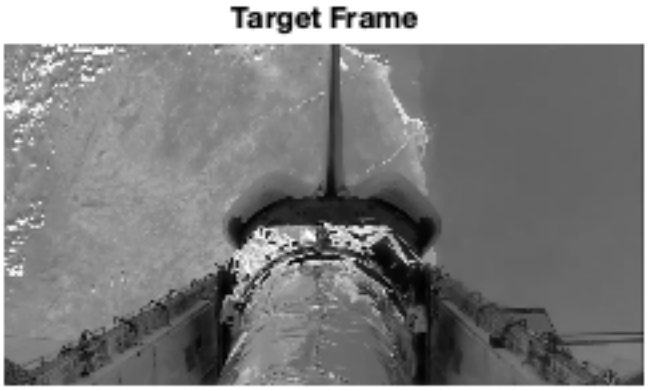
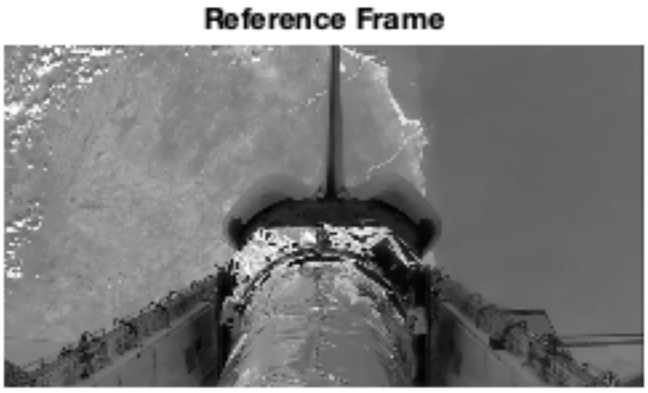
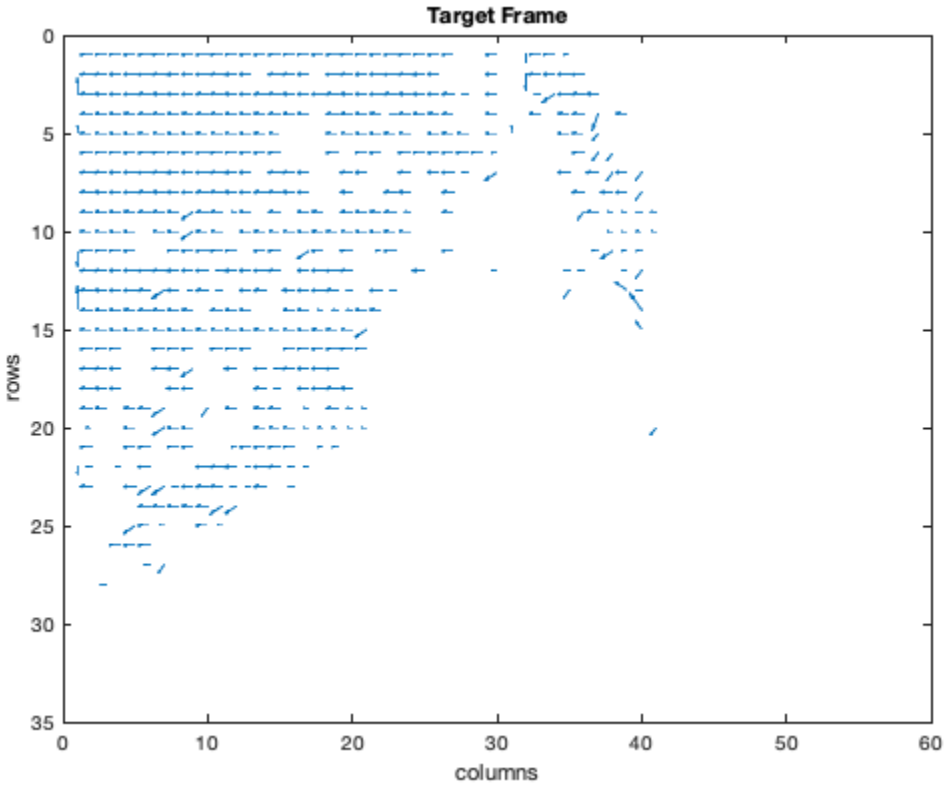
```
% Project: Block matching algorithm
% Submitted By: MANISH SONI

Block_Ref_Img=double(imread('shuttle.tif', 1));
block_size=8; % size of block
search_space=4; % lets make search space 4 up, 4 down,4 left and 4
 right of block position

for img_count=2:16

    % get the target image
    Block_Target_Img=double(imread('shuttle.tif', img_count));
    [row,col]=size(Block_Target_Img);
    % set the row index and colum index to 1 for motion vectors
     row_ind=1;
     col_ind=1;
     % get the rows and columns count for motion vectors
     motion_vector_total_rows=row/block_size;
     motion_vector_total_cols=col/block_size;

 motion_vector_rows=zeros(motion_vector_total_rows,motion_vector_total_cols);

 motion_vector_cols=zeros(motion_vector_total_rows,motion_vector_total_cols);
     threshold=2; % lets set a difference threshold
     % objective of this code is to consider each block and run it
     % from (1,1) to (row,col)
    for i=1:block_size:row
        for j=1:block_size:col
            % motion_vector_target_min stores the motion_vector values
            motion_vector_row_min=0;
            motion_vector_col_min=0;
            % store all the Sum of absolute differences
            SAD=[];
            vectors_rows=[];
            vectors_cols=[];
            itr=0; % iteration counts
            % move the block from -search_space to search_space in x
 and y
            % direction on a target frame so that target and reference
            % frames can be compared
            for space_row_index=-search_space:search_space
                for space_col_index=-search_space:search_space
                 start_target_index=i;
                    start_ref_index=space_row_index+i;
                  if(start_target_index+block_size>row)
                       end_target_index=row;
                       end_ref_index=space_row_index+row;
                   else
                       end_target_index=start_target_index
+block_size-1;
                       end_ref_index=start_ref_index+block_size-1;
                   end
```

```
                    start_target_col_index=j;
                    start_ref_col_index=j+space_col_index;
                    if(start_target_col_index+block_size>col)
                        end_target_col_index=col;
                        end_ref_col_index=space_col_index+col;
                    else
                        end_target_col_index=start_target_col_index
+block_size-1;
                        end_ref_col_index=start_ref_col_index
+block_size-1;
                    end
                    if(end_ref_index<row && end_ref_col_index<col &&
 start_ref_index>0 && start_ref_col_index>0)

 difference=Block_Target_Img(start_target_index:end_target_index,start_target_col_
Block_Ref_Img(start_ref_index:end_ref_index,start_ref_col_index:end_ref_col_index)
                        difference=abs(difference);
                        SAD_VAL=sum(sum(difference));
                        if(itr==1) % for first iteration store it
                            SAD=SAD_VAL;
                            vectors_rows=space_row_index;
                            vectors_cols=space_col_index;
                        else % for second iteration onwards store
 these
                            SAD=[SAD ;SAD_VAL];
                            vectors_rows=[vectors_rows;
 space_row_index];
                            vectors_cols=[vectors_cols;
 space_col_index];
                        end;
                         % increment the iterations
                        itr=itr+1;
                    end
                end
            end
        if(~isempty(SAD))
            min_sad_index=find(SAD==min(SAD));
            motion_vector_row_min=vectors_rows(min_sad_index(1),1);
            motion_vector_col_min=vectors_cols(min_sad_index(1),1);
        end


 sad_exact_points=abs(Block_Target_Img(start_target_index:end_target_index,start_t
Block_Ref_Img(start_target_index:end_target_index,start_target_col_index:end_targe
        sad_exact_points=sum(sum(sad_exact_points));

        %letss set Threshold value so tahat if the same location in
 both
        %the frames have somewhat similar object meaning low value of
        %differences and lower than the threshold than put the row
 index
        %and colindex as 0
        block_total_threshold =threshold*block_size*block_size;
        if(min(SAD)>=sad_exact_points-block_total_threshold)
```

