University of Burgundy

MsCV year 2

Visual Tracking

Tracking a Moving Object II: Mean-Shift

by

Gopikrishna Erabati

Supervisor: Dr. Desire Sidibe

1. Implementation of Mean Shift Algorithm

The mean shift algorithm is implemented and the implementation details are as given by the flowchart in Fig.1

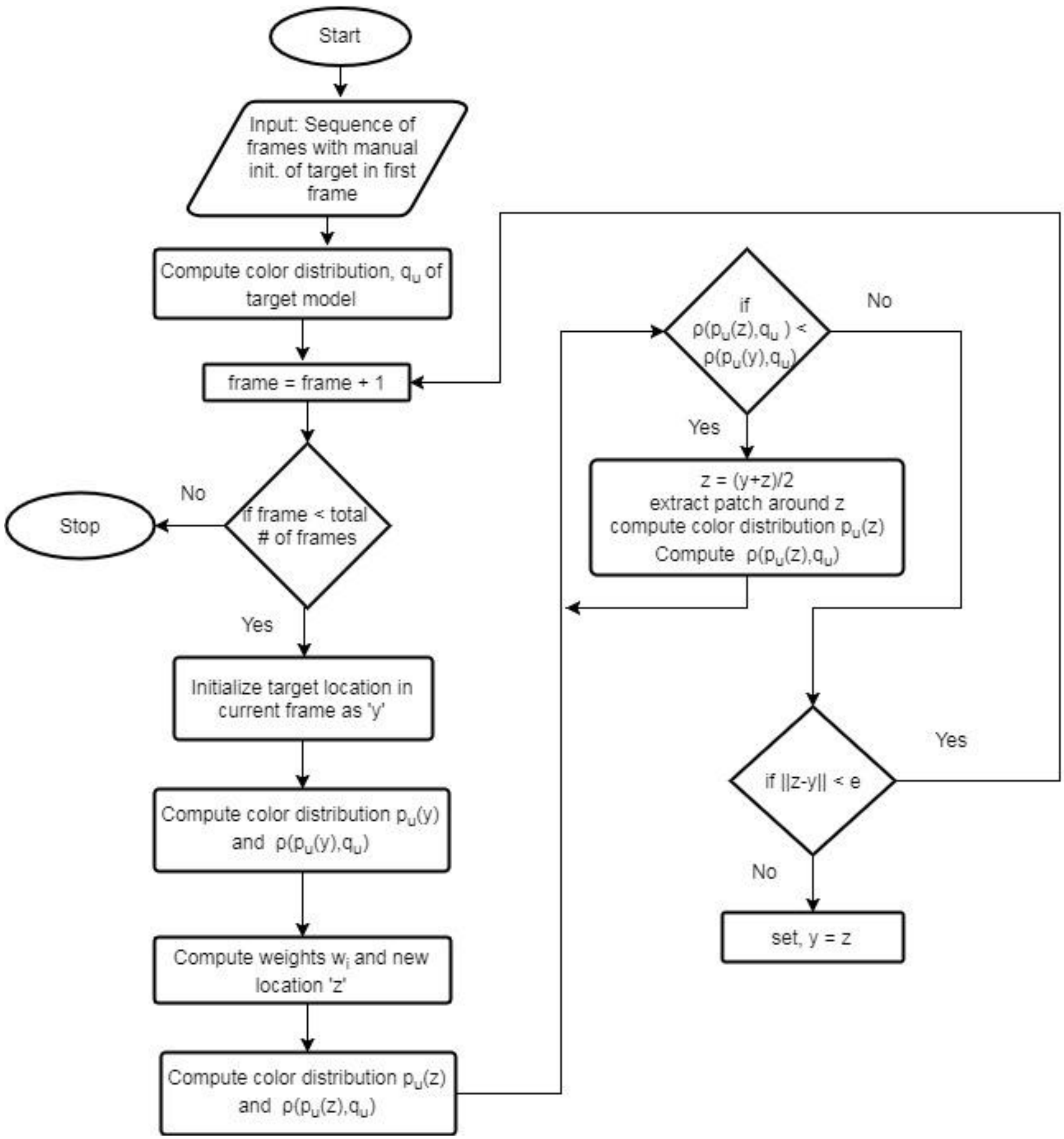
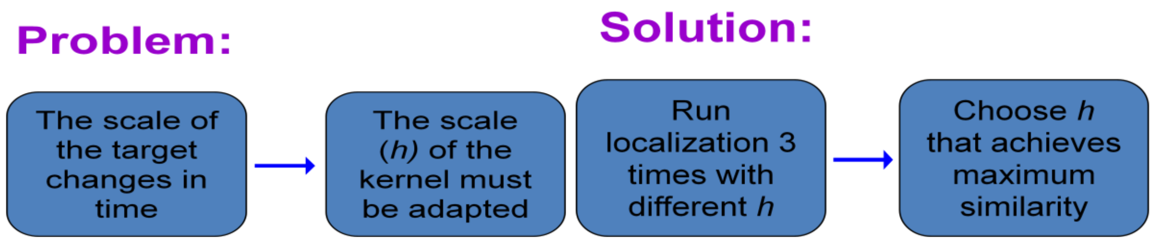


