Project Phase #1

Student: Yu-Hsuan, Chuang
ID: 1211219305
Group members:
Lei, Guo
Yiqian, Zhang
Mihir, Thakkar
Srujan, Kanteti
Rubinder, Singh

Abstract

This project aims to extract the videos data information. The main goal of Task 1 is to cut the frame images into cells and compute the video color histograms. Task 2 is to extract the SIFT vectors of each cells by using the SIFT function in the SIFT library. Finally, Task 3 is to extract the motion in the videos and output the motion vectors.

Keywords: Histogram, SIFT, ffmpeg, Motion vector

Task1

- Introduction
  - I. Terminology

    Histogram
  - II. Goal disruption

    The goal is to cut the frame image into cells and compute the colors histograms.

- Description of the proposed solution/implementation
  - I. Code
    - i. List folder contents by using *dir* and read the corresponding mp4 file. The condition branching (if, else if) is for printing the output result of the video file v_i.

```
list = dir('*.mp4');
for W=1:length(list)
    v = VideoReader(list(W).name);
    if strcmp(list(W).name,'1R.mp4')
        filename = '1';
    elseif strcmp(list(W).name, '2R.mp4')
        filename = '2';
    elseif strcmp(list(W).name, '3R.mp4')
        filename = '3';
    elseif strcmp(list(W).name, '4R.mp4')
        filename = '4';
    elseif strcmp(list(W).name, '5R.mp4')
        filename = '5';
    elseif strcmp(list(W).name, '6R.mp4')
        filename = '6';
    elseif strcmp(list(W).name, '7R.mp4')
        filename = '7';
    elseif strcmp(list(W).name, '8R.mp4')
        filename = '8';
    elseif strcmp(list(W).name, '9R.mp4')
        filename = '9';
    elseif strcmp(list(W).name, '10R.mp4')
```

```
        filename = '10';
    end
```

ii. Here I specify the resolution, `r`, equal to 2, it means that I cut the frame image into 2x2 cells. Then I create a text file in order to store the output value.

```
    r=2;
    fid = fopen('task1-output.txt', 'a');
```

iii. Read every frames of the video and turn the image into grey level image. Record the image size and cut the image into cells by using *mat2cell.*

```
    for index = 1:v.NumberOfFrames;
        frame = read(v,index);
        greyframe = rgb2gray(frame);
        [row column]=size(greyframe);
 cutframe=mat2cell(greyframe,(row/r)*ones(1,r),(column/r)*ones(1,r));
```

iv. Using *imhist* to extract the colors histograms of each cells and print the output result in the text file.

```
    for ROW = 1:r
          for COLUMN = 1:r
              hist = imhist(cutframe{ROW,COLUMN});
              fprintf(fid, '<%s, %d, %d, {%d', filename, index,
COLUMN+(ROW-1)*r, hist(1,1));
              for h = 2:256
                  fprintf(fid, ', %d', hist(h,1));
              end
              fprintf(fid, '}>\n\n');
          end
       end
    end
end
```

II. Output format and results

i. Format

Output: <*i, j, l,* {h_0 , h_1,……, h_{255}}>

where,

i: video file number

j: frame number

l: cell number

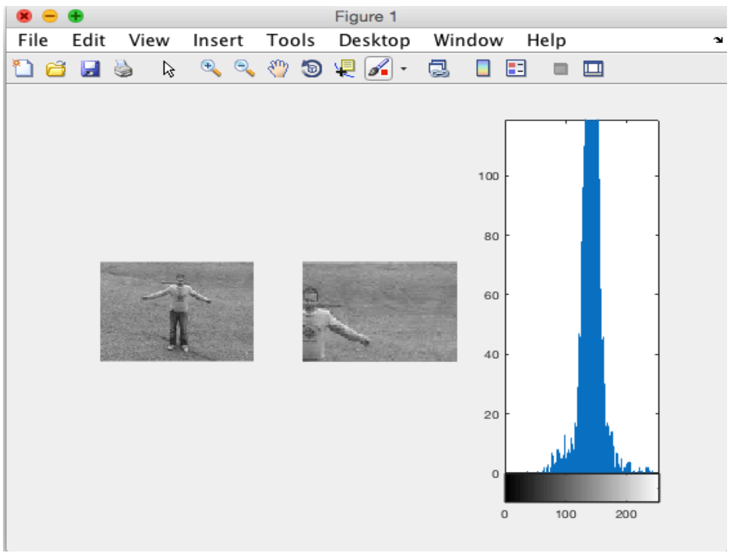
{h_0 , h_1,……, h_{255}}: the color histogram(from grey-level 0 ~ 255)

