Automatic Driver Face State Estimation in Challenging Naturalistic Driving Videos

Brandon M. Smith
Research Associate, Computer Sciences Dept., University of Wisconsin-Madison
1347 Computer Sciences, 1210 W Dayton St, Madison, WI. 53706
bmsmith@cs.wisc.edu Ph: 608 262 5105

Xuan Wang
Graduate Student, Electrical & Computer Engr. Dept., University of Wisconsin-Madison
4618 Engineering Hall, 1415 Engineering Dr, Madison, WI. 53706
xwang554@wisc.edu Ph: 608 886 3605 Fax: 608 262 1267

Yu Hen Hu
Professor, Electrical & Computer Engr. Dept., University of Wisconsin-Madison
3625 Engineering Hall, 1415 Engineering Dr, Madison, WI. 53706
yhhu@wisc.edu Ph: 608 262 6724 Fax: 608 262 1267

Charles R. Dyer
Professor, Computer Sciences Dept., University of Wisconsin-Madison
6379 Computer Sciences, 1210 W Dayton St, Madison, WI. 53706
dyer@cs.wisc.edu Ph: 608 262 1965

Madhav V. Chitturi ***** (**Corresponding author)**
Associate Researcher, Civil & Environmental Engr. Dept., University of Wisconsin-Madison
1241 Engineering Hall, 1415 Engineering Dr, Madison, WI. 53706
mchitturi@wisc.edu Ph: 608 890 2439 Fax: 608 262 5199

John D. Lee
Professor, Industrial & Systems Engr. Dept., University of Wisconsin-Madison
3007 Mech Engineering, 1513 University Ave, Madison, WI. 53706
jdlee@engr.wisc.edu Ph: 608 890 3168 Fax: 608 262 8454

Total Words: 5605 (Text) + 6 (Figures and Tables) = 7105 words

Submitted for Presentation at the 95th Transportation Research Board Annual Meeting
Submitted on: July 29, 2015

ABSTRACT

Driver distraction represents a major safety problem in the United States. Naturalistic driving data, such as SHRP2 Naturalistic Driving Study (NDS) data, provide a new window into driver behavior that promises a deeper understanding than was previously possible. Unfortunately, the current practice of manual coding is infeasible for large datasets like SHRP2 NDS, which contains millions of hours of video. Computer vision algorithms have the potential to automatically code SHRP2 NDS videos. However, existing algorithms are brittle in the presence of challenges like low video quality, under- and over-exposure, driver occlusion, non-frontal faces, and unpredictable and significant illumination changes, which are all substantially present in SHRP2 NDS videos.

This paper presents and evaluates algorithms developed to quantify high-level features pertinent to driver distraction and engagement in challenging videos like those in SHRP2 NDS. Specifically, a novel two-stage video analysis pipeline is presented for tracking head position and estimating head pose, and eye and mouth states. Results on challenging SHRP2 NDS videos are promising. The accuracy of the new head pose estimation module is competitive with the state of the art, and produces good qualitative results on SHRP2 NDS videos.

INTRODUCTION

Driver distraction represents a major safety problem in the U.S., contributing to 10 percent of fatal crashes, 18 percent of injury crashes, and 16 percent of all crashes in 2012 (*1*). The explosion of web-based applications and connected vehicle information makes the issue even more critical in the coming years. Naturalistic driving data, such as SHRP2 Naturalistic Driving Study (NDS) data (*2*), provide a new window into driver behavior that promises a deeper understanding than was ever possible with crash data, roadside observations, or driving simulator experiments. The millions of hours of SHRP2 NDS data presents an unprecedented opportunity to identify the factors contributing to distraction-related crashes. Although the SHRP2 NDS data include detailed vehicle state data, the video record of the driver and surrounding road situation often provide a more revealing account of driver behavior. Each frame of the NDS videos consists of four views (clockwise from upper-left): forward roadway view, driver view (rotated), rear roadway view, and downward steering wheel view as shown in FIGURE 1(a).

(a)

(b)

FIGURE 1: SHRP2 NDS Video: (a) Sample frames of NDS video (*2*), and (b) commonly found challenges.

The current practice of manual coding costs hundreds of dollars per minute of video, making coding of the millions of hours of video infeasible. Computer vision algorithms have the potential to automatically code SHRP2 NDS videos, extracting features from thousands of hours at a fraction of the cost of manual coding. However, using existing algorithms for SHRP2 NDS videos is problematic because of low video quality (e.g., low resolution, low dynamic range, compression artifacts), under- and over-exposure, occlusion, non-frontal faces, and unpredictable and significant illumination changes as shown in FIGURE 1b. The eventual goal of this research is to automatically quantify driver behavior, specifically distraction and engagement, by applying video analytics to the SHRP2 NDS videos. Toward this goal, this paper presents and evaluates algorithms developed to quantify high-level features pertinent to driver distraction and engagement: head pose, eye state, and mouth state.