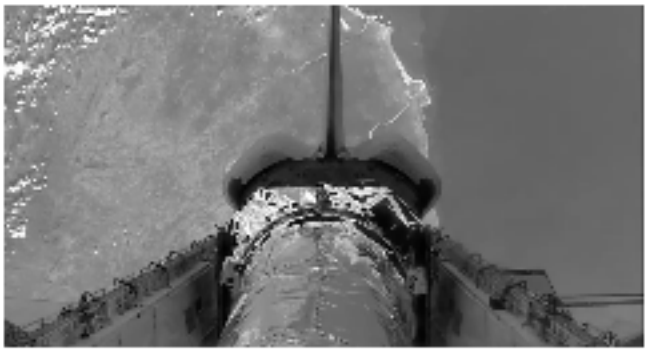
```
            motion_vector_row_min=0;
            motion_vector_col_min=0;
        end
        % store the row and col index for min sad value
        motion_vector_rows(row_ind,col_ind)=motion_vector_row_min;
        motion_vector_cols(row_ind,col_ind)=motion_vector_col_min;
        col_ind=col_ind+1;
        end;
          col_ind=1;
          row_ind=row_ind+1;
    end;
figure,
imshow(mat2gray(Block_Target_Img));
title('Target Frame');
figure,
imshow(mat2gray(Block_Ref_Img));
title('Reference Frame');
figure,
quiver(motion_vector_cols,motion_vector_rows);
xlabel('columns');
ylabel('rows');
% below statement set some settings for axes
set(gca,'ydir','reverse');
title('Vector Fields for Motion');
end;
```

