Multimedia Assignment3

Q1. Try the two search ranges (p=8 and p=16) and two macroblock sizes (8x8 and 16x16) by using the two search methods. The reference image is frame437.jpg, and the target image is frame439.jpg.

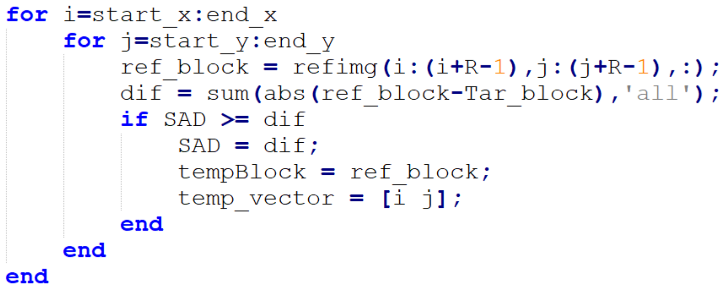
※Implementation

In this homework, it took me a long time to figure out the concept of ME. Not only is the understanding process difficult, "how" to implement it is also a hard task.

I give following steps to demestrate my implementation of full search algorithm and three step search algorithm.

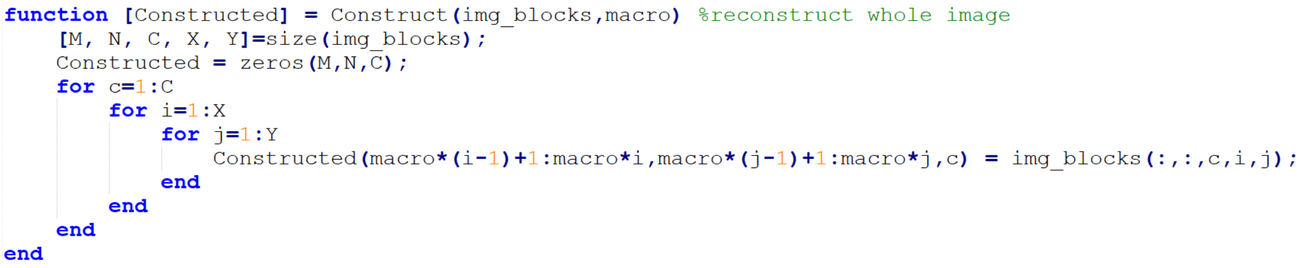
Full search

1. Split the target image into macroblocks due to the input of block size.
2. Set the starting position(x and y index) and the search range of the macroblock on reference image which is the same position as target image.(add range to all around the macroblock.)
3. Start to calculate the SAD value, which is sum(abs(reference - target)),and find the minimum value.



```
for i=start_x:end_x
    for j=start_y:end_y
        ref_block = refimg(i:(i+R-1),j:(j+R-1),:);
        dif = sum(abs(ref_block-Tar_block),'all');
        if SAD >= dif
            SAD = dif;
            tempBlock = ref_block;
            temp_vector = [i j];
        end
    end
end
```

4. Store the motion vectors whenever the calculation of SAD and store them in array to draw the motion vector gragh.
5. Since it's an overlapping precess, the macroblock on reference image shift to left a pixel after one round.(zig zag)
6. After processing all search range(for a block), take the reference block which created the smallest SAD and give it to our predict image to contruct the predict result.

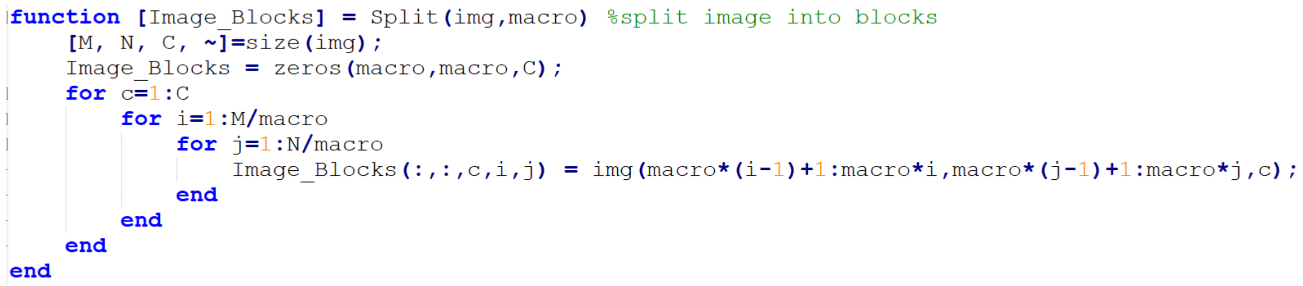


```
function [Constructed] = Construct(img_blocks,macro) %reconstruct whole image
    [M, N, C, X, Y]=size(img_blocks);
    Constructed = zeros(M,N,C);
    for c=1:C
        for i=1:X
            for j=1:Y
                Constructed(macro*(i-1)+1:macro*i,macro*(j-1)+1:macro*j,c) = img_blocks(:,:,c,i,j);
            end
        end
    end
end
```

7. Finishing a series of preccesses, move on to next block(target block and ref block all need to change until traverse the whole image.)

Three step search Residual image construct

1. Split the target image into macroblocks due to the input of block size.



```
function [Image_Blocks] = Split(img,macro) %split image into blocks
    [M, N, C, ~]=size(img);
    Image_Blocks = zeros(macro,macro,C);
    for c=1:C
        for i=1:M/macro
            for j=1:N/macro
                Image_Blocks(:,:,c,i,j) = img(macro*(i-1)+1:macro*i,macro*(j-1)+1:macro*j,c);
            end
        end
    end
end
```

2. Set the starting position(x and y index) and the search range of the macroblock on reference image which is the same position as target image.(add range to all around the macroblock.)

3. Start to calculate the SAD value, which is sum(abs(reference - target)),and find the minimum value.

4. Store the motion vectors whenever the calculation of SAD and store them in array to draw the motion vector gragh.

5. If the search range is 8,then our step goes like 4->2->1 to find the smallest SAD (search range is 16 is like 8->4->2->1),and we can get a final value. The 8 points around the center point need to be traveled, and the range get smaller one after another search until the range becomes 1 and we can take the reference block information which SAD value is the minimum to construct our predict image.

