matting Laplacian (1)

Предположив локальную линейность α, мы можем построить функционал, экстремум которого будет решением:

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in \omega_j} (\alpha_j - a_j I_i - b_j)^2 + \epsilon a_j^2 \right)$$

Тогда

$$\alpha = \arg\min_{\alpha} J(\alpha) = \arg\min_{\alpha} [\min_{a,b} J(\alpha, a, b)]$$

A. Levin, D. Lischinski, Y. Weiss, "A Closed-form Solution to Natural Image Matting," in *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, 2008

Matting Laplacian (2)

$$\min_{a,b} J(\alpha, a, b) = J(\alpha, argmin_{a,b} J(\alpha, a, b))$$

$$argmin_{a,b} J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in \omega_j} (\alpha_i - a_j I_j - b_j)^2 + \epsilon a_j^2 \right) =$$

$$= \sum_k \| G_k \begin{bmatrix} a_k \\ b_k \end{bmatrix} - \vec{\alpha}_k \|^2$$

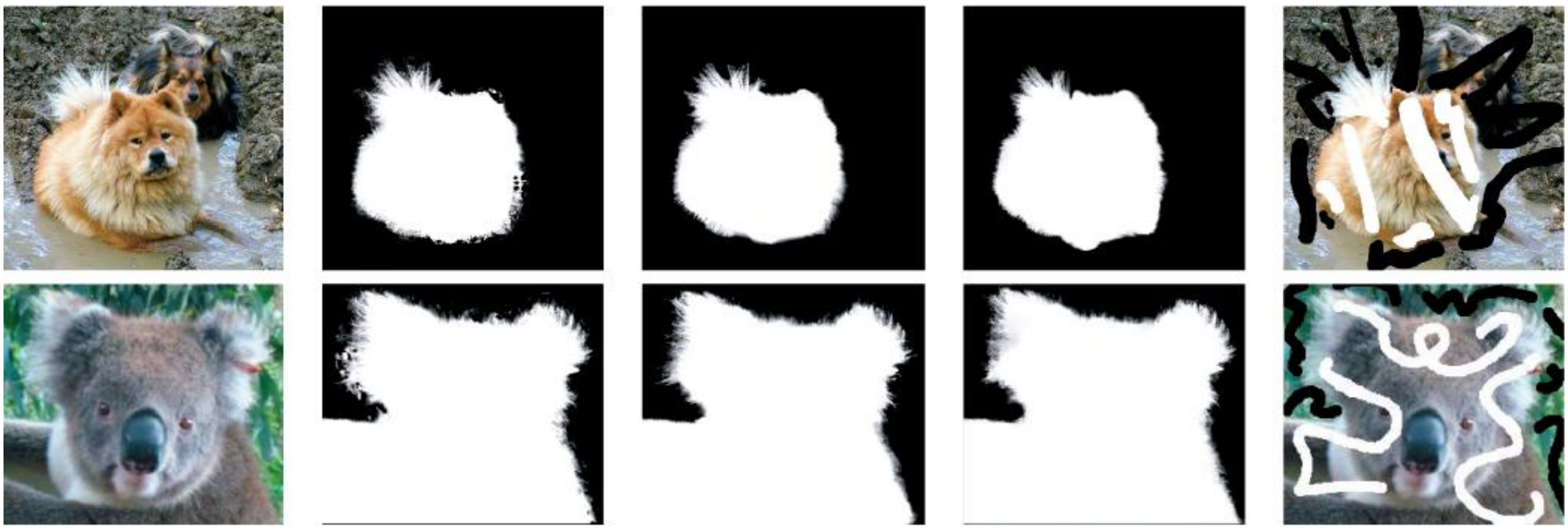
$$\rightarrow J(\alpha) = \sum_k \vec{\alpha}_k^T \hat{G}_k^T \hat{G}_k \vec{\alpha}_k$$

