A Deeper Dive into Multi-Face Tracking in Unconstrained Videos

Ariel Rakovitsky
Princeton University
arielr@princeton.edu

Charles An
Princeton University
ca9@princeton.edu

Abstract

The problem of multi-object tracking is particularly interesting in the scope of videos containing multiple camera angle switches. We follow the approach of Lin and Hung in "A Prior-Less Method for Multi-Face Tracking in Unconstrained Videos"[4] to develop a prior-less multi-face tracking system for use in unconstrained videos. Results achieved with this system closely approximate the results of Lin and Hung. We then edit two components of the system as put forward by Lin and Hung, namely the tracklet generator and the feature extractor. Doing so allows us to observe experimentally that the quality of the tracklet generator, and necessarily the face or body part detector, used in our system has the largest effect on the performance of the system. We conclude that the tracklet generator and associated object detector of a multiple object tracking system have the greatest impact on performance of that system.

1. Introduction

The purpose of a multiple-object tracking (MOT) system is to accurately track a number of suspects in a video. As opposed to single-object tracker, the system must maintain some identifying information about the subjects it is tracking. A multiple-object face tracking system must be able to accomplish this for faces in a video. A prior-less multiple-object face tracking system must be able to do so without any preliminary information. In this paper, we also assume the video is unconstrained, meaning it may feature multiple camera angle switches or camera movements.

1.1. Applications

The applications of a robust system as described above are plenty. The current primary focuses of development are surveillance applications. However, any system that must maintain location of an individual over time can benefit from a strong multiple face tracker. These include sports-analytics applications, business-analytics applications (e.g. the number of times one customer returns to one part of a store), etc.

Figure 1. A successful tracklet generation.

1.2. Challenges

A robust multi-face tracking system faces several key challenges. The first is the issue of frequent face occlusion. Even when a subject is in the frame, their face is often hidden, making identification difficult. Camera-angle switches, as discussed in this paper, alter the appearance of faces and make re-identification of subjects challenging as well. Finally, robust and descriptive methods of representing faces are not trivial and are an open research question.

1.3. Definitions and Key Terms

Tracklet A tracklet is a series of faces in consecutive frames belonging to the same person.
Track A track is a collection of tracklets that represents a single person. Also known as clusters.
Co-occurence Model For our purposes, this is a model that uses multiple body parts to help continue the tracker during moments when faces are not captured by the camera or not detected by the detector, but the person remains in the video frames.[4]

1.4. Background and Previous Work

Chung-Ching Lin and Ying Hung's paper, *A Prior-Less Method for Multi-Face Tracking in Unconstrained Videos*, represents the current state-of-the-art in multi-object tracking. Their model uses a co-occurrence model of multiple body parts to create face tracklets, recursively link them to form a graph, and extract clusters. These clusters represent

individual identities. Lin and Hung improve on the previous *Tracking Persons-of-interest via Adaptive Discriminative Features*[10] by dismissing the need for video priors, such as the expected number of people. They further improve upon existing works by building tracklet generating directly into the system, rather than assuming correct tracklets are provided. Lin and Hung do not, however, discuss the relative importance of different aspects of their system, nor explore alternate feature extractors and tracklet generators.

1.5. Motivation and Goal

The primary focus of our project is a reimplementation of the algorithm described in Lin and Hung's paper in an attempt to replicate the results of the authors. In so doing, we had a stable build from which we could modify aspects of the system in a controlled, experimental fashion. Lin and Hung propose their own algorithm for tracklet linking and data association. But, they rely on two external tools to do so. The first is a body part detector, based on the deformable parts model, used for identifying where a person is in the video. The second is a CNN for feature extraction to associate faces. We experimented with different body part and face detectors and different feature extractors and noted changes in the performance of the algorithm. Such a study allowed us to experimentally show which elements of a multi-object face tracking system have the most effect on results and to what extent the system is reliant on a strong implementation of these components.

1.6. Dataset

We make use of the music video dataset and annotations published in *Tracking Persons-of-interest via Adaptive Discriminative Features*[10]. The dataset features music videos downloaded from YouTube featuring camera angle switches and a variety of tracking circumstances.