Fig.1 Mean shift algorithm flowchart

1.1 Scale Problem

In our above implementation, we did not consider the scale of the object, as due to the dynamic behavior of scene, the object may change its scale from frame to frame, so we have to consider this effect in our algorithm to change the scale of tracking window.



- The scale adaptation : the distance property (Bhattacharya Coefficient) to be invariant to changes in object scale. For this we can simply **modify the window size (by ±10%)** and let the tracking algorithm to converge again and choose the size which gives less distance (i.e., highest Bhattacharya Coefficient).[1]
  This is **Spatial localization for several scales.**

  This is implemented in Mean_Shift_Tracking_scale.m file.

- We can also perform, **simultaneous localization in scale and space** [2].
  Different scales are appropriate for describing different objects in the image, and we may not know the correct scale/size ahead of time.
  Selection of scale is based on T. Linderberg Scale selection principle:
  "*In the absence of other evidence, assume that a scale level, at which some (possibly non-linear) combination of normalized derivatives assumes a local maximum over scales, can be treated as reflecting a characteristic length of a corresponding structure in the data*".

2. Tracking with Color Images

The procedure described above regarding mean shift algorithm and scale, can be also applied to color images. However, some modifications are necessary to handle the color information. The results will also depend on how the color distribution is modeled.

- One approach is to compute the histogram in **each image channel independently** and then increase the size of the q and p vectors to 3m, where 'm' is the number of bins. This approach simplifies the problem, without using explicitly the color histogram, but profiting somehow the color information. The rest is kept as for the gray scale case.
  This implementation is done in *color_distribution_color.m* file.
  The weights are also calculated for three channels and mean shift vector is calculated for three channels and average of mean-shift vector is taken as a final mean-shift vector.
- Another approach for dealing with color is to assume that the **shape of a non-rigid object is approximated by an ellipsoidal** (rectangle or circular or any other kind of

shape) region in an image [3]. The region can then be selected exactly in the same way that the very initial steps for the initial condition were selected.

Let x_i denote a pixel location, θ the initial location of the center of the object in the image, V the variance, M the number of bins in the histogram and let $b(x_i)$ be a function that assigns the color value for each pixel to its bin. The value of the m^{th} bin is calculated by,

$$o_m = \sum_{i=1}^{N_{v0}} \mathcal{N}(x_i;\theta_0,V_0)\delta[b(x_i)-m]$$

In this equation N represents the Gaussian distribution and δ is the Kronecker delta function. The weighting coefficient calculation is exactly the same as the one used in the first part: the farther the lesser weight, and the nearer the larger weight.