APPROACH AND PREVIOUS WORK

The first step of estimating head pose and eye and mouth state is to detect the driver's head. There are many approaches to face detection in the computer vision literature, but the most popular is attributed to Viola and Jones (*3*), which uses a cascade of detectors operating on simple image features (the difference between the sums of adjacent pixel regions) to efficiently detect face regions of interest in an image. Many algorithms (*4*, *5*, *6*), and the one proposed in this paper, use the Viola-Jones face detector as a building block. However, by itself, Viola-Jones and others like it often fail on videos collected in challenging uncontrolled environments (e.g., SHRP2 NDS videos). Boosted exemplar-based face detectors have been proposed in (*7*) and (*8*) to overcome some of the challenges of uncontrolled environments. However, such algorithms have a large memory footprint and are relatively slow. Recently, Li *et al*. (*9*) proposed a faster algorithm based on convolutional neural networks that demonstrated more impressive results on challenging uncontrolled face images. The above methods focus on detecting faces within a single image and hence do not perform tracking. Tracking methods (*10*, *11*, *12*, *13*) can improve the robustness and accuracy of the head location and size estimates in videos. However, these tracking methods require considerable computation and hence are impractical for processing large datasets such as SHRP2 NDS, which contains millions of hours of video.

The goal of head pose recognition is to estimate the orientation of a subject's head, usually with respect to the camera viewpoint. Head pose recognition is often performed in conjunction with, or immediately after, facial landmark localization (*14*, *15*, *16*). Given a detected face, the goal of facial landmark localization is to locate landmarks of interest on the face (e.g., nose tip, mouth corners, and eye centers). Recently, exemplar-based (*17*), and iterative shape regression-based (*18*, *19*) approaches have demonstrated impressive landmark localization results on "in-the-wild" face images. The pipeline presented here uses an extended version of the exemplar-based approach described in (*20*, *21*) for landmark localization and pose recognition. A full review of head pose recognition is outside the scope of this paper; see (*22*) for a review. In the algorithm proposed in this paper, a collection of 3D shape models is fit to the 2D facial landmarks. Yaw, pitch, and roll head rotation angles are then robustly computed by "consensus" of the individual 3D shape fits.

Eye and mouth state (e.g., open/closed) recognition fits within a broader class of work concerned with facial expression and facial action unit recognition, which is typically performed by classification of geometric features (e.g., eye/mouth shape as represented by sets of eyelid/lip landmark locations), motion features (e.g., tracked regions in video), and/or global or local appearance features (e.g., image patches centered on landmarks) (23). Due to the limited resolution of the driver's face and its constituent parts in SHRP2 videos (where a driver's eye fits within a 10 x 8-pixel rectangle), the spatial accuracy of the eyelid and lip landmarks is often not exact enough to reliably estimate eye and mouth openness. Therefore, the algorithm presented here uses only local appearance for eye and mouth state estimation.

METHODOLOGY

A two-stage video analysis pipeline was developed for this project. In Stage 1, the driver's head is detected and tracked. Given the head region of interest, Stage 2 estimates head pose, and eye and mouth state. An overview of the pipeline is shown in TABLE 1. Details are presented in the following sections.

TABLE 1 Overview of the Video Analysis Pipeline

Step	Stage	Procedure	Input	Output
1	Head Detection and Tracking	Face detection	Video Frame	Face detection(s)
2		Spurious face elimination	Face detection(s)	Preserved detection(s)
3		Adaptive template head tracking	Preserved detection(s)	Head bounding box
4	Head Pose, Eye State, and Mouth State Estimation	Low-level feature extraction in region of interest	Image, head bounding box	Low-level features
5		Local landmark hypothesis generation	Low-level features	Landmark response maps
6		Global landmark shape regularization	Landmark response maps	Landmark estimates
7		Head pose estimation	Landmark response maps	Yaw, pitch, roll angles
8		Eye and mouth state estimation	Landmark estimates	Eye/mouth openness

Stage 1: Head Detection and Tracking

The objective of Stage 1 is to develop a computationally efficient algorithm for inference of the driver's head position in each frame. In particular, the algorithm should reliably track the driver's head even when the driver moves quickly and erratically. The head detection and tracking algorithm consists of three steps:

1. Frontal and profile face detection,
2. Spurious face elimination to reject false detections made in the first step, and
3. Adaptive, template-based head tracking.

With this 3-step approach, the driver's head can be tracked even when it is completely turned around, without the need for multiple-view head detection algorithms. Each of the three steps are elaborated below.

Step 1.1: Face Detection

During the first step, the OpenCV Viola-Jones (VJ) face detector (*3*) is applied to each frame independently. In many frames, the VJ detector fails to detect any faces, while in others, spurious faces are also detected, as shown in FIGURE 2(a). The output from this step serves as the input for spurious face elimination.

