Lab 5

IMAGE SEQUENCE PROCESSING

Blanca Bai

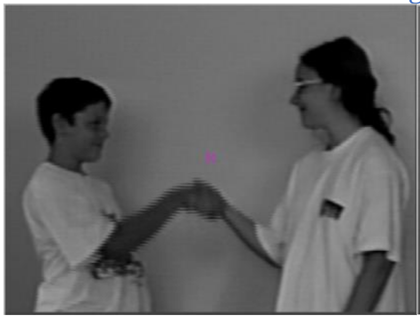
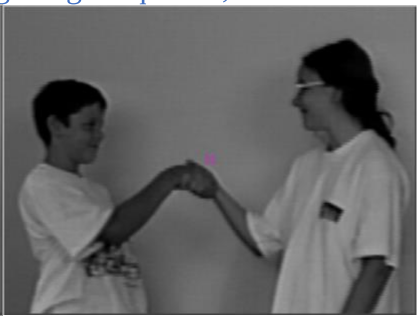
Cornell University Electrical and Compute Engineering

Table of Contents

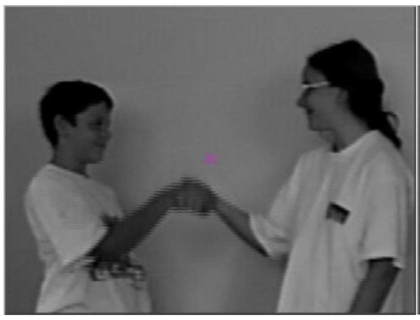
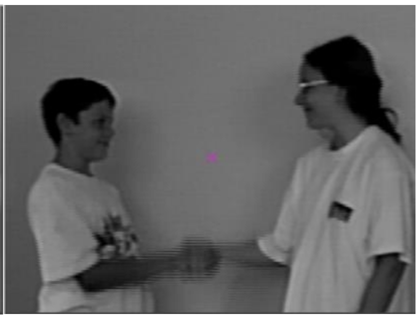
2. Feature Tracking

1.Display the original image sequence using vview:

This is a 15 frames hand shaking image sequence, some frames are:

Frame 1 Frame 5

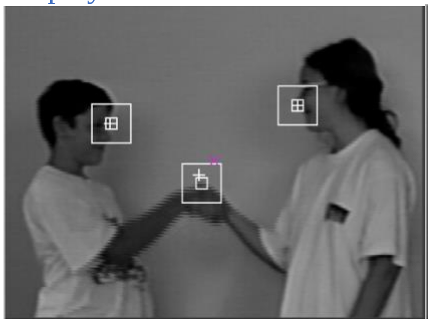
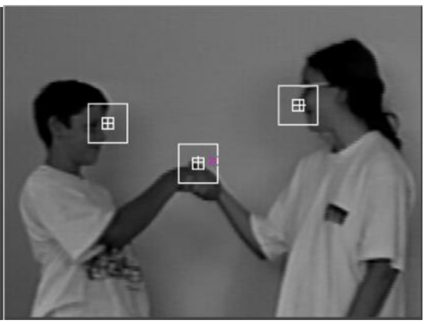
Frame 10 Frame 15

Figure1: the original hand.vs

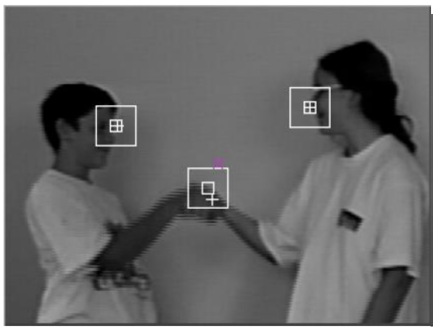
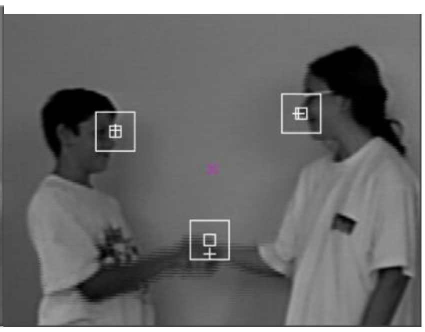
2. Track a set of features

```
vtrack if = hand.vs pf = loc1 h = 10 v = 10 of = trk.vs gf = trk.g
```

Display the results trk.vx with vview:

Frame 1 Frame 5

Frame 10 Frame 15

Figure2: the tracking hand.vs

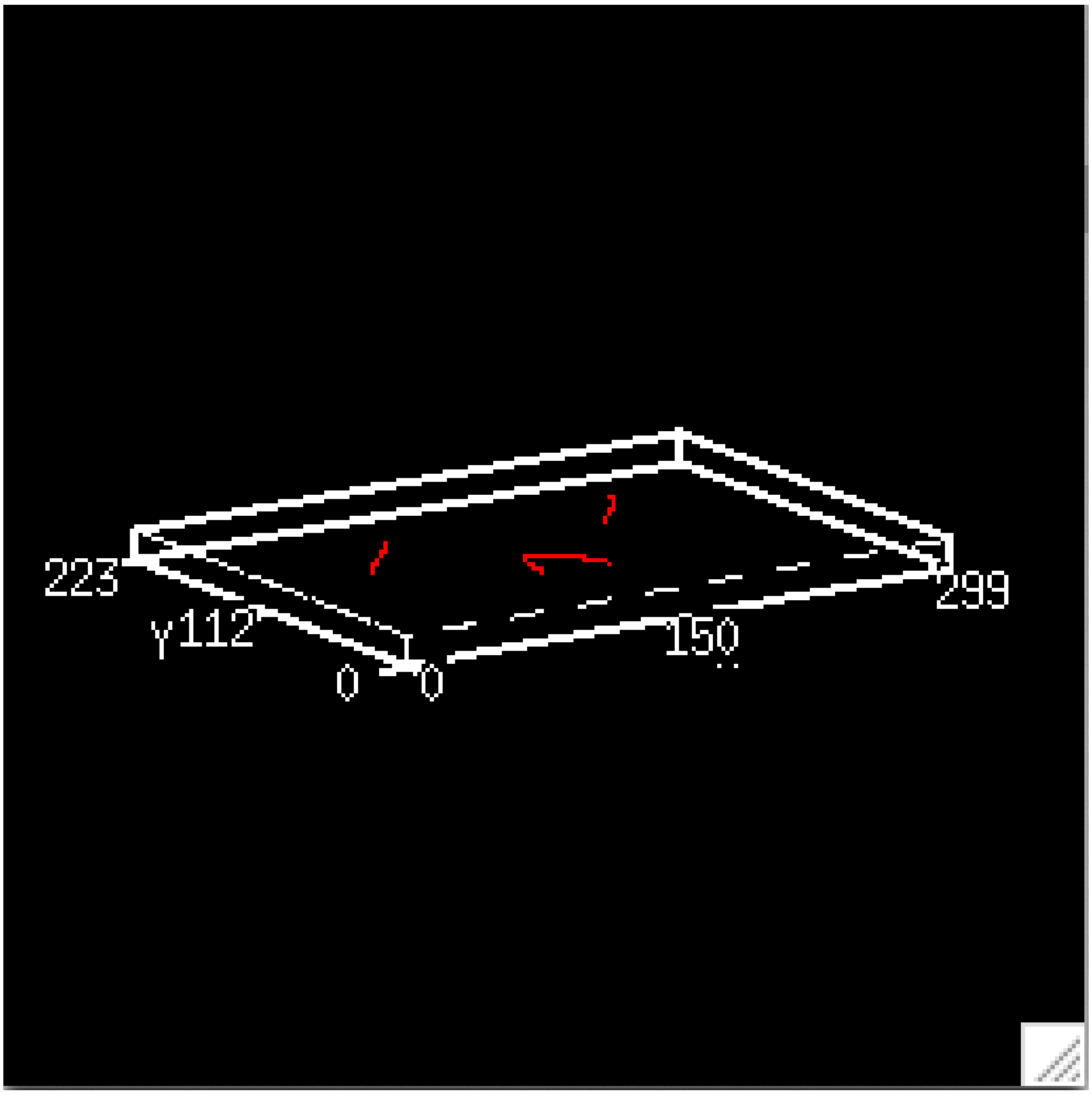


Figure3: the tracking result trk.g

The file trk.g generated by vtrack contains the space-time trajectories for the patches. Each trajectory is represented by a set of 3-D coordinates (x,y,z) where the z for this data indicates the time of the image frame.