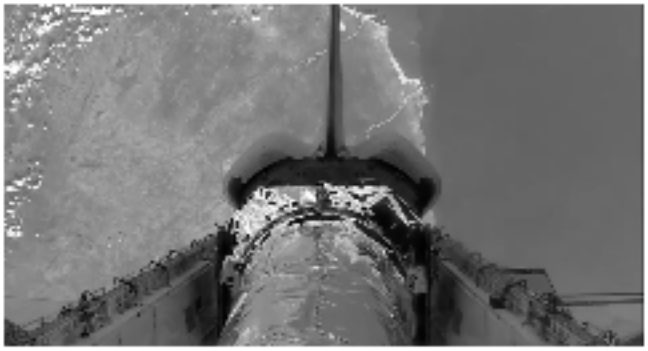


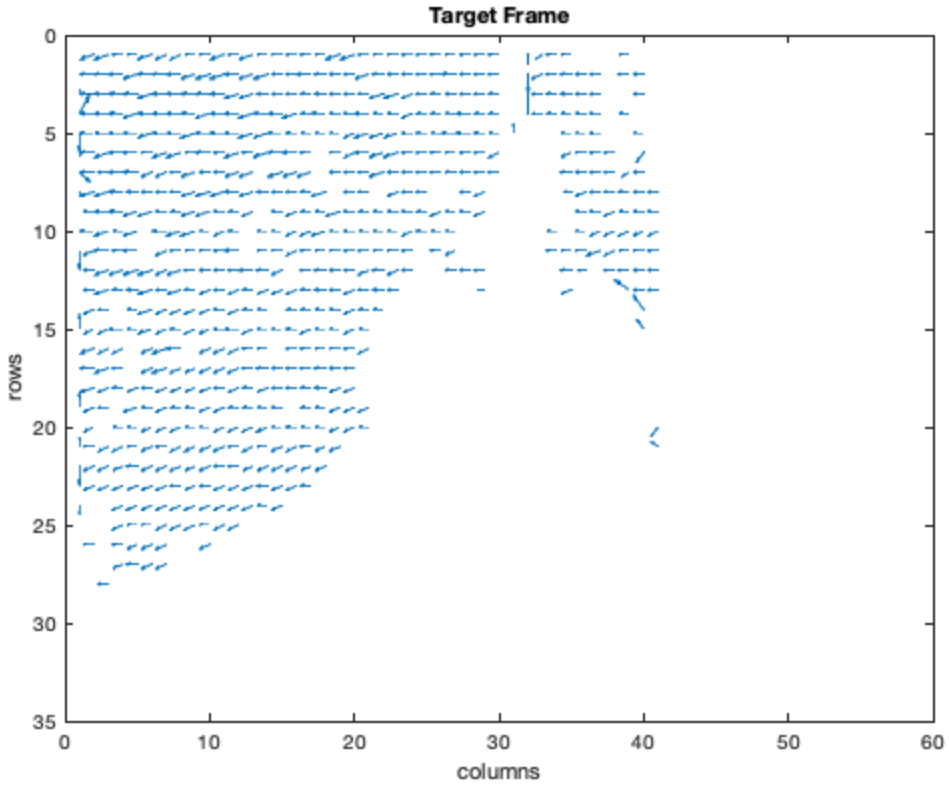

Reference Frame

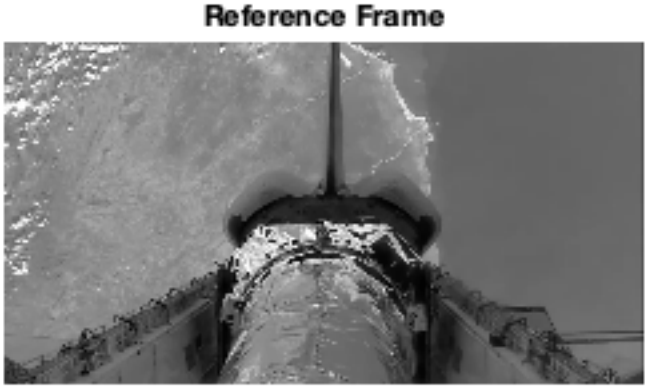

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

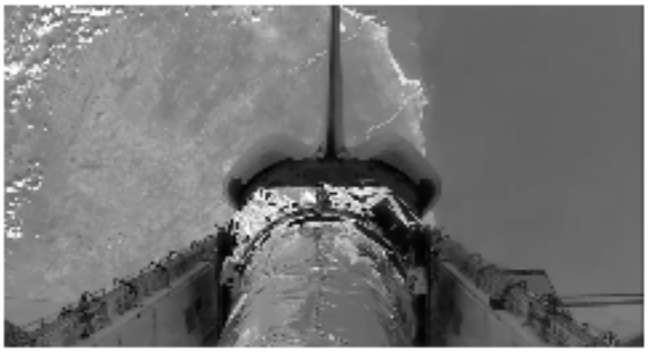
Reference Frame

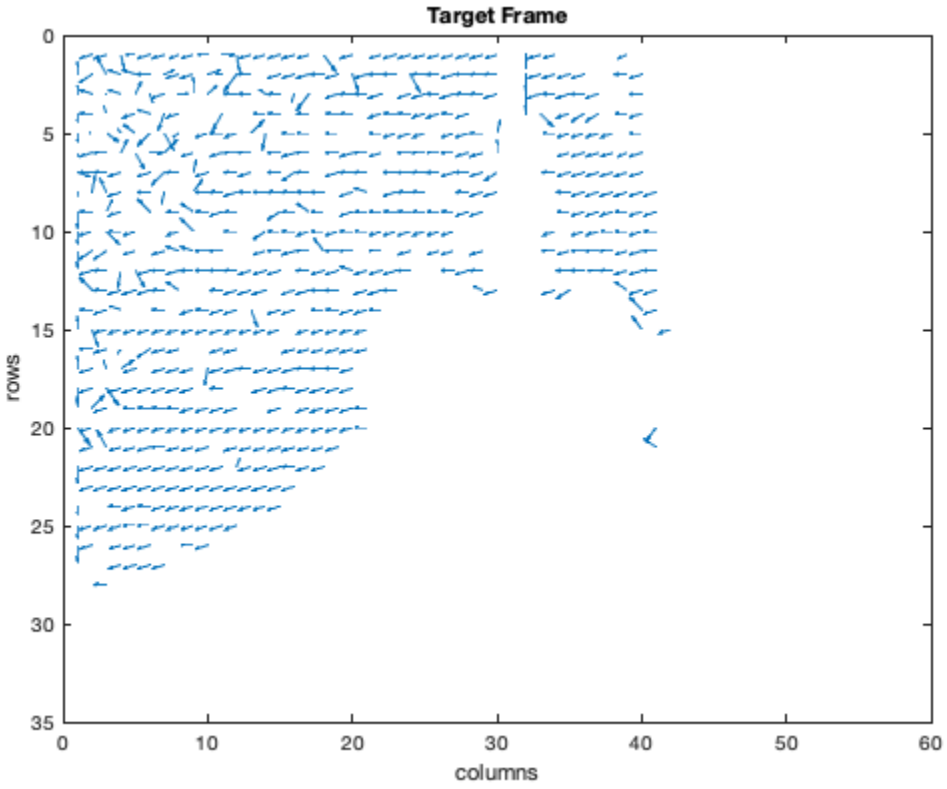


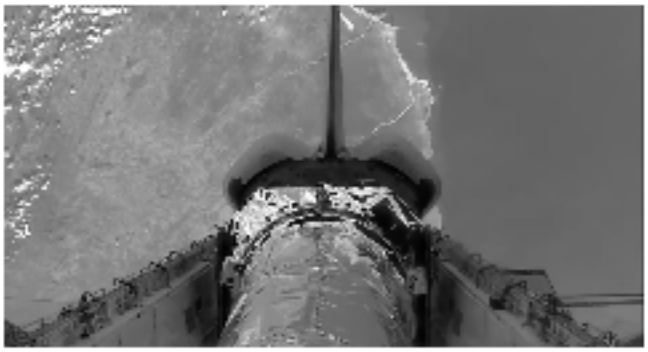



Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

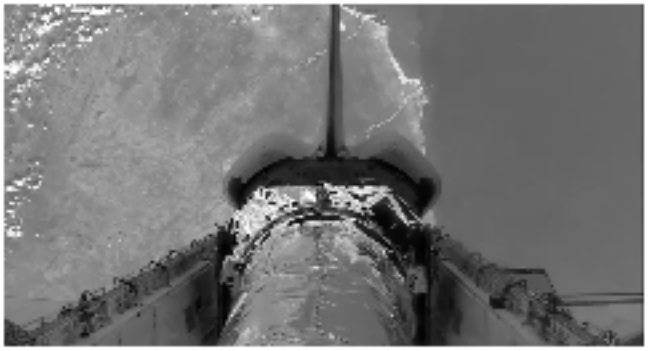

Reference Frame