2. Implementation

2.1. Control

The control for this project was the multiple-object face tracking system as designed by Lin and Hung with the exception of Gaussian process outlier correction.

2.1.1 Tracklet Generating

We began by using OpenPose [7], a publicly available, open source body pose estimator library to extract body and face keypoints, with associated confidences, for each video in our dataset. We connected major adjacent keypoints (e.g. a right wrist and a right elbow) to form lines representing limbs of the body. We then used the length of the limb to estimate a width for a bounding box. In this fashion, we

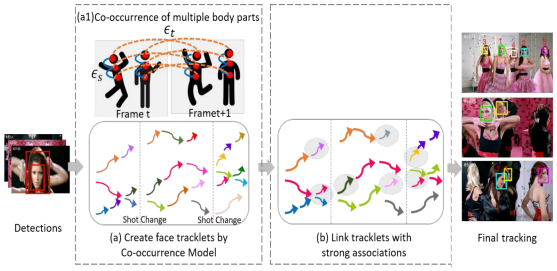

Figure 2. Tracklet Generating and Linking Visualized

generated bounding boxes around limbs and torsos. To generate face bounding boxes, we used multiples of inter-eye distance and eye locations to estimate face position. We were thus able to build a model of each body visible in any given frames out of body part bounding boxes as seen in the top three images of Figure 4. Such a model is an implementation of the deformable parts model (DPM). Body confidences were calculated by averaging keypoint confidences.

Given DPM observations, we compared consecutive frames of the video. If any one body part (right arm, left leg, face, etc.) overlapped by a low threshold with the corresponding body part in the next frame, the corresponding body observations were linked into one tracklet. For any body observations not linked to existing tracklets, new tracklets were initialized if our confidence in the body model exceeded a high threshold.

2.1.2 Tracklet Linking

Once all tracklets were generated, we passed every face in every tracklet through a VGG16 network trained on the VGG-Faces dataset [6]. We extracted the 4096-dimensional vector output of the FC7 layer of this network as a feature descriptor for each face. Every tracklet now was associated with a set of feature descriptors. Because a tracklet should be made up of only one person, these feature descriptors gave us multiple ways of describing a single person's face.

The task of linking tracklets was abstracted as a graph problem. Each tracklet was taken to be an unconnected, lone vertex in a graph containing all tracklets. Any link between tracklets made was expressed as an edge in the graph.

Lin and Hung propose two types of tracklet links:

$\{L_l\}$ and $\{L_c\}$ links. To build $\{L_l\}$ links, we needed to separate the tracklets into two groups based on average face image resolution of each tracklet. We used k-means with two centroids to accomplish this task. Then, within the large group only, we calculated distance between tracklets as follows:

1. Compare every feature descriptor in one tracklet with every feature descriptor in the other tracklet and generate a distance value.
2. Take the minimum of all these distance values to get the computed distance between two tracklets.

Then within the large group, any tracklets whose inter-tracklet distance was less than a threshold '$\{L_l\}$ threshold', were linked.

The other type of link proposed was the $\{L_c\}$ link. First, all the tracklets, both in the large group and small group, were placed into coexisting sets. A coexisting set is defined as a set of tracklets whose frames overlap. A tracklet may be in multiple coexisting sets. Because frame indexes overlap in a coexisting set, we know that tracklets cannot be linked within a coexisting set. The alternative would imply one person is present twice in the same frame, which cannot be. Then, within each coexisting set, a minimum inter-tracklet distance was found. If this value was below an '$\{L_c\}$ threshold', the tracklets in the coexisting set were deemed to be too similar, and that coexisting set was discarded. This prevents false connections. Then, the remaining coexisting sets were pairwise compared, and links were built for tracklets between the two sets whose distance was below the '$\{L_l\}$ threshold'.

Links were expressed as edges in the graph, so a duplicate $\{L_l\}$ and $\{L_c\}$ link between two tracklets did not cause any problems. Clusters in the graph (groups of connected tracklets) were then complete tracks and the program was finished.