3.Display the sequence as a single image

```
vtile trk.vs n = 3, 4 of = trkt.vs
```

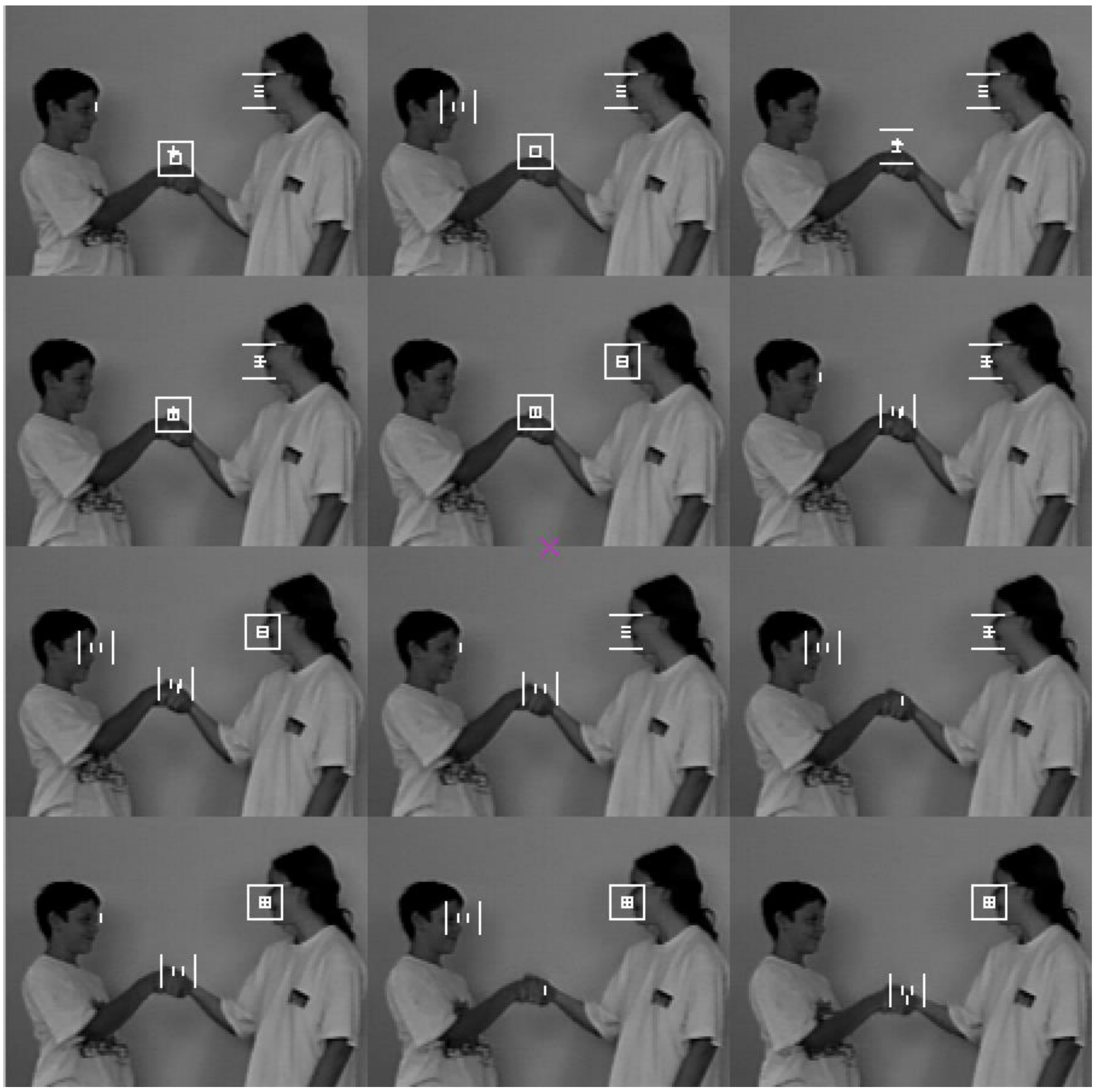
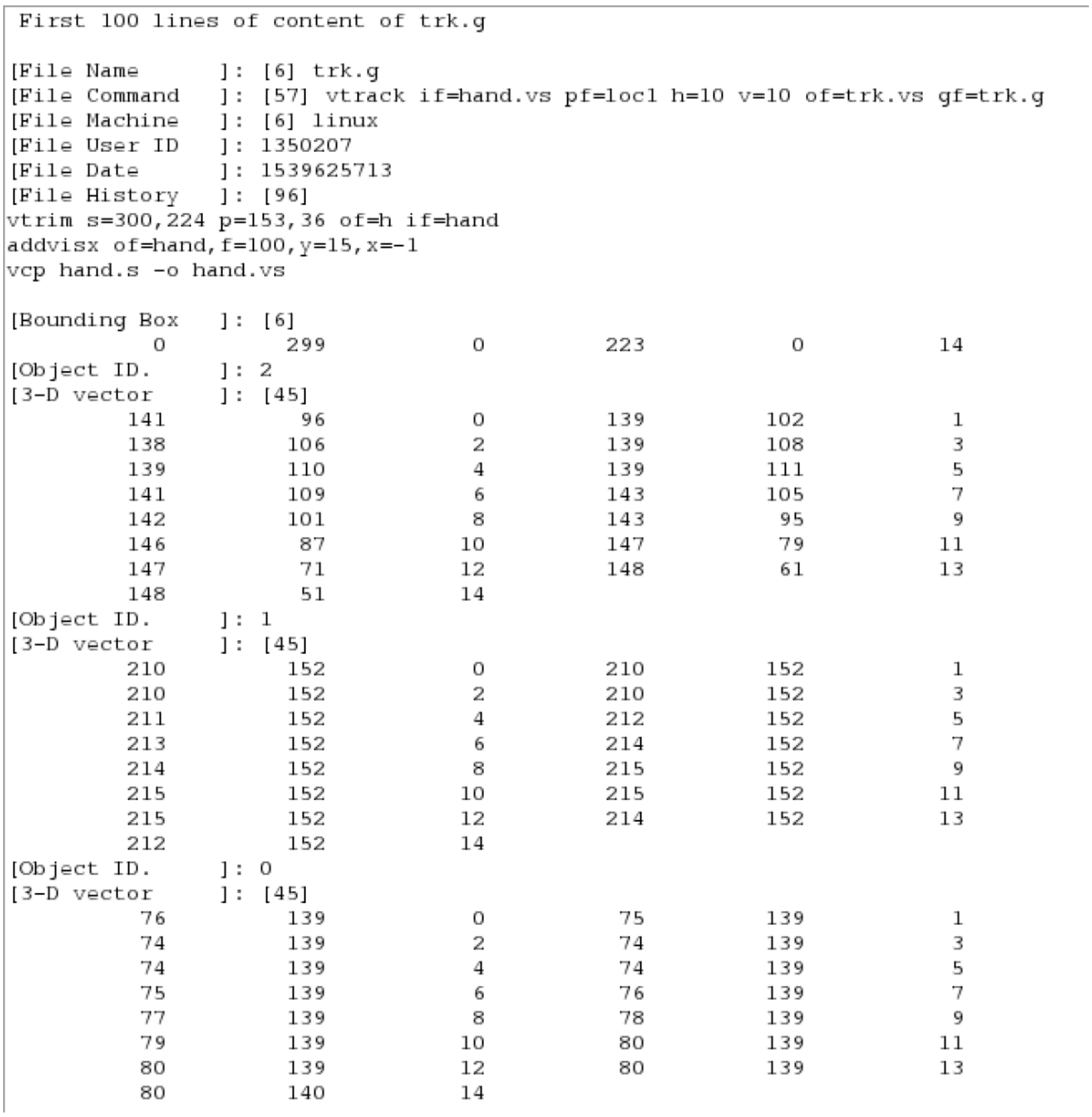


Figure4: the tracking result trkt.vs

4. Study the patch trajectories through time

```
1.  vpr trk.g
```
or

Tools ->Vx File Content



```
 First 100 lines of content of trk.g

[File Name       ]: [6] trk.g
[File Command    ]: [57] vtrack if=hand.vs pf=loc1 h=10 v=10 of=trk.vs gf=trk.g
[File Machine    ]: [6] linux
[File User ID    ]: 1350207
[File Date       ]: 1539625713
[File History    ]: [96]
vtrim s=300,224 p=153,36 of=h if=hand
addvisx of=hand,f=100,y=15,x=-1
vcp hand.s -o hand.vs

[Bounding Box    ]: [6]
           0          299            0          223            0           14
[Object ID.      ]: 2
[3-D vector      ]: [45]
         141           96            0          139          102            1
         138          106            2          139          108            3
         139          110            4          139          111            5
         141          109            6          143          105            7
         142          101            8          143           95            9
         146           87           10          147           79           11
         147           71           12          148           61           13
         148           51           14
[Object ID.      ]: 1
[3-D vector      ]: [45]
         210          152            0          210          152            1
         210          152            2          210          152            3
         211          152            4          212          152            5
         213          152            6          214          152            7
         214          152            8          215          152            9
         215          152           10          215          152           11
         215          152           12          214          152           13
         212          152           14
[Object ID.      ]: 0
[3-D vector      ]: [45]
          76          139            0           75          139            1
          74          139            2           74          139            3
          74          139            4           74          139            5
          75          139            6           76          139            7
          77          139            8           78          139            9
          79          139           10           80          139           11
          80          139           12           80          139           13
          80          140           14
```

Figure5: the tracking result content of trk.g

Use v3d to vview trk.g

```
v3d trk.g cf = v3.in
```

We can use J, k, h keys to control the 3D image, rotates X,Y,X axis to view.

