Project Report on
Block Matching Algorithm
Submitted By: MANISH SONI

Objective:

Implement your own block matching algorithm for the video sequence available here: shuttle.tif

The output should be a report describing your approach, results, and conclusions, and must include a stream of images displayed with the matlab "quiver" command to show how the blocks move from frame-to-frame using the first frame in the data set as the reference.

This paper below describes at spatial correlation-based algorithm for block matching: roggemann_reynolds_oe.pdf

Actions

The references which follow describe MPEG, which uses a "sum of absolute differences (SAD)" algorithm for estimating the shifts:

Introduction:

A Block Matching Algorithm helps in locating macroblocks which are present in subsequent frames of a video frames for motion estimation. The idea is that pattern corresponding to any object present in a frame of video will be available in subsequent frames. Using this technique can help removing temporal redundancy and can help in video compression. In this technique the content of a particular macroblock can be referenced once and can be used to generate an estimation of subsequent frame using detected motion vectors.

Block Matching algorithm divides the current frame of a video into macroblocks and compare these macroblocks in subsequent frame of video to find out the new position in the next frame. During this process a vector is created which keeps the history of movement of a particular block. The movement, calculated for all the macroblocks pointing out the transition from past frame to newer frame of video thus called as a motion estimated in a frame.

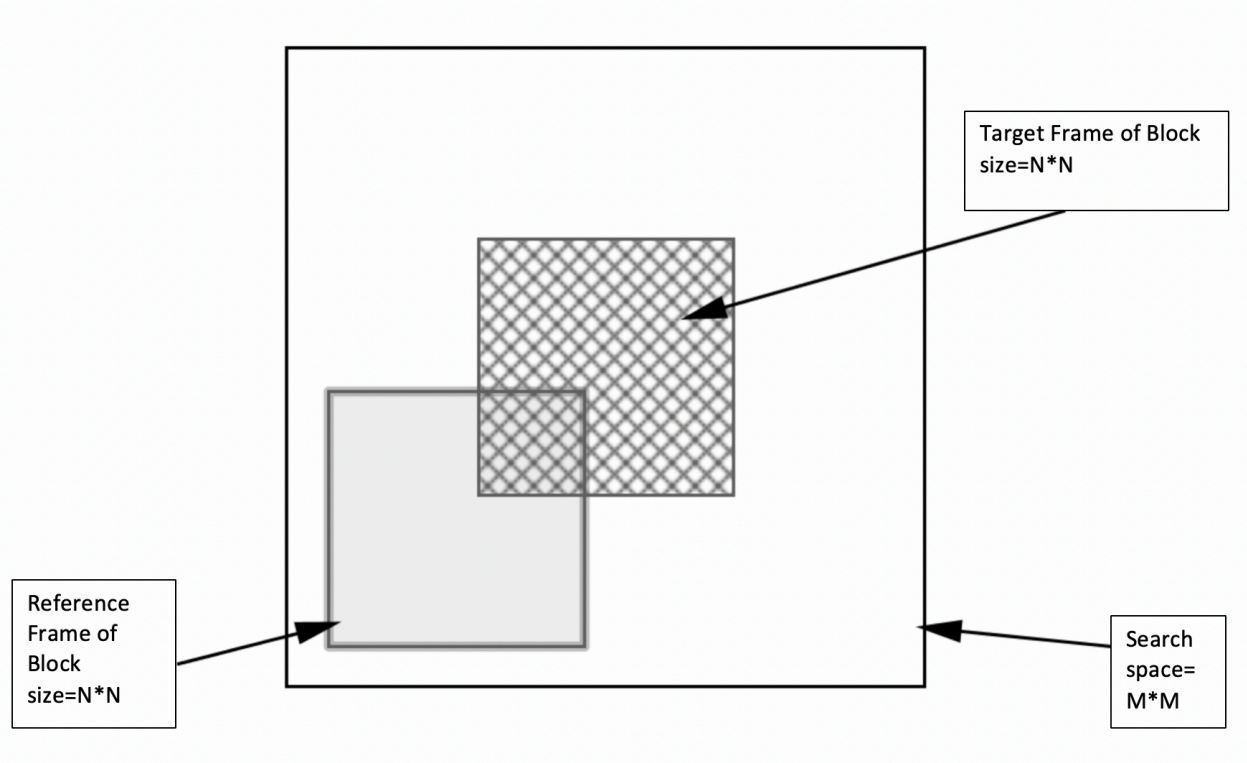

Figure 1: Block Matching Algorithm for Macroblocks

SAD (SUM OF ABSOLITE DIFFERENCE):

Two blocks are compared and the difference of their pixel value is summed up for all the pixels in block for its absolute value.

The objective of SAD is to provide us the minimum value.

Using The SAD value our Objective is to find Global Minima rather than Local Minima in a search space.

Thus SAD value is a key to match out the blocks from Reference frame to Target Frame.

Approach:

Approach is based on Figure 1. We need to target Global Minima for SAD in the entire Search space so to claim it matching block.

1. We decide a search space for a particular macroblock.
2. For a particular Macroblock we navigate it from the 1st row and column of the search block
3. Thus a macroblock keeps on moving through the search space

4. At each point we take SAD (Sum of Absolute differences) between Target Frame and Reference frame for varying postion.
5. If the Reference frame and target frame for that macroblock has less SAD then it means that both the blocks are almost similar.
6. We continue to find the Global Minima for SAD rather than Local Minima in vicinity.
7. At the end of navigating through whole search space. We get the Minimum SAD which represents the matching position.
8. We store the row and column indexes of the matching block in a vector.
9. We continue the Same step from 1-8 for other Macroblocks until the entire frame has been checked
10. At the end of step 9, the saved vectors of matching block positions are used as detected motion vector and fed to quiver command to show the movtion vector representation in image format.

Note: We can use a threshold value for comparison for SAD value so that minimalistic SAD value can be ignored and can be considered as the most matching blocks.

Results:

For Block size= 8
Search space= 4 for Shuttle.tiff

{Please Note: the title of all the images has not got rendered properly by matlab for some reason }{Top image:Vector fields for Motion, Left Image: Reference Image, Right Image: Target Image}

Results of the block matching algorithm are as per follow.