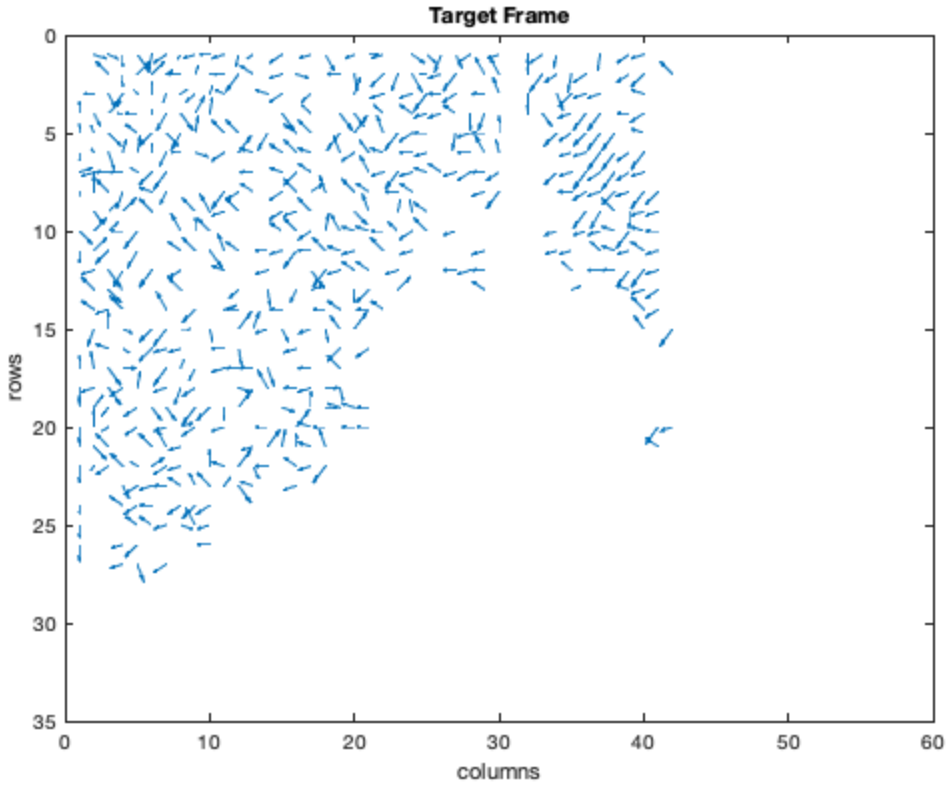


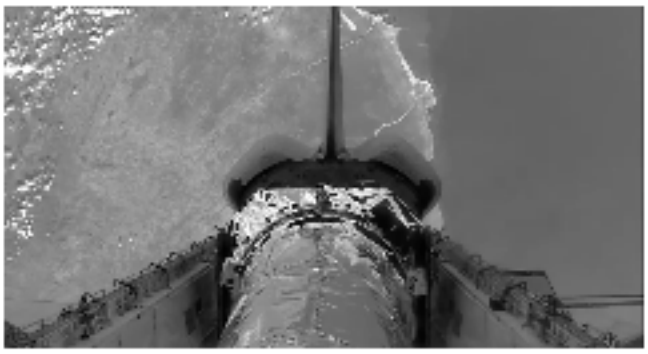

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

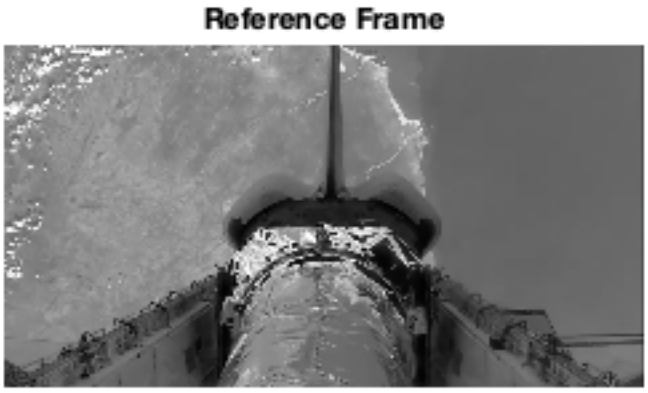


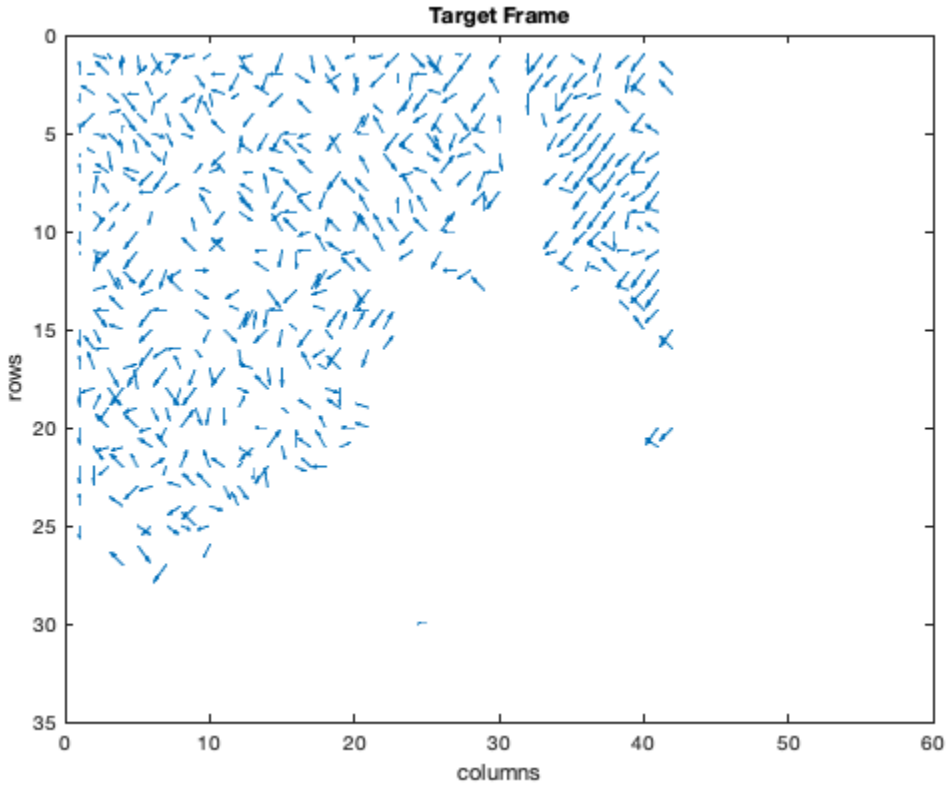
Reference Frame



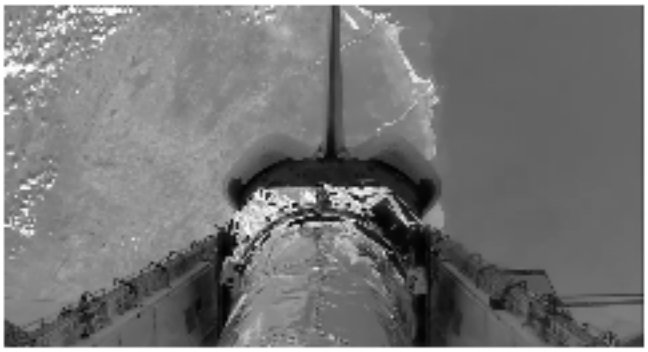

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

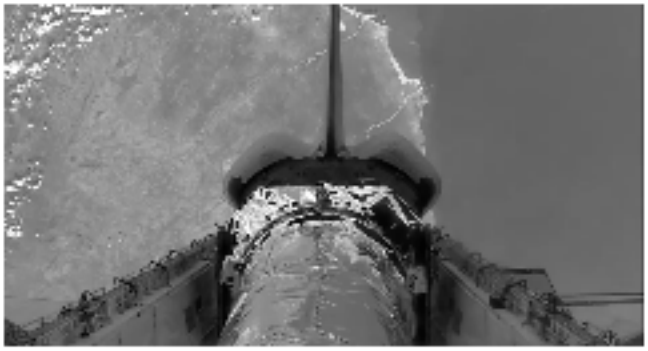


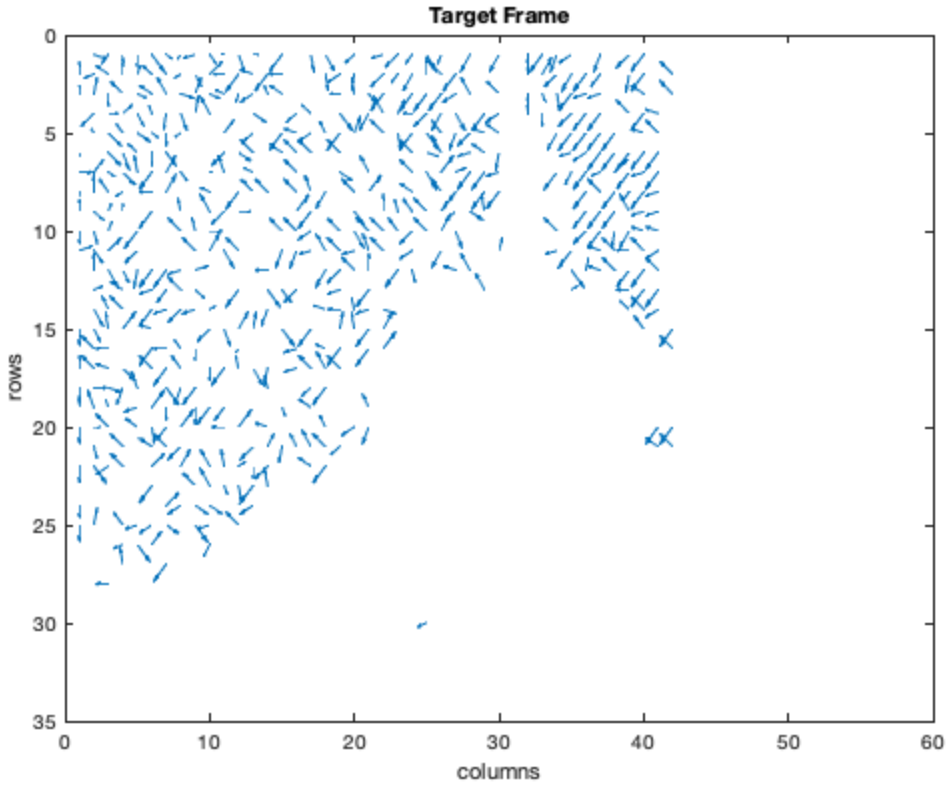

Reference Frame

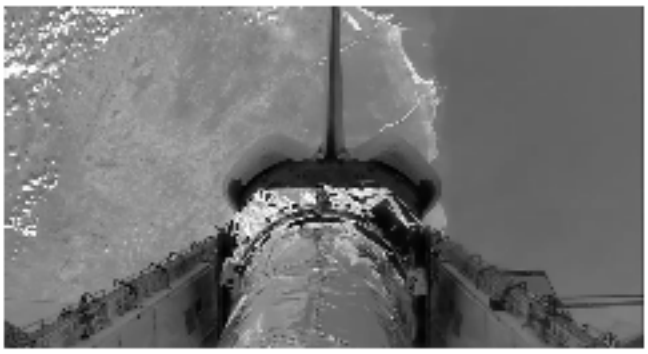