2.2. Experimental Modifications

2.2.1 Tracklet Generating

- CNN Face Detector - Faces were detected using dlib's CNN face detector [3]. Tracklets were then built using just the overlap of face bounding boxes generated by dlib.
- HOG Face Detector - Faces were detected using dlib's HOG face detector [3]. Tracklets were then built using just the overlap of face bounding boxes generating by dlib.
- OpenPose Keypoints With Radius - Rather than transforming OpenPose pose estimation data to bounding boxes, we created a circle of a specified radius around each detected keypoint. Tracklets were generated as for the DPM model, but instead of overlap of bounding boxes, overlap of keypoint circles was used.
- MTCNN Face Detector - Faces were detected using a Multi-Task CNN [9]. Tracklets were then built using just the overlap of face bounding boxes generated by the MTCNN. We were unable to rigorously test this approach as it required a large amount of computing resources.

2.2.2 Tracklet Linking

Instead of using the FC7 layer of VGG16 [8] as the feature descriptor, we used the "flatten_1" layer of ResNet50 [1] and the "classifier" layer of SENet50 [2]. We chose these three networks as we were able to find instances of each pre-trained on the same dataset. They each represent state of the art neural networks. The goal with varying these is not to determine which is best for the task at hand, but rather to see if varying the network used for feature extraction makes a significant impact on results at all. Thus, we will not attempt to analyze the structure or properties of each network specifically.

2.3. Dependencies

This implementation relies on the following open source tools, all available via the pip package manager.

- dlib
- mtcnn
- matplotlib
- sklearn
- numpy
- motmetrics
- shapely
- networkx
- keras_vggface
- opencv-python
- tabulate
- tensorflow

3. Evaluation Metrics

We used two metrics from Lin & Hung's paper to quantitatively evaluate the performance of our model: one evaluating our clustering and another evaluating our tracking. To evaluate clustering, we used a Weighted Clustered Purity (WCP) [5] metric to measure how well our model clustered faces based on identities. WCP

Table 1a: Average CLEAR MOT Metrics. Varying Face Detector

Model	Recall ↑	Precision ↑	F1 ↑	FAF ↓	MOTA↑	MOTP ↑
CNN & ResNet	42.3	49.4	13.5	0.433	9.3	50.5
HOG & ResNet	41.3	46.0	16.3	0.472	10.5	**52.5**
DPM & ResNet	**78.1**	**87.4**	**17.2**	**0.283**	**62.4**	42.1
Lin & Hung	81.7	90.2	85.3	0.27	69.2	86.0

Table 2a: WCP Scores. Varying Face Detector

Model	Westlife	Pussycat Dolls	Hello Bubble	Girls Aloud
CNN & ResNet	0.97	0.96	0.69	0.87
HOG & ResNet	0.98	0.96	0.82	0.89
DPM & ResNet	0.895	0.868	0.982	0.904
Lin & Hung	0.86	0.79	0.70	0.92

is given as $WCP = \frac{1}{N}\sum_{c \in C} n_c \cdot p_c$ where N is the total number of faces detected in the video, n_c is the number of faces in the cluster c∈C, C is the total number of clusters, and p_c is the ratio of the largest number of faces from the same person to n_c. Higher WCP scores indicate better clustering. To calculate the purity of a cluster, p_c, we needed to either annotate and give identities to all face detections by hand or automate the labelling process. Due to resource and time constraints, we opted for the latter. Our automated annotation process involved iterating over every frame and calculating the intersection over union distance between the predicted detections and the ground truth bounding boxes in the frame and if a distance was below a threshold, we labelled the predicted detection with the identity of the "closest" ground truth box.