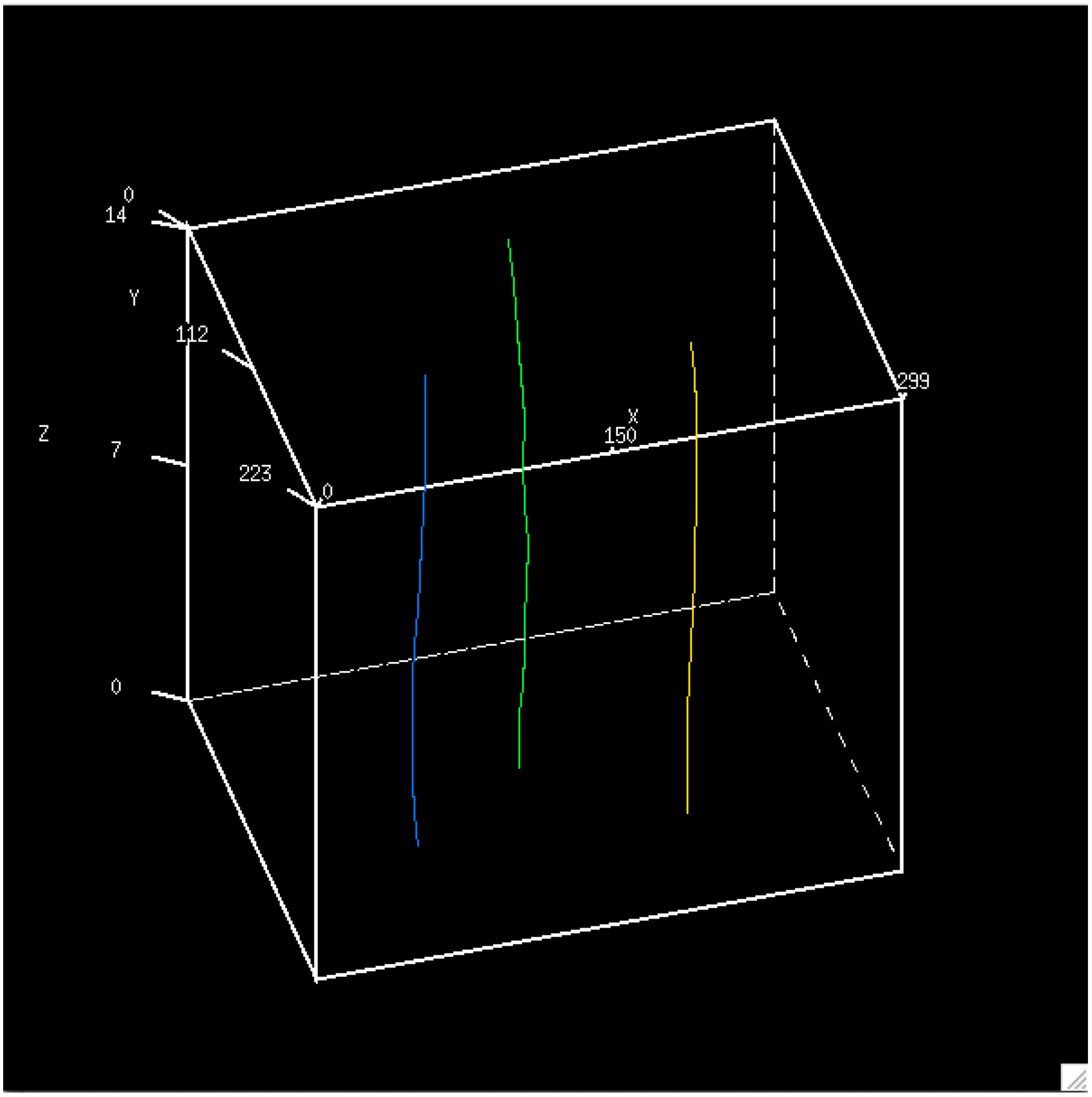


Figure6: the tracking result of trk.g under v3d

5. Test the tracking program with image set taxi.vs

5.1 Use vdview to identify at least 8 suitable tracking locations on the first image frame.

I select 10 points as tracking location, they are circled in orange dash line in the below image.

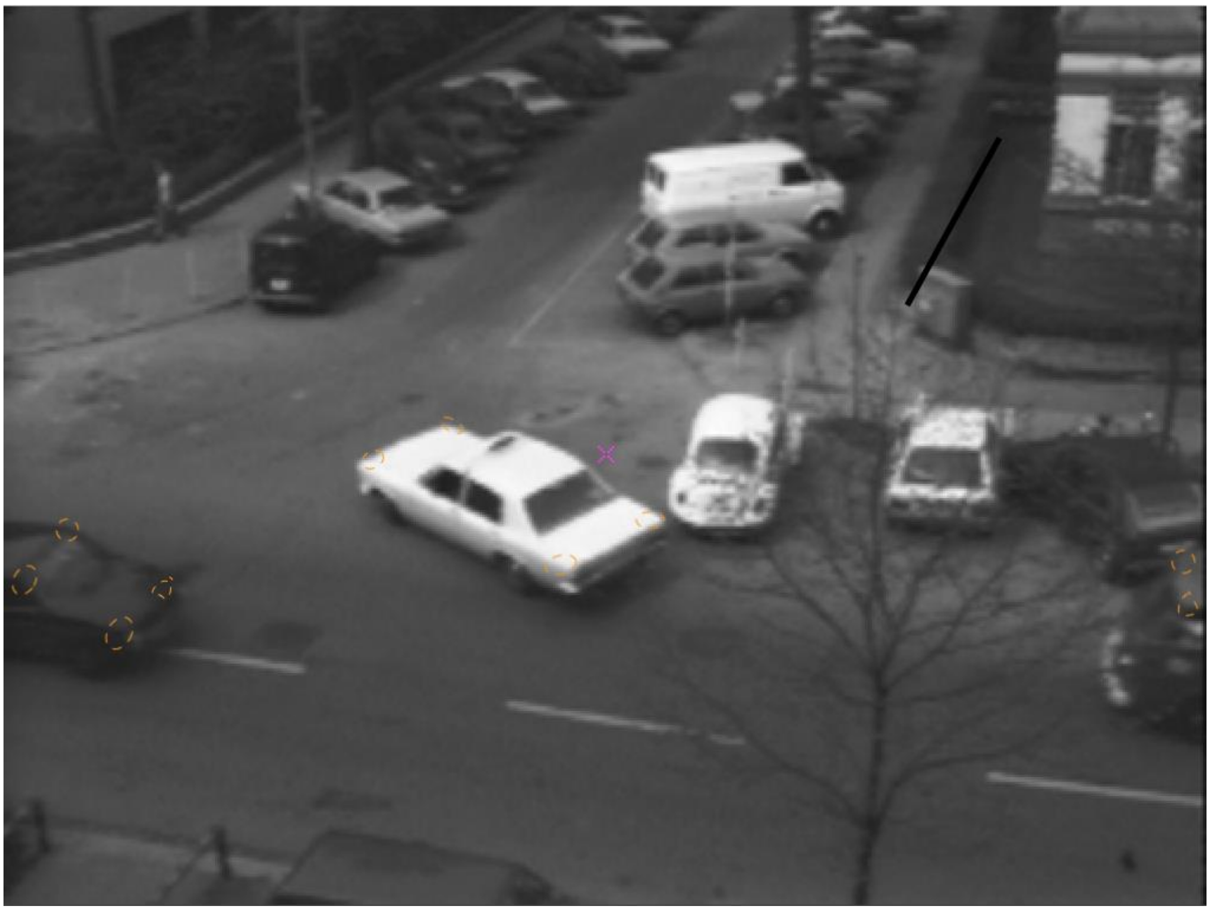
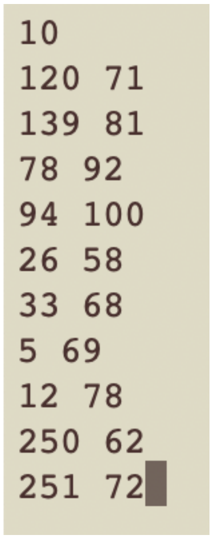


Figure7: my tracking location of taxi

5.2 Make a location file loc2 with the new locations (see the man page on vtrack for the file format).

The location of the 10 points I choose are as below:



```
10
120 71
139 81
78 92
94 100
26 58
33 68
5 69
12 78
250 62
251 72
```

Figure8: 10 tracking location points

5.3 Track your selected features with the vtrack program and appropriate parameters.

```
vtrack if = taxi.vs pf = loc2 h = 3 v = 3 of = trktaxi.vs gf = trktaxi.g
```

Because the image is so small, so I set both the horizontal and vertical search size to 3.