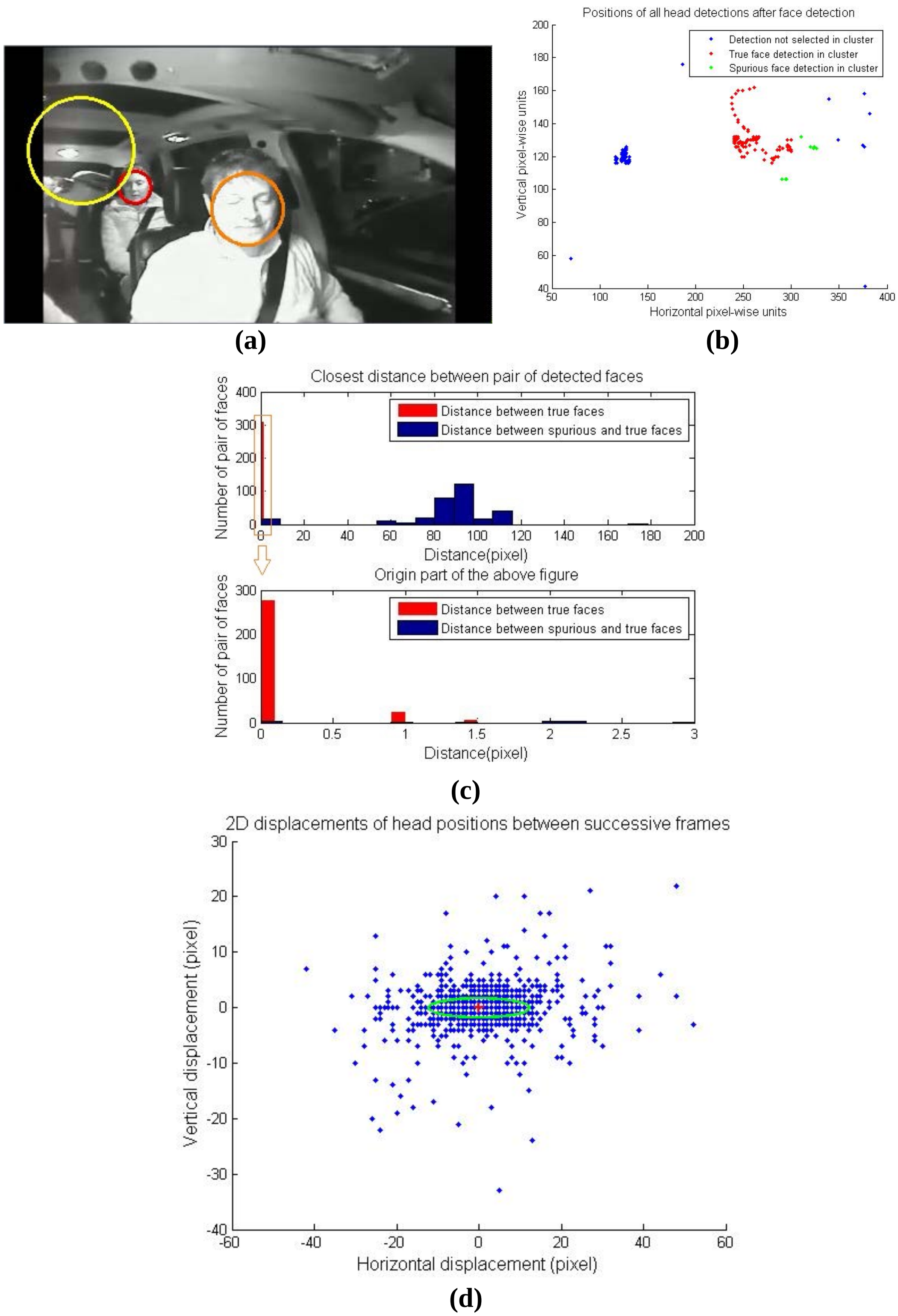


FIGURE 2 Face detection output: (a) Frontal face detection with spurious faces, (b) positions of all detected faces, (c) distance between pairwise detected faces, and (d) 2D displacements of head positions between successive frames.

Step 1.2: Spurious Face Elimination

The VJ detector may detect one or more spurious faces in each frame. Depicted in FIGURE 2 (b) are the true face positions (red) and spurious face positions (blue and green) of all faces detected by the VJ algorithm in one video clip. Note that the cluster of red and green points has an irregular shape due to the movement of the driver's head. Conventional clustering algorithms such as *k*-means (*24*) implicitly assume each cluster has an elliptical shape. Hence it may not be suitable for this kind of application. Instead, we employ a clustering method called density-based spatial clustering of applications with noise (DBSCAN) (*25*) that makes no assumption regarding the shape of the head location distribution.

With DBSCAN, for a given threshold ε, all data within the same cluster shall have at least one nearest neighbor in the same cluster within distance ε. In FIGURE 2 (c), the histogram of pairwise closest L2 distance between true detection positions of the driver's face, and the closest distance between positions of a spurious face and a true face are plotted. FIGURE 2(c) indicates that the choice of ε should be smaller than 20 and greater than 2. However, DBSCAN by itself clusters some spurious detections (green) as true detections (red) because of their proximity, as shown in FIGURE 2(b). Because of this, we need another parameter d_M (Mahalanobis distance threshold) to determine whether a position is too far from the mean position of faces in the cluster and hence is more likely to be a spurious face. Letting μ and S be the sample mean and covariance of the cluster obtained using DBSCAN, d_M of a point p is given by

$$d_M(p) = \sqrt{(p-\mu)^T S^{-1}(p-\mu)}$$

Another parameter, n_M (minimum number of points), determines how small a cluster can be. It is of less importance here. The values of these parameters were chosen empirically from testing video clips using three-fold cross validation: ε = 15, d_M = 3 and $n_M \leq 20$ produced the best results with *precision* = 100% and *recall* = 31.50% on the test data (described in the results section). About 99% of spurious faces were eliminated. However, no faces were detected in about 70% of frames, which is addressed by the head tracking step described next.