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

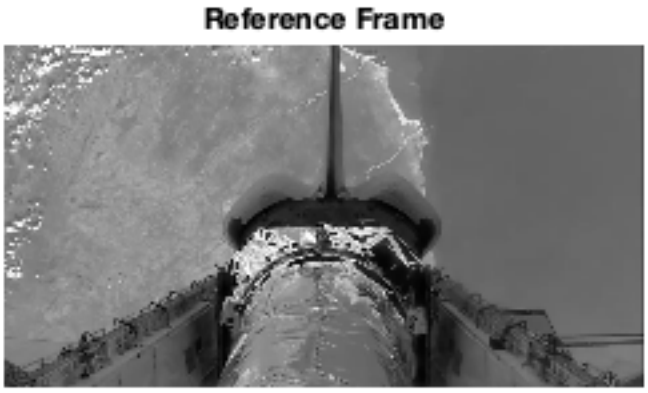
Reference Frame

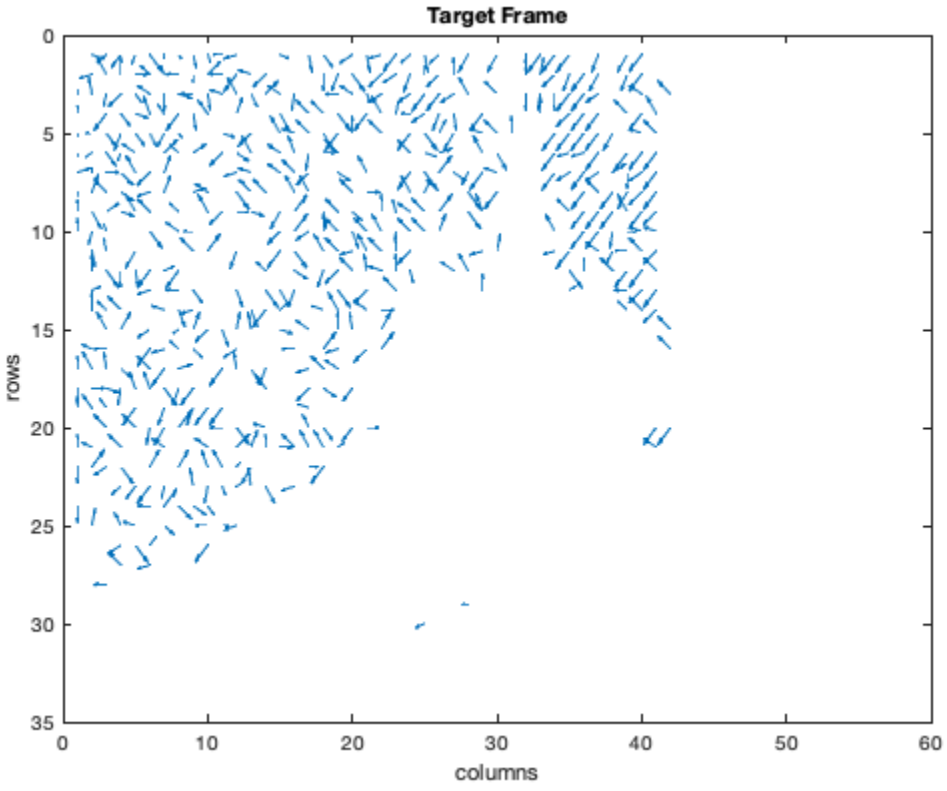




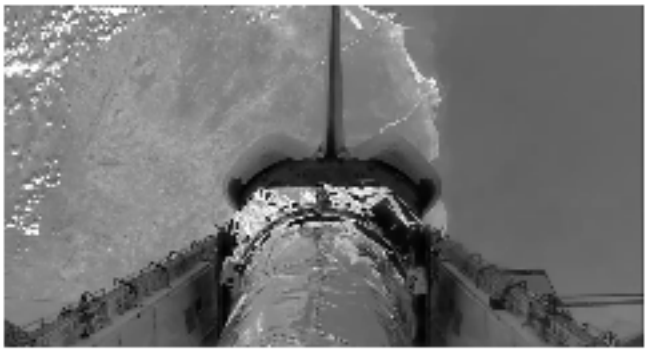

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

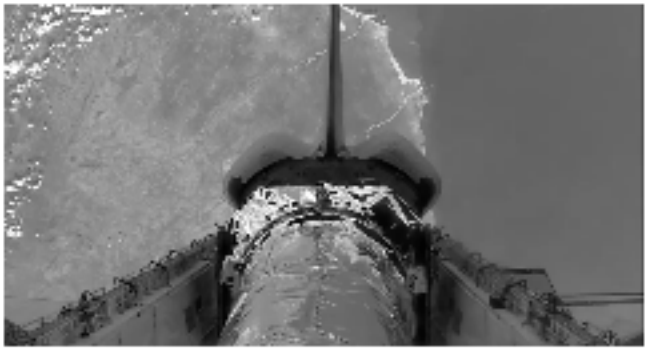


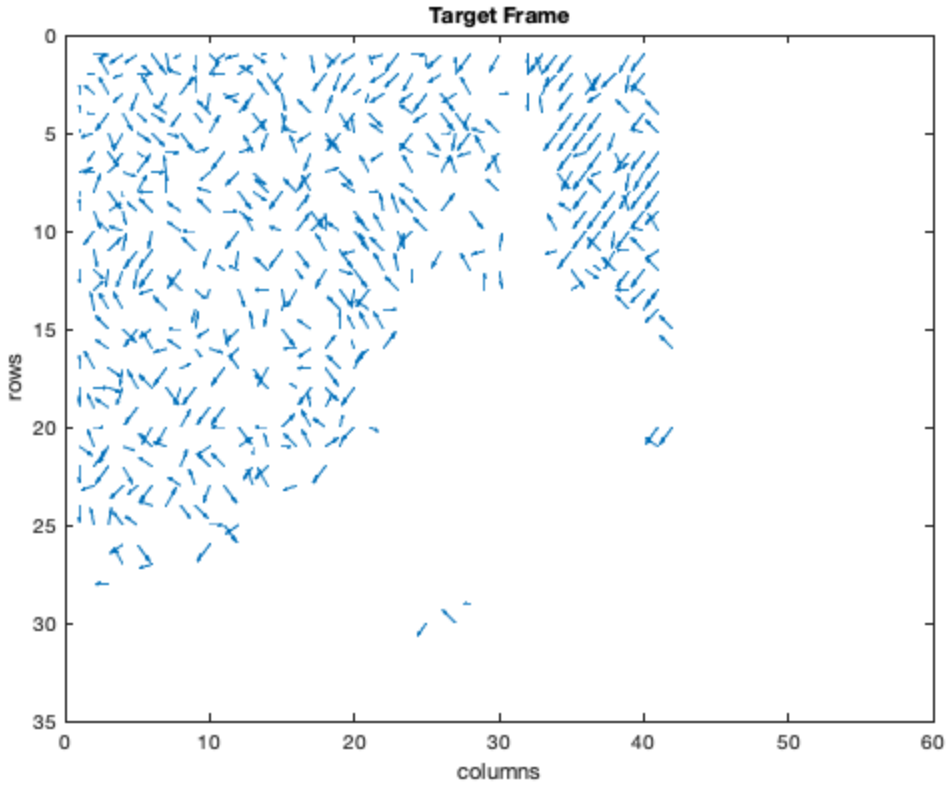

Reference Frame

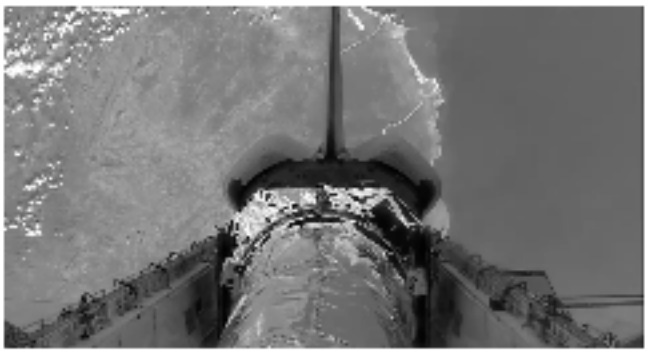

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