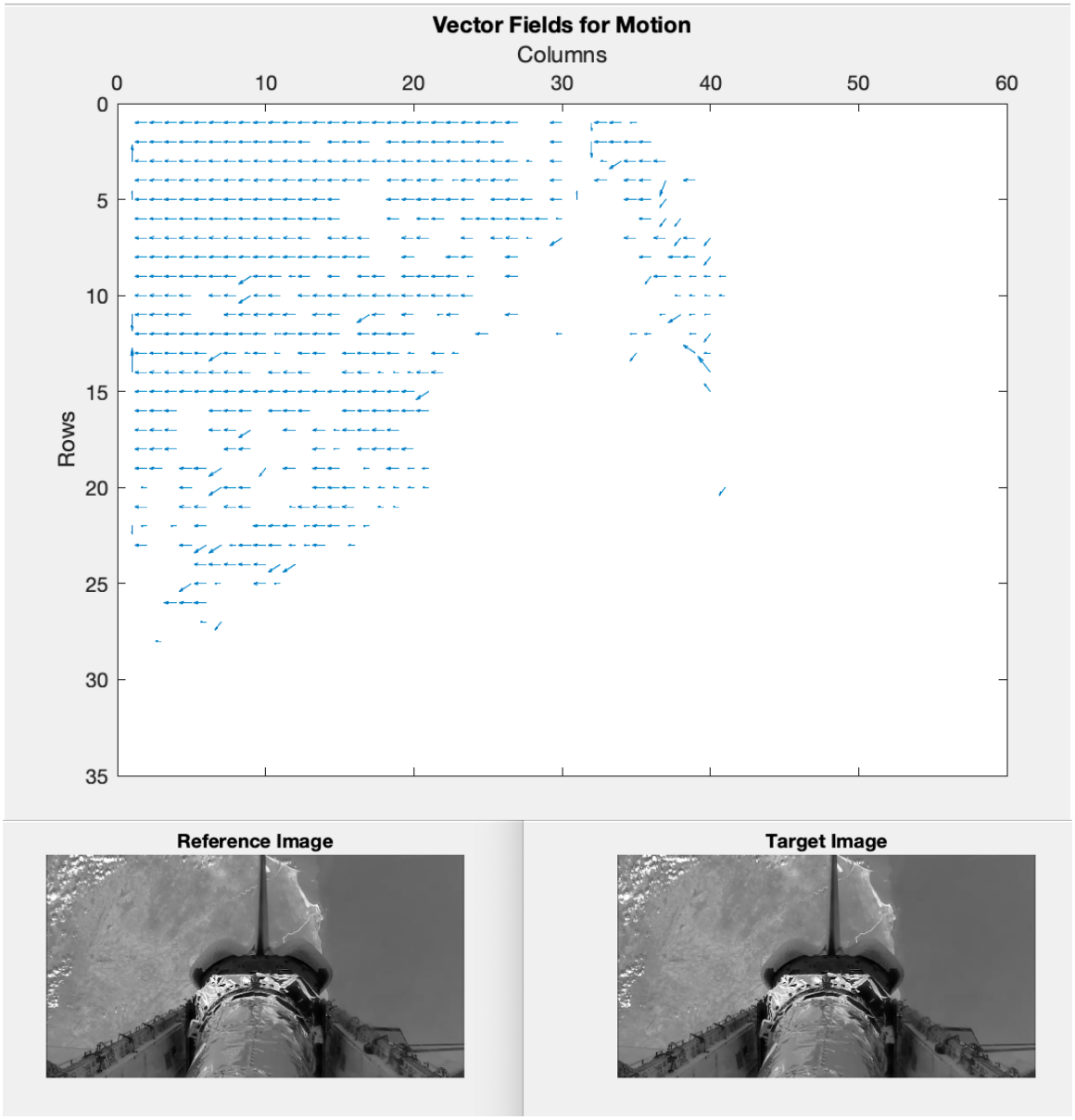

Figure 2: 1st frame and 2nd frame for block matching

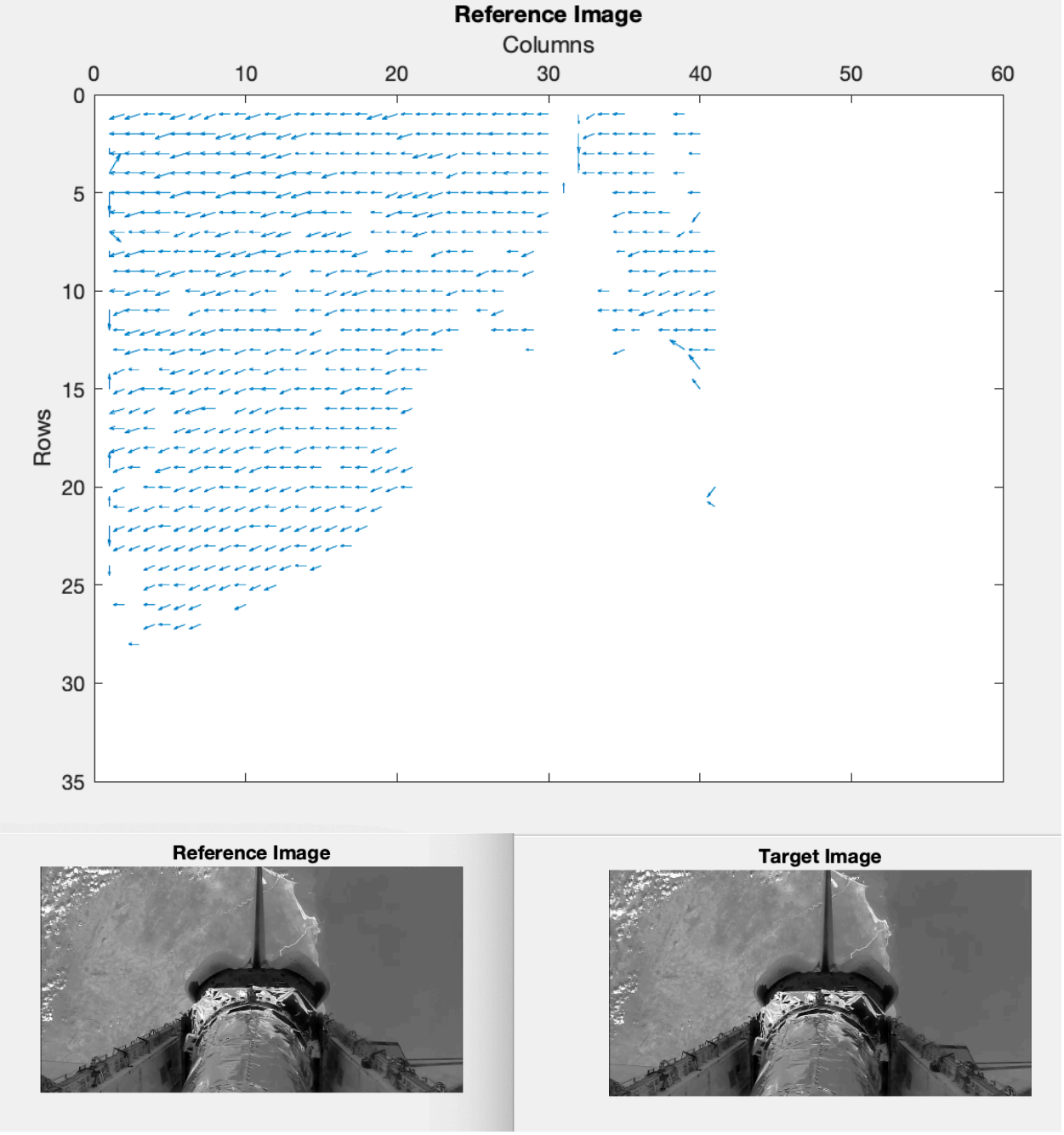

Figure 3: 1st frame and 3rd frame for block matching

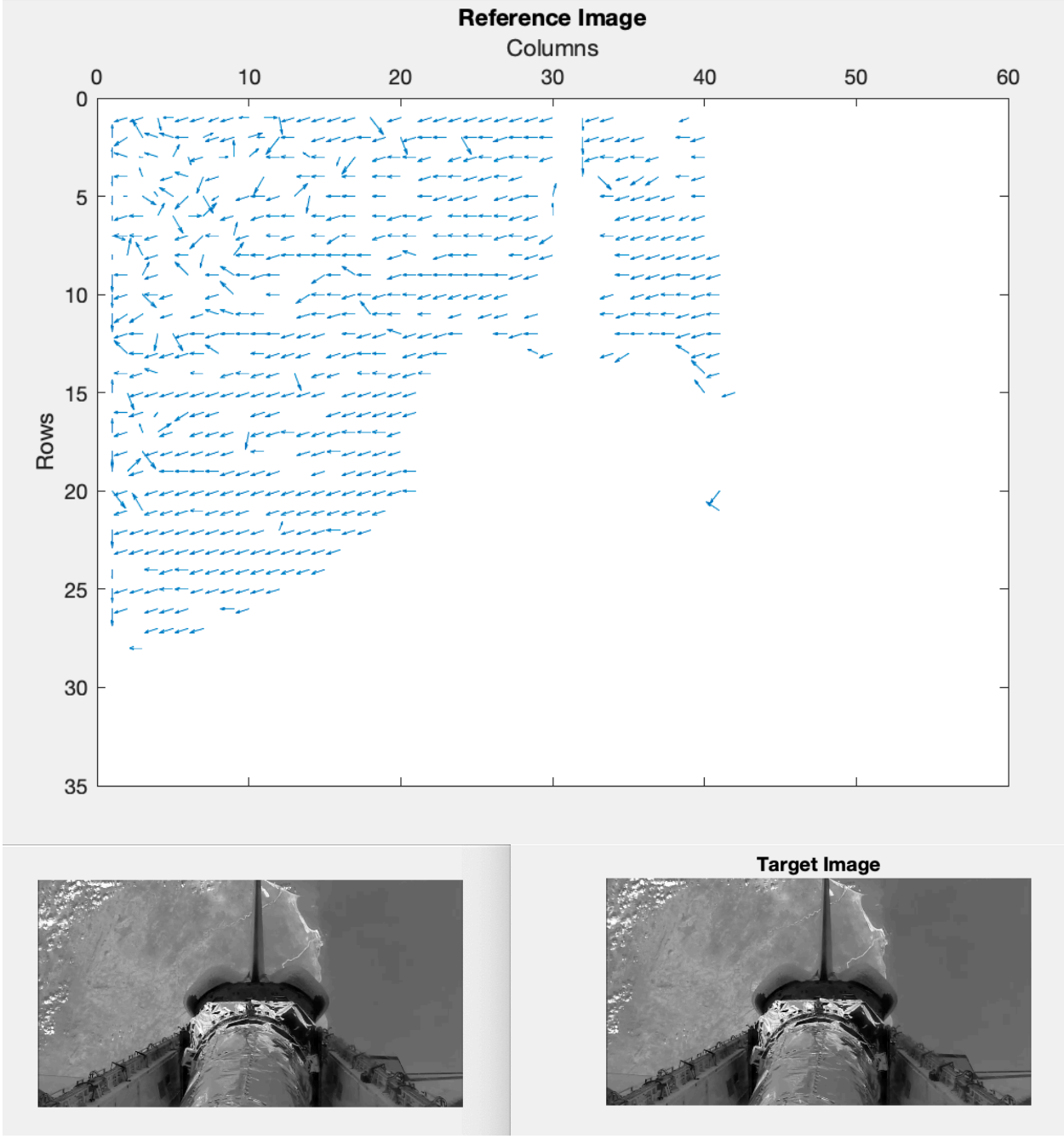

Figure 4: 1st frame and 4th frame for block matching