5.4 Create a file with the first 15 frames of the image sequence as a single tiled image.

(use vtile n=3,5 to get 15 frames and vxto -gif of=out to convert the image to gif.

```
vtile trktaxi.vs n = 3, 5 of = trktaxit.vs
vxto - gif if = trktaxit.vs of = giftrktaxit
```

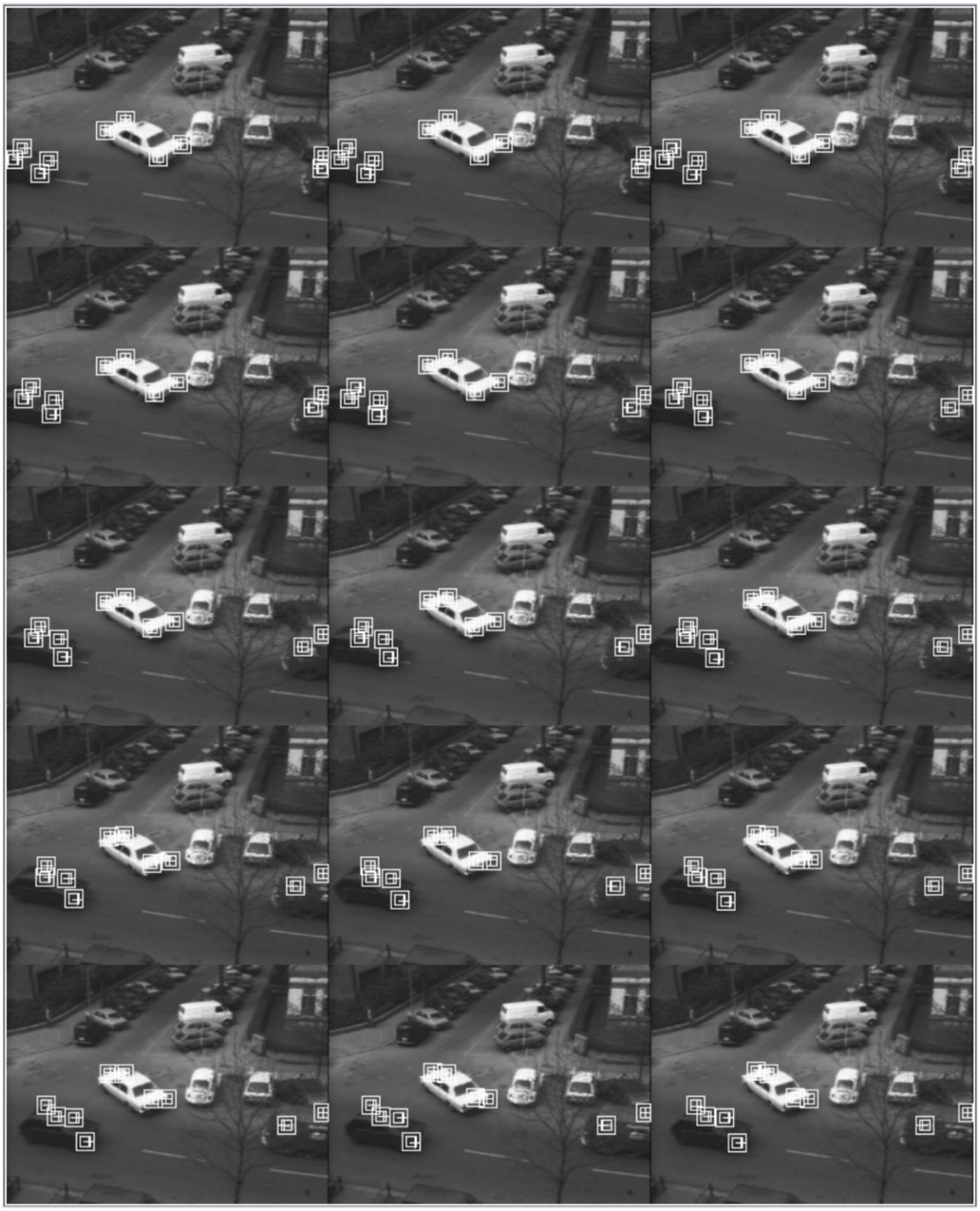


Figure9: tracking results gif

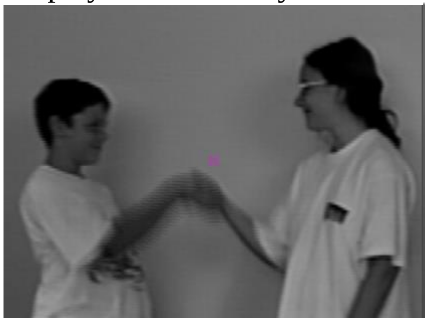
3.Temporal Domain Filter

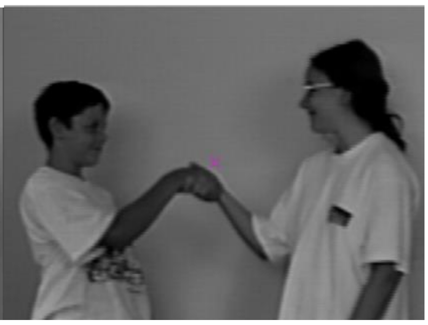
1. What does the temporal mean filter do?

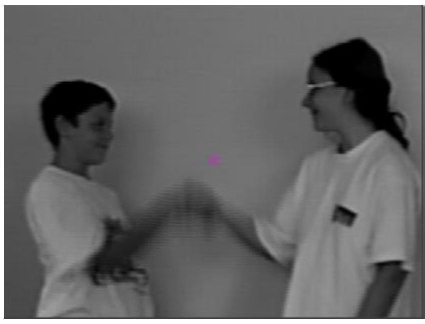
```
vcc vssum.c - o vssum
```

```
vssum hand.vs n = 3 of = hand.m
```

Display the hand.m by vview


Frame 1


Frame 5


Frame 10

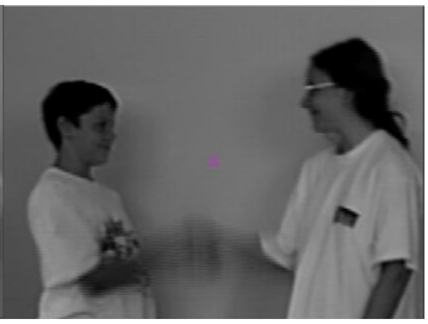

Frame 13

Figure10: tracking results gif

Temporal mean filter reduce the frames from 15 to 13. And also temporal mean filter creates the shaking ghost, druggy sort of effect, particular obvious in the hand shake area. Because it consider all of the pixels from the last few seconds to pull the median from.

2. What is the effect on a large window size?

```
vssum hand.vs n = 5 of = hand.m5
```

This time the window size is 5, and also I set window = 800
When window size increase, the number of frames was reduced to 11. the shake ghost become more obvious, shaking become more severe, and the color of hand is darker and the edge of handshake is clearer. It seems every frame is the overlap of its 5 previous frames, and when the handshake is slower, current frame is more similar to the overlap results of previous frames, so the overlap area is more large, so the color of overlap handshake is darker.