To evaluate tracking, we used some of the most widely accepted evaluation metrics, the CLEAR MOT[10]. The specific metrics used were Recall, Precision, F1, FAF, MOTA, and MOTP. We chose to omit certain metrics from Lin and Hung's paper because those metrics depend on the number of frames used and due to computational restraints that will be discussed later, our volume of data was significantly less than that of the paper's. Recall is the ratio of correct detections to total number of ground truth boxes. Precision is the ratio of correct detections to the sum of correct detections and false positives. The F1 score is the ratio of correct detections to the average number of ground truth and computed detections. FAF is the average number of false alarms per frame. MOTA, or Multiple Object Tracking Accuracy, is a measurement that combines three error sources: false positives, missed targets, and identity switches. MOTP, or Multiple Object Tracking Precision, measures the misalignment between the annotated and the predicted bounding boxes. Good performance relates to high Recall, Precision, F1, MOTA, and MOTP scores and low FAF scores.

4. Results and Analysis

4.1. Replication of Lin & Hung

Our results for the framework using a deformable parts model approximately replicated that of Lin & Hung's. Table 1a shows that recall, precision, and FAF were within 3 or 4 points of Lin & Hung's results. While our MOTA score may seem a bit off from Lin & Hung's score, Table 1b shows that if we use a different feature extractor, we can get results very close to Lin & Hung's. Furthermore, Table 3 shows that some videos, such as Darling and Hello Bubble, showed better or closer results to Lin & Hung's, showing how much of an impact the properties of different videos can have on the performance of the system. In Table 1a and 1b, our F1 and MOTP scores were significantly worse than Lin & Hung's because of how limited our ground truth detections were. Recall that F1 is the ratio of correct detections to the average number of ground truth and computed detections and MOTP measures the miaslignment between the annotated and predicted bounding boxes. Lin & Hung offered ground truth bounding boxes for the videos used, however these ground truth boxes were only for the main singers in each video. In reality, the videos contained many background members, such as the audience, that our model detected and because these predicted bounding boxes don't have corresponding ground truth boxes, our F1 and MOTP scores are going to be worse than Lin & Hung's.

4.2. A Note On OpenPose Keypoints With Radius

We initally used OpenPose Keypoints With Radius, as described in the implementation section, in our project. However, the generated tracklets performed extremely poorly in our initial testing, so we decided not to continue with extensive testing of the method. As opposed to bounding boxes, keypoint circles occupy a circular in space, rather than a rectangular region. Thus, high percentage overlaps with other, unrelated keypoints are more likely than for bounding boxes. Because we build tracklets by using the maximum overlap of any keypoint, erroneous tracklets were generated very frequently with this method. We can conclude that a DPM model with bounding boxes is significantly more effective than a keypoints-overlap model.

4.3. Varying Face Detectors

4.3.1 Tracking

The results from the CNN and HOG face detectors, as shown in Table 1a, are significantly worse than those of the DPM model and Lin & Hung's framework. This is because the detections from the CNN and HOG face detectors resulted in much shorter tracklets because they couldn't handle face occlusion, unlike the DPM model. In the videos, faces would temporarily disappear due to the person turn-

ing their head too far, a hand covering their face, or other forms of occlusion. In these instances, the CNN and HOG face detectors would not detect a face in that frame and due to how our tracklets are built, the tracklet would be terminated. On the other hand, the DPM framework would still be able to detect the person because although the face might be occluded, other body parts may still be visible and thus the tracklet won't be terminated. The longer the tracklet is, the more facial variation there is for the feature extractor to capture, thus resulting in better linking performance. On top of all that, the HOG model was unable to reliably detect faces when only a side of a face was visible. This resulted in shorter tracklets because tracklets would be terminated when a person turned their head.

4.3.2 Clustering

The WCP scores for the CNN and HOG face detectors, as shown in Table 2a, are, on average, significantly higher than Lin and Hung's score despite the overall poor performance by the CNN and HOG face detectors. This is because of a limitation in the way we labeled our detections. We used an intersection over union distance measurement to calculate the distance between a predicted bounding box and the ground truth box in the frame, and if they were close enough, we labeled the detection with the identification of the corresponding ground truth. If the distance between the detection and ground truth box was above a threshold, then no identification was given to the detection. Our model tracks the people in the audience or background of the music videos and our labeling algorithm tries to give every detection an identity, so some faces in the background were given the identity of the main singers. This can raise WCP scores because if these mislabeled faces are in the clusters of the main singers, then the cluster would be given a higher purity score because there would appear to be more faces from the same person.