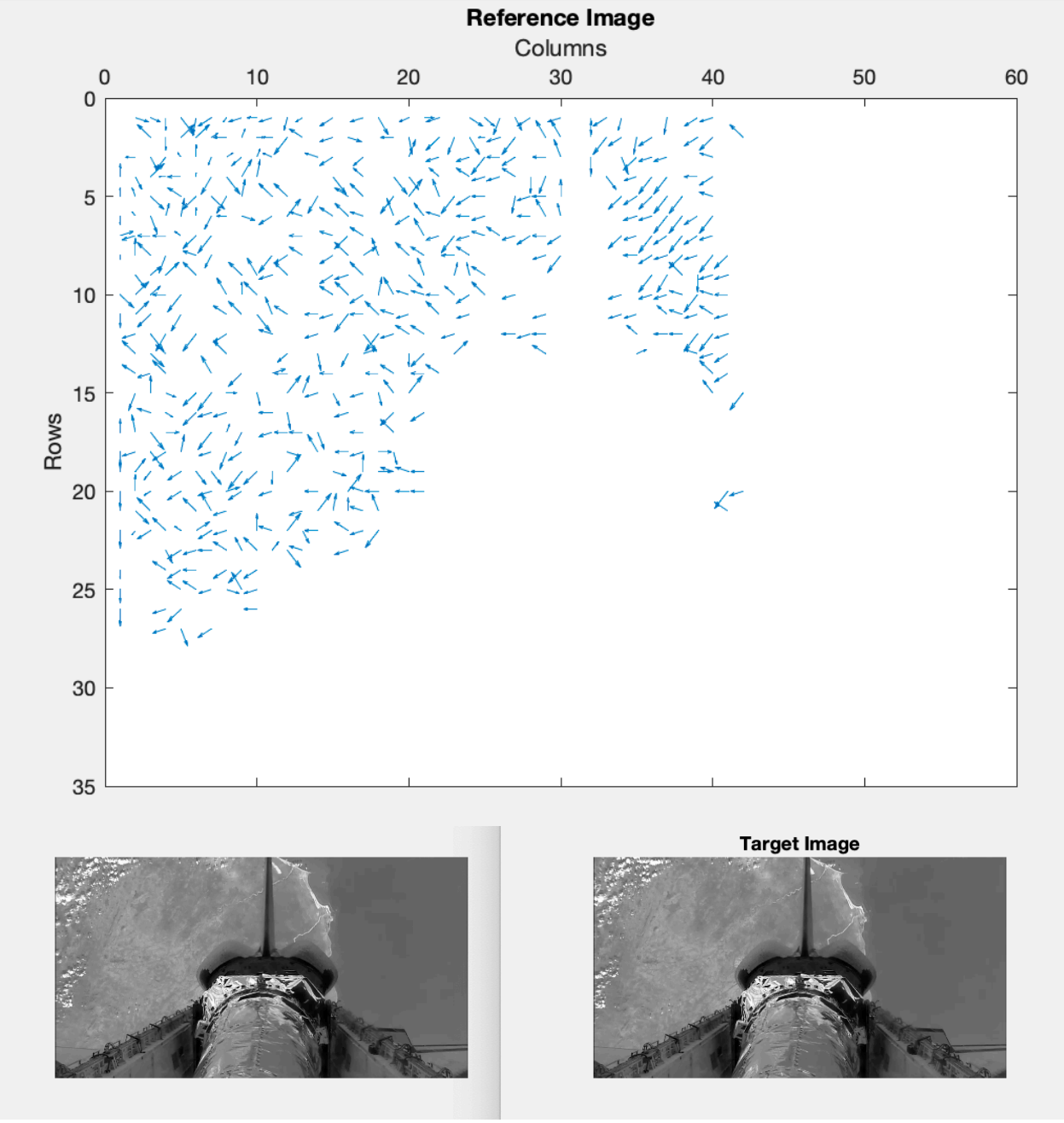

Figure 5: 1st frame and 5th frame for block matching

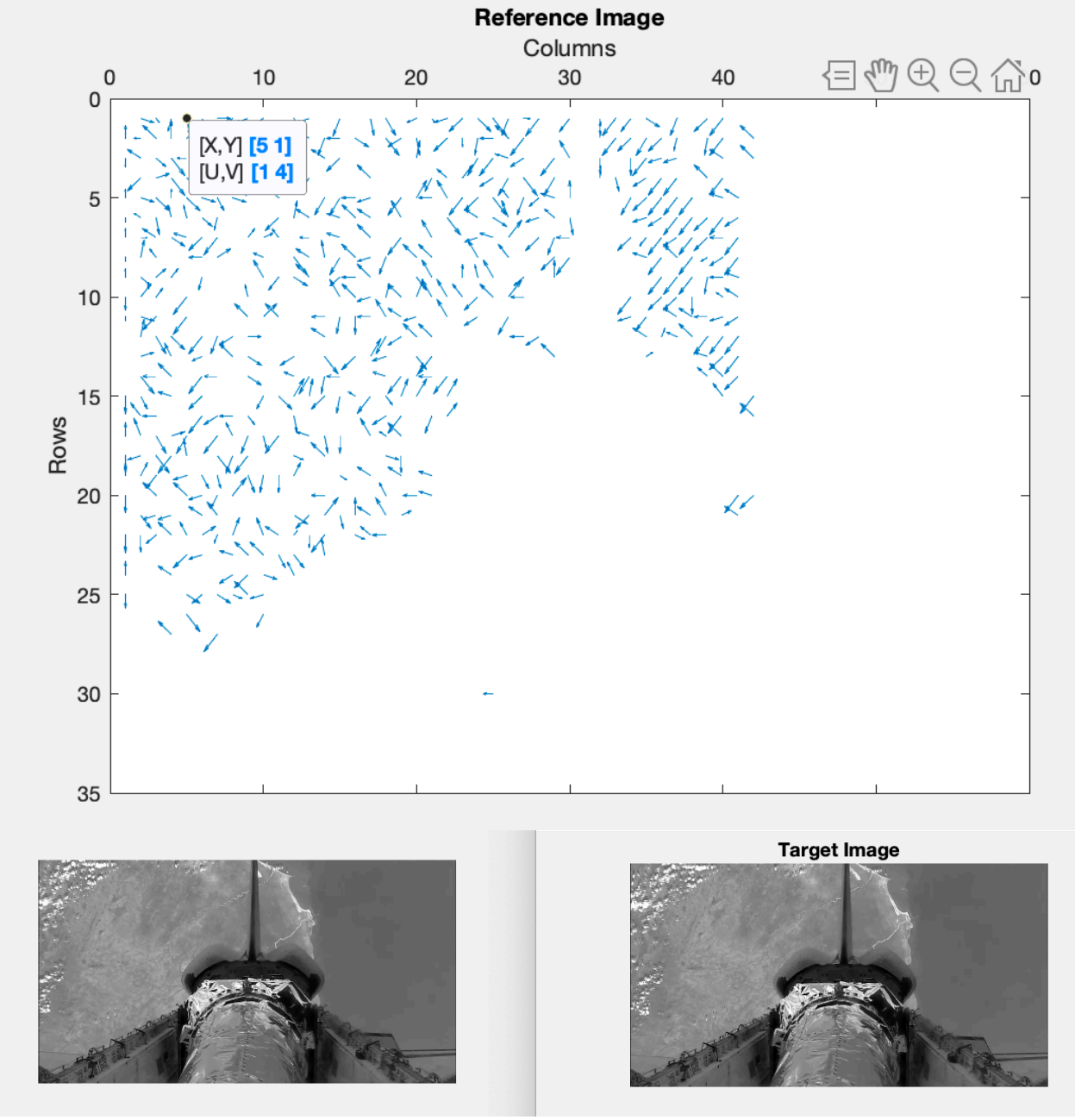

Figure 6: 1st frame and 6th frame for block matching

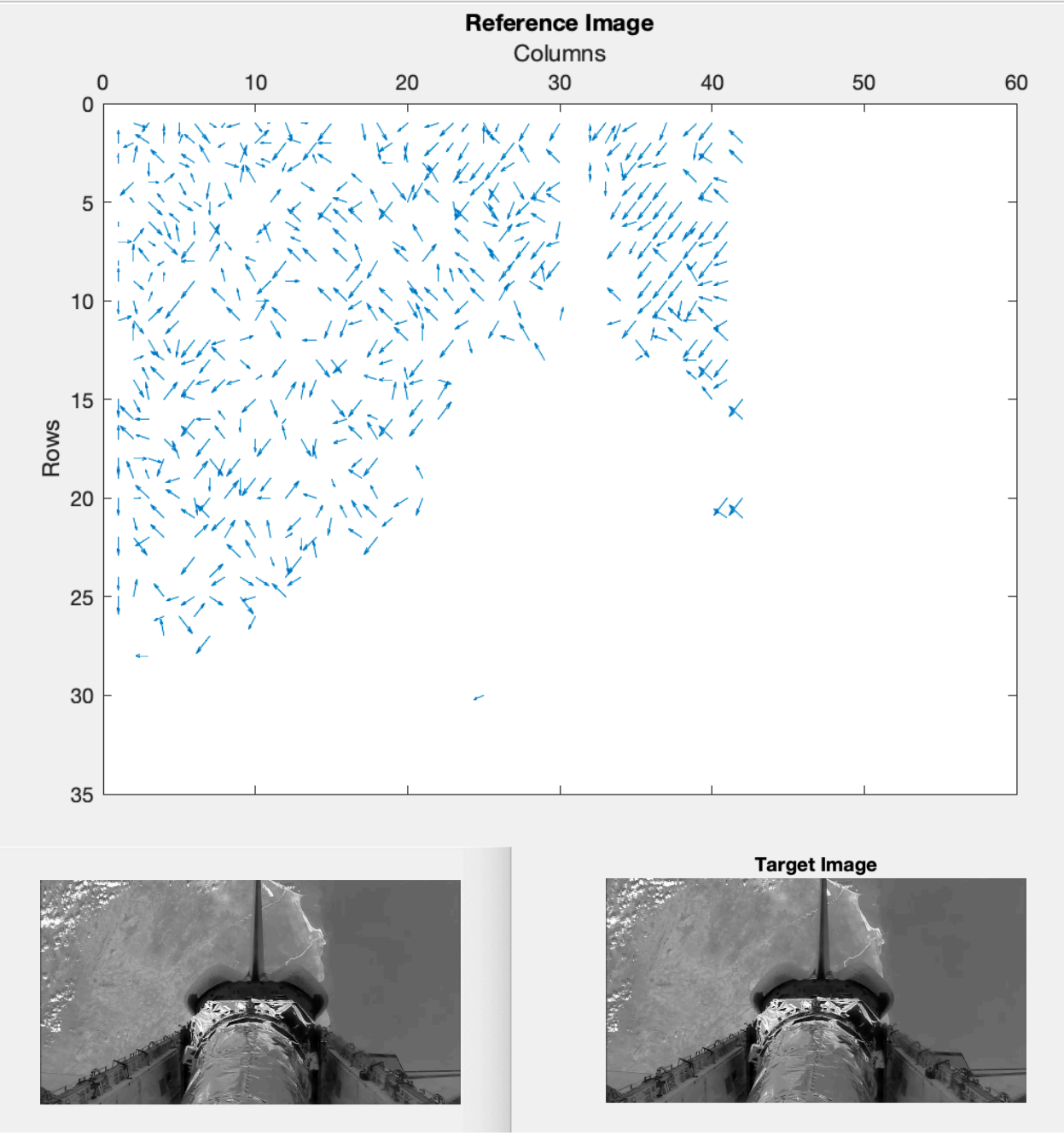

