Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

**Отчёт о лабораторной работе №4**

**Дисциплина:** Методы и средства цифровой обработки информации

Выполнил студент гр. 13541/1 А.В. Пузанов

(подпись)

Руководитель Н.А. Абрамов

(подпись)

« » 2017 г.

Санкт – Петербург

2017

**Программа работы:**

1. Взять видеоролик, извлечь из него два последовательных кадра
2. Вычислить смещение для кадра t-1 относительно t, используя одну из модификаций

Параметры поиска:

Размер блока: 8х8

Stride: 4

Search Window: 40x40

1. В качестве результата представить изображение t-1 с отображенными поверх него векторами смещения
2. Восстановить из изображения t-1 изображение t

**Выполнение работы:**

Алгоритм работы:

1. Изображение t-1 разбивается на блоки размера bSize с шагом Stride
2. Для каждого блока находится область поиска на изображении t, размером Search Window
3. Так как алгоритм полного перебора требует значительных вычислительных затрат, то используется четырехшаговый поиск, для каждого блока возможные блоки-кандидаты внутри области поиска выбираются следующим образом:

Процесс поиска включает в себя следующие ступени.

1 – выбрать девять точек в зоне поиска. Вычислить искажения и найти точку с наименьшими искажениями. Если эта точка является центром зоны поиска, то перейти к шагу 3, иначе – к шагу 2.

2 - сместить центр в точку с наименьшими искажениями. Шаг остается неизменным.

3 – уменьшить шаг в два раза

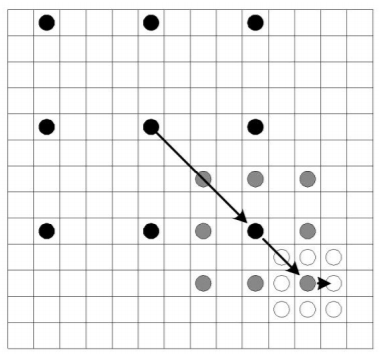


Рис 1. Четырехшаговый поиск

1. В качестве искомого выбирается блок с наименьшим значением метрики
2. Находится вектор смещения

В качестве метрики оценки искажения выбирается:

SEA:

Для того, чтобы вырезать кадр из видео, используем небольшой скрипт:

|  |
| --- |
| % открываем видео-файл  video = VideoReader('Stranger Things S01E01.mp4')    % узнаем параметры открытого видео  videoWidth = video.Width  videoHeight = video.Height  frameRate = video.FrameRate  numOfFrames = video.NumberOfFrames    % читаем пару кадров  % кадры нумеруются с 1  frameNo = 60;  frame1 = read(video, frameNo);  frame2 = read(video, frameNo + 1);    % сохраняем выбранные фреймы  imwrite(frame1,'frame1.jpg');  imwrite(frame2,'frame2.jpg'); |

В результате получаем два последовательных кадра:



Рис. 2. t кадр из кф. «Загадочные события».



Рис. 3. t+1 кадр из кф. «Загадочные события».

Для построения векторов используем следующий скрипт:

