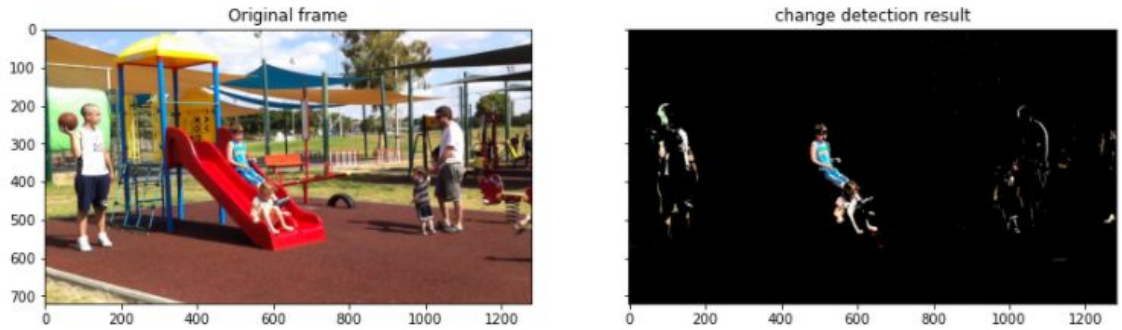
204502926_305498099
Ofir Nesher & Yossi Gavriel

<u>Section A</u>:

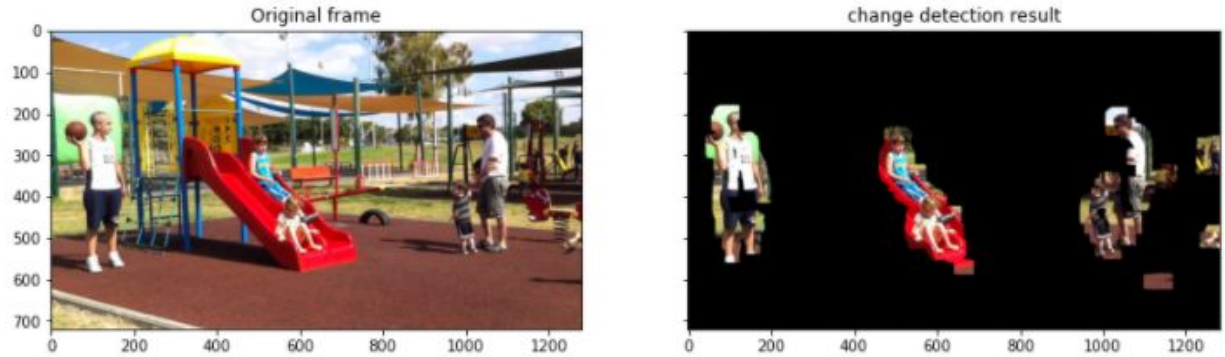
A1- Background reduction image:

```
In [262]: show_imgs(v_origin[65], v_foreground[65], "Original frame", "change detection result")
```



A2 - improving the results of a change detection algorithm - remove noise and fill gaps:

```
show_imgs(v_origin[65], v_foreground[65], "Original frame", "change detection result")
```



<u>Section B</u>:

B5.

Apply and Discuss:

Run the basic_LK_OF, and the affine_LK_OF on one or two videos, one with a static camera and the other with a moving camera. Play with the frames chosen from each video, the algorithm parameters, and the distance between nf1 and nf2.

Answer:

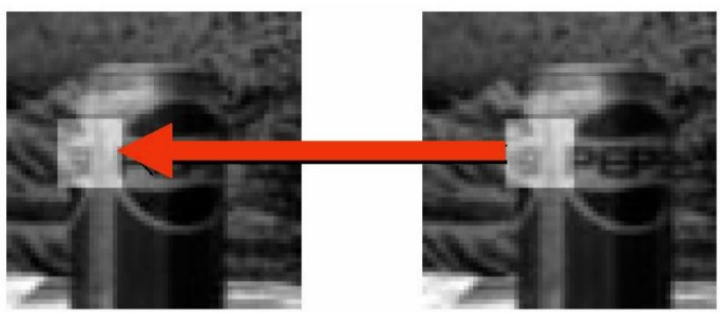
1. **The disparity you compute in HW2 were integers while the OF is not necessarily an integer. Explain why.**

- The **disparity** was calculated by the movement of pixels between two rectified images, with respect to their **indices**, which are naturally denoted by integers.
  We marked how many “steps” each pixel was shifted by.
  With the **OF** matrices, U & V, we mark the **speed** of movement each pixel performs.
  It is also calculated by the number of pixels it moved (as in the disparity), but we divide by time (between frames). This is the formula for calculating speed.