6. Finishing a series of preccesses(25 points are traveled), move on to next block.

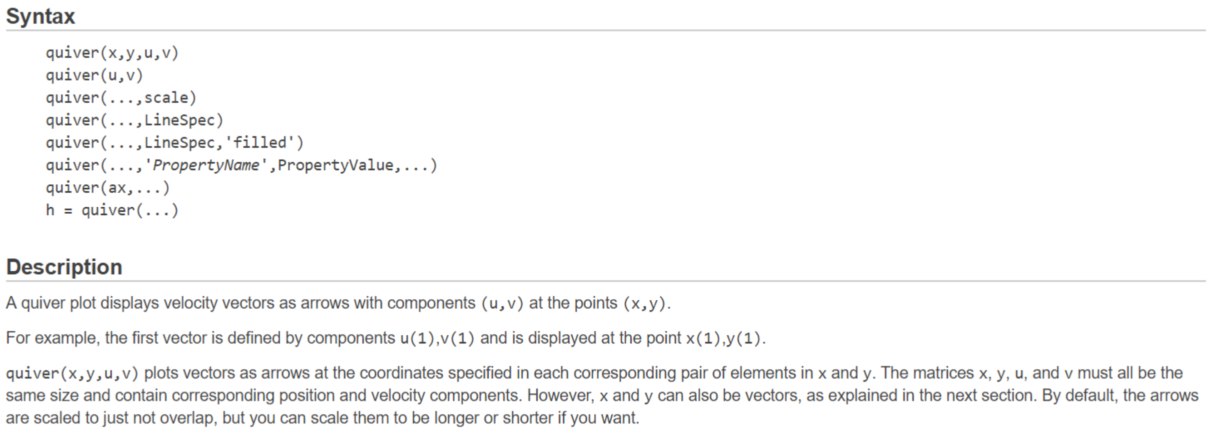
Residual image construct

Just minus the predict image and target image and get the abs value to tell the difference between two images.

Motion vector

Use built-in function "quiver(a,b,c,d)" ,which store starting point index(x,y) and ending index(x,y).

It took me some time to understand this file.



Syntax

```
quiver(x,y,u,v)
quiver(u,v)
quiver(...,scale)
quiver(...,LineSpec)
quiver(...,LineSpec,'filled')
quiver(...,'PropertyName',PropertyValue,...)
quiver(ax,...)
h = quiver(...)
```

Description

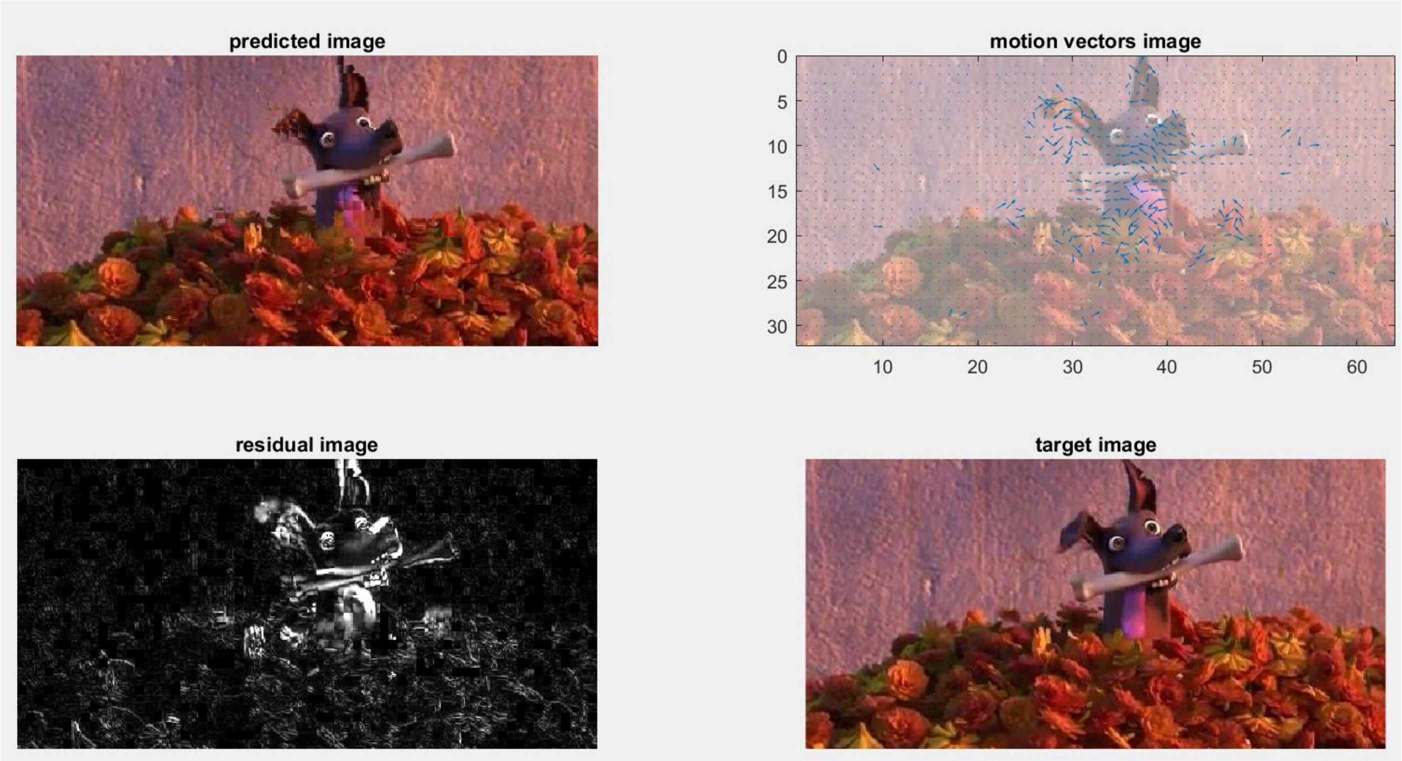
A quiver plot displays velocity vectors as arrows with components (u,v) at the points (x,y).

For example, the first vector is defined by components u(1),v(1) and is displayed at the point x(1),y(1).