Figure 7: 1st frame and 7th frame for block matching

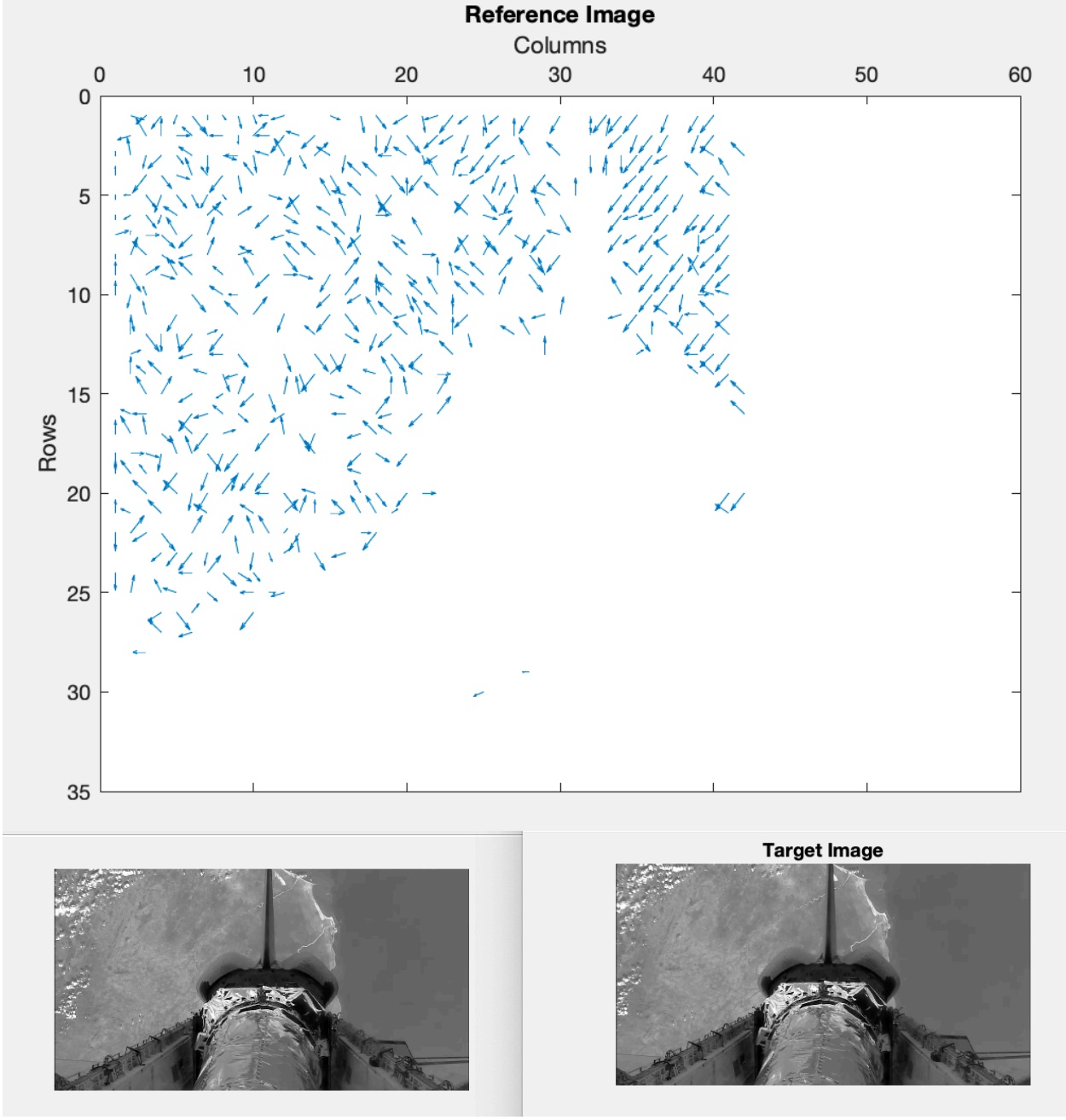

Figure 8: 1st frame and 8th frame for block matching

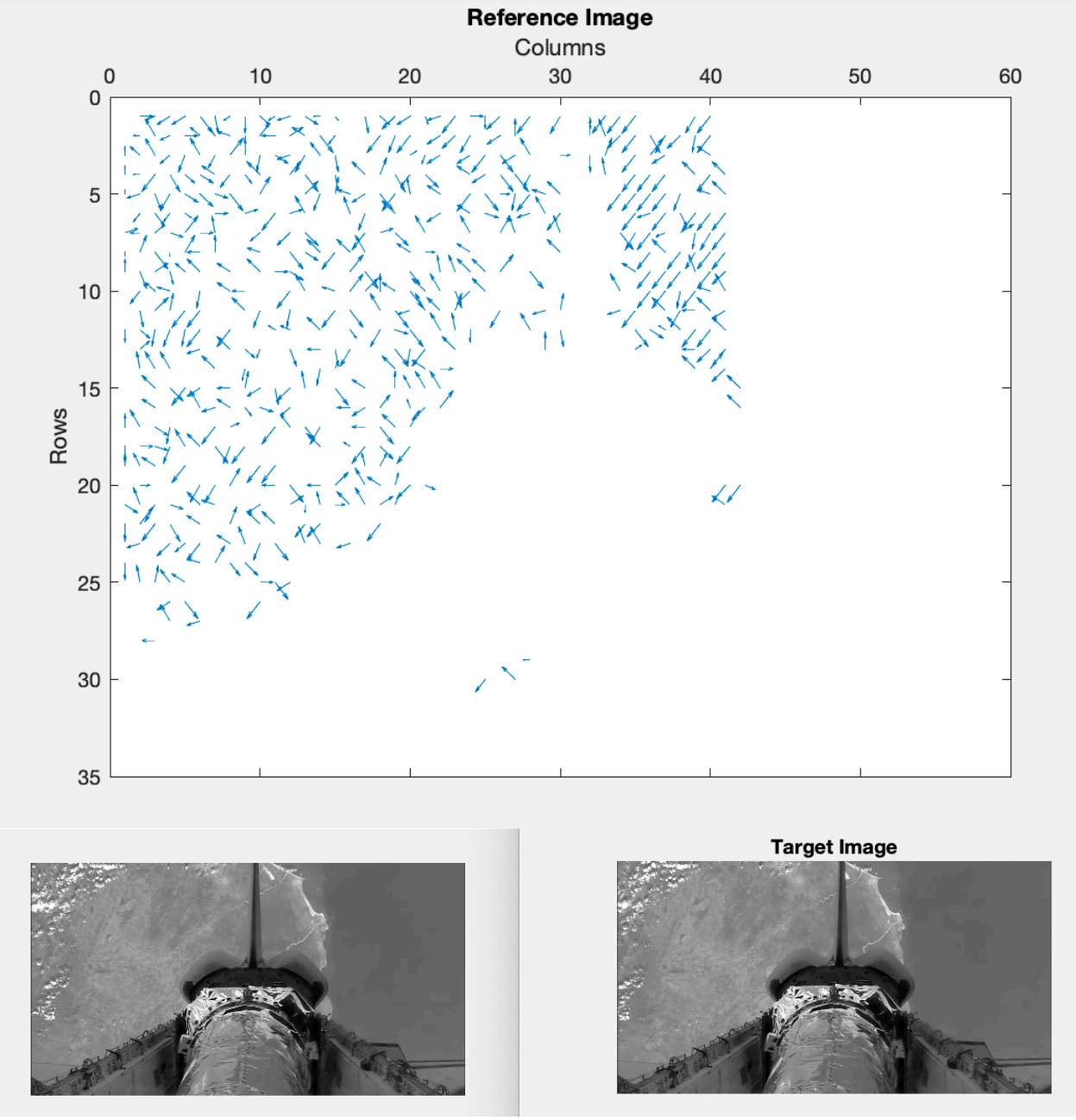

Figure 9: 1st frame and 9th frame for block matching

