Exercises: Object Tracking

AUVSI Foundation: Computer Vision Training

Cell Tracking – Histogram-Based Tracker

Track a cell in a video sequence using histogram-based tracking highlight it in the video by inserting a rectangle around it.

1. Open the script `cellTracking_start`. Here, `VideoFileReader` (`yeast.avi`) and `DeployableVideoPlayer` objects have already been created. . Fill in code according to the comments in the starter code. The following points help in implementing the rest of the code.

2. Create a `HistogramBasedTracker` object and initialize it. For the initialization you must provide the object location. Interactively select the object region using `imrect`.

3. The tracker needs a 2-D feature map as input. The feature map must allow to distinguish the object from background and other objects. Use the `colorThresholder` app to find an appropriate color space and plane.

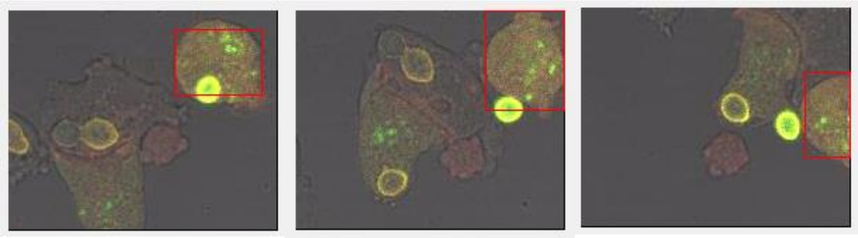
   Hint Use the saturation channel of the HSV color space

4. Implement a loop to process all video frames and highlight the tracked cell using the `insertShape`.

   a. Acquire frame.
   b. Change color space using `rgb2hsv`.
   c. Track using the histogram-based tracker object and `step`.
   d. Visualize using `insertShape` and `DeployableVideoPlayer`.

Solution

```
>> cellTracking
```



Source: **http://www.cellimagelibrary.org/images/7326**

Green Ball Tracking – Kalman Filter

Track a green ball using Kalman Filter and highlight tracked location using a green circle and the detected location using a blue circle..

1. Open the script `greenBallTracking_start`. Here, `VideoFileReader` (`greenBall.avi`) and `DeployableVideoPlayer` objects have already been created. The script uses the `imrect` command to interactively select the initial position of the green ball in the first frame of the video. Fill in code according to the comments in the starter code. The following points help in implementing the rest of the code.

2. Configure the Kalman Filter using the `configureKalmanFilter` function. Start with parameter values similar to the example in the training video.

3. Implement a loop to process all video frames
   a. Acquire current frame.
   b. Track it using `predict`.
   c. Visualize the tracked location with a green circle using the `insertShape`.
   d. Detect ball using the provided utility function `segmentGreenBall`. If found, update the Kalman filter using `correct`. Otherwise, output the original frame.

4. Tune the Kalman Filter parameters to try improve results.

Solution

```
>> greenBallTracking
```