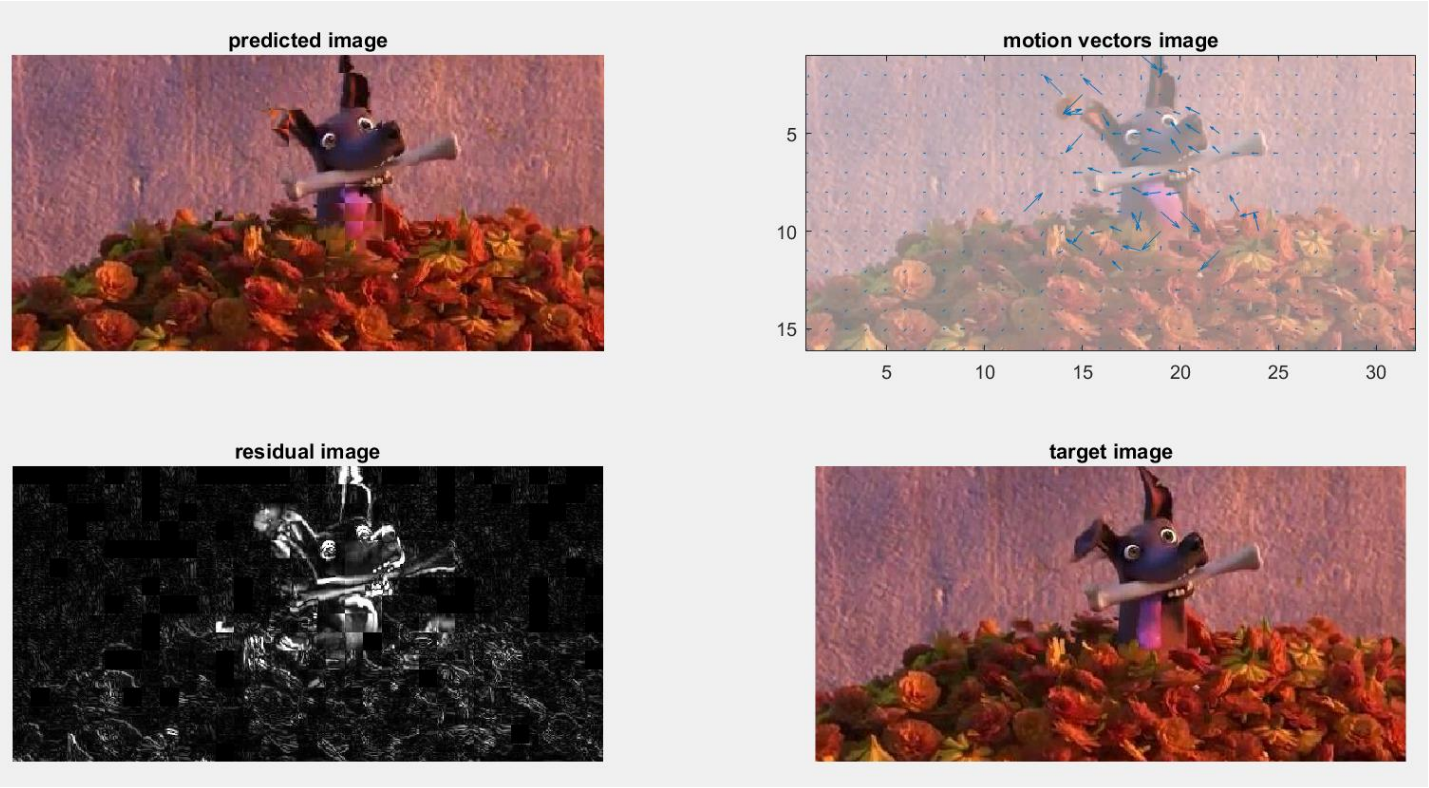
quiver(x,y,u,v) plots vectors as arrows at the coordinates specified in each corresponding pair of elements in x and y. The matrices x, y, u, and v must all be the same size and contain corresponding position and velocity components. However, x and y can also be vectors, as explained in the next section. By default, the arrows are scaled to just not overlap, but you can scale them to be longer or shorter if you want.