2.1 Application of tracking algorithm

The tracking algorithm is applied on the car sequence and the results are shown below in Fig.2

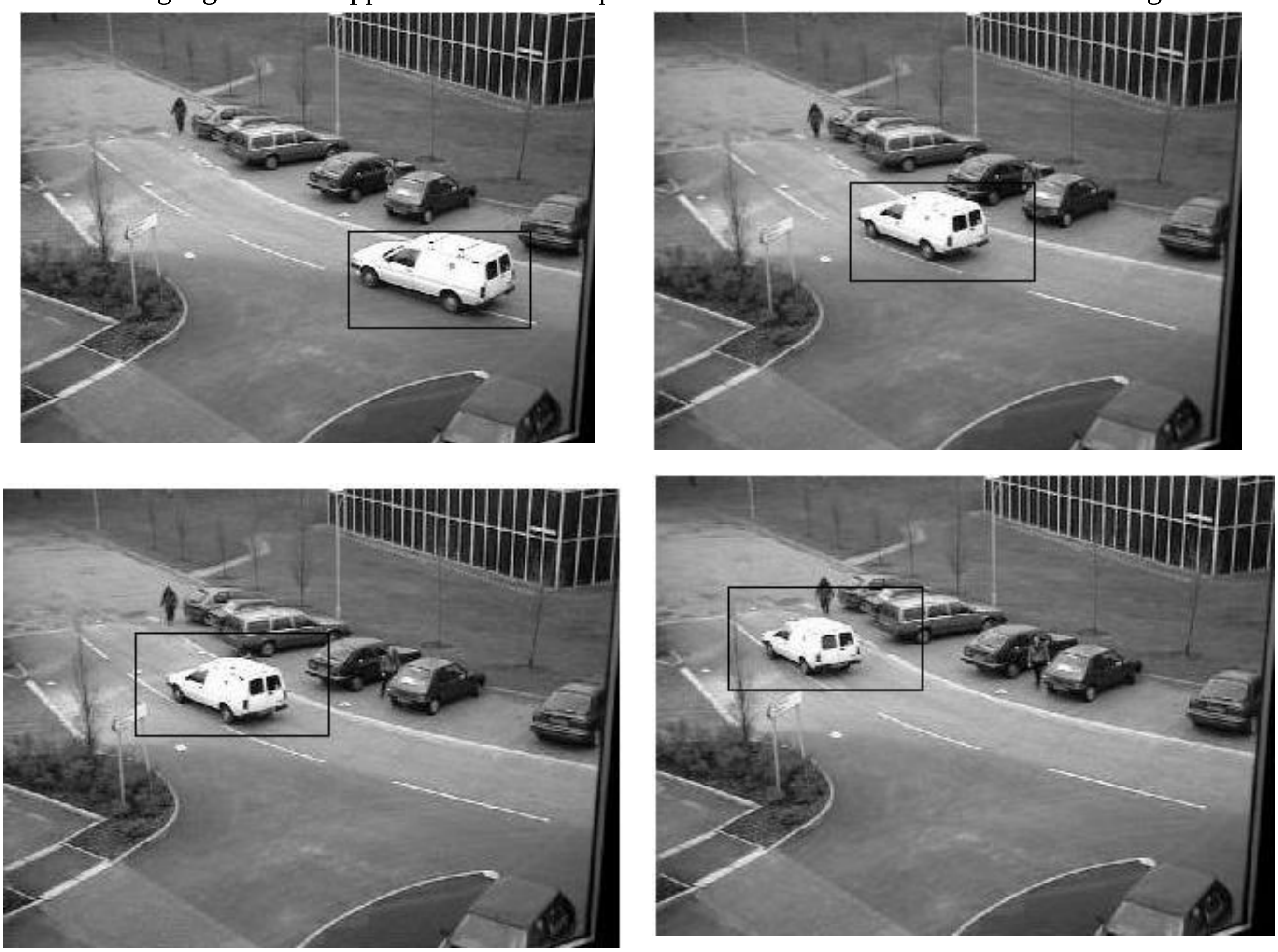


Fig.2 Tracking of car sequence without scaling

The results of the tracking process for different positions of the object is shown in Figure 2. The size of the patch does not change from frame to frame, but the size of the car decreases as the car goes further in the image. Nevertheless, the algorithm is still able to track the car, even though the size of the new car is smaller than the initial one.