$$I_x(p)\frac{dx}{dt} + I_y(p)\frac{dy}{dt} = -I_t(p)$$

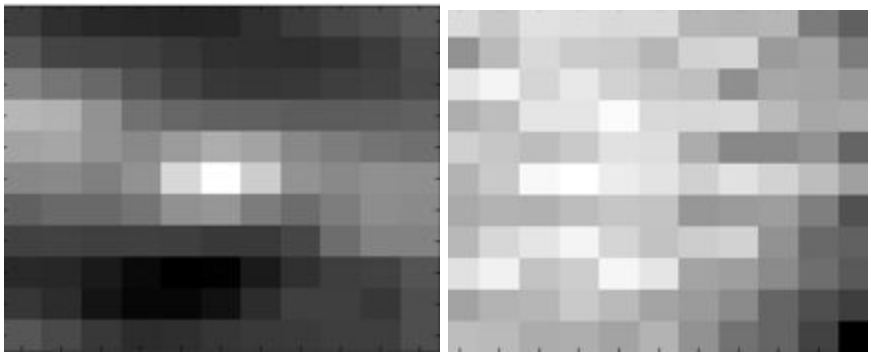
2. **Explain theoretically for which regions the basic_LK_OF is expected to give good results and for which it does not.**

- Theoretically, since the basic_LK_OF is an assumption algorithm, it should work well when its assumptions are true and fail when they are false:
  - <u>Brightness constancy</u>: The intensity of the pixels around the point to track in frame x should be the same as its corresponding pixels in frame x+1



  - <u>Temporal Consistency</u>: The motion between the two frames must be small (1-2 pixels at the most).
  - <u>Local Constraint</u>: Assume constant motion in a small local window.
  - <u>Spatial Coherency</u>: Neighboring pixels belonging to the same surface have similar motion (relies on the window size).

  - The algorithm works well at detecting movement at areas in the picture where a pixel is noticeable among its neighbors in the observed window. If the pixels in the window are homogenous (such as the sky, it may determine incorrect movement, for example in the right picture below)



- Another aspect of the previous scenario is detecting the movement of a point on an edge: The algorithm will fail to determine the correct movement of a one-dimensional spatial structure, such as a bar or edge if it is viewed through a small aperture such that the ends of the stimulus are not visible. This is called the "**Aperture problem**" and we saw examples in class 7.

3. **Demonstrate your answer to (2) by displaying the results of OF overlaid im1 (Quiver overlayed im1), and mark good and bad results.**

- Good results:
    - The dad and the baby are standing in an environment where the pixels are textured (good, not homogenous "Local Constraint"), and their movement is relatively small in the video ("Temporal Consistency"), so the quiver plots are consistent in direction and size and clear.

  "Bad" results:
    - The kids on the slide are moving a lot and fast (bad "Spatial Coherency" and "Temporal Consistency"), at least between these 2 frames (66 & 68) and so the plot is messy, but we can still see that the overall direction makes sense (most arrows are from left-to-right or down-left-to-up-right).
    - Here and there we can see some noise that is displayed by "weird" areas of arrows. This is because the camera is not so steady and also because the brightness in some areas is changing due to moving objects (the teenager twisting towards the sun for example).

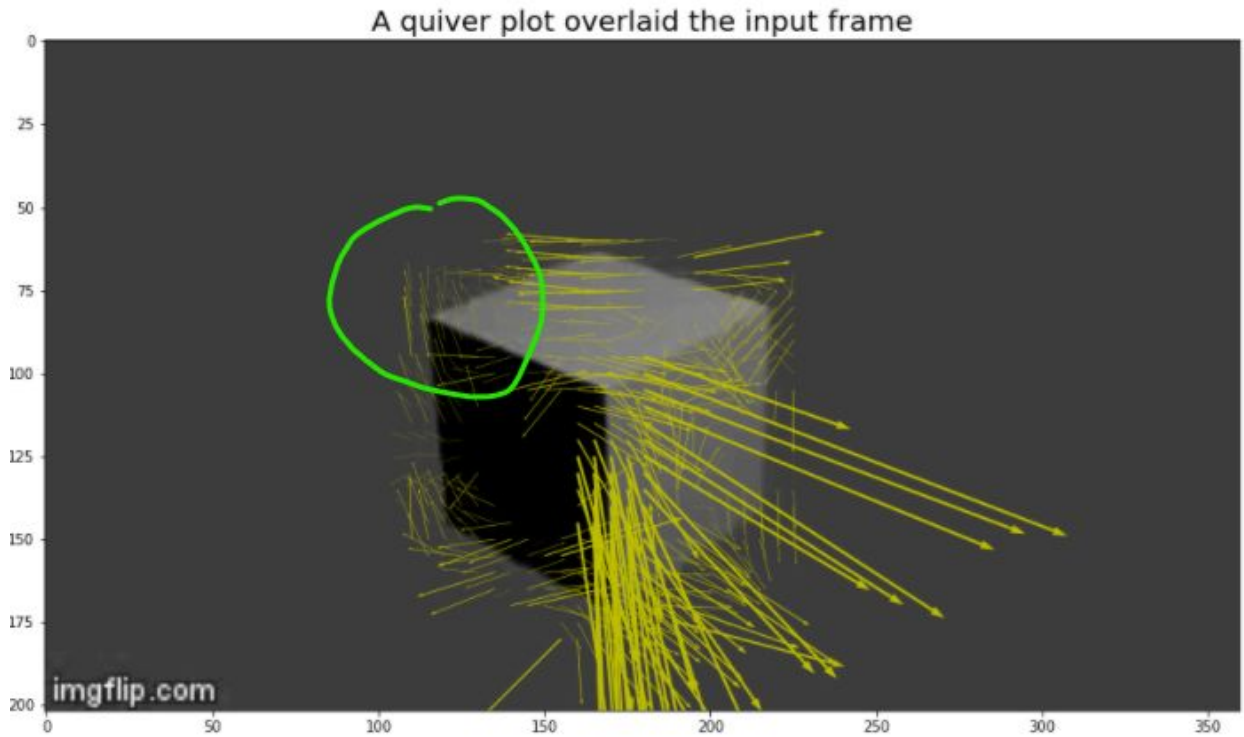
4. **Explain theoretically when the basic_LK_OF is expected to fail while affine_LK_OF works well.**