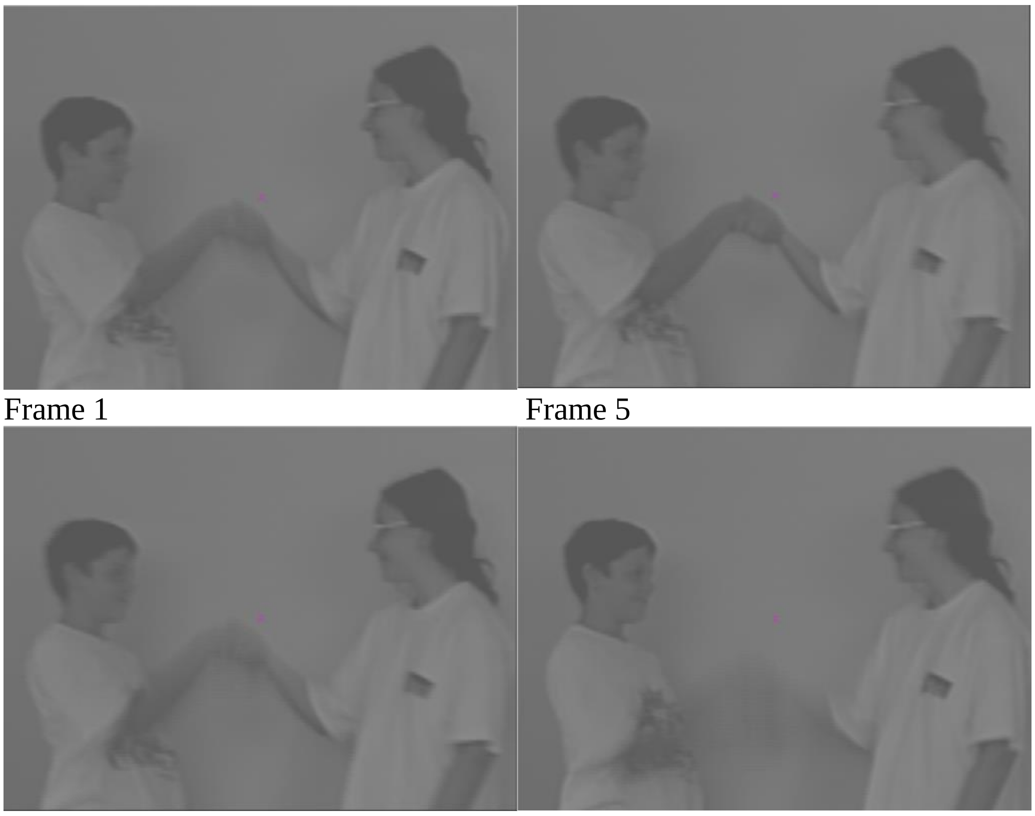


Figure11: large window size mean filter hand results

3. write a temporal median filter program

compute the median of each set of THREE consecutive image frames
Your program need only work with ONE window size of 3, you DO NOT have to make it work for other window sizes. You should only need to change the variable declaration and application specific sections of the program vssum.c. Your program should be simpler than vssum.c .

I use brute force search method to find the median pixel value of 3 frames: im.u[0][y][x], im.u[1][y][x], im.u[2][y][x].

One mistake I made is that I forget to override the value of n.
Because for median filter, we only need 3 frames. n, the variable to record how many frames in total, as the value we input in our command line, if we don't input, the default value of n is 1, for this program, we should fix n as 3.

```c
int main(argc, argv) int argc;
char * argv[]; {
    V3fstruct(im);
    V3fstruct(tm);
    int x, y, z; /* index counters                 */
    int n = 3; /* Number of frames    */
    int sum;
    int avg = (im.zhi + im.zlo) / 2;
    VXparse( & argc, & argv, par); /* parse the command line    */
    n = (NVAL ? atoi(NVAL) : 1); /* read n, default is n=1   */
    while (Vbfread( & im, IVAL, n)) {
        if (im.type != VX_PBYTE || im.chan != 1) { /* check format  */
            fprintf(stderr, "image not byte type\n");
            exit(1);
        }
        for (y = im.ylo; y <= im.yhi; y++) {
            for (x = im.xlo; x <= im.xhi; x++) {
                if (im.u[im.zlo][y][x] < im.u[im.zhi][y][x]) {
                    if (im.u[avg][y][x] < im.u[im.zlo][y][x]) {
                        im.u[0][y][x] = im.u[im.zlo][y][x];
                    } //1 0 2
                    if (im.u[avg][y][x] > im.u[im.zhi][y][x]) {
                        im.u[0][y][x] = im.u[im.zhi][y][x];
                    } //0 2 1
                    im.u[0][y][x] = im.u[avg][y][x]; //0 1 2
                }
                if (im.u[avg][y][x] < im.u[im.zhi][y][x]) {
                    im.u[0][y][x] = im.u[im.zhi][y][x];
                } //1 2 0
                if (im.u[avg][y][x] > im.u[im.zlo][y][x]) {
                    im.u[0][y][x] = im.u[im.zlo][y][x];
                } //2 0 1
            }
        }
        V3fwrite( & im, OVAL); /* write the oldest frame */
    }
    exit(0);
}
```

4. Test your program with the test image sequence **`hand.vs`** and comment on your results.

```
vcc vsmed.c - o vsmed
vsmed if = hand.vs of = vsmedhand.vs
vtile n = 3, 5 if = vsmedhand.vs of = vsmedhandt.vx
```

After median filter, the number of frames is reduced to 13. The ghost and shaking was less than mean filter.

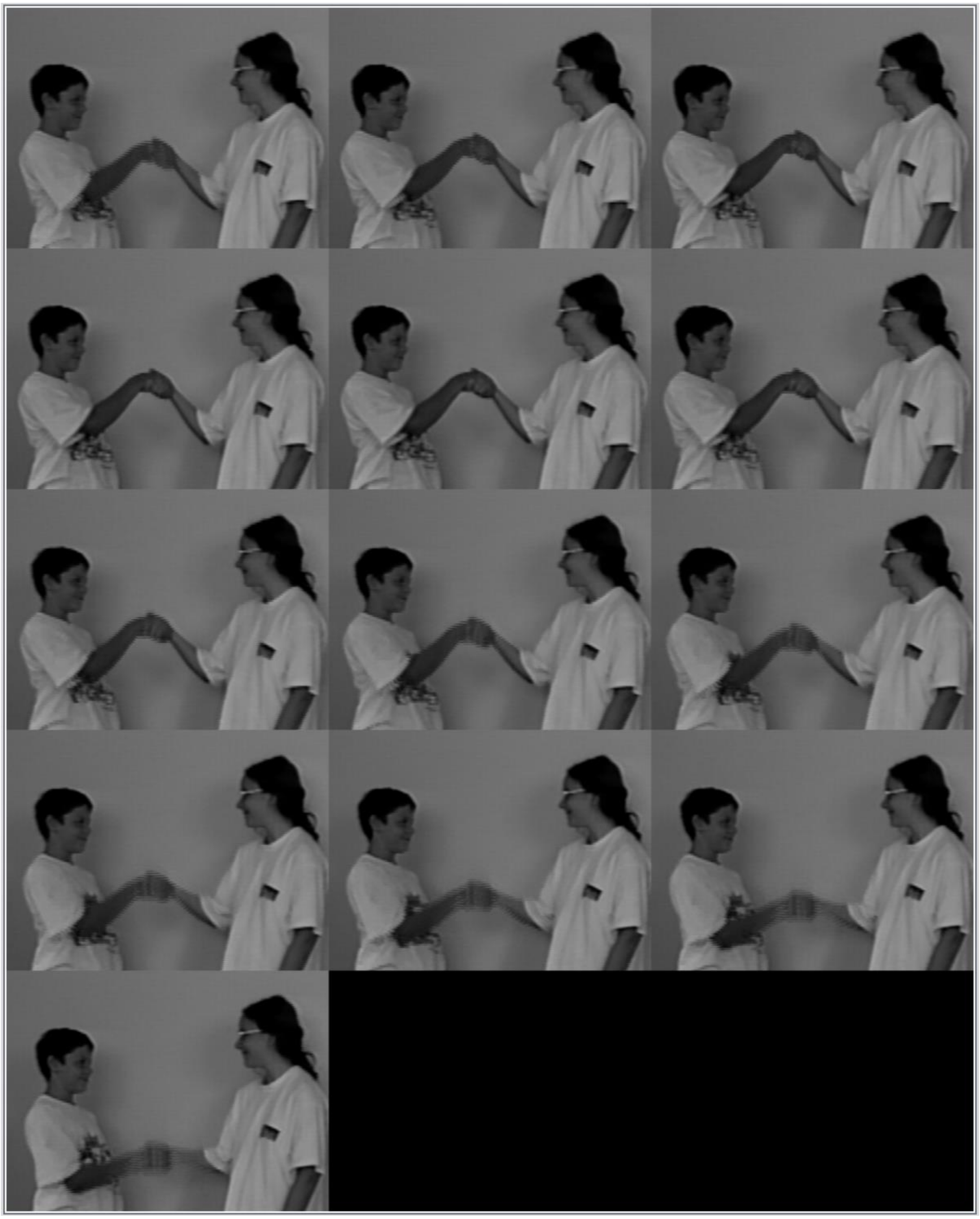


Figure12: median filter hand results

5. Compare the result of filtering the image sequence **`lb5.vs`** with both a *mean filter* with a window size of *three* and your *median filter*. Explain the results in each case.

```
vsmed if = lb5.vs of = vsmedlb5.vs
vssum lb5.vs n = 3 of = vssumlb5.vx
vtile n = 2, 3 if = vsmedlb5.vx of = vsmedlb5t.vx
```

