ENPM 673 - Perception for Autonomous Robots

Lucas Kanade Tracker

Submitted towards completion of Project-4

Arjun Gupta
Vishnuu AD
Abhishek Banerjee

19th April 2020

1 Introduction

In this project we have implemented a Lucas-Kanade (LK) template tracker first proposed in [1]. The image alignment algorithm is one of the most studied algorithm for tracking a template across the sequential set of images. The major understanding of this work is derived from [2]. The goal of the algorithm is to align a chosen template image (say $T(x)$) to the input image $I(x)$, where x is the image coordinate vector. The output of the algorithm is the subregion in the image $I(x)$ at t=2 which most closely corresponds to $T(x)$ at t=1.

In order to achieve this task we first define the vector for affine transformation that warps the pixel x in the frame T to the sub pixel location in frame I. We denote this transformation as **W(x;p)** where $p = (p_1, p_2, ..., p_n)^T$ as defined in [3]. The affine warp of a large image patch moving in 3D is given by:

$$W(x;p) = \begin{pmatrix} (1+p_1) & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{1}$$

where the $p_i's$ denote the 6 parameters required to define the affine warp and the vector p can be defined as $p = (p_1, p_2, p_3, p_4, p_5, p_6)^T$.

2 Algorithm

The goal of the problem can be mathematically defined as an optimization problem where the goal is to minimize the sum of squared error between the template image and the warped image I (onto the template image T). The equation below describes the formulation:

$$\sum_x [I(W(x;p)) - T(x)]^2 \tag{2}$$

where $I(W(x;p))$ denotes the warped image. The warping of the image requires the interpolation of the pixel intensities which in our implementation is taken care by the in built function of *OpenCV*. Since the pixel intensities are not the linear function of the coordinates, the above mentioned

optimization problem becomes non linear and the obtained problem can be iteratively solved for small increments in parameters p denoted as Δp and the parameters are updates as:

$$\sum_x [I(W(x;p+\Delta p)) - T(x)]^2 \tag{3}$$

$$p \leftarrow p + \Delta p \tag{4}$$

The above obtained problem can be linearised by using taylor series expansion around Δp and the jacobian of $W(x;p)$ is given by:

$$\frac{\partial W}{\partial p} = \begin{pmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{pmatrix} \tag{5}$$

Since the optimization problem is linear and has a closed from solution, by taking derivative w.r.t Δp we get:

$$\sum_x \left[\nabla I \frac{\partial W}{\partial p}\right]^T [I(W(x;p)) - T(x) + \nabla I \frac{\partial W}{\partial p} \Delta p] \tag{6}$$

where $\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$. In order to solve for Δp we equate this equation to 0 and thus the obtained solution is:

$$\Delta p = H^{-1} \sum_x \left[\nabla I \frac{\partial W}{\partial p}\right]^T [T(x) - I(W(x;p))] \tag{7}$$

where H is the $n \times n$ Hessian matrix:

$$H = \sum_x \left[\nabla I \frac{\partial W}{\partial p}\right]^T \left[\nabla I \frac{\partial W}{\partial p}\right] \tag{8}$$

The above mentioned process is broken down, summarised and presented in steps for implementation in 1

Algorithm 1 Calculate **p**

Require: $error = Max\_INT, \quad threshold = \epsilon, \quad max\_iter = Maximum\_iteration$
  while error $\geq threshold$ **do**
    Warp I with W(x;p) to compute I(W(x;p))
    Compute the error image $T(x) - I(W(x;p))$
    Warp the gradient ∇I with W(x;p)
    Evaluate the jacobian $\frac{\partial W}{\partial p}$ at (x;p)
    Compute the steepest descent images $\nabla I \frac{\partial W}{\partial p}$
    Compute Hessian using equation 8
    Compute Δp by substituting calculated Hessian in equation 7
    Update p using 4
    if iteration $\geq$ max_iter or $||\Delta p|| \leq threshold$ **then**
      break
    else
      continue
    end while

3 Implementation - Image Pipeline

For the given datasets (found here: Usain_bolt, Car and Baby_vs_Dragon) we have unique challenges posed. We first discuss our generic pipeline. Thereafter, we discuss the unique challenges and parameter selection for each of the datasets.

