Drone Project - winter 2014-2015

Noa Myara 305260895, Shraga Levzion 311597330

Mr. Eli Hayoon, Mr. Aaron Welzer

Prof. Ron Kimmel

In this project, our goals were to track moving objects and recognize missing objects. To achieve these goals we used the following algorithms: Blob detection, optical flow and TLD for Tracking and MSER and dense optical flow for recognition.

- MSER - Maximally Stable Extremal Regions

  We used this algorithm for smooth planes recognition, as can be seen below.

  The MSER algorithm is used to find correspondences between image elements from two images with different viewpoints, by enumerate extremal regions in consequtive frames and find matches by robust similarity.

  The enumeration of extremal regions proceeds as follows: First, pixels are sorted by intensity. After sorting, pixels are placed in the image and the list of connected components and their areas are computed using Union Find algorithm. Finally, intensity levels that are local minima of the rate of change of the erea function are selected as thresholds, producing maximal stable extremal regions.

  The robust similarity is computed as follows: For each M_A^i on region A, k, regions $B_1, \dots, B_k$ from the other image with the corresponding i-th measurement $M_{B_1}^i, \dots, M_{B_k}^i$ nearest to M_A^i are found and a vote is cast suggesting correspondence of A to each of $B_1, \dots, B_k$. Votes are summed from all measurements, and using probability analysis, we pick out the 'good measurements', as it's assumed that 'corrupt measurements' spread their votes randomly.

  Pros:

  - Simple
  - Worked also on different camera views

  Cons:

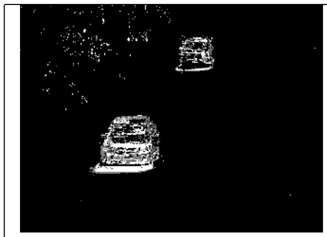
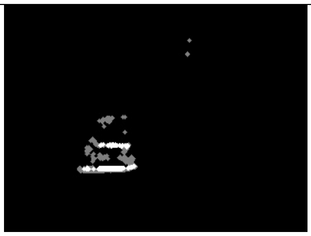
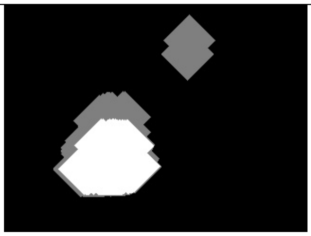
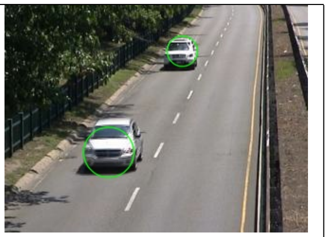
  - There is no global optimal threshold
  - MSER's parameters needs to adjust according to image size, thexture etc.
  - Slow - we used imresize in order to deal with the high computation time.

  Litrature: J. Matas, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions", 2002

- Simple Blob Detector

  Blob detection refers to mathematical methods whose purpose is to detect regions in a digital image that differs in properties, such as brightness or color, compared to areas surrounding those regions. Informally, a blob is a region of a digital image in which some properties are constant or vary within a prescribed range of values.

  In order to differ foreground (moving objects) from background, we subtract a frame from the previous one. We also used morphological operation in order to create a more accurate blob for every moving object.

background subtraction	after erosion	after morph. op.	blob detection
			

We were asked to think of a data structure which will provide a simple way to match blobs to elements that leave for few frames and return afterward. We suggested to solve a bipartite matching problem, between tracked moving elements to their identifiers - blobs, each side of the bipartite graph will augmented with null nodes, so an element can go unmatched for several frames (when dealing with elements which left the frame).

Pros:

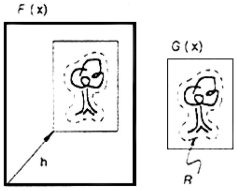
- simple

Cons:

- Static camera only
- Not accurate, depends on the segmentation.
- Close blobs frequently merge
- It is difficult to segment the whole object as one homogeneous blob.