|  |
| --- |
| I1=imread('frame1.jpg');  I2=imread('frame2.jpg');  [Ysize, Xsize, kol]=size(I1)  for x=1:Xsize  for y=1:Ysize  Imag1(y,x)=I1(y,x,1)\*0.299+I1(y,x,2)\*0.587+I1(y,x,3)\*0.114;  Imag2(y,x)=I2(y,x,1)\*0.299+I2(y,x,2)\*0.587+I2(y,x,3)\*0.114;  end  end  imshow(Imag1);  bSize=8;  stride=4;  SWindow=40;  kol=1;  for x=bSize/2:stride:Xsize-bSize/2  for y=bSize/2:stride:Ysize-bSize/2  It=0;  f=0;  for i=x-bSize/2+1:1:x+bSize/2  for j=y-bSize/2+1:1:y+bSize/2  It=It+double(Imag1(j,i));  f=f+double(Imag2(j,i));  end  end  Size\_i=SWindow;  x\_center=x;  y\_center=y;  It\_1=-1;  if (abs(It-f)>5)  while Size\_i>=bSize  x\_search1=x-Size\_i;  if x\_search1<0  x\_search1=1;  end  x\_search2=x+Size\_i;  if x\_search2>Xsize  x\_search2=Xsize;  end  y\_search1=y-Size\_i;  if y\_search1<0  y\_search1=1;  end  y\_search2=y+Size\_i;  if y\_search2>Ysize  y\_search2=Ysize;  end  x\_c(1)=x\_center; y\_c(1)=y\_center;  x\_c(2)=x\_search1+bSize/2; y\_c(2)=y\_center;  x\_c(3)=x\_search1+bSize/2; y\_c(3)=y\_search2-bSize/2;  x\_c(4)=x\_center; y\_c(4)=y\_search2-bSize/2;  x\_c(5)=x\_search2-bSize/2; y\_c(5)=y\_search2-bSize/2;  x\_c(6)=x\_search2-bSize/2; y\_c(6)=y\_center;  x\_c(7)=x\_search2-bSize/2; y\_c(7)=y\_search1+bSize/2;  x\_c(8)=x\_center; y\_c(8)=y\_search1+bSize/2;  x\_c(9)=x\_search1+bSize/2; y\_c(9)=y\_search1+bSize/2;  for l=1:1:9  flag=0;  for i=x\_c(l)-bSize/2+1:1:x\_c(l)+bSize/2  for j=y\_c(l)-bSize/2+1:1:y\_c(l)+bSize/2  flag=flag+double(Imag2(j,i));  end  end  if (It\_1==-1)|(abs(It-flag)<abs(It-It\_1))  It\_1=flag;  x\_center=x\_c(l);  y\_center=y\_c(l);  end  end  Size\_i=fix(Size\_i/2);  end  hold on  end  plot([x x\_center], [y y\_center])  MasVect(1,kol)=x;  MasVect(2,kol)=x\_center;  MasVect(3,kol)=y;  MasVect(4,kol)=y\_center;  kol=kol+1;  end  end |

Полученный результат:

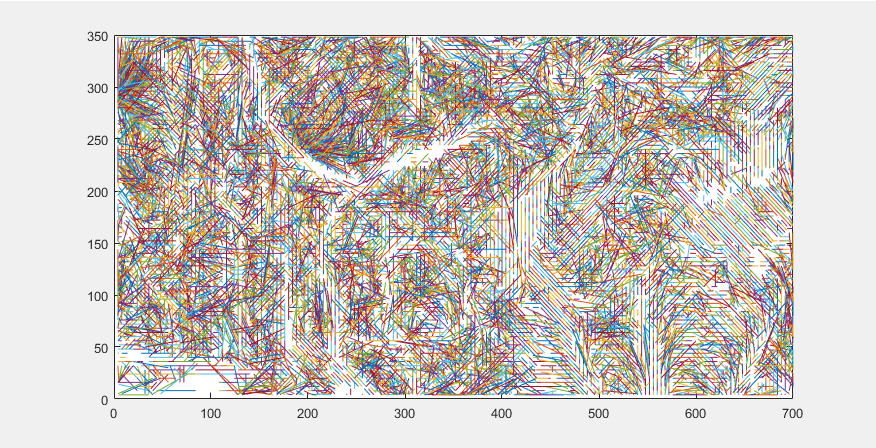


Рис 4. Векторы смещения

Восстановим изображение по векторам и посмотрим на результат. Для этого продолжим выполнение скрипта следующими строчками:

|  |
| --- |
| for l=1:1:kol-1  for i=-bSize/2+1:1:bSize/2  for j=-bSize/2+1:1:bSize/2  if ((MasVect(4,l)+j)<=Ysize)&((MasVect(3,l)+j)<=Ysize)&((MasVect(2,l)+i)<=Xsize)&((MasVect(1,l)+i)<=Xsize)  ImagChange(MasVect(4,l)+j,MasVect(2,l)+i)=Imag1(MasVect(3,l)+j,MasVect(1,l)+i);  end  end  end  end  imwrite(ImagChange, ’result.jpg'); |

Полученные результаты:



Рис 5. Восстановленное изображение

**Выводы:**

Алгоритм полного перебора для поиска векторов смещения требует значительных вычислительных затрат, однако обеспечивает высокую точность полученных результатов. Используя модификацию поиска – четырехшаговый поиск мы теряем в точности полученных результатов в пользу уменьшения вычислительных затрат.