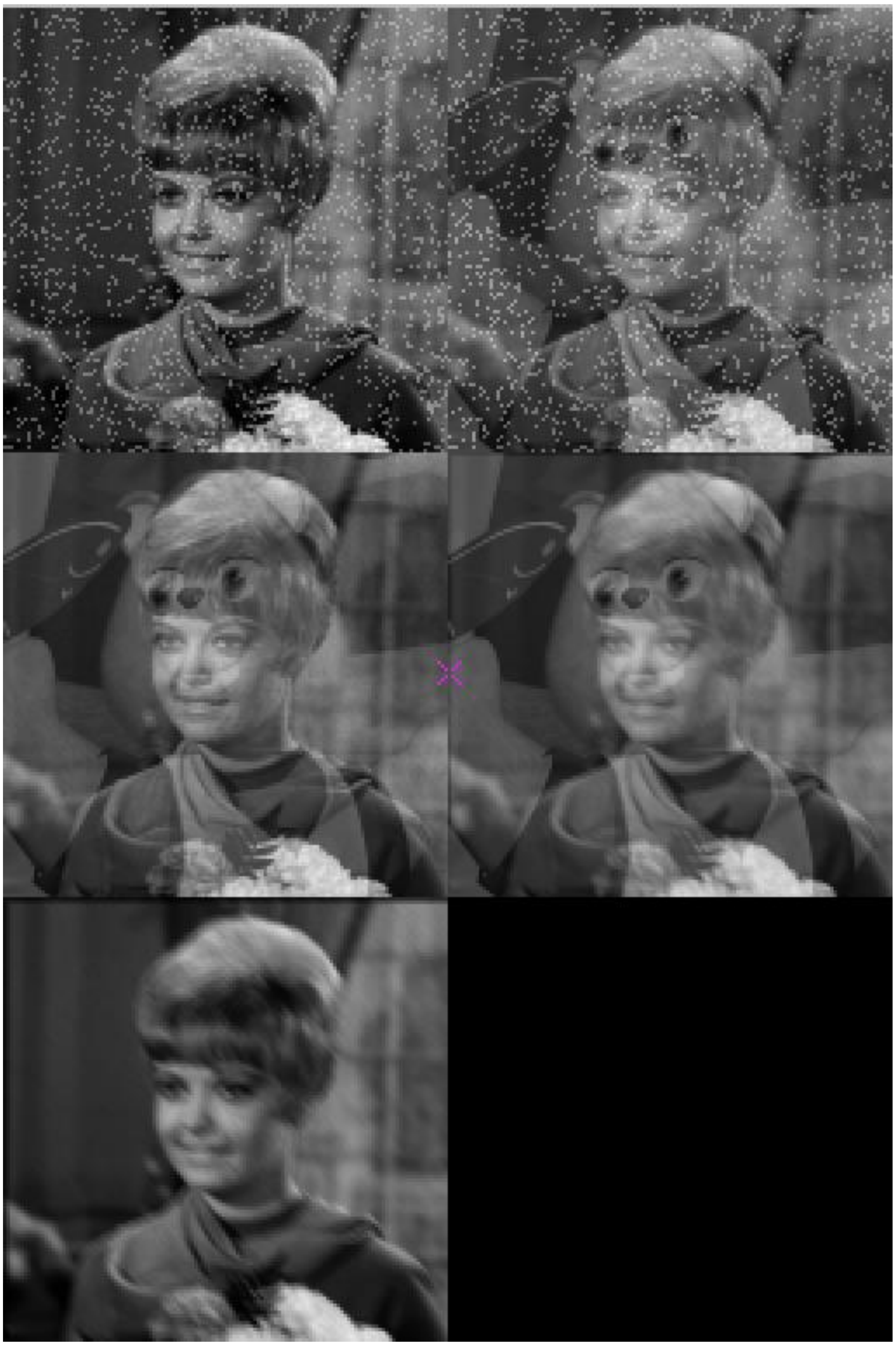


Figure13: mean filter lb5 results

After the mean filter, frames have overlaps with other frame.

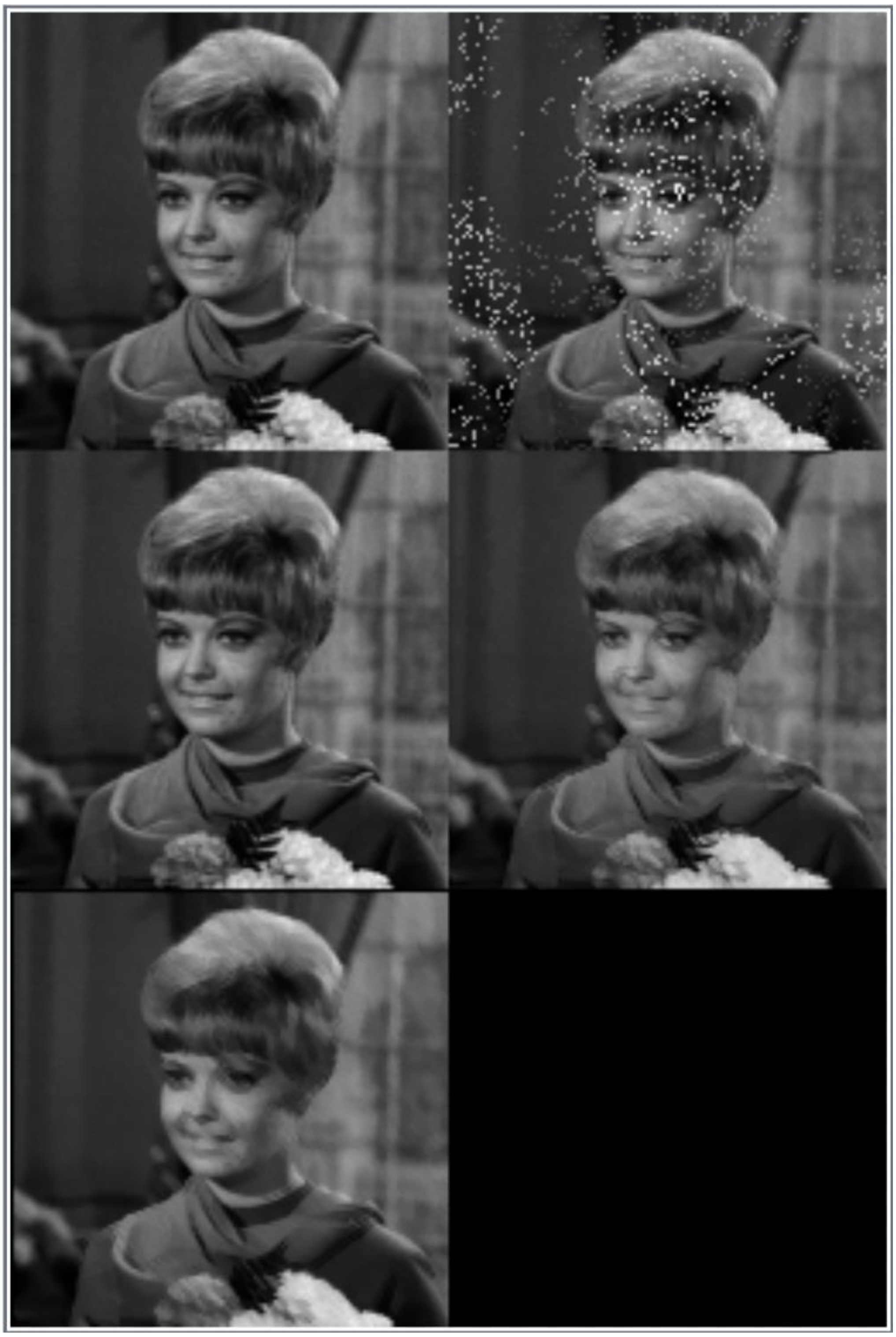


Figure14: median filter lb5 results

After the median filter, the number of frames was reduced to 5, and the noise in second frame was reduced, the teddy bear frame was removed.

3.Change Detection

1. Write a temporal binary difference filter program vsdif.c

First I get the difference of two frames: im.u[im.zhi][y][x], im.u[im.zlo][y][x],
Then if the difference is smaller my threshold, I set the frame im.u[0][x][y] = 128,

if the difference is bigger than my threshold, I set the frame im.u[0][x][y] = 255

one tip I think deserve mention is to set the threshold variable th as global variable input variable, so to change the threthold, we don’t need to change the variable in our program and compile again to see the results, we only input the th value in our command line.

```
VXparam_t par[] = /* command line structure              */
{
    {
        "if=", 0, " input file vsdif: compute temporal mean"
    }, {
        "of=", 0, " output file "
    }, {
        "th=", 0, " threshold "
    }, {
        0, 0, 0
    }
};
#define IVAL par[0].val# define OVAL par[1].val# define thVAL par[2].val int main(argc, argv) int argc;
char * argv[]; {
    V3fstruct(im);
    V3fstruct(tm);
    int x, y, z; /* index counters                 */
    int th;
    int n = 2; /* Number of frames to average    */
    int sum;
    VXparse( & argc, & argv, par); /* parse the command line    */
    th = (thVAL ? atoi(thVAL) : 1); /* read n, default is n=1   */
    while (Vbfread( & im, IVAL, n)) {
        if (im.type != VX_PBYTE || im.chan != 1) { /* check format  */
            fprintf(stderr, "image not byte type\n");
            exit(1);
        }
        for (y = im.ylo; y <= im.yhi; y++) {
            for (x = im.xlo; x <= im.xhi; x++) {
                if (im.u[im.zhi][y][x] - im.u[im.zlo][y][x] >= th || im.u[im.zlo][y][x] - im.u[im.zhi][y][x] >= th) im.u[0][y][x] = 255;
                else im.u[0][y][x] = 128;
            }
        }
        V3fwrite( & im, OVAL); /* write the oldest frame */
    }
    exit(0);
}
```

Implement temporal binary difference filter on taxi

2.Test your program with the test image sequence **`taxi.vs`** using different threshold values. What effect does the *threshold* have on the filtered sequence? What is a good threshold for this image set?