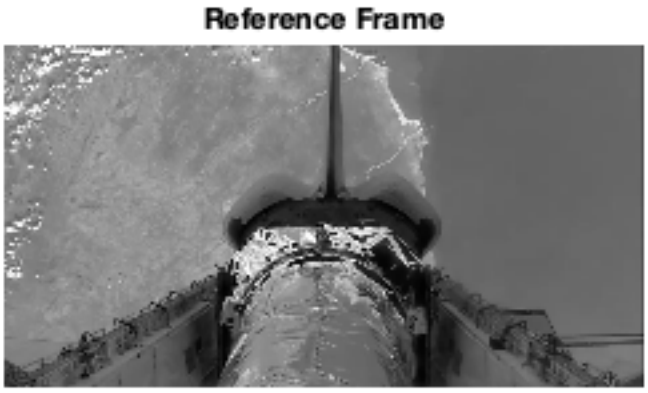
Reference Frame

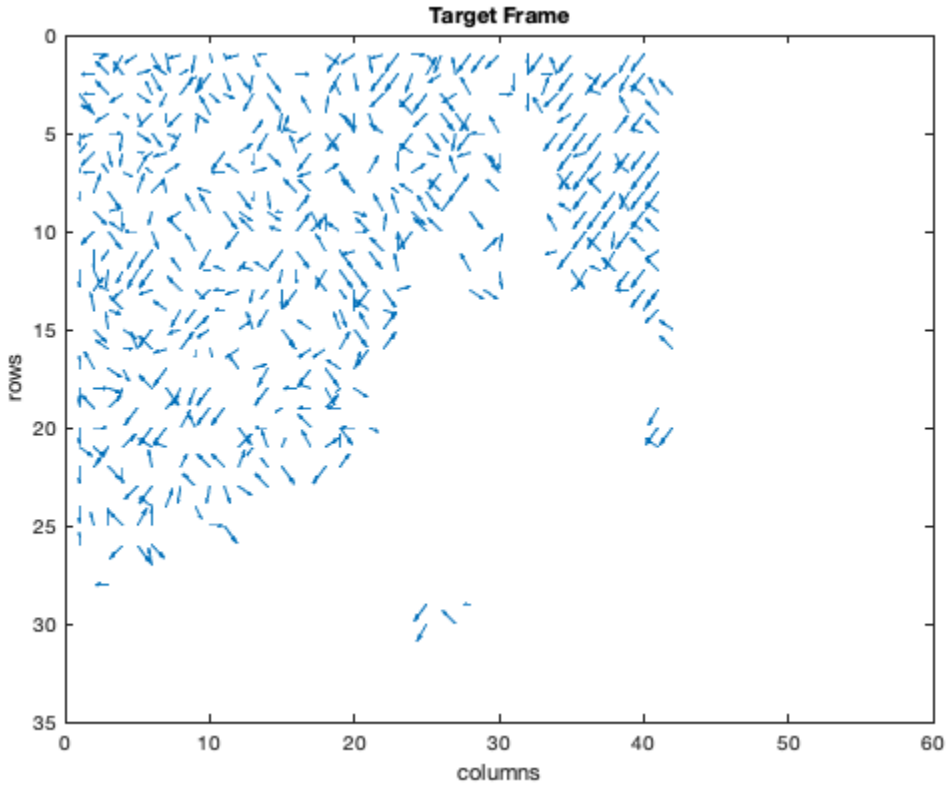




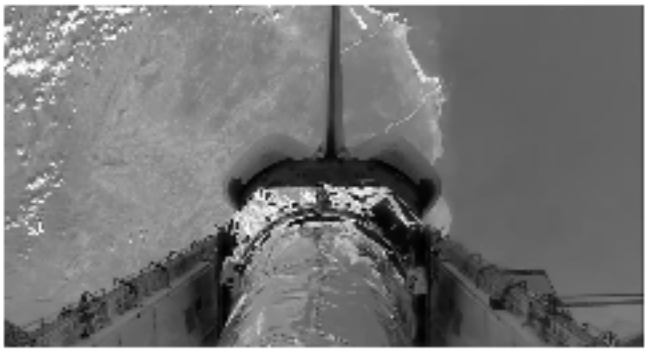

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

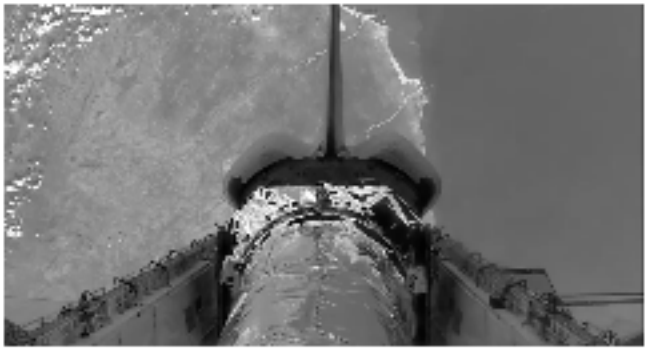


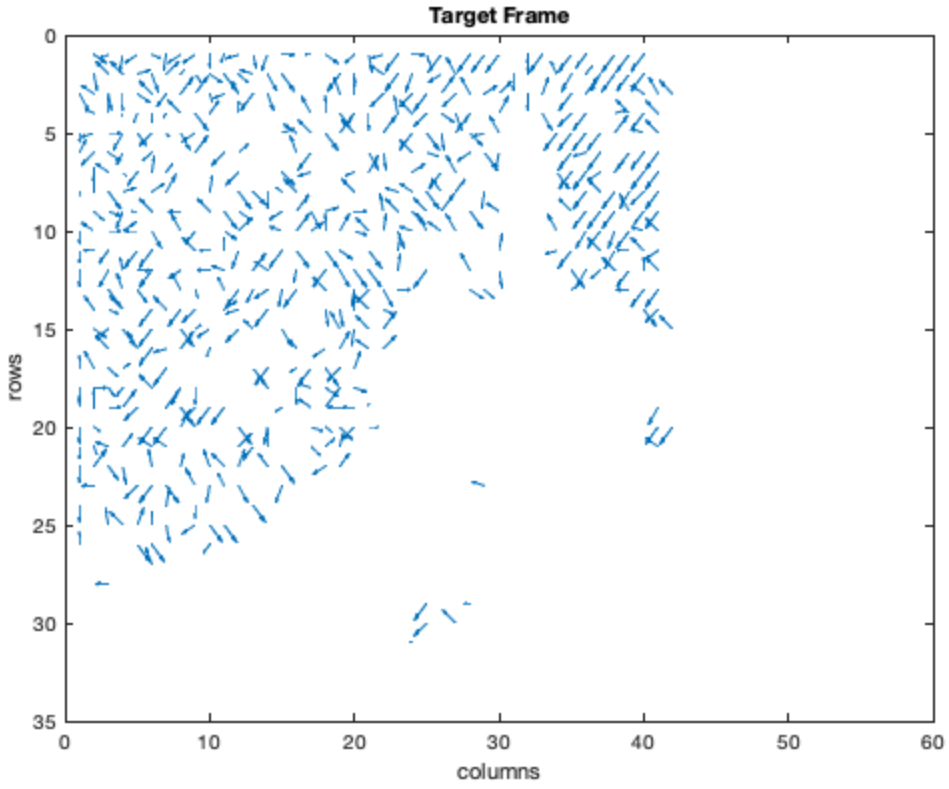

Reference Frame

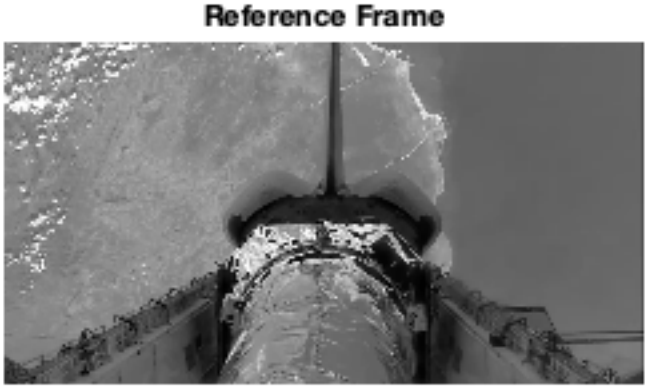

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

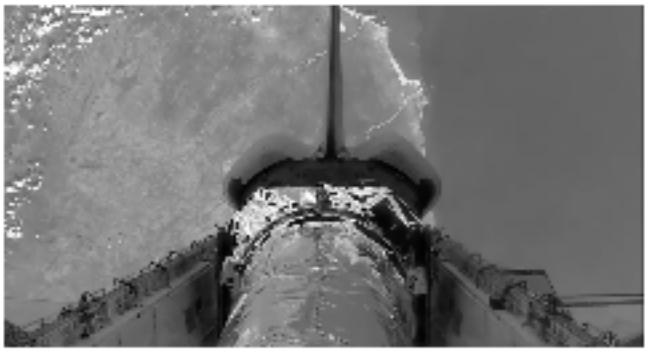
Reference Frame

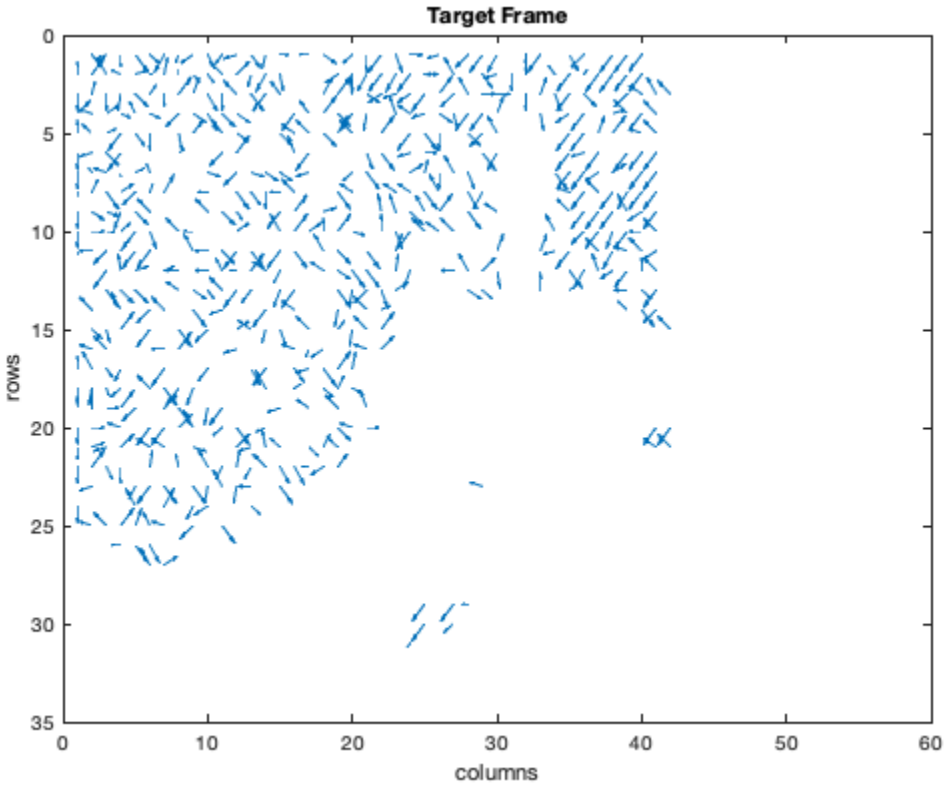