Figure 1: Template Image

Figure 2: Warped Image

3.1 Pipeline

- Choose template(s) for the given dataset by choosing an image which is roughly the average of all images in the dataset. Or one could choose the first image as well.
- Get bounding box
- **If using Histogram equalisation**
  - Histogram equalise the entire image.
- Extract the region(s) inside the bounding box(s) as the template(s).
- Initialise P(affine transform parameters) matrix as

$$P = (0, 0, 0, 0, 0, 0) \tag{9}$$

For each image in the dataset we run the following,

- **If using Histogram equalisation**
  - Histogram equalise the entire image.
- Calculate the gradient $\nabla I = [I_x \ , \ I_y]^T$ of the current frame/image.
- For **500 iterations, and each template** perform the steps mentioned in **Algorithm 1** while using **Huber Loss** if needed. Visualisation of the intermediate steps are shown above.
- We also calculate the L_1 norm of the error between the each of template and warped image. i.e

$$Error = ||T(x) - I(W(x; p))||_1 \tag{10}$$

- Update the global P matrix array(Individual P for each template) for that template which gave the least L1 error.
- The update is also subject to the condition that P has converged in 500 iterations and the L_2 norm of P is below a certain threshold. If these conditions are not satisfied, we simply use the previous P matrix.
- Given the calculated P, we forward warp the corresponding template's bounding box onto the current image frame and then draw a bounding box on the image using $cv2.rectangle()$.
- A moving average filter is used to give a better estimate of the bounding box by limiting the sudden changes in bounding box predictions due to few images.

Sample results are shown below.

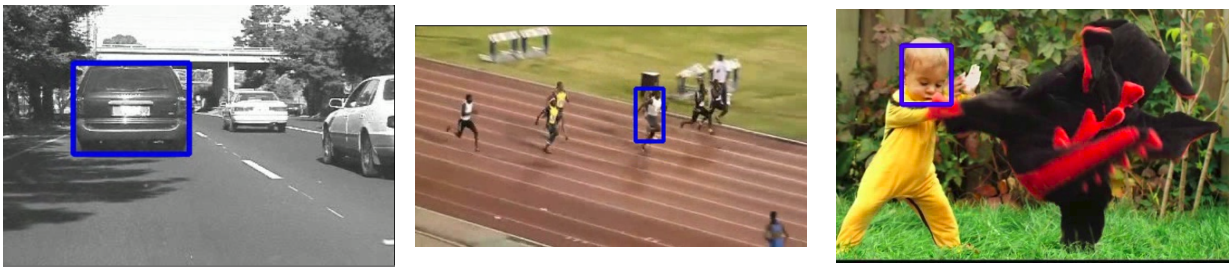

Figure 3: From left to right; (a) Car Tracking, (b) Bolt tracking, (c) Baby Tracking

3.2 Robustness to Illumination

- **Histogram Equalisation:** In order to enhance the contrast we use histogram equalization to evenly spread the intensities and thus enhance global contrast. It is a fairly straight forward method which enhances and uniformly spread the intensities resulting in better contrast.
- **Huber loss** This is simply a smooth l1-loss. Huber loss demonstrates the properties of l1-loss for most of the region of operation except when below a threshold where it exhibits the property l2-loss. We employ huber loss as it is robust to outliers and is differentiable for small values also.

$$L(y, f(x)) = \begin{cases} \frac{1}{2}\left[y - f(x)\right]^2 & \text{for } |y - f(x)| \leq \delta, \\ \delta\left(|y - f(x)| - \delta/2\right) & \text{otherwise.} \end{cases} \tag{11}$$

where we have considered $|y - f(x)|$ as the error between the template and warped image, and the δ is the standard deviation of the calculated error.

- **Sample comparison for Car, Usain Bolt and Baby** We present below the images for comparisons between the output of different pipelines. Please note that we do not show the images of just huber loss without histogram equalization for car and baby because of Hessian Matrix getting singular in several frames. Also, **due to the time limitations and heavy time required by videos to be generated we could not generate the full videos with huber loss and histogram equalization for some cases**.