- The basic_LK_OF is expected to fail while the affine_LK_OF works well on objects whose movements are not just translations but also moving in an “affine” way which means it detects pixels that are moving in different directions in the same patch.

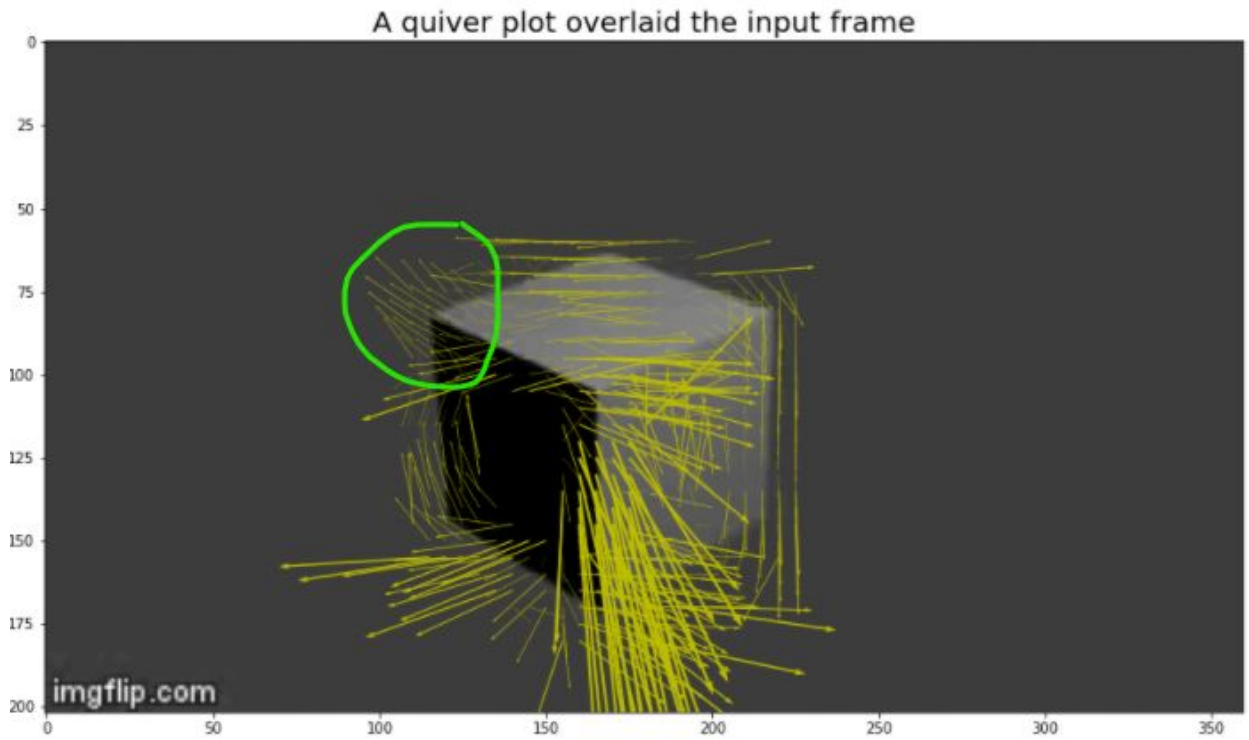
  It makes sense because the affine_LK_OF’s matrix degrees of freedom is 6 while the basic is only 2.

  In basic_LK_OF in the next section, the cube’s image is rotated in the right direction but the algorithm detects the upside left corner as moving up.

5. **Find an example for (4) (at least a region in the scene) and display it.**



While the affine_LK_OF detects it much better:



6. **When two OF vectors have the same magnitude, are they necessarily correspond to 3D points that move at the same speed?**

- Actually **NO**: Consider 2 points - one closer to the camera and a farther one.
  They can both have the same vector magnitude if they both passed a “similar distance” on the picture plane (2 pixels for example), but it is clear that the farther point had to move faster to make this distance, while the closer one moved slower.
  The trick is with their depths (distances on the Z-axis).