Step 1.3: Adaptive Template Head Tracking

After Step 2, the driver's head was detected with high confidence in only about 30% of the video frames. To improve this, Step 3 capitalizes on two observations: between successive frames (a) the driver's head position displacement is limited and (b) the changes in the appearance of the driver's head are relatively small. These observations motivate the use of head tracking to fill in missing detections from Step 2.

FIGURE 2(d) shows a scatter plot of displacements of head positions between successive frames in blue, mean displacement in red, and the covariance of displacement in green for 24 video clips. This provides an empirical estimate of the state transition probability $P(\mathbf{x}_t|\mathbf{x}_{t-1})$ of head position $\mathbf{x}$ from time t-1 to t. It shows $P(\mathbf{x}_t - \mathbf{x}_{t-1})$ can be modeled by a Gaussian distribution. Therefore, given the position of the driver's head in the current frame ($\mathbf{x}_{t-1}$), the position of the driver's head in the next frame ($\mathbf{x}_t$) may be limited to a search region, $S = \{\mathbf{x}_\tau \mid P(\mathbf{x}_t|\mathbf{x}_{t-1}) > 0\}$. In practical implementation, S is approximated by a rectangular region and $P(\mathbf{x}_t|\mathbf{x}_{t-1})$ is approximated by a uniform distribution over S.

We measure the similarity between a head template $\mathbf{y}_t$ and a candidate head region at $\mathbf{x}_t$ using cross correlation. The similarity scores are likely to vary with time: larger when the driver's head is stationary and smaller when the head is turning or the body is moving. By tracking the trend of the similarity score, one may determine a similarity score threshold at the current frame to determine the similarity of the templates. The computed similarity score is an empirical estimate of the likelihood of the head template is observed at the position of the candidate head region $\mathbf{x}_t$, i.e. $P(\mathbf{y}_t|\mathbf{x}_t)$. The posterior probability $P(\mathbf{x}_t|\mathbf{y}_t)$ then can be evaluated as

$$P(\mathbf{x}_t|\mathbf{y}_t) = \int_S P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{x}_{t-1})d\mathbf{x}_{t-1}$$

where the integration is over the search region *S.* The maximum posterior probability (MAP) estimation of the position of the driver's head at the current frame *t* is then found by

$$\mathbf{x}_t^* = \underset{\mathbf{x}_t}{\operatorname{argmax}} P(\mathbf{x}_t|\mathbf{y}_t)$$

Results for Head Detection and Tracking

Twenty four short (10-30 seconds) sample clips from SHRP2 NDS Insight videos (*26*) were selected for evaluation. Each clip exhibits challenging characteristics as demonstrated in FIGURE 1(b).

Evaluation was performed using two metrics:

- *Precision* = TP/(TP + FP) (a.k.a. *positive predicted value*)
- *Recall* = TP/(TP + FN) (a.k.a. *sensitivity*),

where TP is the number of true positive detections, FP is the number of false positive detections, TN is the number of true negative detections, and FN is the number of false negative detections. For each frame in these videos, the true head location was manually marked to define ground truth for each step. The confusion matrices of the three steps are given in FIGURE 3. Precision is high (about 99%) in Step 1, and does not decrease through Step 3. Recall is low (about 28%) in Step 1, but increases significantly to about 88% after Step 3.

Step 1

	H	NH
H	3133	8022
NH	23	91

Precision: 99.27%
Recall: 28.03 %

Step 2

	H	NH
H	3513	7642
NH	0	114

Precision: 100%
Recall: 31.50%

Step 3

	H	NH
H	9843	1312
NH	0	114

Precision: 100%
Recall: 88.24%

FIGURE 3 Confusion matrix for each step on 24 clips. H=head, NH=no head.

Stage 2: Head Pose, Eye, and Mouth State Estimation

Similar to the approach in Stage 1, Stage 2 also uses a pipeline to take the head information for each frame from Stage 1 and extracts head pose, eye and mouth states. It is important to note that, given the gamut of challenges in SHRP2 NDS videos, the automated pipeline is not perfect. Therefore, in each step of Stage 2, the pipeline produces a confidence value that can be used, for example, to highlight potentially problematic videos and frames for manual evaluation or coding.

An overview of the face analysis pipeline for Stage 2 is shown in TABLE 1; additional details are given below.

Step 2.1: Low-Level Feature Extraction

Dense SIFT (Scale Invariant Feature Transform) feature descriptors (*27*) are extracted in the region of interest (ROI) at regular three-pixel intervals. SIFT descriptors encode local image structure (e.g., points and edges) into 128-element histograms of image gradient intensity and orientation.

Step 2.2: Local Landmark Hypothesis Generation

A weighted, generalized Hough voting strategy (*28*) is used to map low-level features to landmark location hypotheses. Offline, a database of {low-level image feature, facial landmark} pairs from a large collection of training images was constructed using approximately 18,000 face images from the CMU Multi-PIE Face Database (*29*). Each {feature, landmark} pair has a spatial offset associated with it that maps the low-level feature to a landmark location. For example, a feature near the tip of the nose and a landmark at the center of the top lip might have an offset of $x = 0$, $y = 10$ that indicates the lip landmark is 10 pixels below the nose tip feature. At test time, each low-level feature descriptor is matched to similar features in the database. According to the example, a feature near the nose would "vote" for a lip landmark 10 pixels below it. Due to noise and inherent ambiguities in the image, these local votes may be noisy. However, because there are many {feature, landmark} pairs, votes will tend to pile up at the correct landmark locations. After spatial smoothing, the votes generate a landmark probability map for each landmark type.