※Result display & Discussion

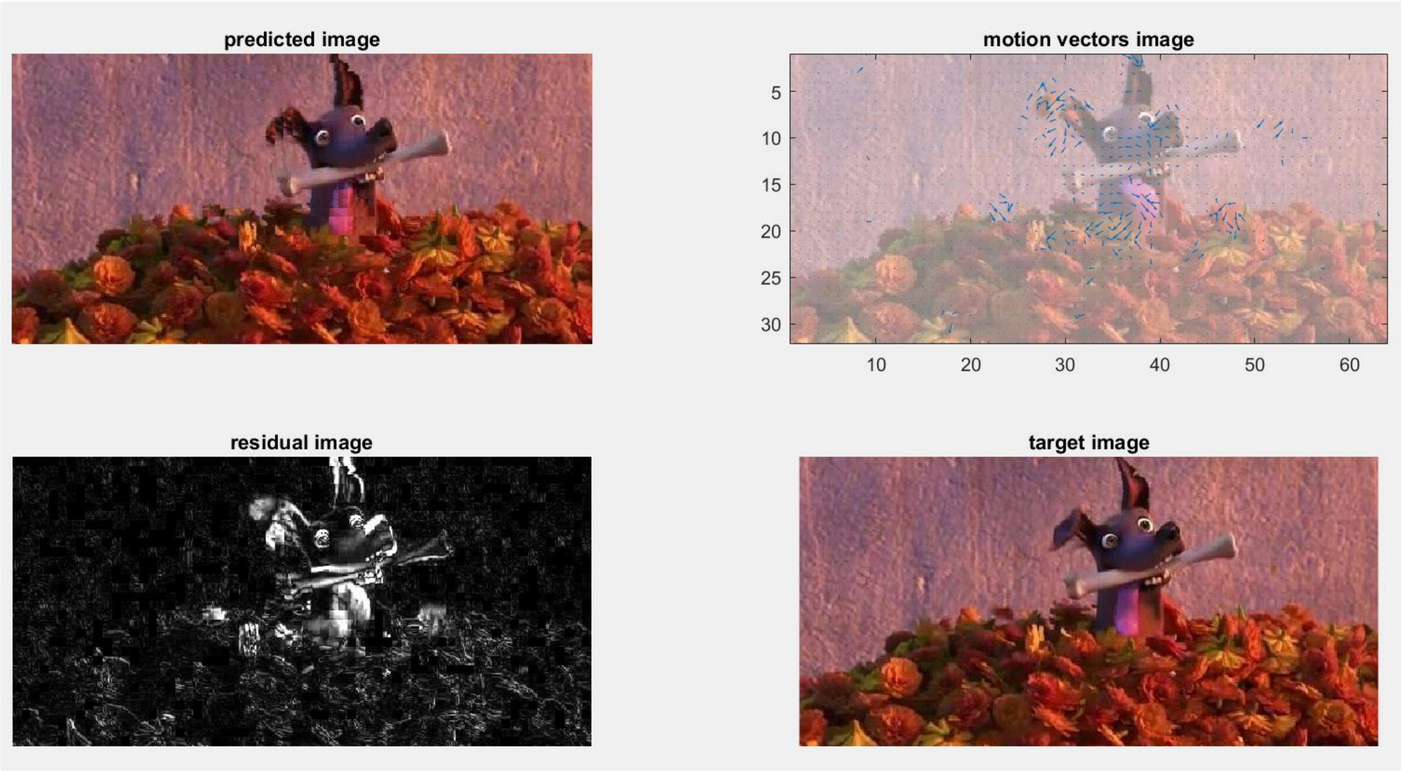
FULL Macroblock:8 Search range:8



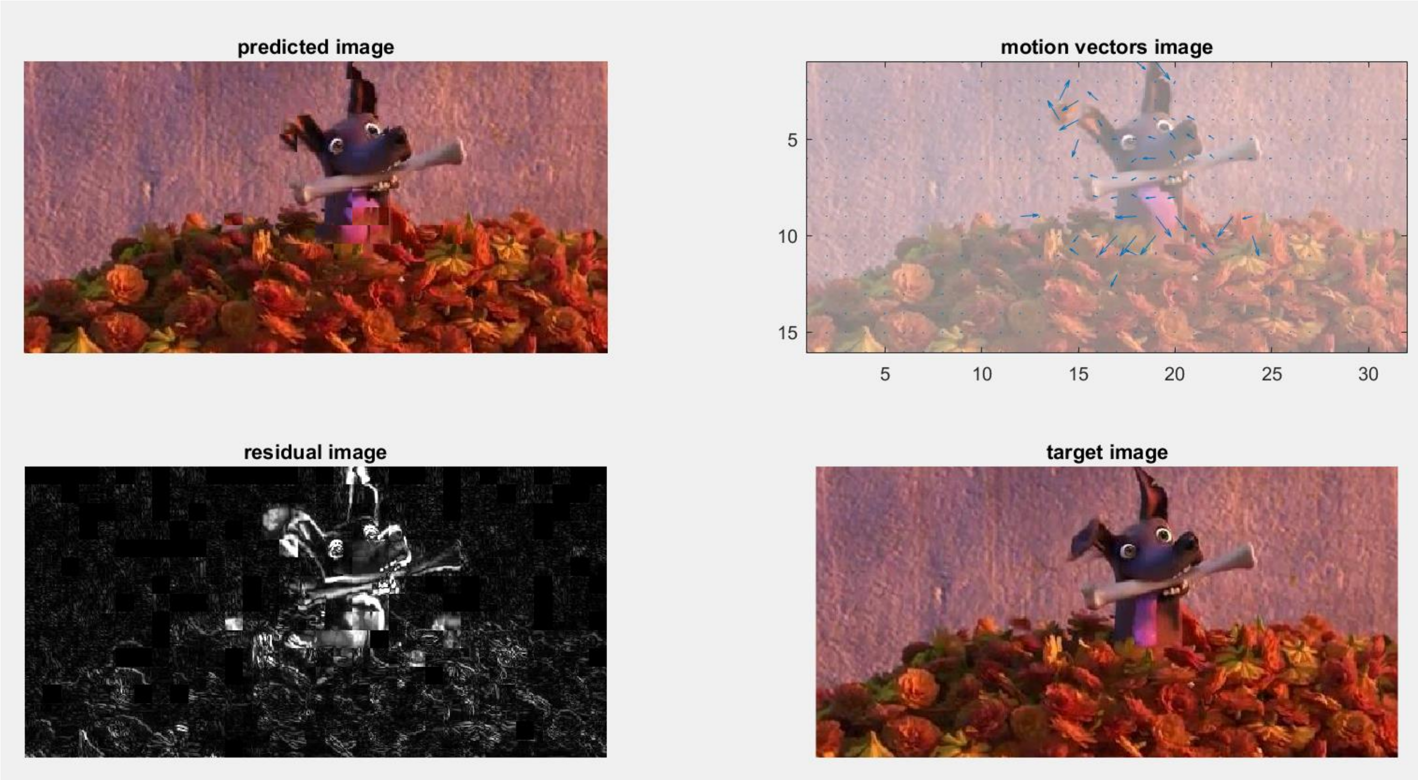
FULL Macroblock:16 Search range:8



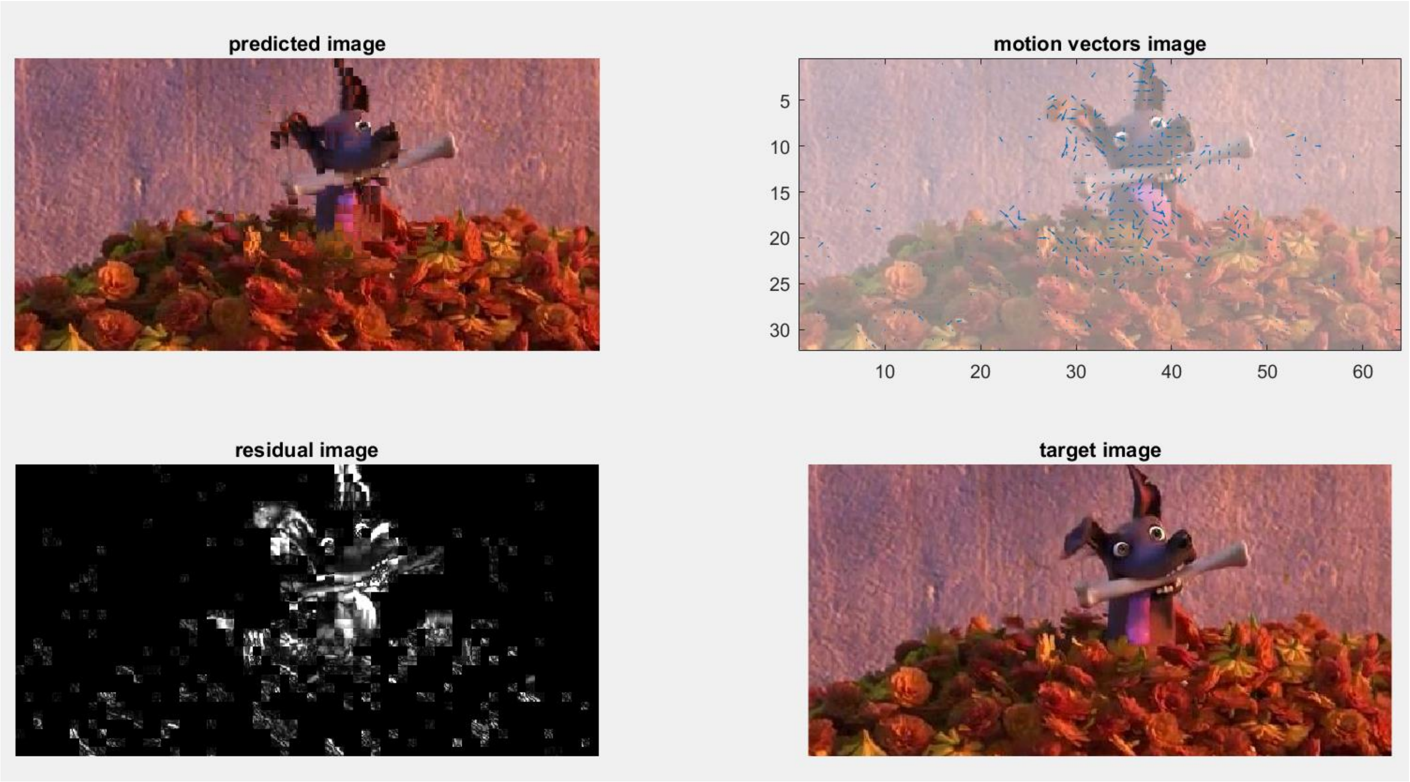
FULL Macroblock:8 Search range:16



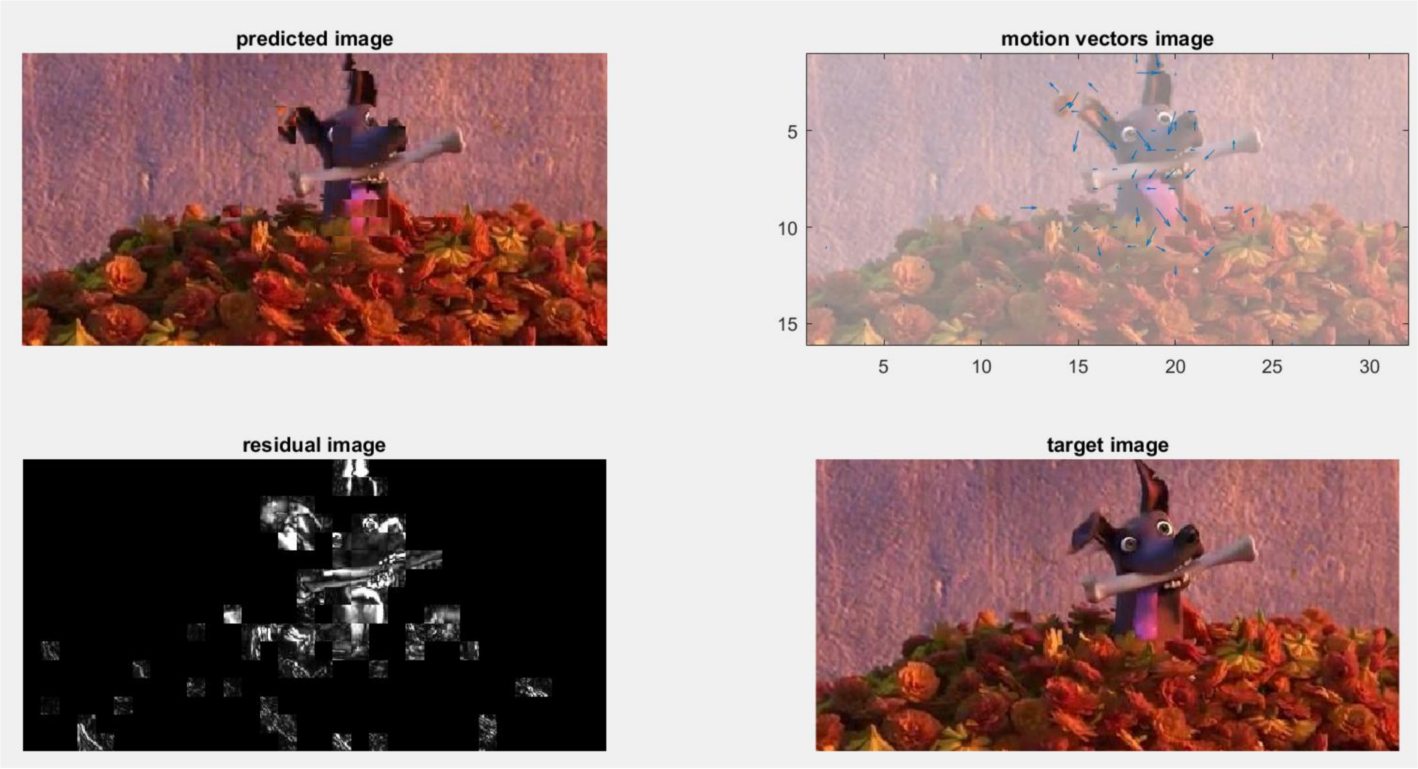
FULL Macroblock:16 Search range:16



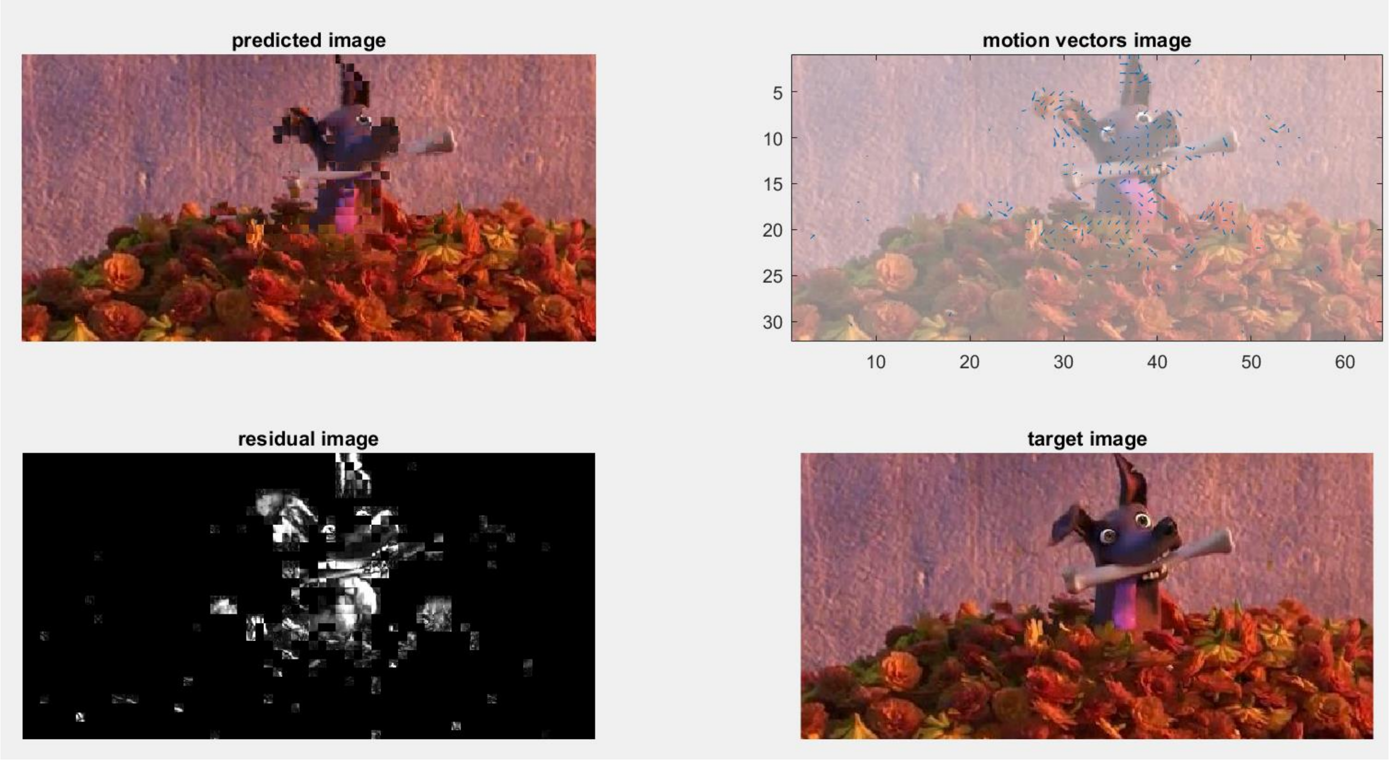
3 STEP Macroblock:8 Search range:8



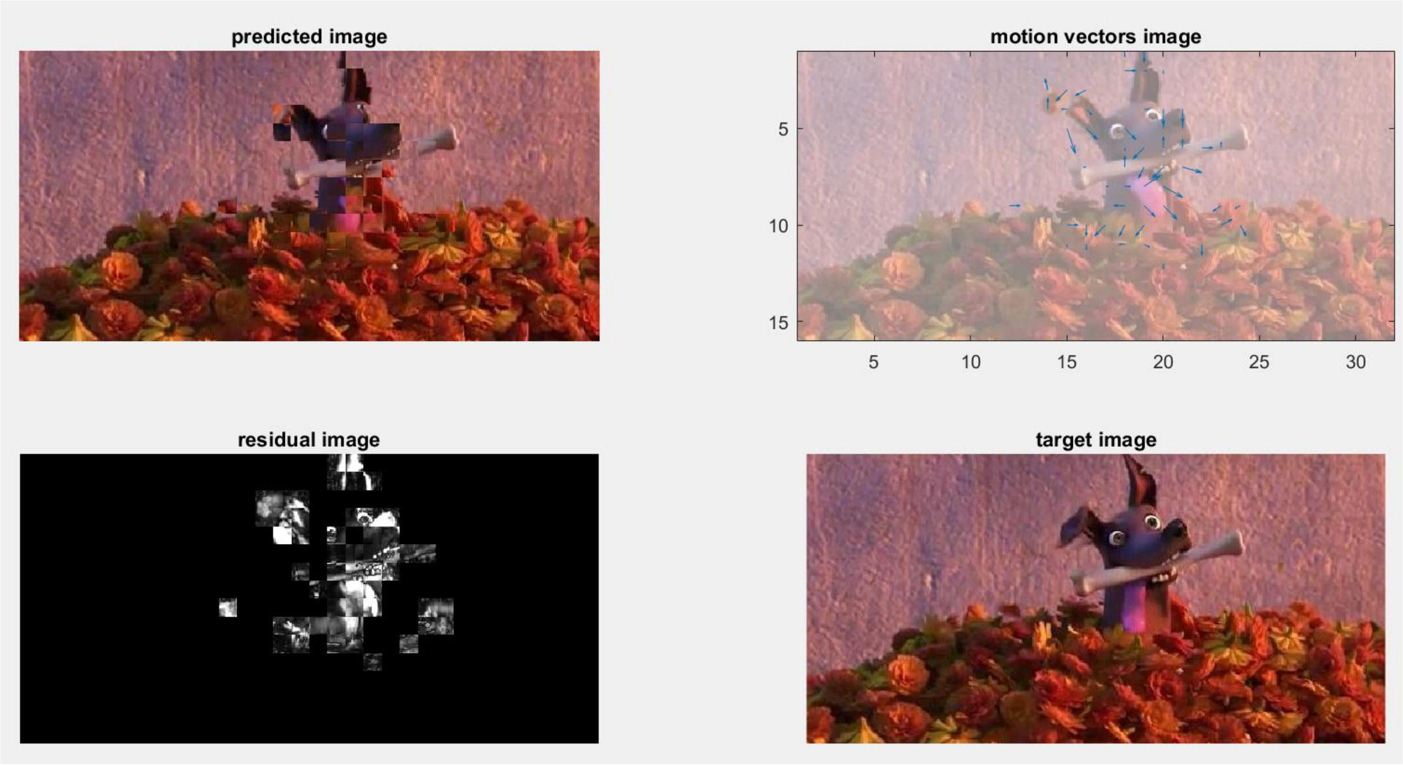
3 STEP Macroblock:16 Search range:8



3 STEP Macroblock:8 Search range:16



3 STEP Macroblock:16 Search range:16