Figure 3. Examples of successful re-identification after occlusion and after camera angle switches.

Furthermore, because the ground truth only had bounding boxes for a few people in the videos, the WCP score was only calculated on a handful of clusters, namely the clusters tracking the main singers. Clusters that were tracking the people in the background will have faces that don't have identities because their intersection over union distance was too high, and thus these clusters won't contribute to the WCP score; without identifications and labels, there is no way to tell how many faces in a cluster are from the same person. Therefore, the high WCP score doesn't necessarily represent the model's results as a whole. If we were to annotate every detection by hand and gave each face a proper identity, we expect our WCP scores to be more accurate and reflective of our model's performance.

Table 1b: Average CLEAR MOT Metrics. Varying Feature Extractor

Model	Recall ↑	Precision ↑	F1 ↑	FAF ↓	MOTA ↑	MOTP ↑
DPM & ResNet	**78.1**	87.4	17.2	**0.283**	62.4	42.1
DPM & VGG	76.6	**88.9**	15.9	0.311	**72.3**	**47.1**
DPM & SENet	73.9	82.4	**19.2**	0.404	54.0	42.1
Lin & Hung	81.7	90.2	85.3	0.27	69.2	86.0

Table 2b: WCP Scores. Varying Feature Extractor

Model	Westlife	Pussycat Dolls	Hello Bubble	Girls Aloud
DPM & ResNet	0.895	0.868	**0.982**	**0.904**
DPM & VGG	**0.933**	**0.876**	0.794	0.890
DPM & SENet	0.875	0.847	0.807	0.829
Lin & Hung	0.86	0.79	0.70	0.92

Table 3: DPM CLEAR MOT Metrics

Video	Recall ↑	Precision ↑	F1 ↑	FAF ↓	MOTA ↑	MOTP ↑
Apink	60.8	94.8	17.6	0.0589	55.3	33.0
Bruno Mars	72.7	87.9	12.8	0.362	58.9	43.0
Darling	92.7	91.2	11.3	0.2144	79.3	43.4
Girls Aloud	87.9	96.3	13.5	0.3224	57.0	46.4
Hello Bubble	71.2	96.9	16.6	0.0382	66.2	37.3
Pussycat Dolls	71.0	85.4	21.1	0.3224	57.0	46.4
Westlife	81.5	70.1	19.1	0.7302	45.6	43.7

Figure 4. Top images visualize DPM detections. Bottom images demonstrate effectiveness of DPM in creating longer tracklets.

4.4. Varying Feature Extractor

Table 1b and 2b demonstrate that several different neural network feature extractors produce slightly different results. As stated earlier, our goal is not to explain these disparities, but rather to show that changing the neural network used has far less of an effect than changing the face or body part detector used to generate tacklets. In general, there wasn't much variation in the scores between feature extractors; most of the scores were relatively similar. However, as seen previously, varying our face or body part detector resulted in significantly different results. We can then conclude that the most influential step in our framework is in fact the tracklet generating step.

4.5. Qualitative Results

From Figure 3, we can see that our model does a good job in re-identifying faces and linking tracklets. In the top six images of Figure 3, despite the singer's hand occluding her face and the camera changing to a different shot and back, which terminated tracklets, our framework was still able to successfully link these tracklets and correctly identify the singer. The bottom six images of Figure 3 also demonstrate this. Figure 4 shows how much better the DPM framework performs compared to the CNN and HOG framework. In the second and third frame on the bottom, the singer's hand occludes her face and for the CNN and HOG face detector, this would have terminated the tracklet. However, the DPM was able to use other parts of the singer's body to continue the tracklet. Overall, our framework was able to generate invariant face identities and reliably track them across different shots in the unconstrained videos.