For efficiency, all feature descriptors are quantized into visual words before they are used for landmark voting. Each visual word is identified by a unique integer ID and represents a cluster of similar feature descriptors in the training database. A fast, approximate nearest neighbor algorithm (*30*) is used to map each feature descriptor to a visual word ID. For efficient retrieval from the exemplar database, each {feature, landmark} pair is stored in an inverted index by visual word ID number.

Each landmark vote is weighted. This is key to the success of the algorithm. Intuitively, some features in the image are better at predicting landmarks than others. For example, features on the cheek are locally ambiguous and should therefore be down-weighted; features on the upper nose are more unique and can better predict eye landmarks and should therefore be up-weighted. In previous work (*20*), weights were computed in a highly data-intensive way. In the current implementation, an online feature weighting method replaces the offline one. The weight of each vote is inversely proportional to (a) the vote offset distance and (b) the variance among the offsets generated by features that map to the same visual word ID. Intuitively, this gives more weight to low-level image features that are both near landmarks and consistently vote for the same landmark location. Technical details are presented in Smith and Zhang (*20*). Computing weights online incurs a modest computational cost and a small decrease in accuracy, but reduces the memory footprint of the database by a factor of 10.

Step 2.3: Global Landmark Regularization

Local landmark estimates can be noisy and ambiguous (e.g., sunglasses occlude eye landmarks). Shape regularization addresses this problem by imposing global structure over the spatial arrangement of landmarks. Informally, the regularization algorithm attempts to find a set of landmark hypotheses that agree well with a consensus of exemplar face shapes. Belhumeur *et al.* (*17*) introduced this general idea, but used 2D exemplar shapes. Instead, 3D exemplars are used in this work. The regularization procedure consists of the following 6 steps:

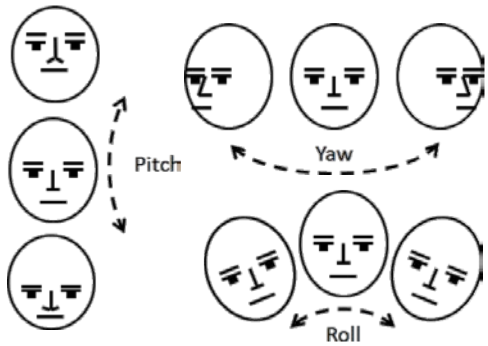
1. Select four landmark types at random, and one candidate at random for each type.
2. Select a 3D exemplar shape at random.
3. Compute a weak perspective projection P_j that projects the 3D exemplar shape onto the 2D image using the four landmark correspondences as constraints. This generates one face shape candidate, S_j.
4. Compute a score for S_j. Each landmark i = 1,2,… N in S_j has a probability, v_{ji}, equal to the value in the probability map (generated by the weighted Hough voting step) at the landmark location. The score for S_j is $\log(v_{j1}) + \log(v_{j2}) +\ldots+ \log(v_{jN})$.
5. Repeat Steps 1-4 many times. Save the top-scoring T = 100 face shape candidates.
6. Compute the final landmark locations. For each landmark type, compute the median location among the top-scoring T face shape candidates.

A confidence value is computed for the final landmark estimate by measuring v_i (the value in landmark *i*'s probability map at each landmark location), and then averaging. Note that four landmark candidates are selected in Step 1 because computing a weak perspective projection requires a minimum of eight constraints (an *x* and a *y* from each landmark): scale, *x*-translation, *y*-translation, absolute yaw angle, yaw sign, absolute pitch angle, pitch sign, and roll angle. The yaw and pitch angles are ambiguous up to a sign change, but the roll angle is not. FIGURE 4 (a) shows the three types of pose rotation angles.

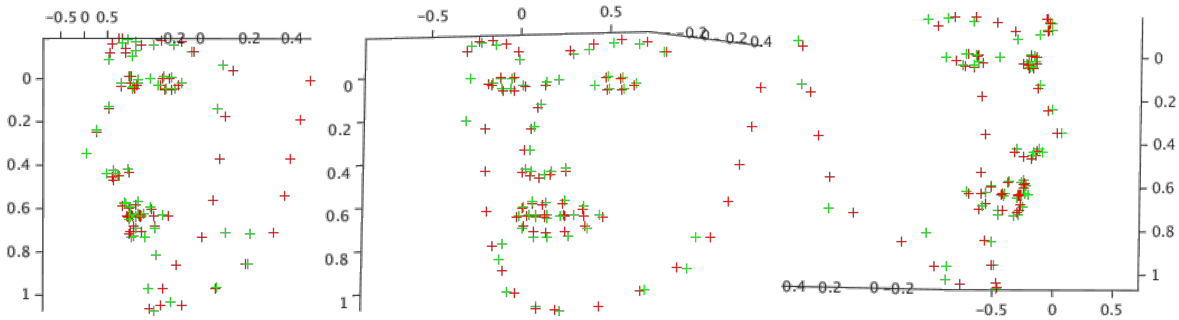
Approximately 800 3D exemplar shapes were generated from sets of 2D landmarks. Each 3D shape was computed by a structure-from-motion (SfM) algorithm (*31*) applied to a set of manually annotated 2D landmarks from the Multi-PIE Face Database; each set of 2D landmarks depicted the same face from different viewpoints. Expectation maximization (EM) (*32*) and principal component analysis (PCA) (*33*) are used to fill in missing points and reduce spatial noise in the computed 3D exemplar shapes, as shown in FIGURE 4(b). The noisy raw points from the SfM algorithm are shown in green. The EM+PCA results are shown in red.