Fig.3 Tracking of car sequence with scaling

The results obtained using the scale adaptation are shown in Figure 3. Comparing these results with Figure 2, the improvement in the scale selection can be appreciated, and is particularly noticeable for the last image.

3. Limitation of the algorithm

1. This algorithm is quite **sensitive** to the **initial conditions**, i.e., if the bounding box is initially not located in a good place, it will go to a wrong place after some movement of the object. However, tracking will be still possible, although less accurate

2. In sense of **illumination and contrast** , if the **difference** between the background and the object is small, it is hard to track the object with the color model that was defined. If there is a good amount of difference between background and the object then the algorithm works well due to a lot of redundant information.

3. The **main limitation** of simple mean shift tracking is the case of total **occlusion**. The proposed way to solve this problem in this paper was to use the **Kalman filter** to predict the position of the tracked object when it is totally occluded.

3.1 Handling Occlusions

In order to increase the accuracy and robustness of the tracking, and to deal with total occlusion, we can use the Kalman filter in the body of the algorithm. The idea is to predict the position of the tracked object in the new frame based on the object's previous motion.

Methodology

Let the states of filter be

$$X = [x_c, y_c, \dot{x}_c, \dot{y}_c], where\ \dot{x}_c and\ \dot{y}_c\ represent\ velocity\ in\ x_c\ and\ y_c$$

The discrete time process model will be given by,

$$X_t = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X_{t-1} + N(0, Q)$$

where Δt is the sampling time and N(0, Q) represents a normal distribution for the model with a covariance given by Q.

For initialization, the initial positions are set to the initial centers of the ROI (computed in the very first frame) and the velocities are arbitrarily initialized.

The discrete-time measurement model is given by,

$$Y_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} X_t + N(0, R)$$

where N(0, R) is the zero-centered normal distribution corresponding to the uncertainty in the measurement, with R as the covariance matrix for the measurement.

After the initialization of the values, the basic iteration of the mean-shift algorithm is performed for the second frame, but before assigning the new center of the ROI (the tracked object), the prediction part of the Kalman filter is computed with the very well known Kalman equations. After the prediction step, the value of the distance ρ is checked.

- If $\rho > \rho_{min}$, the measurement is consistent enough and it is used to update the prediction.
- If $\rho < \rho_{min}$, then the new center calculated using mean-shift is not reliable, and no update is performed. This happens, for instance, when there is complete occlusion. In this situation, the new center relies completely on the prediction of the Kalman Filter, since no measurement is taken into account.

4. Conclusion

The mean-shift tracking algorithm works well when there is a lot of information in the frames, a noticeable difference in contrast, and illumination without presence of noise. This algorithm might cause problems when tracking with objects with partial or complete occlusions or dealing with varying size of objects. One way to solve this problem is by incorporating Kalman filter, which has good performance in real-time even with presence of noise.

References

[1] Dorin Comaniciu, Visvanathan Ramesh and Peter Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift", in IEEE CVPR 2000.

[2] Robert Collins, " Mean-Shift Blob Tracking through Scale Space", in IEEE CVPR 2003.

[3] Oscar Efrain Ramos Ponce, Mohammad Ali Mirzaei, Fr´ed´eric Merienne. "Tracking in Presence of Total Occlusion and Size Variation using Mean Shift and Kalman Filter", 2011 IEEE/SICE International Symposium on System Integration, Dec 2011, Kyoto, Japan.