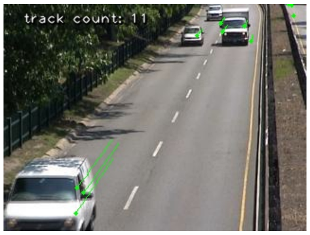
Litrature: Gonzales, “Digital Image Processing”, chapter 9, “Morphological Image Processing”
Forsyth, ”computer vision a modern approach”, chapter 11 “Tracking”

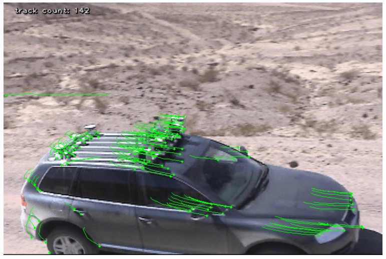
- Optical Flow



To understand the concept of optical flow, consider two frames of a motion sequence produced by a moving camera. For a small movement, we will see relatively few new points, and lose relatively few points, so we can join each point in the first frame to its corresponding point on the second frame with an arrow. The arrows are known as the optical flow.

1. Lucas Kanade Method (improved)





We also tried to segment the “flow” for better and specific recognition of the whole object.
Pros:

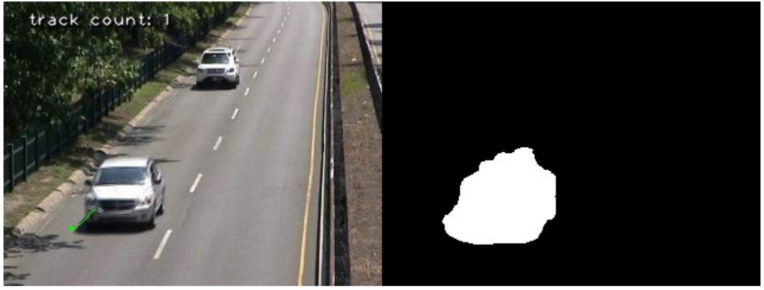
- Accurete
- Tracks also far objects

Cons:

- Every object is identified by few flows but we were asked to identify the whole object by one.

Litrature: Lucas, Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", 1981
Bouget, “Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm”

2. K means - We try to segment moving objects by their gradient orientation, and apply optical flow on the segmented frames in order to identify the moving object with one flow.



```
K-Means Algorithm

Choose k data points to act as cluster centers
Until the cluster centers change very little
    Allocate each data point to cluster whose center is nearest.
    Now ansure that every cluster has at least one data point
    Replace the cluster centers with the mean of the elements in their cluster
end
```

For our purpose, we used 2-Means algorithm to distinguish between moving objects and the background by their flow direction (they are 'moving' in opposite directions).

Pros:

- Accurate segmentation
- Worked also on different camera views

Cons:

- Applying optical flow algorithm on the segmented frame, yields less accurate results than the first method.
- The detection occurs only when the item is close enough.

Litrature: Forsyth, "computer vision a modern approach", chapter 9.3.3 "Segmentation by Clustering - Segmentation Using K-means"

3. Homography + RANSAC - We used homography transform in order to find a more accurate match for optical flow points of previous frames and used RANSAC for reject outliers.

```
RANSAC

Determine:
    $n$ - the smallest number of points required for fitting the model
    $k$ - the number of iteration required
    $t$ - the threshold used to identify a points that fits well
    $d$ - the number of nearby points required to assert a model fits well

Until $k$ iterations have occured
    Draw a sample of $n$ points from the data uniformly and at random
    Fit to that set of $n$ points
    For each data point outside the sample
        Test the distance from the point to the stracture against $t$;
            if the distance is less then $t$, the point is close
    end
    If there are $d$ or more points close to the stracture then there is a good fit.
    Refit the structure using all these points. Add the result to a collection of good fits.
end
Use the best fit from this collection, using the fitting error as criteria.
```

Pros:

- Very accurate
- RANSAC improves the fitting of the moving object

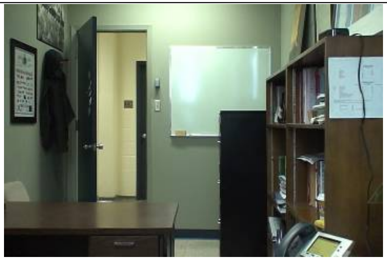
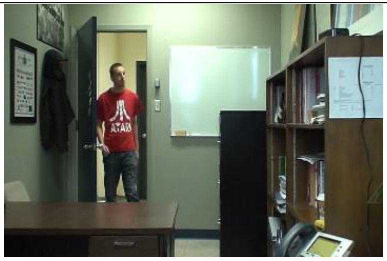
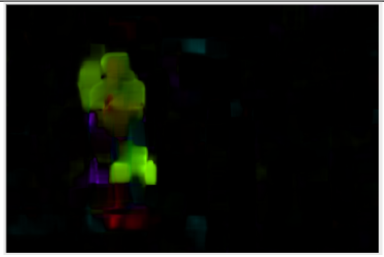
Cons:

- Works under the assumption that the image is approximately plannar.
- The object is still identified by many flows.