Step 2.4: Head Pose Estimation

Each of the T = 100 top face shape hypotheses in the shape regularization step has an associated weak perspective projection, which includes yaw, pitch, and roll angles. Head pose is expressed using these three angles. The final yaw angle is computed by taking the median of the yaw angles from the T=100 top weak perspective projections. The consensus of yaw angles among the T = 100 top weak perspective projections is used to compute a confidence value. Specifically, $confidence = 1 - std((\text{angle}_1, \text{angle}_2, \ldots, \text{angle}_{100}))/M$, where *std* is standard deviation and M is set empirically. Pitch and roll angles are computed similarly. Experimentally, yaw angle estimates were found to be consistently too small in magnitude. Therefore, the final yaw angle is multiplied by 1.3, set by minimizing the error between estimated and ground truth yaw angles.



(a)



(b)

(c)

FIGURE 4: (a) Head pose described by pitch, yaw, and roll angles, (b) different views of a 3D shape exemplar computed from 2D landmarks, and (c) closed eye (left) and open eye (right) from a SHRP2 NDS sample video.

Step 2.5: Eye and Mouth State Estimation

The eye and mouth state estimation module is executed after landmark localization is complete. FIGURE 4(c) shows an example from one of the InSight SHRP2 NDS sample videos illustrating the challenge with eye state detection. The two frames shown in FIGURE 4(c) are qualitatively very similar to frames typically found in the much larger SHRP2 NDS dataset. Eye state estimation is particularly challenging in the SHRP2 videos because they have low resolution and low dynamic range. The eye fits within a small 10 x 8 pixel window, and the differences between a closed eye (left) and an open eye (right) are subtle, which makes eye state estimation particularly challenging. For concreteness, eye state estimation is described here, and mouth state estimation is performed in the same way.

A straightforward approach to eye state estimation would be to compute eye openness as the distance between the upper and lower eyelid landmarks. However, this would require consistent subpixel landmark accuracy, which is often unrealistic in SHRP2 videos. Therefore, all of the pixel intensity information around each eye is used directly to estimate the state.

Specifically, the system extracts a patch of pixel intensity values centered on the centroid of the eye landmarks. The intensity values are normalized to reduce the impact of global illumination variation. The system then performs *k*-nearest neighbors classification to compute the state of the eye, which is given as a relative distance (between eyelids) and a confidence value. The system computes the cross correlation between the test patch and a collection of exemplar patches, which each have a known eyelid gap. A weighted cross correlation similarity measure is used, where the weight of each pixel is determined by an isotropic Gaussian function centered on the patch; this emphasizes pixels near the center of the eye and de-emphasizes others.

The final eye state estimate is the median eyelid gap among the top *k* closest exemplar patches (*k* was set at 10 based on cross-validation experiments). If desired, a threshold can be applied to the estimated eyelid gap to produce a binary "open" or "closed" state estimate. The confidence value is a function of the level of consensus (quantified by standard deviation) among the top *k* eyelid gaps. The assumption was that poorly matched patches would be more randomly distributed than well matched patches. To improve robustness to landmark errors, several different patch offsets (e.g., $x = -5$ to $x = 5$ pixels) are tried and the offset with the best match is chosen. The algorithm computes a confidence value for the estimate by measuring the consensus among the *k* closest exemplar patches: $confidence = 1 - std((\mathrm{gap}_1, \mathrm{gap}_2, \ldots, \mathrm{gap}_{10}))/N$, where *std* is standard deviation and *N* is set empirically.

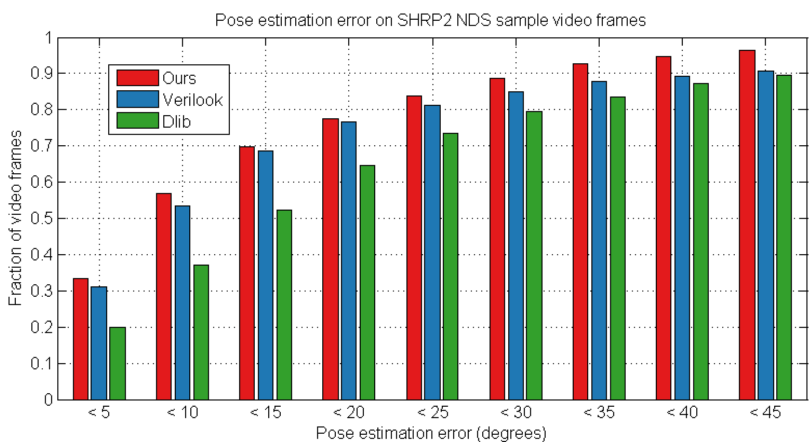
Results for Head Pose, Eye, and Mouth State Estimation