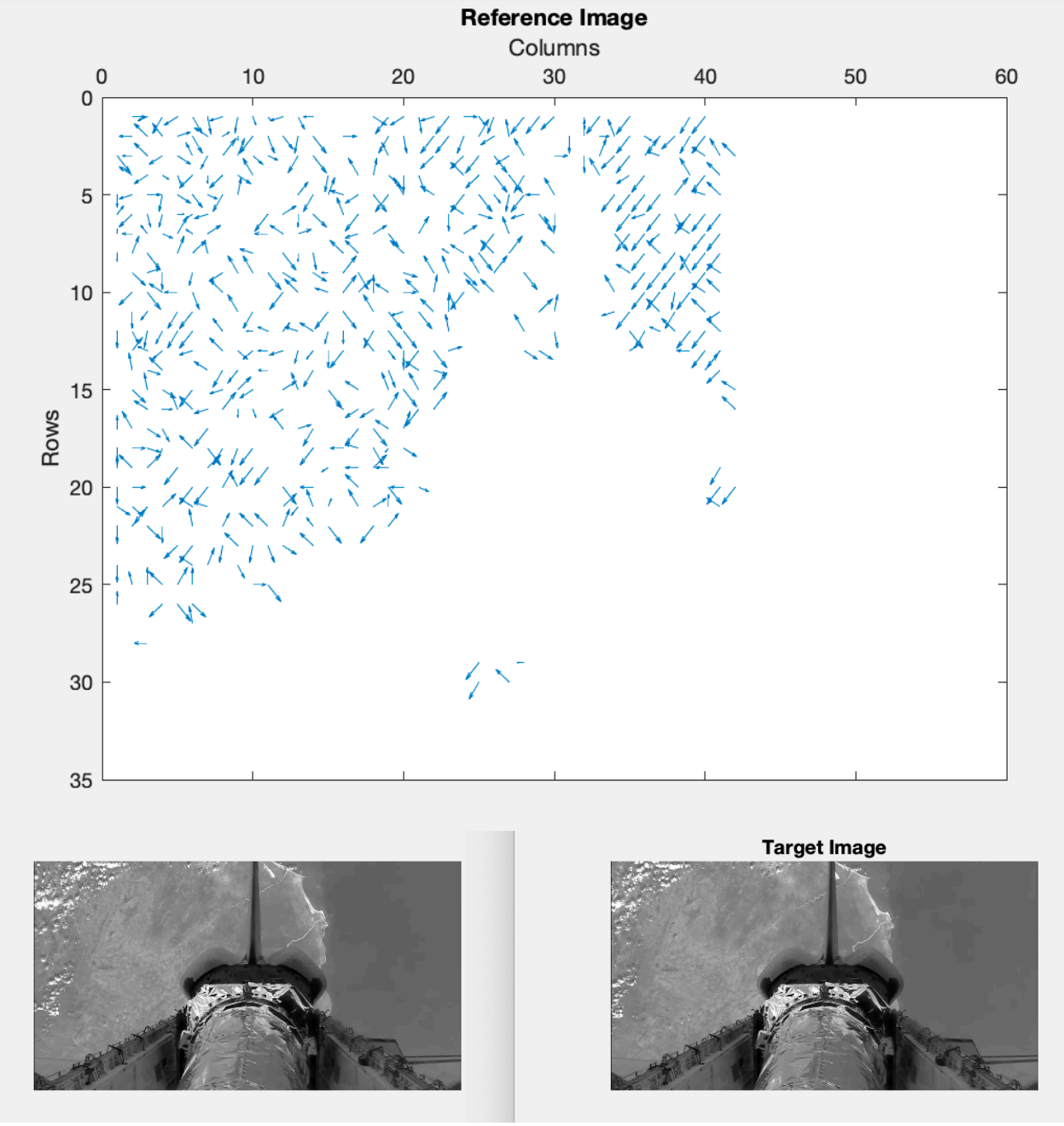

Figure 10: 1st frame and 10th frame for block matching

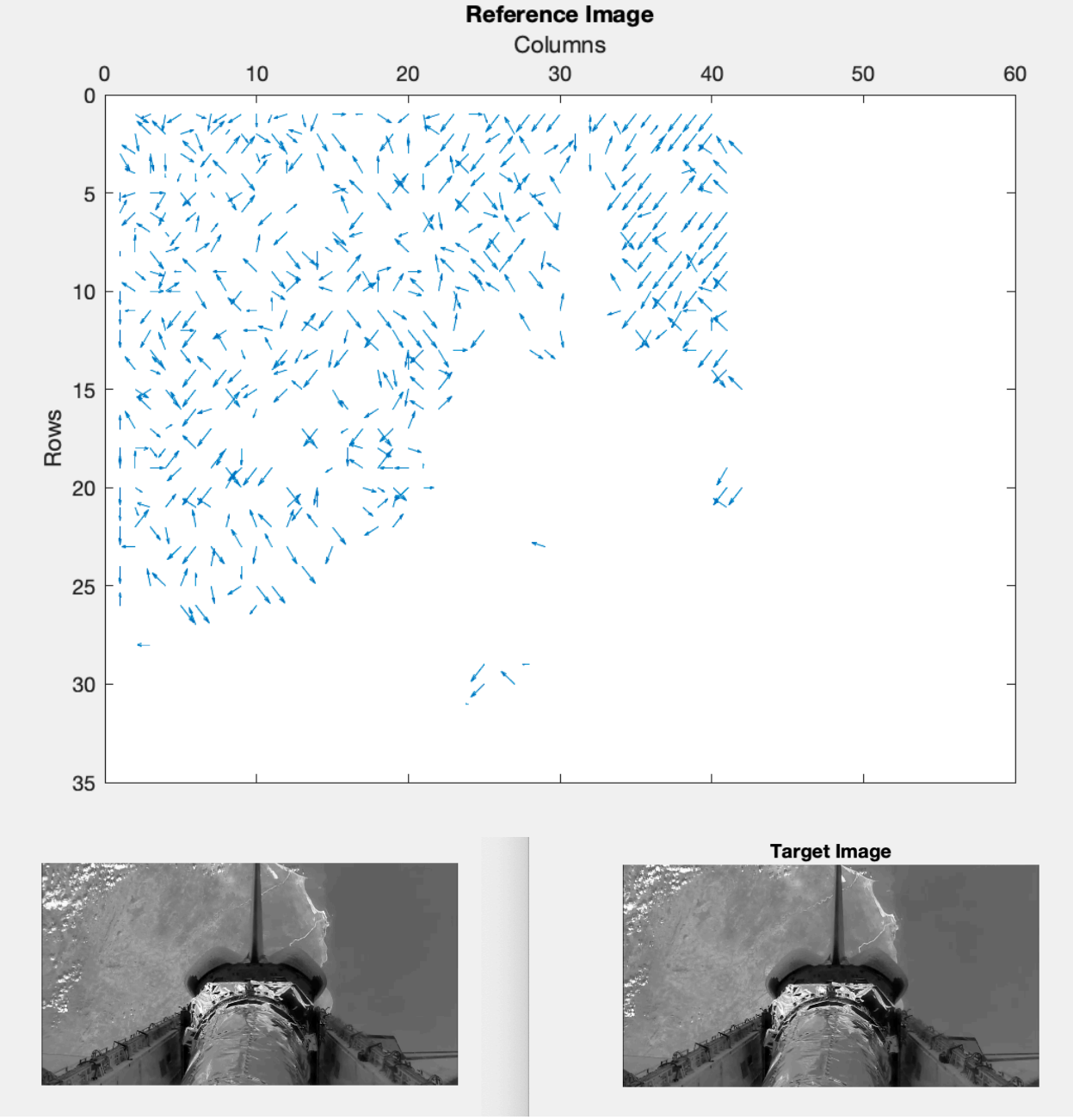

Figure 11: 1st frame and 11th frame for block matching

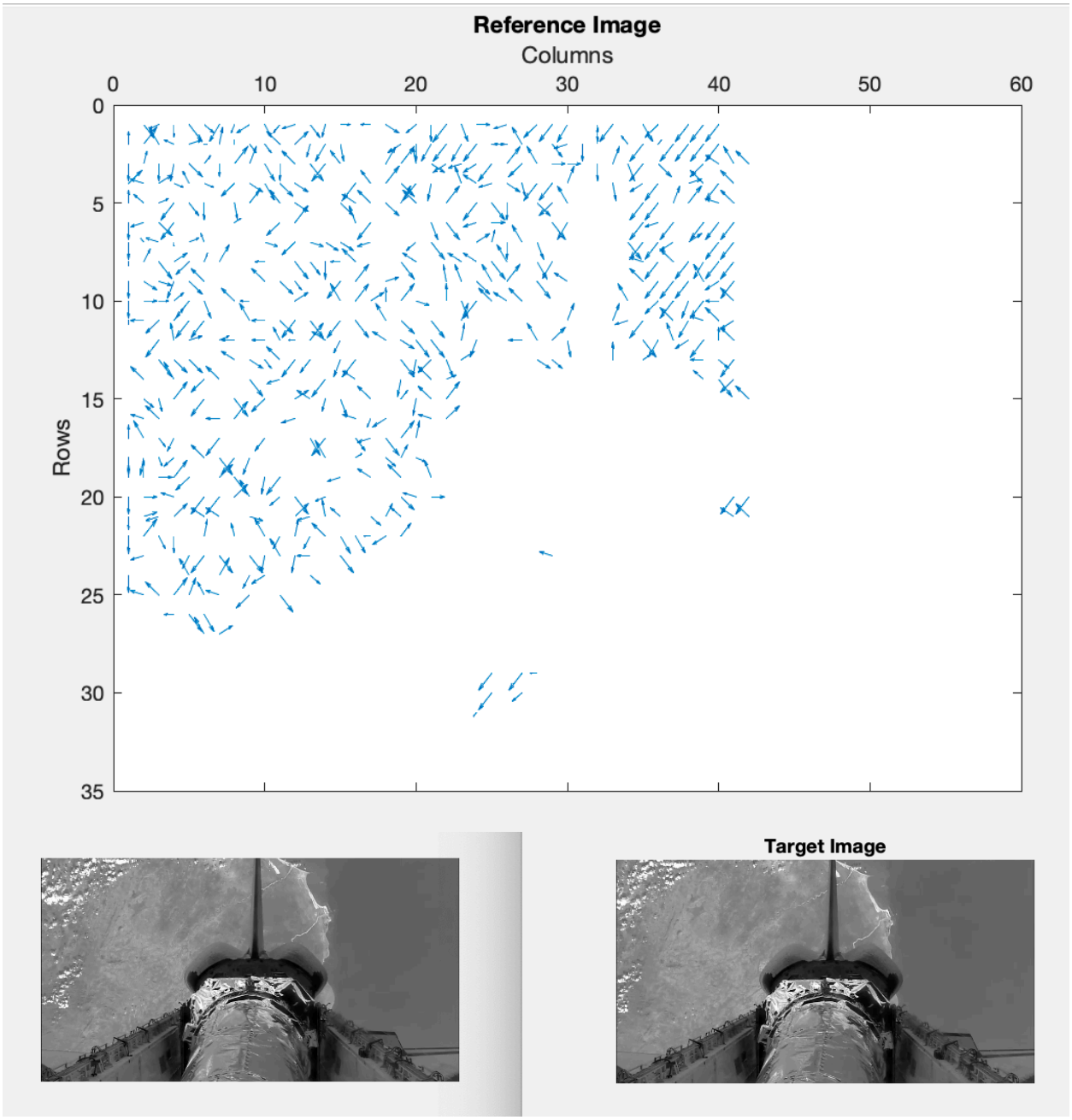

Figure 12: 1st frame and 12th frame for block matching