где $\hat{G}_k = I - G_k (G_k^T{}' G_k)^{-1} G_k^T$

$$\rightarrow J(\vec{\alpha}) = \vec{\alpha}^T L \vec{\alpha}$$

A. Levin, D. Lischinski, Y. Weiss, "A Closed-form Solution to Natural Image Matting," in *(PAMI)*, 2008

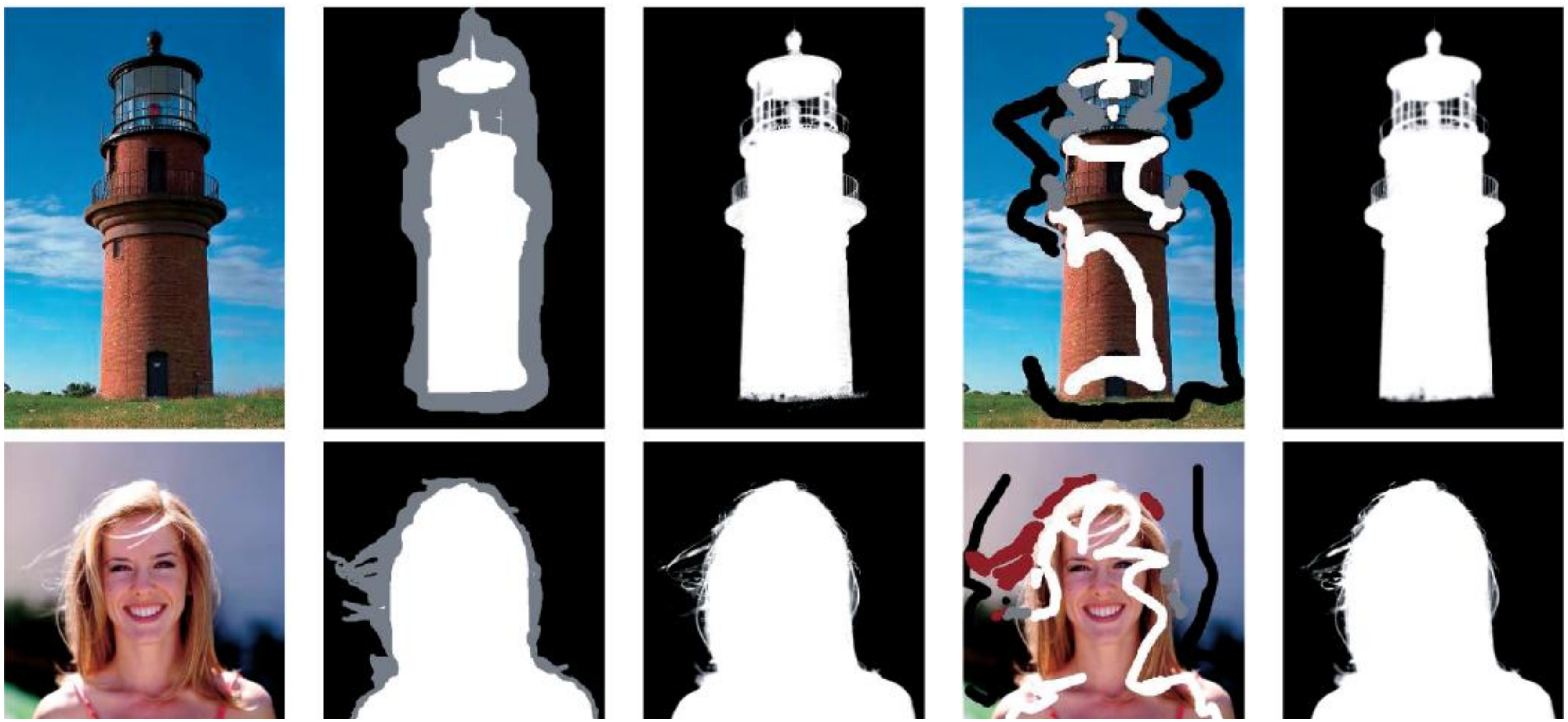
Results (1)



Input image | Bayesian Matting | Poisson matting | Closed-form matting | Scribbles