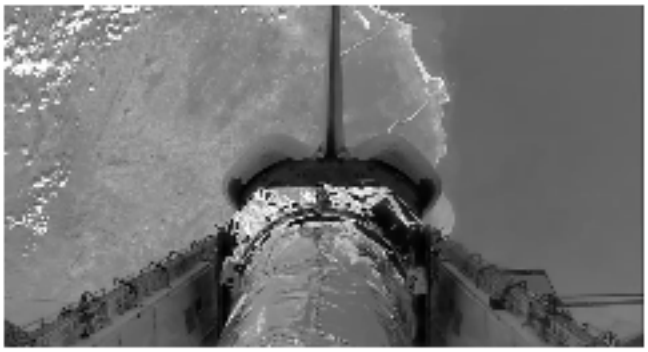

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

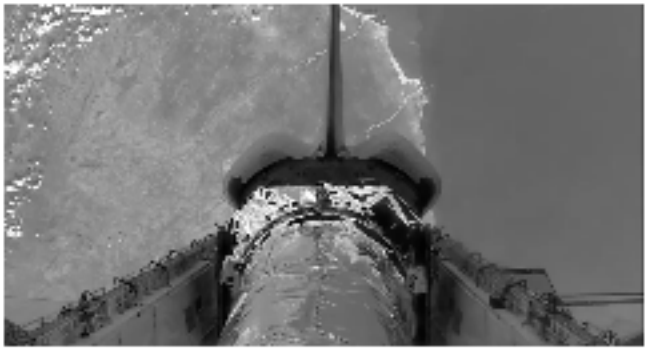

Reference Frame

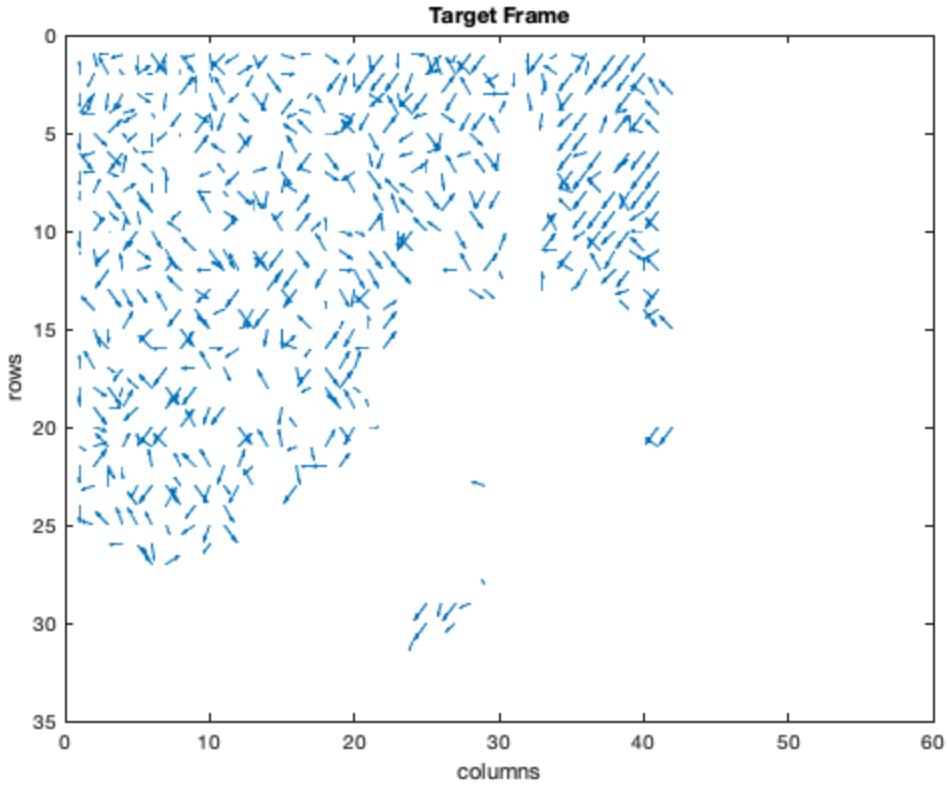


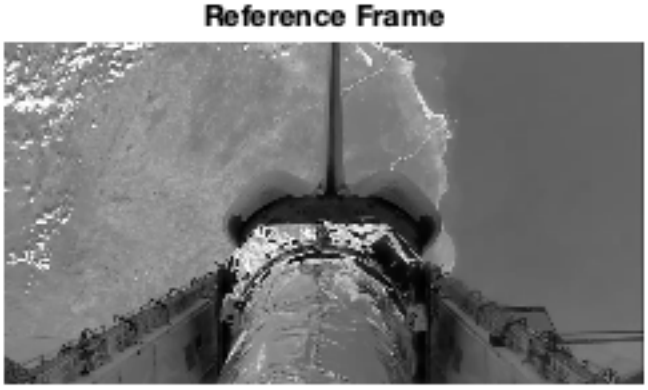

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

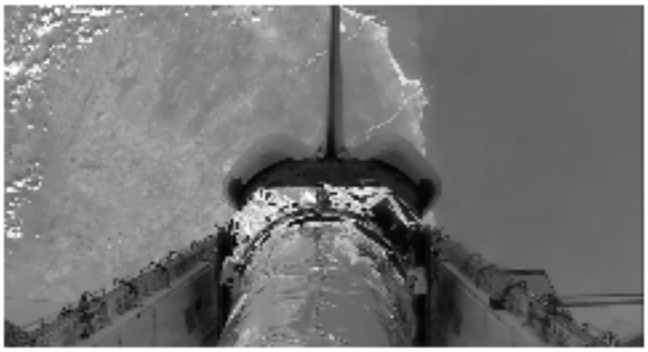
Reference Frame

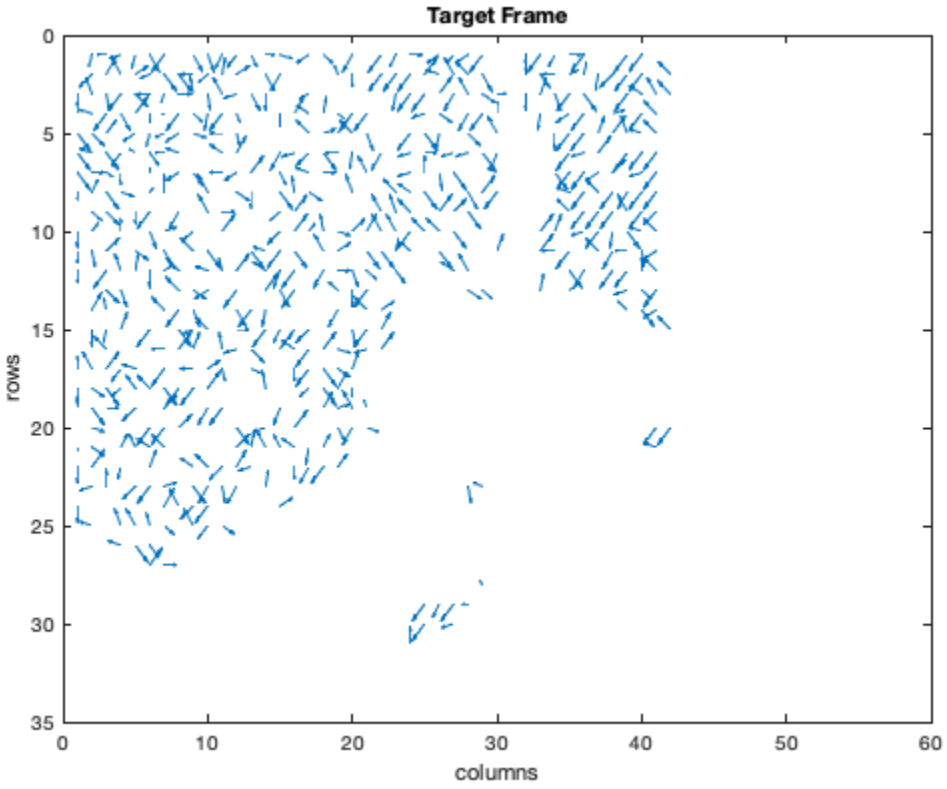




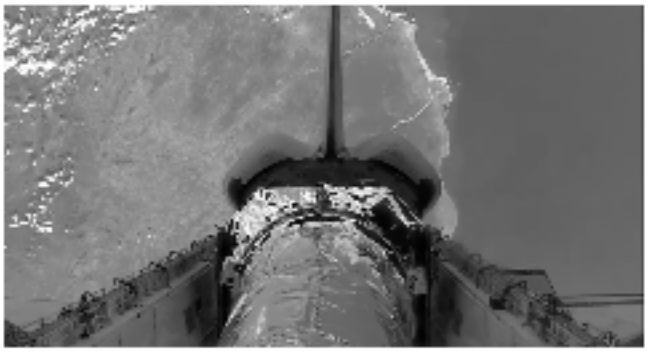