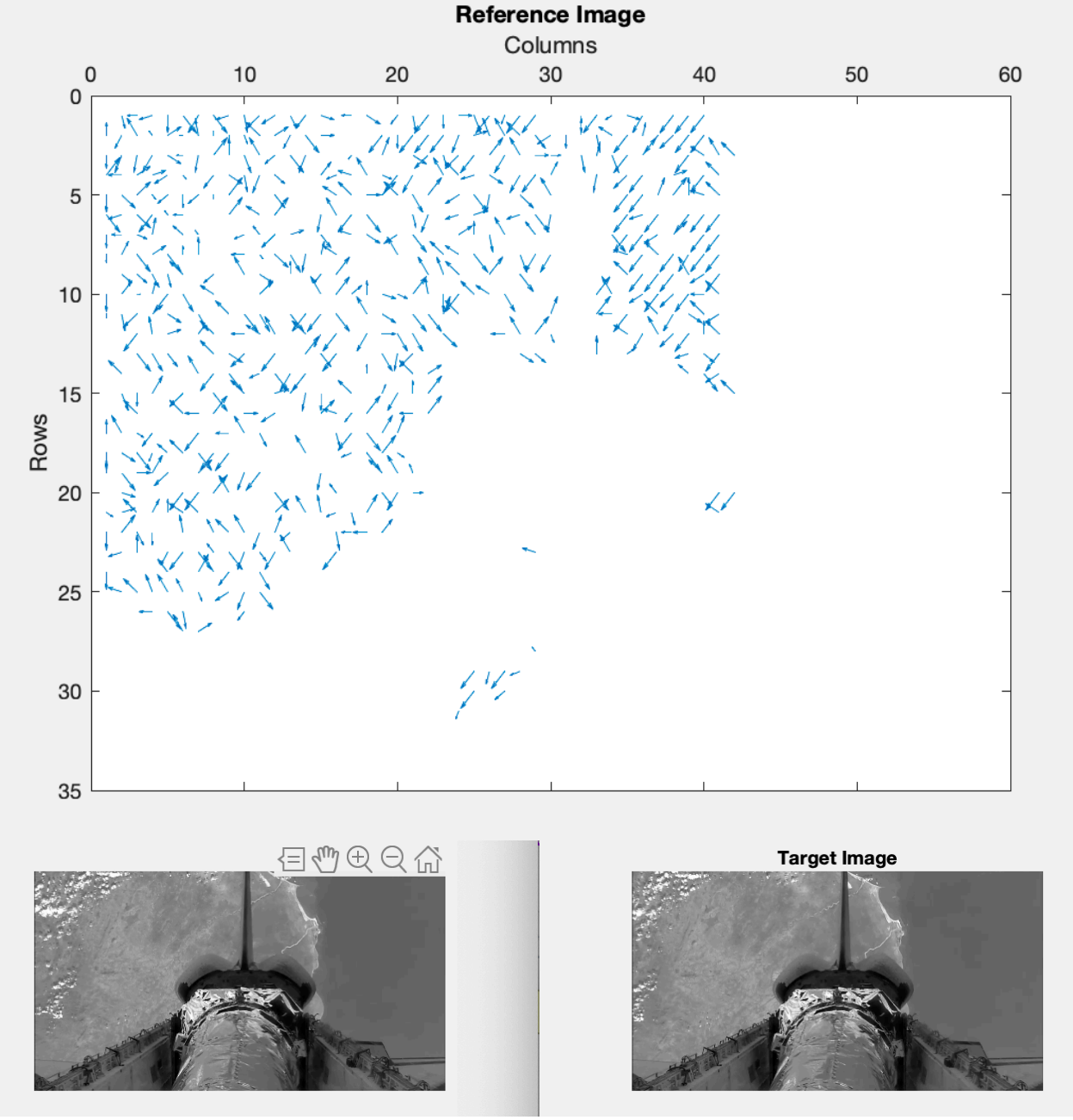

Figure 13: 1st frame and 13th frame for block matching

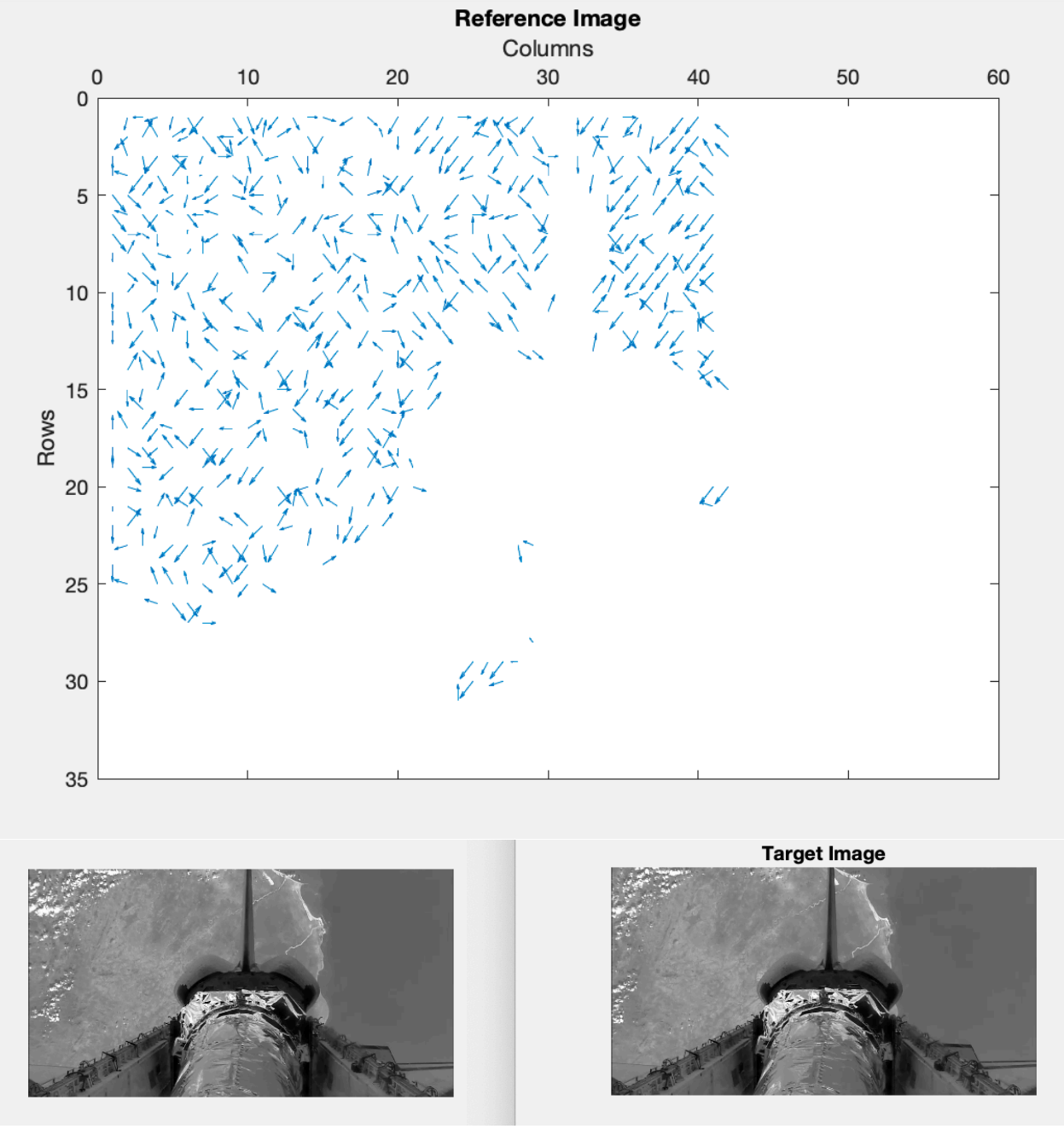

Figure 14: 1st frame and 14th frame for block matching

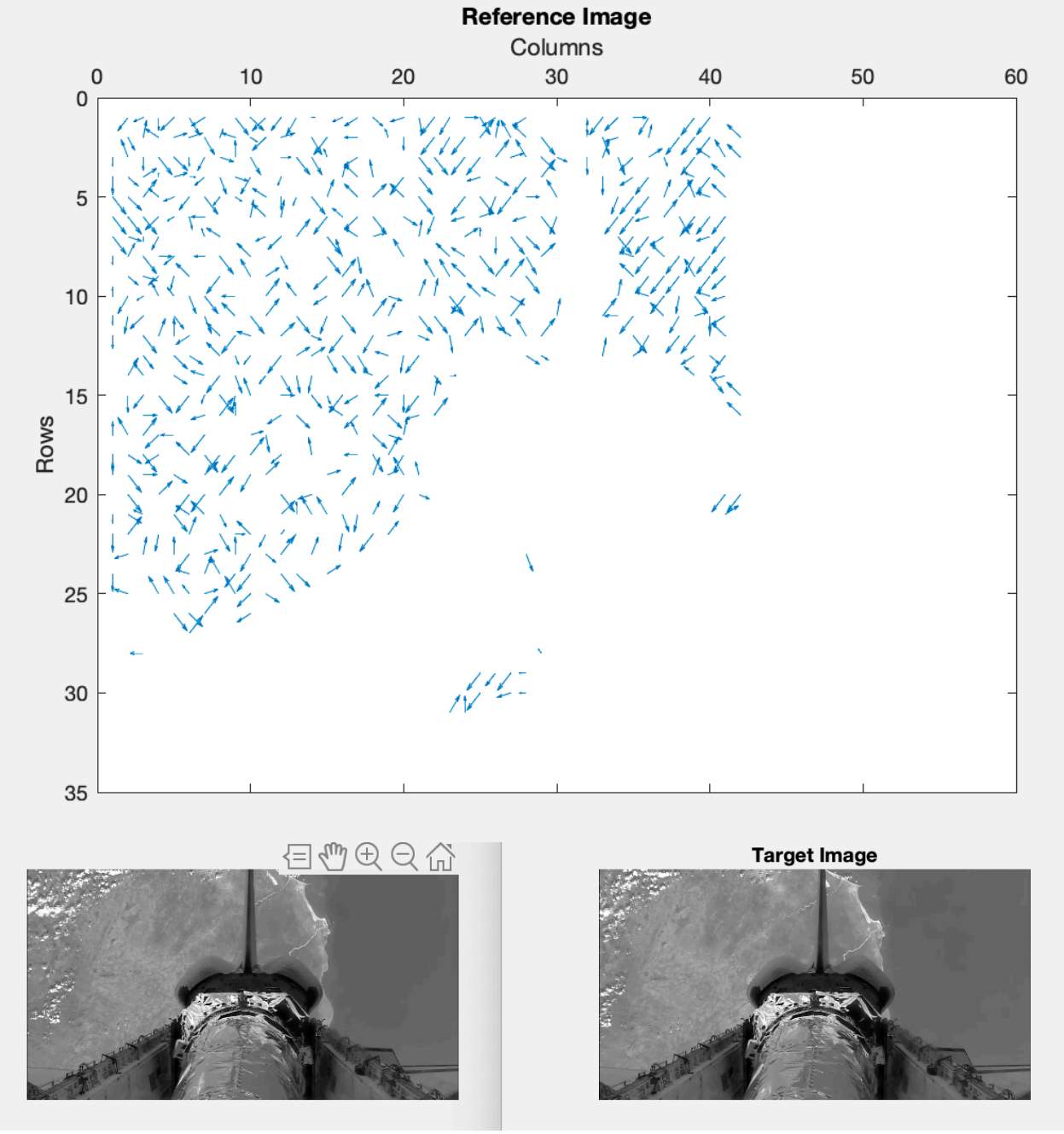