A. Levin, D. Lischinski, Y. Weiss, "A Closed-form Solution to Natural Image Matting," in *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, 2008

Results (2)



Input image | Trimap | Bayesian matting | Scribbles | Closed-form matting

A. Levin, D. Lischinski, Y. Weiss, "A Closed-form Solution to Natural Image Matting," in *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, 2008

Results (3)

Input image

Matting result

Foreground reconstruction

Background reconstruction

New background

A. Levin, D. Lischinski, Y. Weiss, "A Closed-form Solution to Natural Image Matting," in *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, 2008

Содержание

- Introduction
- Guided Filter
- PatchMatch
- Closed-form Matting
- Nonlocal Matting
- **Alpha Flow**
- Conclusion

Предположение о природе движения в видео

M. Sindeev, A. Konushin, and C. Rother, "Alpha Flow for Video Matting," *Asian Conference on Computer Vision* (*ACCV*), 2013

Шаги Алгоритма (1)

1. Инициализация альфа потока значениями оптического потока для RGB
2. Обработка occlusions: построение цепочек (суперпикселей во времени)
3. Перерасчет значений прозрачности

M. Sindeev, A. Konushin, and C. Rother, "Alpha Flow for Video Matting," *Asian Conference on Computer Vision* (*ACCV*), 2013

Шаги Алгоритма (2)

4. Фильтрация (Guided Filter), чтобы подавить артефакты в виде больших полупрозрачных областей (в конце этот шаг может быть пропущен)
5. Вычисление альфа потока
6. Переход на шаг 2

M. Sindeev, A. Konushin, and C. Rother, "Alpha Flow for Video Matting," *Asian Conference on Computer Vision* (*ACCV*), 2013

Вычисление оптического потока (1)

$$E(V) =$$

$$= \sum_{x,y,t} (\alpha(x,y,t) - \alpha(x + V_x, y + V_y, t + 1))^2 +$$

$$+\lambda(|V_x|^2 + |V_y|^2)$$

Первое и третье слагаемое оптимизируются попеременно $E_D(V) + \frac{\lambda}{2\theta}(U - V)^2 + E_S(V)$

M. Sindeev, A. Konushin, and C. Rother, “Alpha Flow for Video Matting,” *Asian Conference on Computer Vision* (*ACCV*), 2013

Вычисление оптического потока (2)

Первое слагаемое:

PatchMatch-based Motion Estimation

Второе слагаемое:

Решение линейной системы уравнений (если я не ошибаюсь)

Вообще, говоря условие гладкости может быть включено и в PatchMatch [Besse 2012]

M. Sindeev, A. Konushin, and C. Rother, "Alpha Flow for Video Matting," *Asian Conference on Computer Vision* (*ACCV*), 2013

Обработка occlusions

Оптический поток вычисляется в обе стороны

Если вектора в обоих направлениях приблизительно совпадают (если я не ошибаюсь), то они образуют ненаправленное ребро в нашем графе

Попробуем соединять последовательные вектора в цепочки

M. Sindeev, A. Konushin, and C. Rother, "Alpha Flow for Video Matting," *Asian Conference on Computer Vision* (*ACCV*), 2013