As we can see in the above images , we have some discussion by sorting them in search range, macroblock size and search methods:

	Full search	**Three step search**
P=8 macroblock=8	Since traversing the image by small size of macroblock (or we can say split image into smaller blocks) and having smallest search range, we can get the closest result comparing to the target image. That's obvious.	Also in most "carefully" search scale, so the result is the best comparing to other size of search range or macroblock in three step search algorithm. But it's not so good compares with full search since three step search only get 25 points.
P=16 macroblock=8	I can't tell any clearly difference between this residual image and the last one. Therefore ,I suppose that the search range is not "very" important to the prediction.	The dog's eyes distorted and its head and ears has obvious unsmooth lines, many strange color blocks appear.
P=8 macroblock=16	The dog's tongue in the predict image is the most obvious place that I found the difference when comparing to before results. It gets blur due to macroblock size gets larger, the accuracy of prediction decreases.	Places around the dog's tongue is very messy, and the bone is torn in half. Very bad result.
P=16 macroblock=16	The dog's tongue gets worse when all sizes become 16, that is , inaccuracy rate rises.	Distort condition getting worse. Now, we can hardly tell it's a dog hanging a bone. Very bad result.

What motion vectors tell us:

You can find out that when macroblock is 8, the motion vector graphs show more vectors and vector size is shorter while macroblock in 16 are less and sizes are longer. It's because that when we travel image in smaller "steps", we can record more direction whenever we move, even very small movements can be caught and thus short.

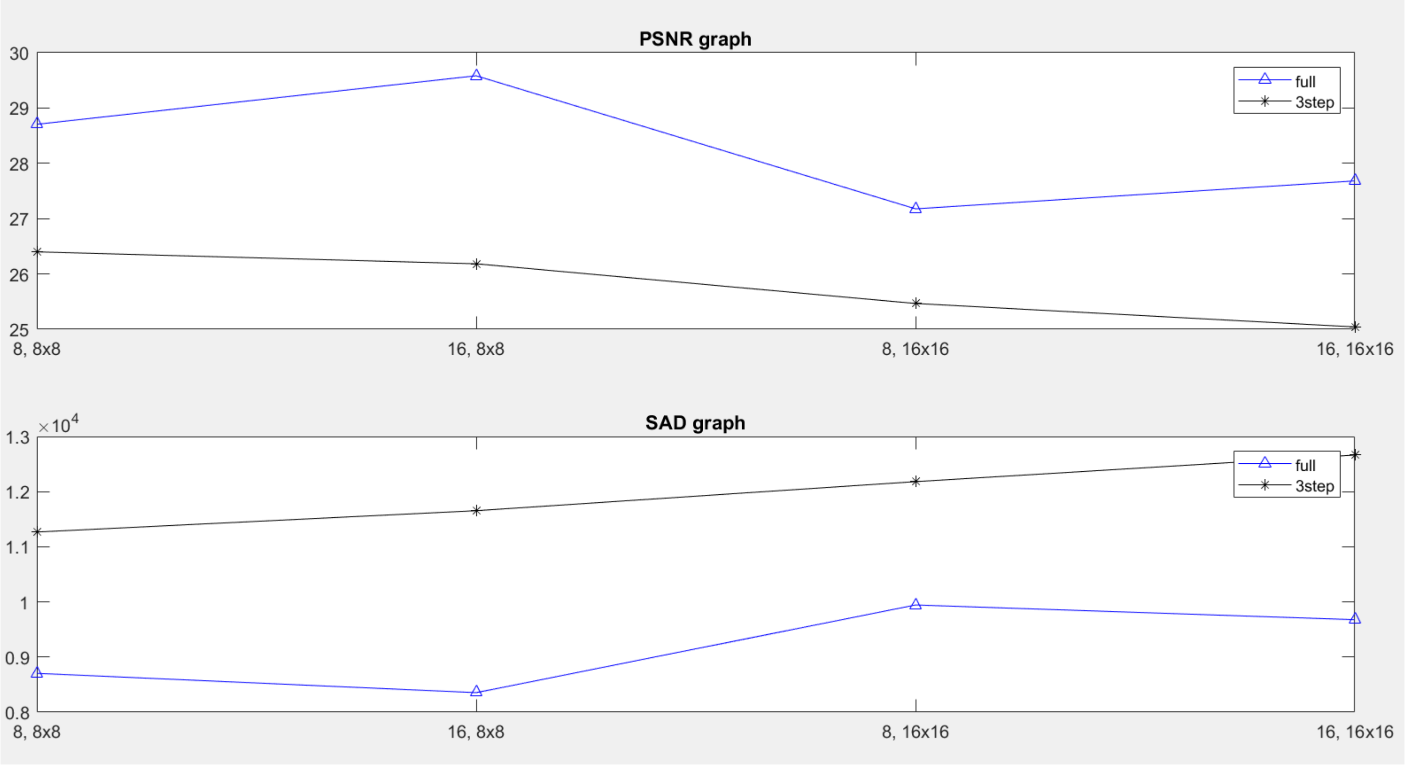
But when macroblock is larger, the recording times decrease and less information we can keep. Also the length of vector increases.

