Facial features Detection and Reenactment Onto Computer Generated Imagery

Abhishek Pande & Arpit Gupta

Department Of Electronics And Communication Engineering , IIT Roorkee

Motivation and problem statement

- We aim to extract real time facial features, emotions, wrinkles, head pose estimation and transfer and animate them upon Computer Generated Imagery (CGI).
- A significant percentage of this CGI usage is seen in animating the faces of the CGI models. Be it movies, cartoons, television, or video games, a considerable amount of time and money are invested in providing human like feel and realism to the animated faces.
- Thus the above-stated problems find its applications in various diverse fields.
- This poster presents a novel method of facial feature detection and the reenactment of a monocular video target sequence in real-time.

Methodology

1. **Facial feature detection:**
   - Using dlib to detect facial feature in the incoming target video stream (68 point shape predictor data set).
   - The 68 points obtained are part of the iBUG 300-W data set which was used to train the dlib facial landmark predictor.
   - The predictor provides us with the method to obtain a bounding box of the face (i.e. the (x, y-coordinates of the face in the video stream frames. Given the face region the algorithm begins to detect the key facial structures in the face region.

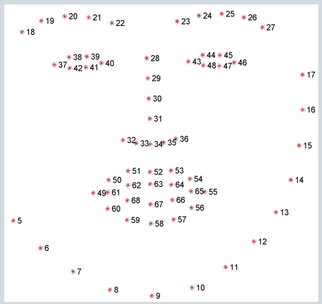

Figure 1. Landmarks

2. **Head Pose Estimation:**
   - Movement of head is an important feature, the position of our head affects the important facial feature/landmarks and how to capture them.
   - Head pose estimation is a Perspective-n-Point problem or PNP problem.The target of the problem is to detect the orientation of the face with respect to camera when we have the n-3D points of the face and their corresponding 2D projection.
   - solvePnP function provided by OpenCV is used by passing flag named Direct Linear Transform (DLT).

Results & Discussion

- The end result is a shape predictor that functions in real time.
- We run the algorithm on the incoming video frames in real time and obtain the location of the 68 landmark points location in the form of 2D coordinates.
- The change in landmark coordinates is detected and the change is passed to smoothening or normalization function which calculates the intensity of the animation to be triggered.
- Changing the number of points from 68 to other value affect the detailing in capturing v/s the time taken by the predictor.
- The accuracy in facial feature detection depends upon the distance threshold.
- The predictor used identifies points on the face such as the corners of mouth, along the eyebrows, on eyes and so forth.
- The face detector and predictor algorithm provided by the dlib library is created using techniques such as HoG filters, linear classifiers, image pyramid and sliding window detection schemes.
- HOG uses direction of oriented gradients as features to detect objects since their magnitude is large around edges and corners and pack more information compared to flat surfaces.
- Support Vector Machine is a supervised machine learning algorithm which is mostly used for classification problems by maximizing the margin by learning a suitable decision boundary.
- Image pyramid is used when we work on images at different resolution generally for the task of searching objects in the image.
- Head pose estimation requires the location of the following six facial landmarks:
  - Tip of the nose
  - Chin
  - Left corner of left eye
  - Right corner of right eye
  - Left corner of mouth
  - Right corner of mouth
- Intrinsic parameters of the camera:
  - Optical Center: By the center of the image.
  - Focal length: Width of the image in pixels

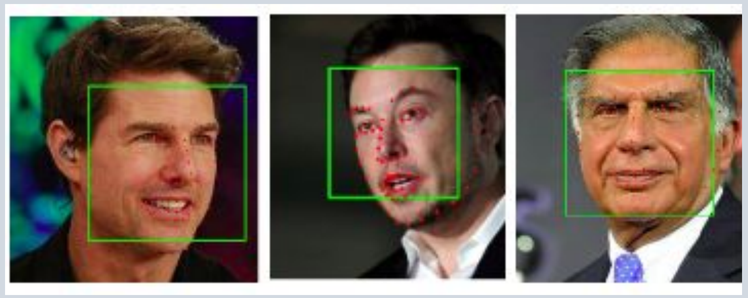

Figure 2. Detection of 68 facial landmarks

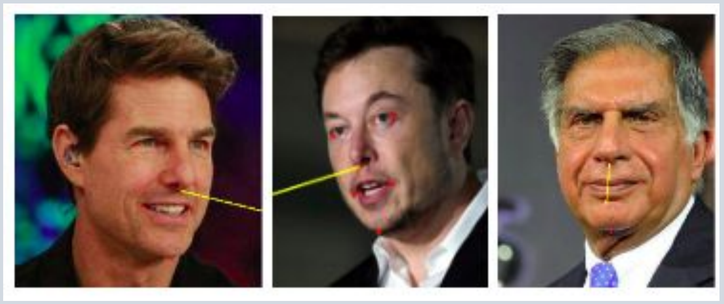

Figure 3. Head pose estimation

Conclusion and Future work

1. The presently developed approaches perform decently in terms of the computational costs as it uses efficient and robust algorithms.
2. The dlib shape predictor does not provide us with wrinkle level details nor does it help in estimation of head pose.
3. The PNP approach used can be further optimised by not computing head pose for every frame. This will lead us to trade off between execution speed and accuracy.
4. All the facial features such as 68 facial landmarks, head pose estimation, wrinkle level details will be reenacted on the Computer Generated Imagery using Blender functionality.
5. We can also include eye gaze detection and reenactment. The algorithm works correctly for fixed movements of the head. The accuracy reduces when head pose changes abruptly. Therefore to detect head pose changes beyond speed and angles, we can use some pretrained data to detect the head pose.

References

1. B. A. Efraty, M. Papadakis, A. Profitt, S. Shah and I. A. Kakadiaris, "Facial component-landmark detection," 2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG), Santa Barbara, CA, 2011, pp. 278-285, doi: 10.1109/FG.2011.5771411.
2. Q. Wang and L. Shi, "Pose estimation based on PnP algorithm for the racket of table tennis robot," 2013 25th Chinese Control and Decision Conference (CCDC), Guiyang, 2013, pp. 2642-2647, doi: 10.1109/CCDC.2013.6561387.
3. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on CVPR'05, San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.
4. Kazemi, V. and J. Sullivan. “One millisecond face alignment with an ensemble of regression trees.” 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014): 1867-1874.
5. T. Vatahska, M. Bennewitz and S. Behnke, "Feature-based head pose estimation from images," 2007 7th IEEE-RAS International Conference on Humanoid Robots, Pittsburgh, PA, 2007, pp. 330-335, doi: 10.1109/ICHR.2007.4813889.