Алгоритм построения цепочек

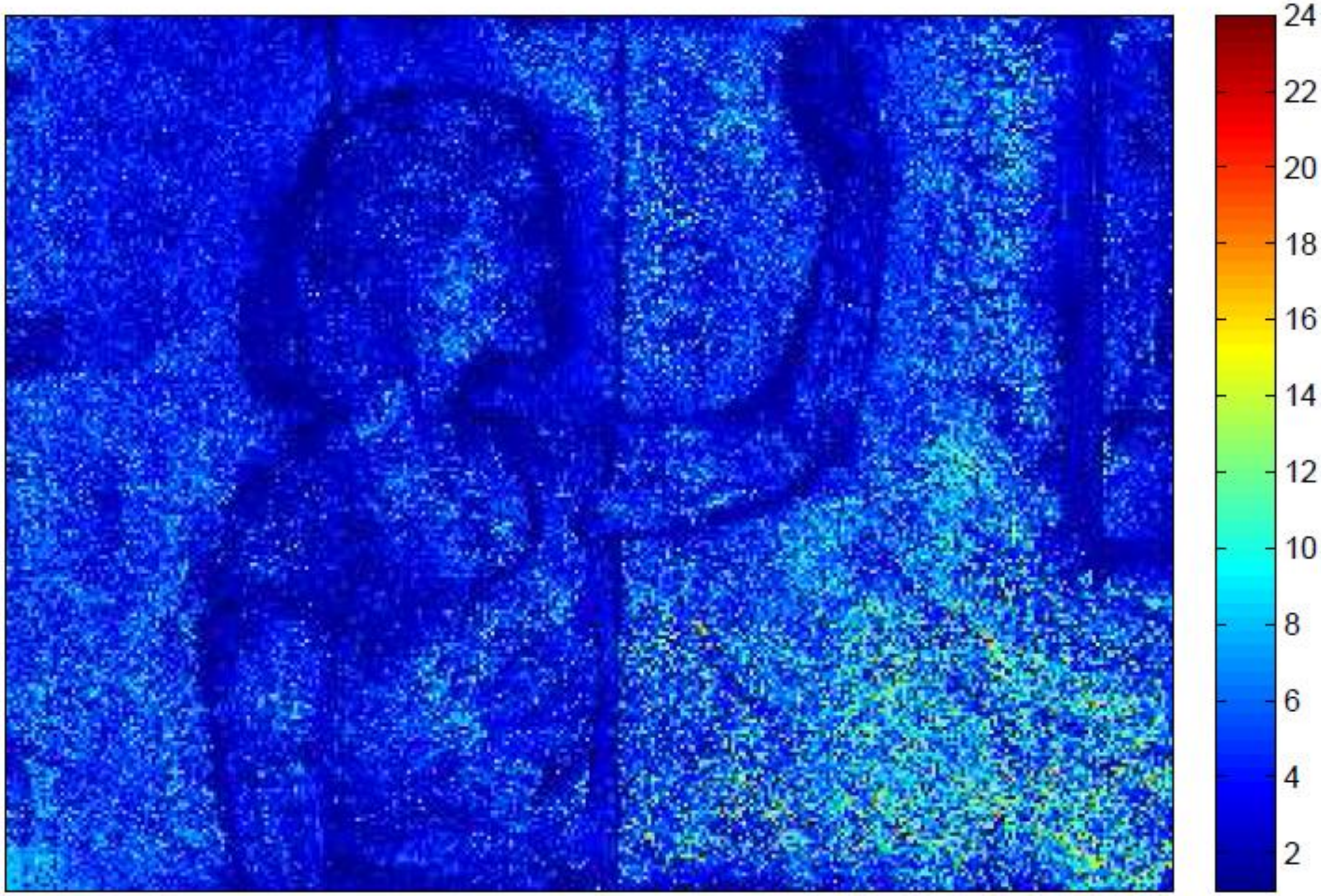
Цепочки должны начинаться и заканчиваться в occlusion'ах

На каждом шаге будем «жадно» выбирать две цепочки, которые соединяем, пока цена (приращение оптимизируемой функции) не станет отрицательна

Каждая цепочка дает вклад в зависимости от дисперсии цветов пикселей вдоль нее

M. Sindeev, A. Konushin, and C. Rother, "Alpha Flow for Video Matting," *Asian Conference on Computer Vision* (*ACCV*), 2013

Распределение длин цепочек (Avg = 2–4)



M. Sindeev, A. Konushin, and C. Rother, "Alpha Flow for Video Matting," *Asian Conference on Computer Vision* (*ACCV*), 2013