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

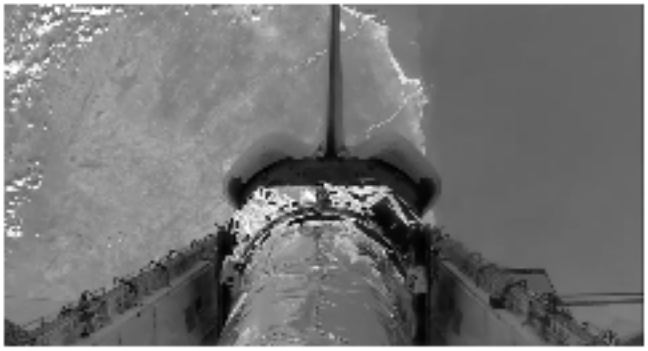

Reference Frame

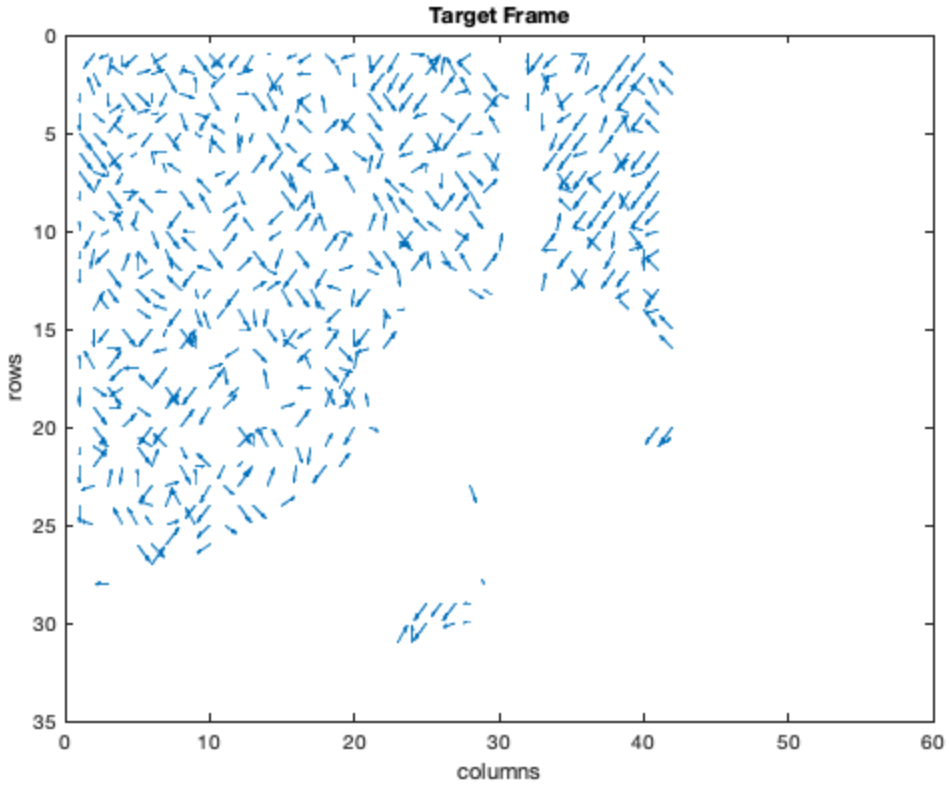


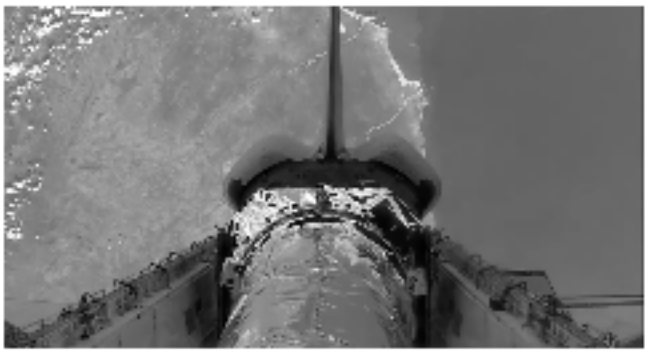

Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60

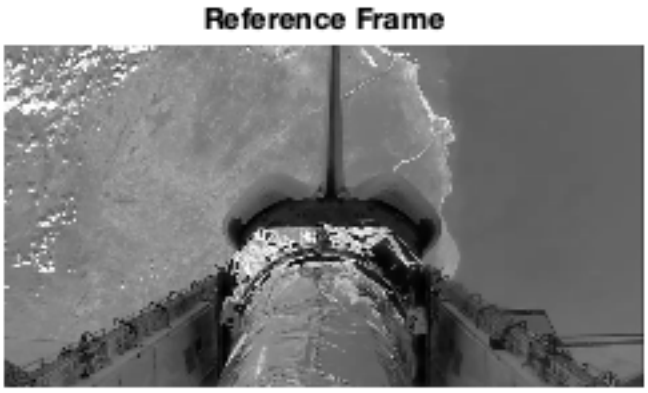
Reference Frame

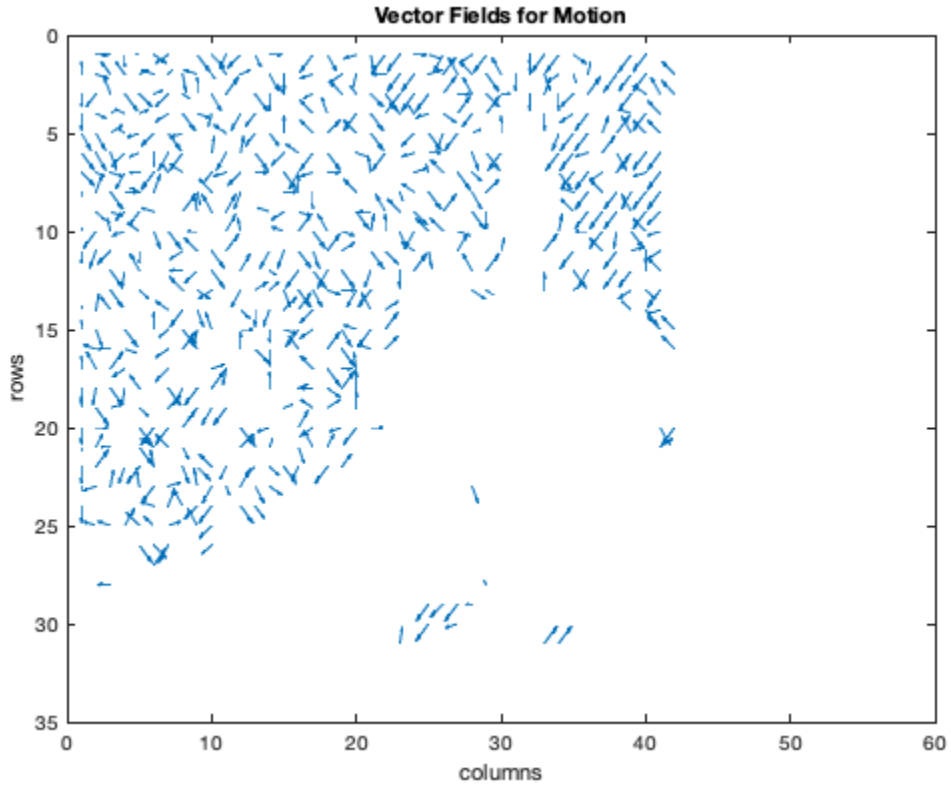





Target Frame
rows
columns
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60







Published with MATLAB® R2018b