ii. Results

The following example output is the histogram result of the 2nd cell of the 1st frame in the video 10R.mp4.

```
<10, 1, 2, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
2, 0, 0, 2, 0, 1, 2, 3, 0, 0, 1, 2, 3, 7, 3, 6, 3, 0, 4, 4, 3, 4, 8, 8, 0, 7, 4, 4, 5, 2, 5, 0, 6, 7,
13, 5, 3, 5, 0, 7, 7, 8, 1, 7, 12, 0, 10, 7, 8, 4, 16, 17, 0, 16, 21, 29, 24, 31, 47, 0, 46,
57, 78, 83, 96, 110, 0, 123, 124, 148, 161, 147, 154, 0, 175, 174, 158, 174, 164,
178, 0, 171, 148, 169, 175, 160, 128, 142, 0, 109, 125, 99, 87, 62, 57, 0, 45, 46,
33, 30, 20, 16, 0, 17, 14, 16, 15, 10, 13, 0, 14, 8, 7, 14, 9, 4, 0, 2, 7, 3, 3, 7, 1, 0, 3, 2,
1, 1, 5, 1, 0, 2, 2, 2, 0, 1, 3, 0, 4, 2, 4, 0, 1, 4, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 2, 1, 0, 1, 0,
1, 1, 0, 0, 0, 0, 2, 0, 2, 0, 0, 2, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}>
```

The following picture is the histogram chart corresponding to the above example.



- System requirement/installation and execution instructions

  Matlab

- Related work

  Histogram

  A histogram is a representation of the number of pixels that have color in the range from 0 to 255 in an image. We can get the information of color distribution from the color histogram, then do some analysis or extract some useful data according to the histogram.

Task2

- Introduction
  - I. Terminology

    SIFT vector
  - II. Goal disruption

    The goal is to cut the frame image into cells and extract the SIFT vectors of each cell.

- Description of the proposed solution/implementation
  - I. Code **(it would run a few minute to get the result)**
    - i. (Same as task1) List folder contents by using *dir* and read the corresponding mp4 file. The condition branching (if, else if) is for printing the output result of the video file v_i.

```
list = dir('*.mp4');
for W=1:length(list)
   v = VideoReader(list(W).name);
   if strcmp(list(W).name,'1R.mp4')
      filename = '1';
   elseif strcmp(list(W).name, '2R.mp4')
      filename = '2';
   elseif strcmp(list(W).name, '3R.mp4')
      filename = '3';
   elseif strcmp(list(W).name, '4R.mp4')
      filename = '4';
   elseif strcmp(list(W).name, '5R.mp4')
```

```
        filename = '5';
    elseif strcmp(list(W).name, '6R.mp4')
        filename = '6';
    elseif strcmp(list(W).name, '7R.mp4')
        filename = '7';
    elseif strcmp(list(W).name, '8R.mp4')
        filename = '8';
    elseif strcmp(list(W).name, '9R.mp4')
        filename = '9';
    elseif strcmp(list(W).name, '10R.mp4')
        filename = '10';
    end
```

ii. (Same as task1) Specify the resolution, `r`, equal to 2, it means that I cut the frame image into 2x2 cells. Then I create a text file in order to store the output value.

```
    r=2;
    fid = fopen('task2-output.txt', 'a');
```

iii. (Same as task1) Read every frames of the video and turn the image into grey level image. Record the image size and cut the image into cells by using *mat2cell.*

```
    for index = 1:v.NumberOfFrames;
        frame = read(v,index);
        greyframe = rgb2gray(frame);
        [row column]=size(greyframe);
cutframe=mat2cell(greyframe,(row/r)*ones(1,r),(column/r)*ones(1,r));
```

iv. Using *sift* to extract the sift vectors ([frames, descr], frames = [x, y, scale, orientation], descr = $[a_1, \ldots, a_{128}]$) of each cells and print the output result in the text file.

```
                for ROW = 1:r
                    for COLUMN = 1:r
                        [frames,descr] = sift(cutframe{ROW,COLUMN},
'Verbosity', 1 ) ;
```

```
                    [m,n] = size(descr);
                    fprintf(fid, '<%s, %d, %d, {', filename, index,
COLUMN+(ROW-1)*r);
                    for q = 1:n
                        fprintf(fid, '[');
                        for w = 1:4
                            fprintf(fid, '%f, ', frames(w, q));
                        end
                        fprintf(fid, '%f', descr(1, q));
                        for e = 2:128
                            fprintf(fid, ', %f', descr(e, q));
                        end
                        fprintf(fid, ']');
                    end
                    fprintf(fid, '}>\n\n');
                end
            end
        end
    end
```