The larger the thresthold I set, the less the number of 255 pixels, so the less the object area shows, the more the background area shows.

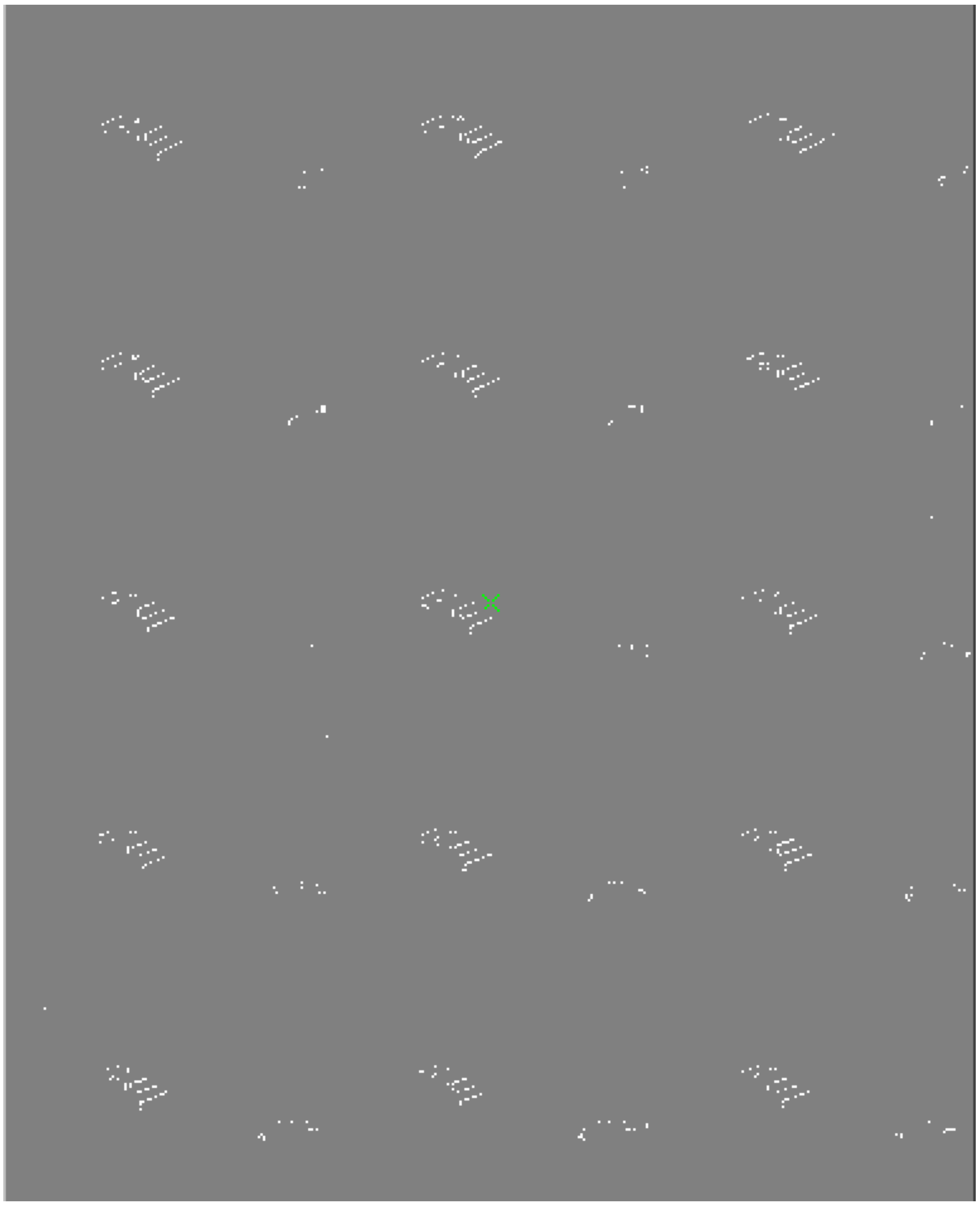


Figure15. temporal binary difference filter taxi results (Th=50)

When threthod is 50, the two cars' moving is very obvious, and contains nearly no noise. But the shape of car is opeaque and almost unrecognized. So I think 50 is so large.

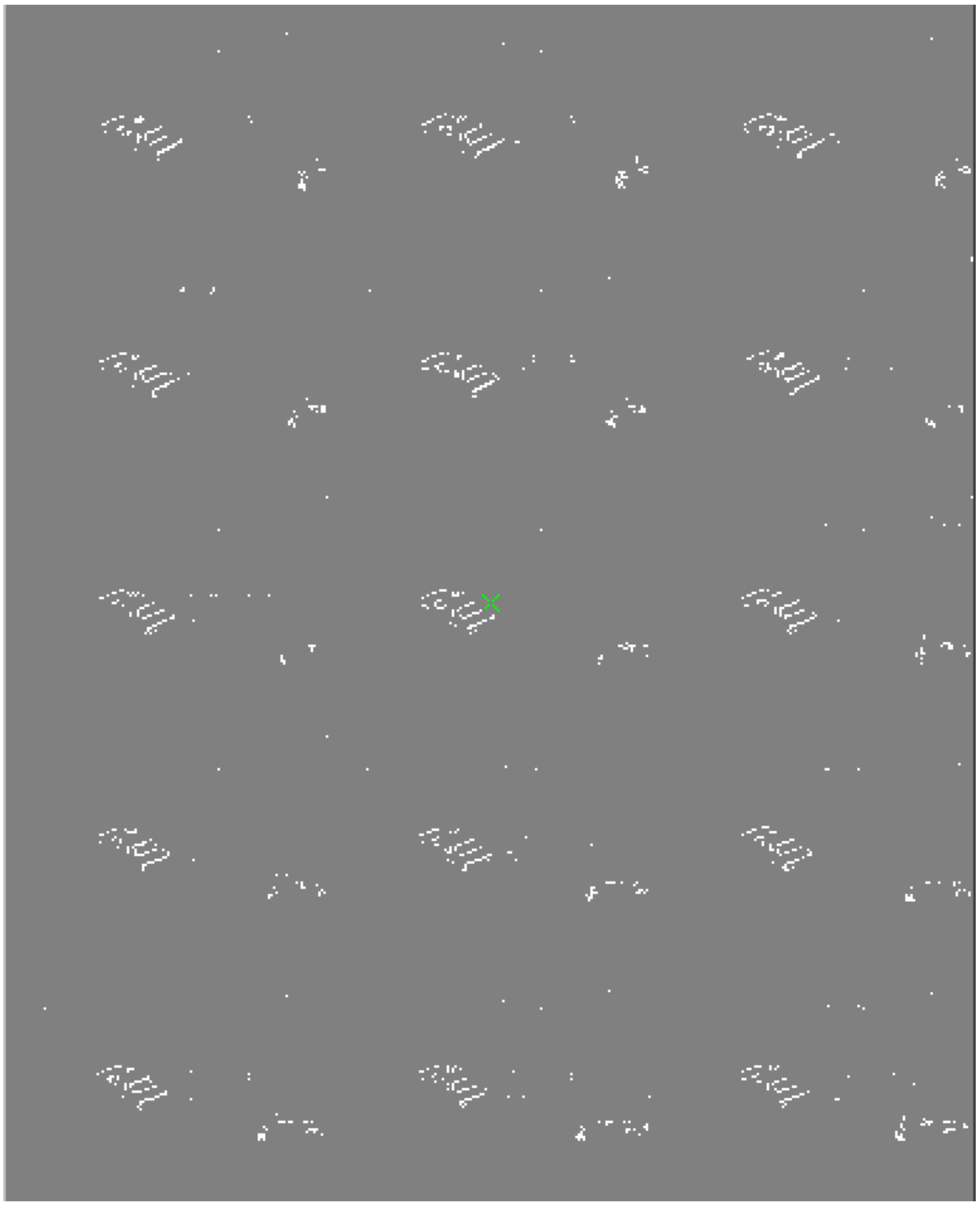


Figure16. temporal binary difference filter taxi results (Th=30)

When threthod is 30, the two cars' moving is very obvious, and contains very little noise. The shape of the car in the center is at least recognizable. So I think 30 is a good choice.