Figure 15: 1st frame and 15th frame for block matching

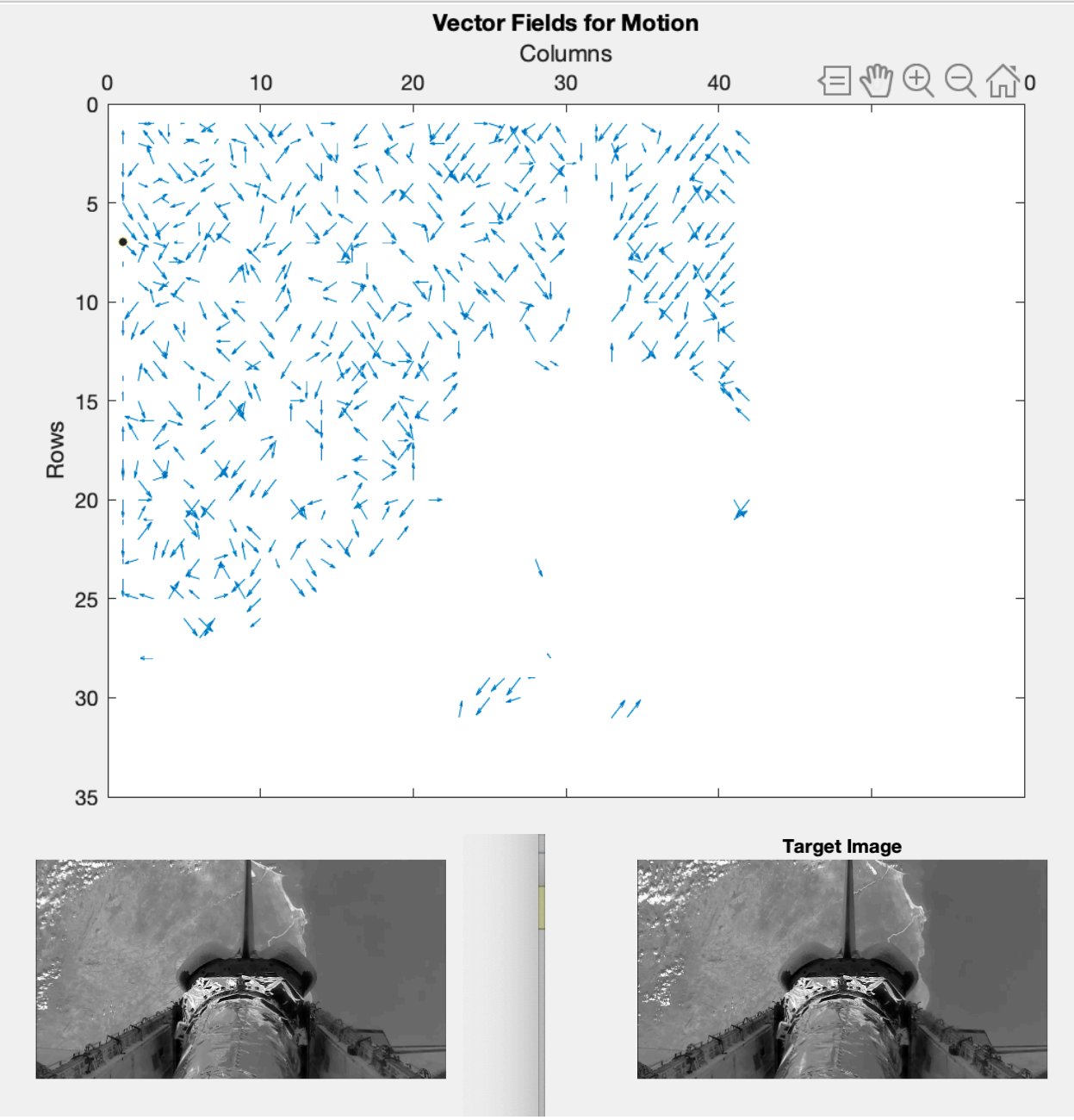

Figure 16: 1st frame and 16th frame for block matching

Observation:

Motion vector overall depends on the frame to frame and specially reference frame to target image.