Figure 4: From left to right (a) Our result, (b) With Histogram Equalization, (c) With Huber Loss only, (d) With Huber Loss and Histogram Equalization

Figure 5: From left to right (a) Our result, (b) With Histogram Equalization, (c) With Huber Loss and Histogram Equalization

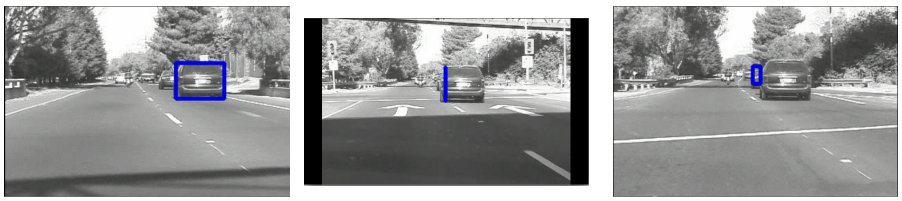

Figure 6: From left to right (a) Our result, (b) With Histogram Equalization, (c) With Huber Loss and Histogram Equalization

3.3 Dataset

- **Usain Bolt**:
  - As seen in the sequence of images, Usain bolt initially has a crouched pose, later on has a full upright pose and finally we see a side view of his Pose towards the end of this run. Due to changing features throughout the video we use 3 templates taken at **Image001**, **Image0072** and **Image0190**.
  - We also tried re-initialisation of P to zeros for every image and found it to be better as the conventional approach of using the previously calculated P. The conventional approach resulted in drift and poor estimation of further bounding boxes when there was a single poor estimate.
- **Car**:
  - Throughout the sequence of images, it was seen that the pose of the vehicle remains almost constant throughout the images. Hence we decide to just have a single template taken at **Image001**.
  - We use the conventional method of using the previously calculated P for the next frame.
- **Baby vs Dragon**:
  - The fighter baby was seen to perform complex moves throughout the video where the his face turns multiple times and it can be broadly classified into 3 poses: Front face, Side face and Back of the head. Hence we take three templates at **Image0001**, **Image0018** and **Image001**.
  - Here we re-initialised P to zeros for every image and found it to be better as the conventional approach of using the previously calculated P. Results are shown in the Results and discussion section.

4 Results Discussion

All the final videos can be found here here.

- **Comparison of result for Bolt**

Figure 7: From left to right (a) Our result, (b) With Histogram Equalization, (c) With Huber Loss only, (d) With Huber Loss and Histogram Equalization

- **Comparison of result for Parameter convergence vs Least error** The figures below show the results when the termination condition is the fastest convergence to the threshold error set by us and the other one is when the best out of the three templates is chosen.

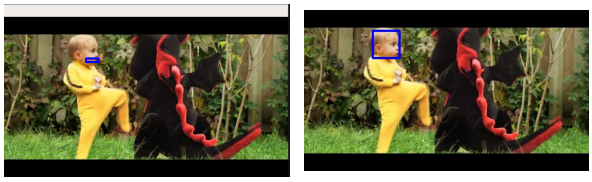

Figure 8: From left to right, comparison of results for different termination conditions (a)Choose P of template with fastest Parameter convergence (b) Choose P of template with Least Photometric Loss

- **Comparison of results for baby Original output vs Hist equalisation vs Huber loss** As discussed above also the images show the difference between the output with different illumination tackling techniques.

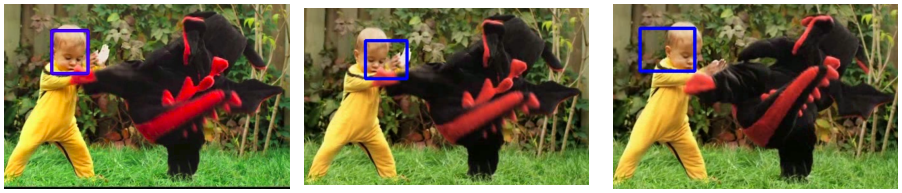

Figure 9: From left to right (a) Our result, (b) With Histogram Equalization, (c) With Huber Loss and Histogram Equalization

- **Comparison of results for baby: With and without Moving average filter and Parameter Re-Initialization** Since in some detections even though the box center is correct the affine parameters are such that the size of the box is either very large or very small with respect to the previous frames. Since the frames are consecutive, we assume that the size of the object do not change much. In order to facilitate this, we take the moving average of past five box sizes along with the newly detected one to predict the new bounding box.

Figure 10: From left to right (a) With no moving average and parameter re initialization, (b) With Moving average and no parameter re initialization, (c) With moving average over 3 frames and parameter re initialisation.

References

[1] Bruce D Lucas, Takeo Kanade, et al. "An iterative image registration technique with an application to stereo vision". In: (1981).

[2] Simon Baker and Iain Matthews. "Lucas-kanade 20 years on: A unifying framework". In: *International journal of computer vision* 56.3 (2004), pp. 221–255.

[3] James R Bergen et al. "Hierarchical model-based motion estimation". In: *European conference on computer vision*. Springer. 1992, pp. 237–252.