For initial testing, the Annotated Faces in the Wild (AFW) dataset (*15*) was used, which includes 468 faces in a wide variety of real-world conditions. FIGURE 5(a) shows qualitative results from the proposed algorithm on AFW faces. Although some mistakes are inevitable (bottom row), our approach is robust to a wide variety of "in-the-wild" conditions. AFW faces include accurate ground truth annotations: 68 landmarks and yaw, pitch, and roll head rotation angles for each face. To minimize the differences between AFW images and SHRP2 video frames, all AFW images were converted to grayscale and resized all faces to the typical size of SHRP2 faces (30-pixel inter-ocular distance (IOD)) using the face detection result. Note that the results in FIGURE 5(a) were computed on these more difficult, smaller grayscale faces; however, the algorithm outputs landmark estimates that are rescaled to the original image resolution, and so they are simply shown overlaid on the original images. In previous work (*20*), quantitative results showed that the proposed landmark localization algorithm produces results favorable in accuracy to several state-of-the-art approaches on AFW faces.

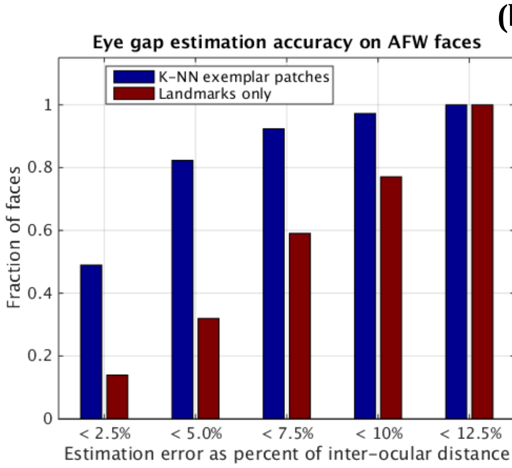
FIGURE 5(b) shows quantitative results for pose estimation on challenging clips (2,600 frames total) from several SHRP2 NDS videos. Accuracy is computed relative to manually-annotated "ground truth" yaw angles. For approximately 70% of the test frames our algorithm estimates the yaw angle of the driver's head to within 15 degrees. The yaw angle estimation accuracy of our algorithm compares favorably to two commercial software libraries applied to the same clips: Verilook (*34*) and Dlib (*35*).

FIGURE 5(c) and (d) show quantitative results for eyelid and mouth gap estimation, respectively (large eyelid gap implies an open eye state, and small eyelid gap implies a closed eye state). Due to lack of eye or mouth state ground truth with SHRP2 data, results are shown for AFW faces, which include detailed eyelid and lip landmarks from which ground truth eyelid and

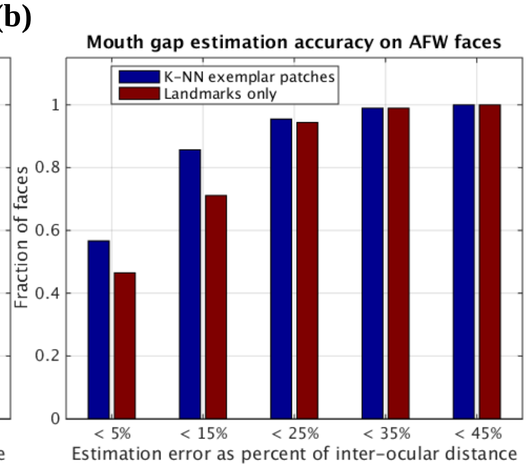
(a)



(b)



(c)



(d)

FIGURE 5: (a) Qualitative results on AFW faces. Cumulative error distributions of: (b) yaw head pose on SHRP2 NDS sample video frames, (c) vertical eyelid gap on AFW faces, and (d) vertical mouth gap on AFW faces.

mouth gaps can be computed. We see that, due to the low resolution of the test faces (similar to SHRP2 resolution), *k*-nearest neighbors classification of eye and mouth patches outperforms gap estimates using only the eyelid and mouth landmarks. For 85% of AFW faces, our algorithm estimates eyelid openness to within 1.5 pixels from ground truth, and to within 4.5 pixels for mouth openness. In all cases, faces were resized to 30 pixels inter-ocular distance (IOD), which is similar to the size of SHRP2 driver faces.

CONCLUSIONS & RECOMMENDATIONS

The challenging nature of SHRP2 NDS videos requires the development of innovative approaches for ultimately achieving the goal of automatic feature extraction for quantifying driver distraction and engagement. Experience shows that clips most relevant to distraction and disengagement are likely to be those that are most difficult to code automatically. Therefore, all the algorithms presented in this paper produce a confidence value associated with each estimate to identify where manual involvement might be necessary.

A flexible, two-stage video analysis pipeline for tracking head position and estimating head pose, and eye and mouth states was developed. A novel template matching approach was designed to address the challenge of driver movement, off-center head position, and head rotation. Results on challenging SHRP2 NDS videos are very promising; specifically, no false positives and false negatives below 1%. Previous landmark localization work by the authors was adapted and extended to better handle the challenges of SHRP2 videos. The accuracy of the new head pose estimation module is competitive with the state of the art, and produces good qualitative results on SHRP2 NDS videos. Eye state estimation is particularly challenging in the SHRP2 videos because they have low resolution and low dynamic range. Therefore, an exemplar approach was developed for eye and mouth state estimation. Based on the initial quantitative evaluation on challenging low-resolution “in-the-wild” faces and the qualitative evaluation on SHRP2 video frames, this approach to eye and mouth state estimation shows promise. Work to date has focused on implementing proof-of-concept solutions. Future work will continue to improve the robustness, accuracy and runtime of the video analysis pipeline.