II. Output format and results

i. Format

Output:

$<i, j, l, \{[x, y, scale, orientation, a_1, \dots\dots, a_{128}]_1, \dots\dots$

$[x, y, scale, orientation, a\_1, \dots\dots, a\_128]_n\}>$

Where,

i: video file number

j: frame number

l: cell number

$[x, y, scale, orientation, a_1, \dots\dots, a_{128}]$*:* The SIFT vectors

ii. Results

The following example output is the SIFT vectors result of the 1st

cell of the 36th frame in the video 8R.mp4.

<8, 36, 1, {[19.098232, 45.831364, 0.941145, 4.303974, 0.054537, 0.008408, 0.011961, 0.015838, 0.018147, 0.030003, 0.078008, 0.241225, ……, 0.011532, 0.013135, 0.056885]}>

- System requirement/installation and execution instructions

  Matlab: should contain SIFT library

  Before running the code, the sift_compile.m should be run first.

- Related work

  SIFT vector

  SIFT (Scale-Invariant Feature Transform) is a computer vision algorithm that can detect the key points and get the description of the features in an image. The SIFT vector shows the position, scale, orientation and the keypoint descriptor.

Task3

- Introduction
  - I. Terminology

    Motion vector, ffmpeg
  - II. Goal disruption

    The goal is to cut the frame image into cells and extract the motion vectors of each cell. This task is run on the visual studio(C).

- Description of the proposed solution/implementation
  - I. Code (base on the ffmpegMV.cpp on the blackboard)

    This task I only run the one video at once.

```
for (int h = 1; h <= r; h++){
    for (int w = 1; w <= r; w++){
        if (mv->dst_x < (frame->width / r)*w && mv->dst_x >=
(frame->width / r)*(w - 1) && mv->dst_y < (frame->height / r)*h &&
mv->dst_y >= (frame->height / r)*(h - 1)){
```

```
            fprintf(fp, "1, %d, %d, {%d %d %d %d %d %d %d %d}\n",
video_frame_count, w + (h - 1)*r, mv->source, mv->w, mv->h, mv->src_x,
mv->src_y, mv->dst_x, mv->dst_y, mv->flags);
        }
    }
}
```

II. Output format and results

i. Format

Output: $<i, j, l, \{source, w, h, src_x, src_y, dst_x, dst_y\}>$

Where,

i: video file number

j: frame number

l: cell number

$\{source,\ w,\ h,\ src_x,\ src_y,\ dst_x,\ dst_y\}$*:* the motion vectors

ii. Results

The following example output is the SIFT vectors result of the 1st cell of the 2nd frame in the video 1R.mp4.

```
1, 2, 1, {-1 16 16 8 8 8 8 0}
1, 2, 1, {1 16 16 8 8 8 8 0}
1, 2, 1, {-1 16 16 24 8 24 8 0}
1, 2, 1, {1 16 16 24 8 24 8 0}
1, 2, 1, {-1 16 16 40 8 40 8 0}
.
.
.
```

- System requirement/installation and execution instructions

  Visual studio: should contain ffmpeg library

- Related work

  Motion vector

  Motion vector is the key data in the motion estimation process. Motion

vector estimation means for estimating a motion vector having a minimum difference between reference picture data and current picture data. The motion vector shows source, width of block, height of block, absolute source position, x, absolute source position, y, absolute destination position, x, absolute destination position, y.

- Conclusions

  During this project, I have learned some basic methods to extract useful data of a video that would help us doing further analysis.

- Bibliography

1. Vedaldi, Andrea, “An open implementation of the SIFT detector and descriptor”, in vedaldi2007open *UCLA CSD*, 2007
2. Thyagarajan, Kadayam S, “Still image and video compression with MATLAB”, in thyagarajan2011still, 2011
3. Mulani, Ms Arifa U and Mahajan, Ms Varsha V and Wadghule, Ms Swati B and Shirbhate, Radha S, “Features Preserving Video Event Detection using Relative Motion Histogram of Bag of Visual Words”, in mulanifeatures