We can see here that 1st and 2nd frame and even 3rd and 4th frame are perfectly showing us the motion vector but after that there has been more random vector or the estimation of minimal SAD is pointing in right direction.

It is because of Block size and Search space. Since First and 2nd frame are consecutive thus maximum displacemtn of same object in 2nd frame is still in the range of search space. If that object has moved in another location and is not in search space than motion vector points out towards the lowest SAD value. Which is not abnormal.

For **Block size=16** and **Search_Space=8**

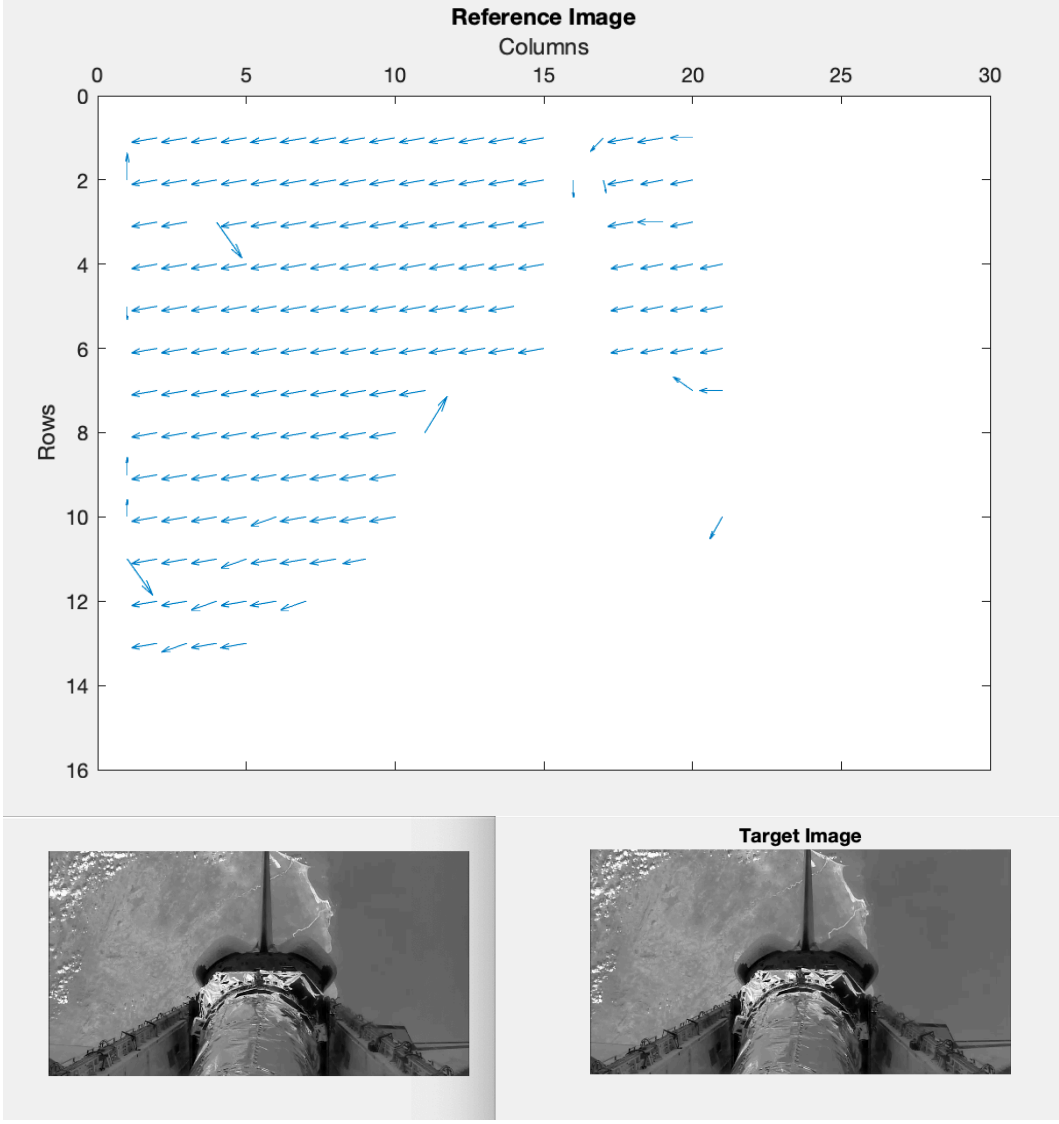

*Figure 17: 1st frame and 6th frame for block matching(**Block size=16** and **Search_Space=8**)*

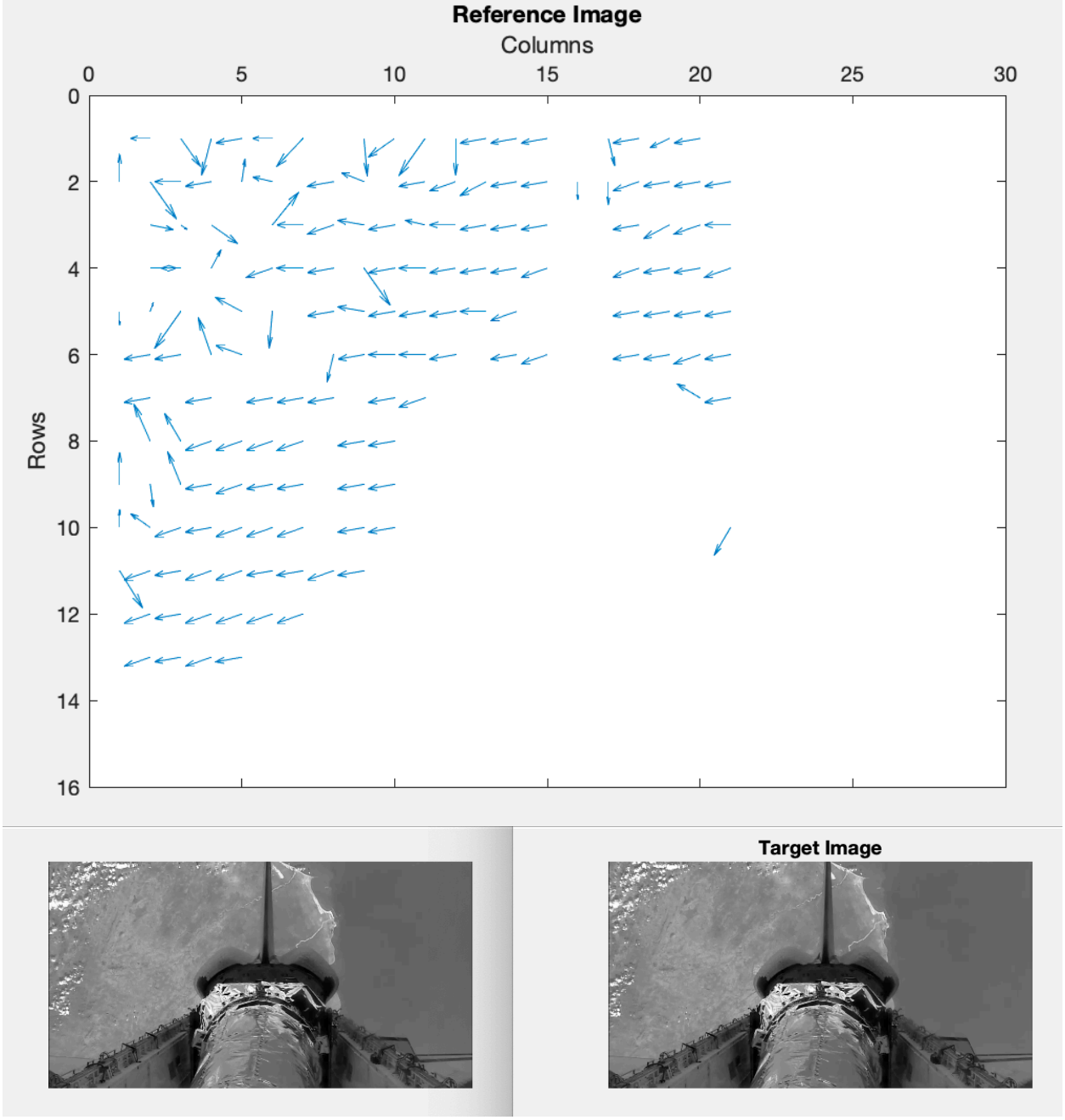

*Figure 18: 1st frame and 7th frame for block matching (**Block size=16** and **Search_Space=8**)*

Conclusion:

Block matching algorithm helps us to get an estimated idea of objects movement from frame to frame. Using this idea an estimated image can be generated thus the frames having same content in various frames can have one reference frame and multiple motion vectors to guide it to generate estimated images using reference frames. Overall, Block matching helps us to get the motion vectors of objects present in one frame to another frame. This technique helps us in compressing the video.

We can see in the project that subsequent frames are more easier to provide us the motion vector rather than lets say 14^{th} frame because of Search space and block size constraints if we have put those.

Throughout the project we have implemented an exhaustive block search algorithm which tries to look out for same object in different position in subsequent frame from the reference frame in search space.

I've finished this Project and have attached relevant snapshots as well as code is attached in the submission with relevant comments.