ACKNOWLEDGEMENTS

The research presented in this paper is based upon work supported in part by the Federal Highway Administration under contract number DTFH6114C00011 and the National Science Foundation under grant number IIS-0916441.

REFERENCES

1. National Highway Transportation Safety Administration (NHTSA). *Traffic Safety Facts: Distracted Driving 2009*. Washington, DC. Retrieved from http://www.distraction.gov/research/pdf-files/distracted-driving-2009.pdf in 2014.
2. Campbell, K. The SHRP2 Naturalistic Driving Study. *TR News*, 282, Sep-Oct 2012.
3. Viola, P, and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.
4. Wu, J., and M. M. Trivedi. A two-stage head pose estimation framework and evaluation. *Pattern Recognition*, Vol. 41, Issue 3, 2008, pp. 1138-1158.
5. Murphy-Chutorian, E., A. Doshi, and M. M. Trivedi. Head pose estimation for driver assistance systems: a robust algorithm and experimental evaluation. *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, 2007, pp. 709-714.

6. Vatahska, T., M. Bennewitz, and S. Behnke. Feature-based head pose estimation from images. *7th IEEE-RAS International Conference on Humanoid Robots*, 2007, pp. 330-335.
7. Li, H., Z. Lin, J. Brandt, X. Shen, and G. Hua. Efficient boosted exemplar-based face detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp.1843-1850.
8. Shen, X., Z. Lin, J. Brandt, and Y. Wu. Detecting and aligning faces by image retrieval. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3460-3467.
9. Li, H., Z. Lin, X. Shen, J. Brandt, and G. Hua. A Convolutional Neural Network Cascade for Face Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5325-5334.
10. Tu, J., T. Huang, and H. Tao. Accurate head pose tracking in low resolution video. *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, 2006, pp. 573-578.
11. Murphy-Chutorian, E., and M. M. Trivedi. HyHOPE: hybrid head orientation and position estimation for vision-based driver head tracking. *Proceedings of the IEEE Intelligent Vehicles Symposium*, Eindhoven, The Netherlands, 2008.
12. Comaniciu, D., V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 142-149.
13. Stauffer, C., and W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, No. 8, August 2000.
14. Gu, L., and T. Kanade. 3D alignment of face in a single image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1305-1312.
15. Zhu, X., and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2879-2886.
16. Horprasert, T., Y. Yacoob, and L. S. Davis. Computing 3d head orientation from a monocular image sequence. *Proceedings of the 25th Annual International Society for Optics and Photonics Applied Image and Pattern Recognition (AIPR) Workshop on Emerging Applications of Computer Vision*, 1997, pp. 244–252.
17. Belhumeur, P. N., D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 12, 2013, pp. 2930-2940.
18. Xiong, X., and F. De la Torre. Supervised descent method and its applications to face alignment. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 532-539.
19. Ren, S., X. Cao, Y. Wei and J. Sun. Face alignment at 3000 FPS via regressing local binary features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1685-1692.
20. Smith, B.M., J. Brandt, Z. Lin, and L. Zhang. Nonparametric Context Modeling of Local Appearance for Pose- and Expression-Robust Facial Landmark Localization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2014, pp. 1741-1748.

21. Smith, B.M., and L. Zhang. Collaborative Facial Landmark Localization for Transferring Annotations Across Datasets. *Proceedings of the European Conference on Computer Vision*, 2014, pp. 78-93.
22. Murphy-Chutorian, E., and M. M. Trivedi. Head Pose Estimation in Computer Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 4, 2009, pp. 607-626.
23. Coen, J. F., and F. De la Torre. Automated Face Analysis for Affective Computing. *The Oxford Handbook of Affective Computing*, Oxford University Press, 2014.
24. MacQueen, J. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, CA, Vol. 1, 1967, pp. 281-297.
25. Ester, M., H-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226-231.
26. Virginia Tech Transportation Institute. The 2nd Strategic Highway Research Program Naturalistic Driving Study InSight Dataset (Version 1.0) [Video], 2015. doi:10.15787/VTT1CC72. Retrieved from: https://insight.shrp2nds.us.
27. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Vol. 60, No. 2, 2004, pp. 91-110.
28. Ballard, D. H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, Vol. 13, No. 2, 1981, pp. 111-122.
29. Gross, R., I. Matthews, J. F. Cohn, T. Kanade, and S. Baker. MultiPIE. *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.
30. M. Muja and D. G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2009.
31. Snavely, N., S. M. Seitz, and R. Szeliski. Photo Tourism: Exploring image collections in 3D. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006), 2006.
32. Dempster, A. P., N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, Vol. 39, No. 1, 1977, pp. 1-38.
33. Hotelling, H. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, Vol. 24, No. 6, 1933, 417-441.
34. Verilook SDK: Face identification for stand-along or web applications, Neurotechnology. Retrieved from http://www.neurotechnology.com/verilook.html
35. Real-Time Face Pose Estimation, Dlib C++ library. Retrieved from http://dlib.net/