Litrature: Forsyth, ”computer vision a modern approach”, chapter 10.4.2 “Robustness - RANSAC: Searching for Good Points”

- Dense Optical Flow

We used the dense optical flow algorithm in order to detect changes in the same scene at different times. The algorithms estimates the motion in every neighborhood of the two frames, using polynomial expansion and an estimation of the movement.

Pros:

- Very accurate, we can see the missing object quite clear.

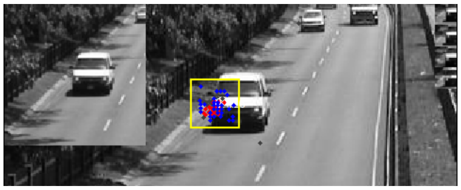
Cons:

- Slow
- Depends on exact camera view, if there will be rotation, translation and zoom differences between the frames it will be much harder to achieve an accurate detection of missing objects.

Litrature: Farneback, “Two Frame Motion Estimatiom Based on Polynomial Expansion”, 2003

- TLD - Tracknig Learning Detection

  We have used TLD algorithm to achieve motion tracking with a moving camera.



  This algorithm takes a bounding box defining the object of interest in a single frame and automatically determins whether the object is visible in every frame that follows.

  TLD consists three main components:

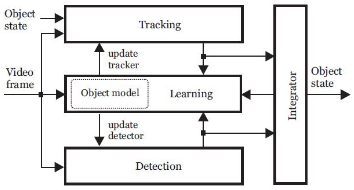


Fig. 8. Detailed block diagram of the TLD framework.

  - Tracker - which estimates the object motion between consequtive frames, under the assumption that the frame-to-frame motion is limited and the object is visible.
  - Detector - which at every frame performs a full scanning of the images to localize all apparences that have been observed. A detector makes two types of errors, false positive and false negative which the next component tries to minimize.
  - P-N Learning - which observes the performance of the tracker and detector, estimates the detector's errors and generates training examples to avoid these errors in the future.

    The P-N Learning is decomposed of two experts, each for one error of the detector.

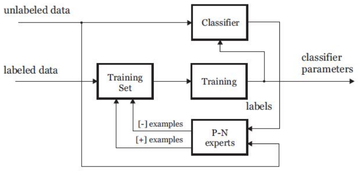


Fig. 3. The block diagram of the P-N learning.

    P-expert identifies only false negative (missed detection). The P-expert assumes that the object moves along a trajectory, remembers the location in the previous frame and estimates the object location in the current frame using a frame-to-frame tracker. If the detector labeled the current location as negative, the P-expert generates a positive example.

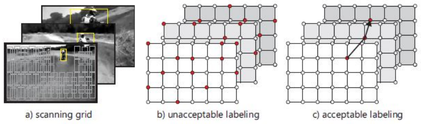


Fig. 6. Illustration of a scanning grid and corresponding volume of labels. Red dots correspond to positive labels.

    N-expert identifies only false positive (false alarms). The N-experts assumes that the object can appear only at a single location. The N-expert analyzes all responses of the detector in the current frame, and the response produced by the tracker, and selects the one that is the most confident.

    The training process proceeds as follows : The initial positive training set is generated by applying "Nearest Neighbor" algorithm over the initial bounding box, finding the most similar objects to it, applying geometric transformations and convolve them with gaussian noise. The initial negative training set is generated by collecting patches from the bounding box surrounding. The training set is then passed to supervised learning which train a classifier. The learning process then proceeds by iterative bootstraping. The classification is analyzed by the P-N experts who estimate examples which have been classified incorrectly. These examples are added to the training set with changed labels.

    Update: A patch is added to the collection only if its label is estimated by Nearest Neighbor classifier is different from the lable given by P-N experts.

For choosing the initial bounding box, we apply optical flow algorithm, as mentioned earlier, detect a motion and return a bounding box around the flow.

Pros:

- Works also on different camera views
- Detection of an object when reappears
- Accurate

Cons:

- It's hard to formulate a criteria for finding a reliable bounding box automatically
- Tracks only a single object at a time

Litrature: Z. Kalal, J. Matas, "Tracknig-Learning-Detection", 2010