Иллюстрация объединения цепочек

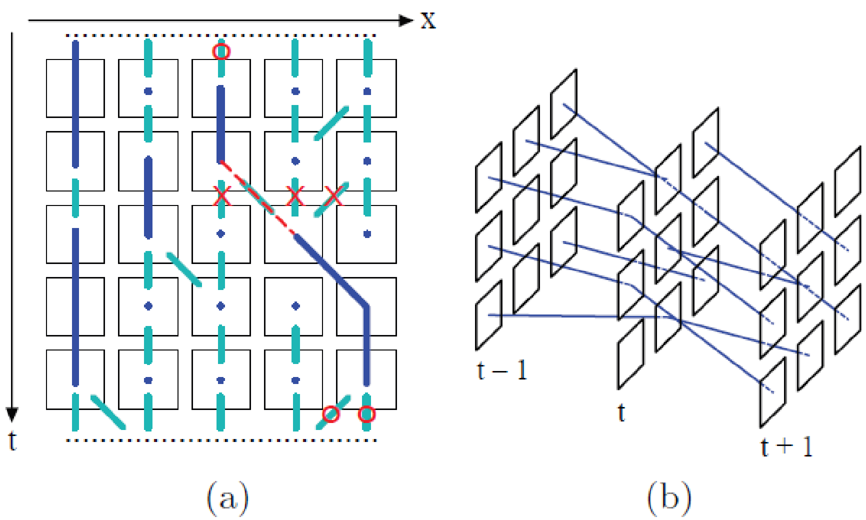


Fig. 6. (a) Superpixel merging procedure. Cyan lines are undirected flow vectors (candidates for merging the trajectories). If the dashed candidate is chosen for merging, adjacent candidates are deleted (marked with red 'x'). Merging gain values are then updated for the candidates adjacent to opposite ends of the merged trajectories (marked with red 'o'). (b) Temporal connections (blue lines) are hard constraints made up from temporal pixel grouping. Spatial connections (not shown) are the same for every frame and define a grid of soft constraints with edge weights based on Laplacian matrices of each frame.

M. Sindeev, A. Konushin, and C. Rother, "Alpha Flow for Video Matting," *Asian Conference on Computer Vision* (*ACCV*), 2013

Перерасчет значений прозрачности

Энергия, которую мы оптимизируем на этом шаге:

$$E(\alpha) =$$

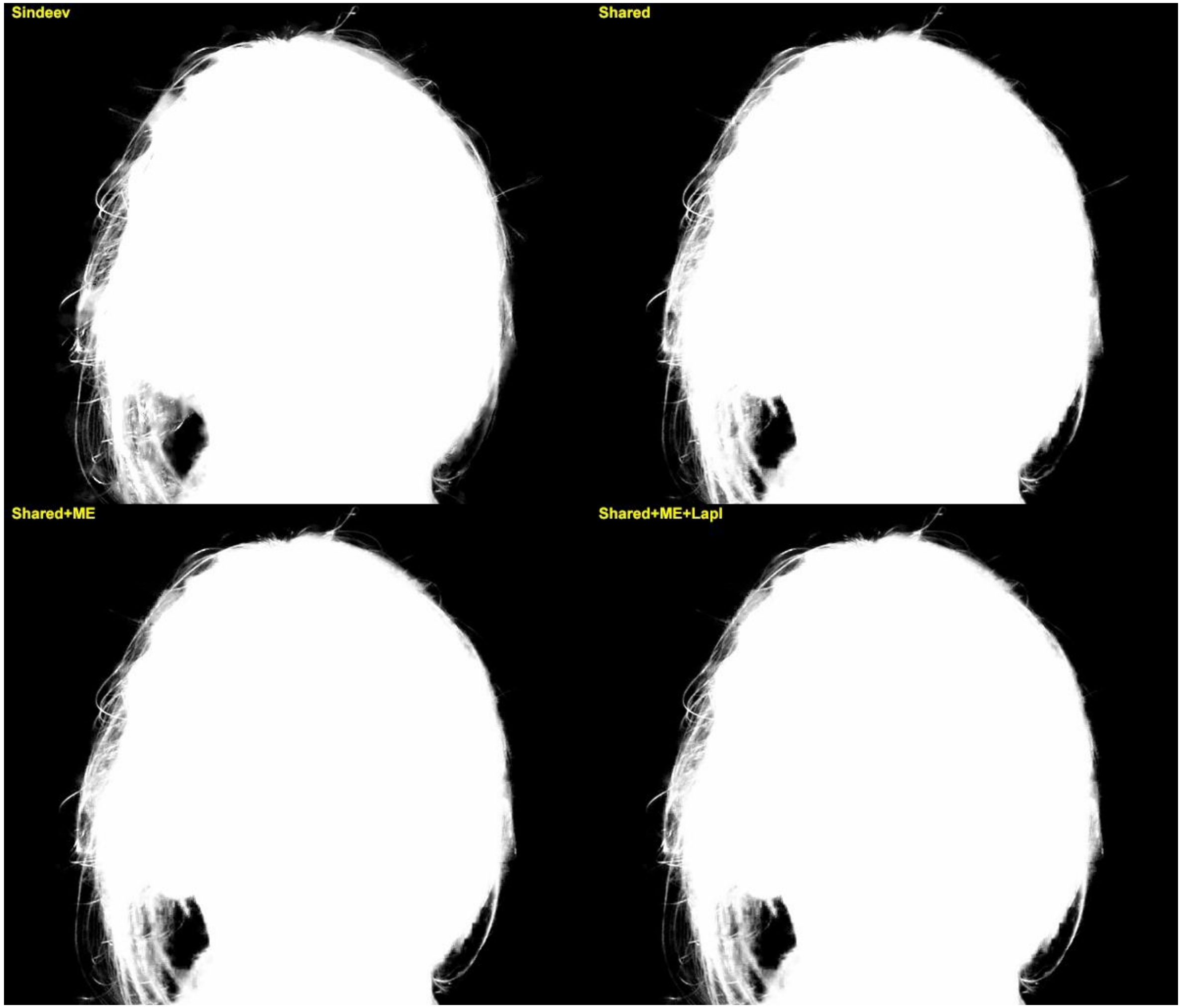
$$= \sum_{x,y,t} \omega_\alpha (\alpha(x,y,t) - \alpha(x + V_x, y + V_y))^2 +$$

$+\vec{\alpha}^T L_t \vec{\alpha}$, где L_t matting Laplacian для полученных суперпикселей