That's why two macroblock sizes influence the motion vectors number.
Or we can say that it resembles less to the target image when the number of motion vectors is less.
Furthermore, the direction of motion vectors shows the tendency of the direction that image going to change.

What residual images tell us:

I found out that in general, the residual values are high in my result and I don't know if it's normal or just because my results are wrong. However, there're still some differences between them. It's obvious that from full search to three step search, the residual amount increase (white places form from lines to patches)
However, an interesting thing I discovered is that, the search range isn't likely to have high relationship to the accuracy of predict images since there's few clearly difference between p=8 macroblock=8 and p=16 macroblock=8 or p=8 macroblock=16 and p=16 macroblock=16.
On the other hand, difference gets more when macroblock size changes. So I thought that macroblock is the key influencing factor of prediction.

What PSNR and SAD values tell us:



Three step search's PSNR steadily decrease when macroblock size increase (bigger gap between macroblock size change from 8 to 16 while smaller change while search range change from 8 to 16, which prove my assumption just said.)
Even the search range increase doesn't let PSNR increase because three step search has only

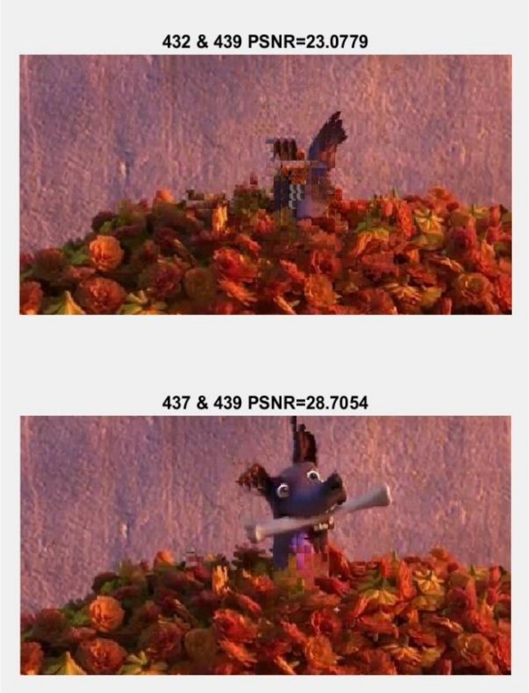
few steps, little difference can the search range do to influence the results.
However, there's obvious increase of PSNR in full search when search range change from 8 to 16, That's because full search travel pixels one by one, therefore, the influence of search range gets larger.
Like three step search, PSNR still decrease when macroblock turns to 16, which means not so accurate.

Total SAD values is the sum of all smallest SAD when a macroblock operation finish.
High the value is, lower the accuracy gets.
We can see that three step search's SAD values increase steadily and are far high than full search, which is also easily to deduce.
As for full search, SAD value goes down when search range getting bigger. (but in slight scale)
But when macroblock becomes to 16, the value gets high rapidly. So we can say that the macroblock is truly the key point to the accuracy of prediction.(meets my guess)

In conclusion, to full search, either search range or macroblock size has certain impact on predicting a image. We get up-and-down line in PSNR and SAD values. For three step search, since it's already not so accurate enough, search range changes influence little on it so we get steady upward line on SAD value and steady downward line on PSNR.

Interesting thing to know, although the scale of SAD and PSNR are totally different , they are quite symmetric to each other!!!! (PSNR goes up and SAD goes down.)

Q2. Try the full search method with search range p=8 and macroblock sizes = 8x8. The reference image is frame432.jpg, and the target image is frame439.jpg.



You can see that PSNR is lower when we use frame432 than we use frame437, that's because frame432 is the frame far before frame439, so it gets harder to predict the right image when using too unsimilar images. Therefore, predict image beside when using frame432 is terrible, we can hardly tell what's in the image.

Q3: Analyze the time complexity

My assumption:

Full search (search range = p)

⇨ $(2 * p + 1)^2$ SAD search points and $(2 * p + 1)^2$ two-term comparisons are calculated to find the best match.

⇨ So, time complexity is $O(p^2)$

Search range =8 -> 8*8 = 64

Search range=16 -> 16*16 = 256

Three step search (search range = p)

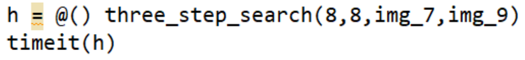
⇨ The candidate points are data-dependent—the current step result decides the to-be-evaluated search points in the next step. Therefore, each step has to be performed sequentially. In total, there are $\log_2$(p+1) search steps and 1+8 log2(p+1) SAD search points for each image block.

⇨ So, time complexity is O(log2(p+1))

Search range =8 -> log2(9) ≒ 3

Search range=16 -> log2(17) ≒ 4

I used this code to get the real execution time to testify my assumption.



```
h = @() three_step_search(8,8,img_7,img_9)
timeit(h)
```

Execution time are below:

	p=8	P=16
Full search	2.6165 (s)	8.2333 (s)
Three step search	0.6412 (s)	0.7216 (s)

By my theoretical calculation, we can see that in full search method, time complexity when p=16 is exactly 4 times to p=8 .

And real execution time is 8.2333 and 2.6165 , which is 3.14 times more.

As for three step search, theoretical result is 4 and 3 when p=16 and p=8 ,that is 1.333 times.

And real execution time is 0.7216 and 0.6412 , which is 1.1254 times more.

Mostly correct assumption!!!!!

Reference:

quiver usage:

https://www.dummies.com/programming/matlab/how-to-display-velocity-vectors-in-matlab-using-quiver/

ME:

https://ir.nctu.edu.tw/bitstream/11536/68124/1/251601.pdf

https://graphics.cmlab.csie.ntu.edu.tw/~itct/ref/7_ITCT_Motion_compensation.pdf

https://en.wikipedia.org/wiki/Block-matching_algorithm

graph plot:

http://mirlab.org/jang/books/matlabProgramming4beginner/03-3_axisControl.asp?title=3-3%20%B9%CF%B6b%AA%BA%B1%B1%A8%EE