5. Weaknesses of Model

The top three images in Figure 5 show a weakness of our model. Those three images are consecutive frames of the same video and the faces in those frames are in roughly the same position. Due to how we generate tracklets, those three faces are placed into the same tracklet and this is not ideal because those faces belong to three different people. Because we have multiple people in the same tracklet, the feature extractor extracts from three distinct people which makes linking frames difficult. Situations like these are very difficult for our model to handle.

The performance of our model depends heavily on the quality and length of the tracklets we generate. Videos with very frequent camera shot changes or videos with objects that temporarily occlude the entirety of a person, such as in the bottom three images of Figure 5, lead to short tracklets that result in low quality feature extraction and thus poor linking. Videos with these properties are also very difficult for our model to handle.

The quality of the input video also plays a big role in the performance of our model. Original video files must be converted to a sequence of images representing frames, however, despite using software that advertised lossless conversion, we noticed that the frames still had noticeably less quality than the original video. Lower quality images make it difficult to reliably detect faces in frames and this results in lower quality tracklets and linking. Therefore, poor video quality leads to poor performance; high quality videos are highly recommended.

6. Conclusion

Using a DPM, like Lin & Hung used, we were able to approximately replicate the results of the paper. We conclude that the deformable parts model has strong advantages over using convolutional neural network and histogram of oriented gradients face detectors in building tracklets based on overlap algorithm. The deformable parts model creates longer tracklets and this proves to be very influential in the results of the model because the longer the tracklets are, the greater the number of possible facial variations of each person can be captured, thus improving the linking. We noticed that changing the feature extractors displayed less variance than changing the face or body part detectors, demonstrating that the most important part of the model is the tracklet generation. Furthermore, results had a strong dependence on the properties of the video that the system was run on.

7. Painpoints and Bottlenecks

A major pain point and limitation that hindered our progress was our access to computational resources. Due

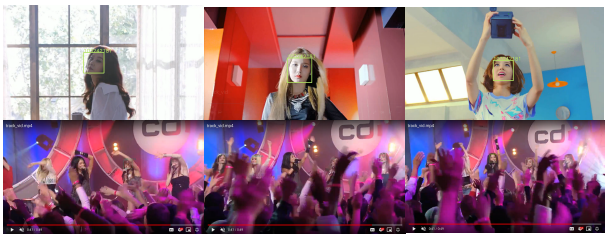

Figure 5. Examples demonstrating challenging video properties.

to budgetary constraints, we were limited to using free resources such as Google Colab. Although Google Colab appears to have sufficient hardware, such as a powerful GPU and CPU, if a task is being run for an extended period of time, Colab will throttle the performance of the GPU, making it extremely slow and ineffective. Because of our lack of access to a strong GPU, we were limited in the scope of our dataset and were only able to use a relatively small fraction of the video frames to run our framework on. Our lack of access to hardware also placed constraints on the face detectors and feature extractors that we could use; for example, we weren't able to use more powerful face detectors, like MTCNN, because we didn't have the computational resources to run it on a significant number of frames. Instead, we opted for lighter face detectors that did not perform as well. Strong face detection and feature extraction is instrumental in the performance of our framework.

8. Acknowledgements

We would like to thank Dr. Olga Russakovsky and Felix Yu for advising us throughout our project.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[2] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. 2018.

[3] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[4] C. Lin and Y. Hung. A prior-less method for multi-face tracking in unconstrained videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 538–547, June 2018.

[5] A. Milan, K. Schindler, and S. Roth. Challenges of ground truth evaluation of multi-target tracking. In *CVPR Workshops*, pages 735–742. IEEE Computer Society, 2013.

[6] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In M. W. J. Xianghua Xie and G. K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 41.1–41.12. BMVA Press, September 2015.

[7] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[8] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[9] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. *CoRR*, abs/1604.02878, 2016.

[10] S. Zhang, Y. Gong, J.-B. Huang, J. Lim, J. Wang, N. Ahuja, and M.-H. Yang. Tracking persons-of-interest via adaptive discriminative features. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 415–433, Cham, 2016. Springer International Publishing.

9. Project Code

The code for this project is online and can be found at: github.com/chuckan13/multi-face-tracking-project