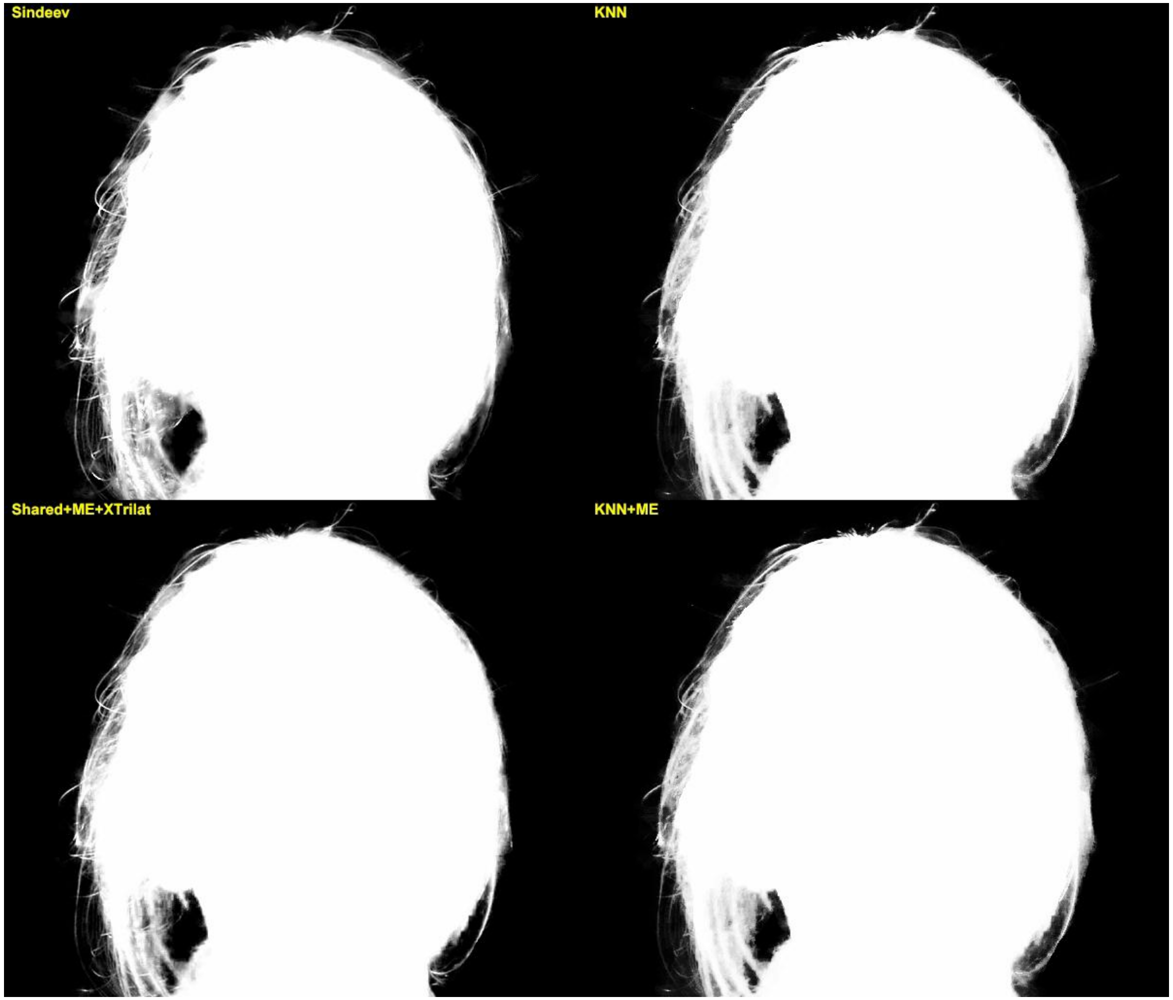
Содержание

- Introduction
- Guided Filter
- PatchMatch
- Closed-form Matting
- Nonlocal Matting
- Alpha Flow
- **Conclusion**

Matting algorithms comparison by M. Erofeev (1)



Matting algorithms comparison by M. Erofeev (1)



Литература

1. M. Sindeev, A. Konushin, and C. Rother, "Alpha Flow for Video Matting," *Asian Conference on Computer Vision* (*ACCV)*, 2013.
2. K. He, J. Sun, X. Tang, "Guided Image Filtering," in *European Conference on Computer Vision (ECCV), 2010*, pp. 1–14.
3. C. Barnes, E. Shechtman, A. Finkestein, D. Goldman, "PatchMatch. A Randomized Correspondence Algorithm for Structural Image Editing," in *ACM Transactions on Graphics (TOG)*, 2009, vol. 28, p. 24.
4. F. Besse, C. Rother, A. Fitzgibbon, J. Kautz, "PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation," in *British Machine Vision Conference (BMVC)*, 2012.
5. A. Levin, D. Lischinski, Y. Weiss, "A Closed-form Solution to Natural Image Matting," in *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, 2008, vol. 30, pp. 228–242.

Литература

6. P. Lee, Y. Wu, "Nonlocal Matting," in *IEEE Computer Vision and Pattern Recogntion (CVPR)*, 2011, pp. 2193–2200.
7. X. Bai, J. Wang, D. Simons, "Towards Temporally-coherent Video Matting," in *IEEE Mirage*, 2011, pp. 63–74.
8. I. Choi, M. Lee, Y.W. Tai, "Video Matting Using Multi-Frame Nonlocal Matting Laplacian," in *European Conference on Computer Vision (ECCV)*, 2012.