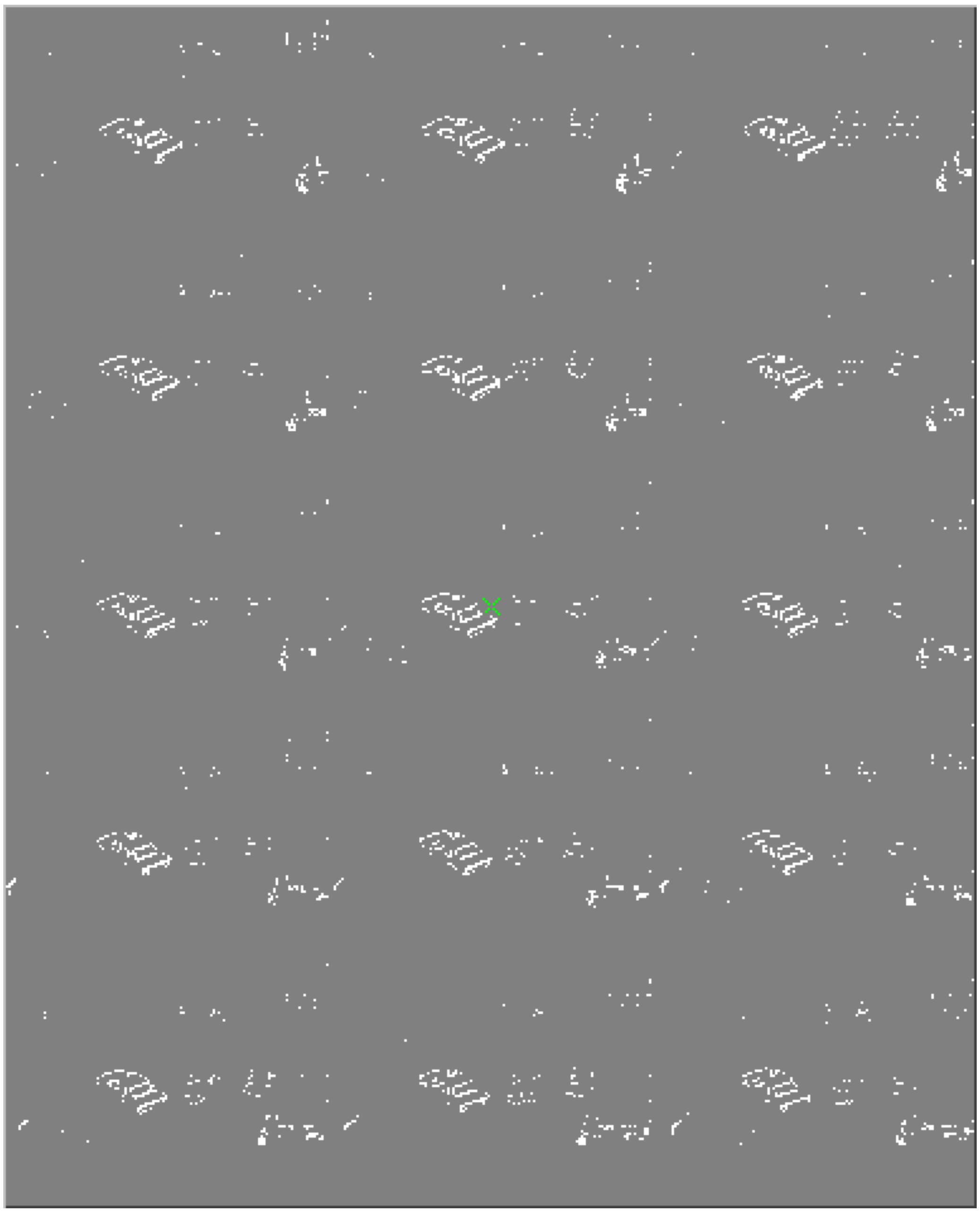


Figure17. temporal binary difference filter taxi results (Th=20)

When threthod is 20, the two cars’ moving is very obvious, and contains more noise. The shape of car is mostly recognizable. But for our goal of tracking, I think the shape is not so important, the noise is a bigger challenge for object and background dividing as well as following tracking. So I think 20 is so small.

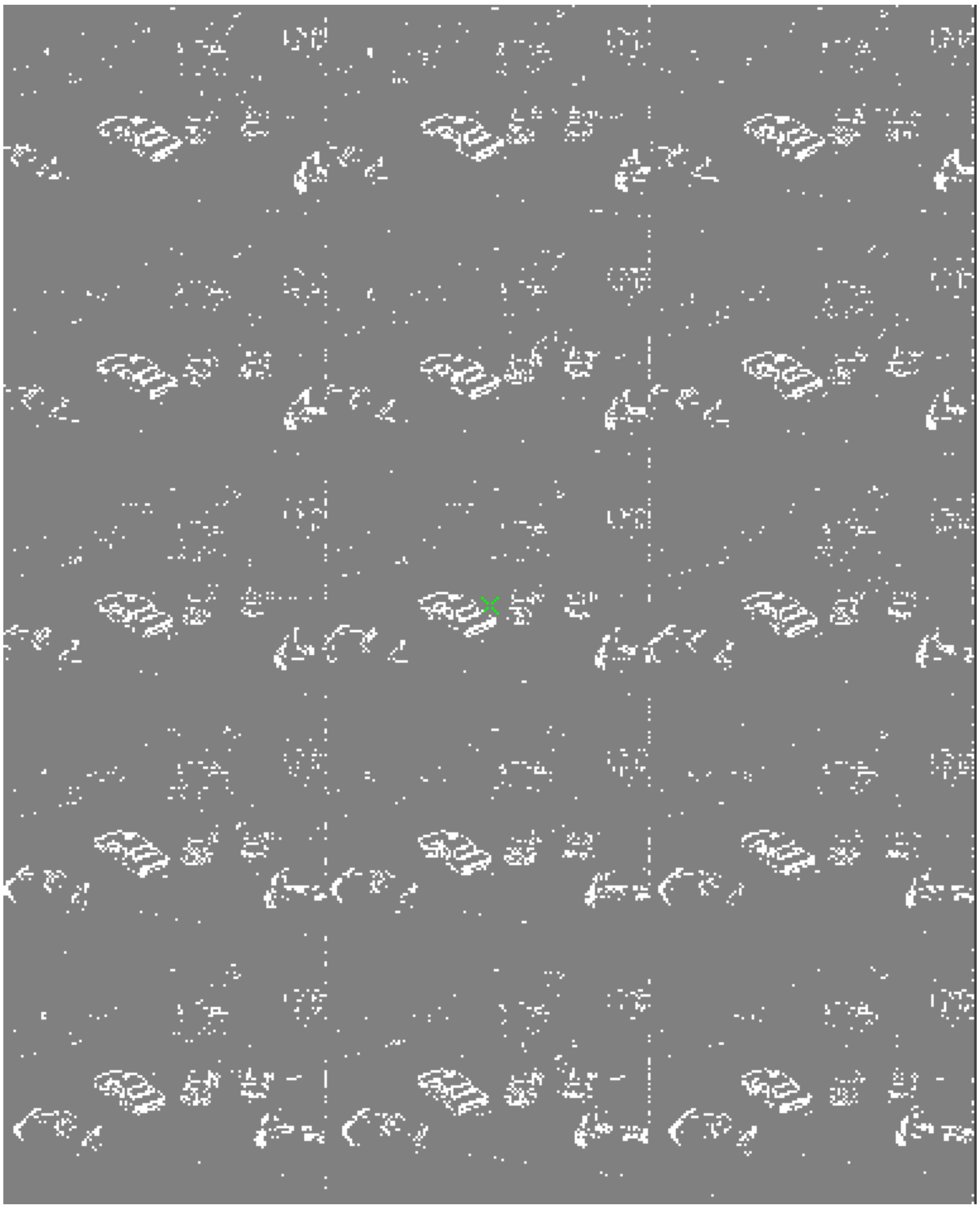


Figure18. temporal binary difference filter taxi results (Th=10)

Again, when threthod is 10, the two cars' moving is extremely obvious, and contains a lot of noise. The shape of car is absolutely recognizable. But for our goal of tracking, I think the shape is not so important, the noise is a bigger challenge for object and background dividing as well as following tracking. So I think 10 is so small.

So, as a result, I think 30 is the best threthold among all threthhold I tests. Because it shows the moving car which is in the center very clear, and shows part of the right button car, so we can tell the right button car are moving, also, when threshold is 30, the noise is very little, almost the two car are the only objects.

3.Create an image with the first 15 frames of the filtered image sequence (using the best threshold) as a single tileds image.

```
vsdif if = taxi.vs th = best_value | vtile n = 3, 5 | vxto - gif of = output
```

```
vsdif if = taxi.vs th = 30 | vtile n = 3, 5 | vxto - gif of = vsdiftaxi30
```