Visual Monitoring of Driver and Passenger Control Panel Interactions

Toby Perrett, Majid Mirmehdi, *Senior Member, IEEE*, and Eduardo Dias

***Abstract*—Advances in vehicular technology have resulted in more controls being incorporated into cabin designs. We present a system to determine which vehicle occupant is interacting with a control on the center console when it is activated, enabling the full use of dual-view touchscreens and the removal of duplicate controls. The proposed method relies on a background subtraction algorithm incorporating information from a superpixel segmentation stage. A manifold generated via the diffusion maps process handles the large variation in hand shapes, along with determining which part of the hand interacts with controls for a given gesture. We demonstrate superior results compared with other approaches on a challenging dataset.**

***Index Terms*—Computer vision, driver monitoring, hand-gesture recognition, infotainment, user determination.**

I. Introduction

Determining which vehicle occupant is interacting with controls on the cabin's center console is of growing interest to vehicle manufacturers. Dual-view touchscreens are beginning to be included in cabin designs, which for example allow the driver to see a GPS display while the passenger sees a movie. One current drawback is that the driver and passenger are confined to interacting with their "half" of the screen. Technological advances and competition between manufacturers has also led to a larger variety of controls being included for the purposes of safety, comfort and entertainment. This can result in a cluttered interface, with duplicate functionality for both the driver and passenger (e.g., multiple temperature dials).

A number of challenges are presented when trying to design and implement a system that monitors control interactions with the required reliability to be included in production vehicles. As it will be necessary to monitor occupants' arms and hands, such a system must be able to handle different combinations of skin color, jewelry, and clothing (including gloves). In more complicated cases where gesture information is needed to make a decision, different hand shapes and sizes must be accounted for. There will also be a high inter- and intra-person variability in performing these interaction gestures. In cases where there is occlusion, it will be necessary to predict how the hand shape, and thus the area being interacted with, changes.

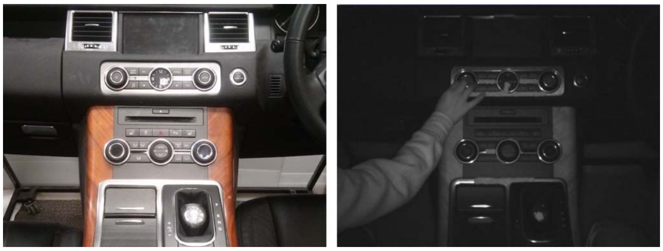


Fig. 1. Control panel. (Left) Under normal lighting conditions with an RGB camera. (Right) With a near-infrared camera with near-infrared illumination.

Weight sensors in seats [1], RGB cameras [2], or depth sensors [3] could be considered to monitor control interactions. We have chosen to use a near infra-red (NIR) camera as it has the required fidelity, is well suited to an automotive environment and is cost effective. These issues are addressed further in Section II. Fig. 1 gives examples of the center console using RGB and NIR cameras.

Other works have attempted driver hand tracking [4], [5] or have determined if an occupant's hand is in the vicinity of a control panel [6], but none have attempted to establish which occupant is interacting with specific controls. In order to do this accurately, a method of modeling hand shapes and gestures is needed. Accurate hand shape modeling has been achieved with depth cameras [3], but most relevant are works that model certain gestures in low light conditions using standard infra-red cameras [7], [8].

We propose a system, using a single camera mounted in the ceiling of the cabin, to determine which occupant is interacting with which control on the center console. This would enable the full use of the dual-view touchscreen for both driver and passenger. Another advantage is that duplicate controls would no longer be needed (e.g., a single temperature or air conditioning dial would be sufficient in luxury cars with multiple climate control zones), resulting in a cleaner cabin design and the space to add a choice of other controls.

We adopt a background subtraction algorithm as the first stage in the proposed method. The foreground mask is then cleaned up using superpixel voting and the hand contours are extracted, using optical flow if necessary. Our main contribution lies in the next stages, where hand outlines are modeled with a manifold generated via the diffusion maps process. The manifold is constructed using a difference measure that takes

Manuscript received November 4, 2014; revised June 1, 2015, December 14, 2015, and March 9, 2016; accepted May 7, 2016. Date of publication June 8, 2016; date of current version February 1, 2017. This work was supported by Jaguar Land Rover Research. Data access statement: Due to a confidentiality agreement with Jaguar Land Rover, the data supporting this article is not openly available. The work of T. Perrett was supported by an Engineering and Physical Sciences Research Council/Janguar Land Rover Industrial Cooperative Awards in Science & Technology Award. The Associate Editor for this paper was C. Wu.

T. Perrett and M. Mirmehdi are with the Department of Computer Science, University of Bristol, Bristol BS8 1UB, U.K. (e-mail: tp8961@bristol.ac.uk; majid@cs.bris.ac.uk).

E. Dias is with the Research and Technology Department, Jaguar Land Rover, Coventry CV3 4LF, U.K. (e-mail: edias@jaguarlandrover.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2567540

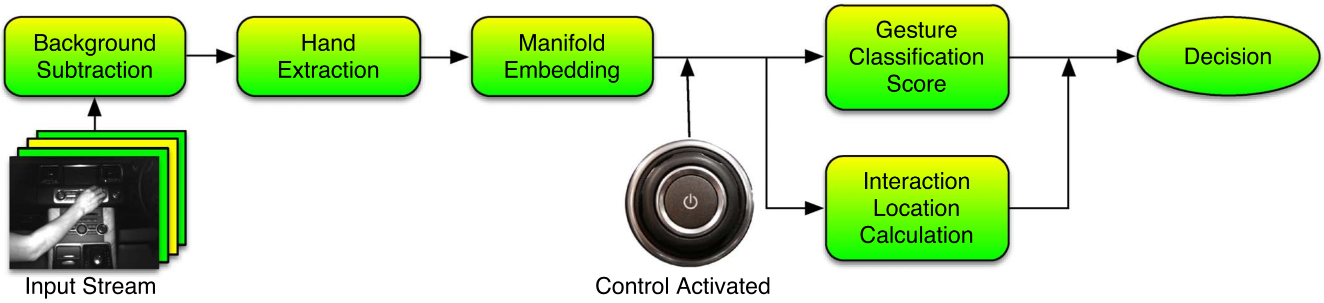


Fig. 2. Schematic overview of the proposed method, which gives a decision on which occupant is interacting with a control when it is activated.

into account both overall hand shape and the parts of the hand which interact with different controls. This gives an accurate location for a sample hand's interaction, and the path through this manifold is used to provide gesture information. When a control is activated, each hand is scored based on these attributes in addition to the position of the control. The hand using the control is determined to be the one with the highest score. Fig. 2 illustrates an overview of the proposed method.

The rest of this paper is laid out as follows. Section II presents an account of related literature, Section III gives an overview of data we work with, and we detail the various stages of our methodology in Section IV. Section V outlines our experiments and results, and we present our concluding arguments in Section VI.

II. Background

The majority of computer vision based monitoring systems for use in vehicles have focussed on safety [2], [4], [9]–[19]. Examples of this include drowsiness monitoring by observing head pose and gaze direction [9]–[11] or via the PERCLOS measurement [12]–[15] and occupant classification for automatic airbag suppression [16]–[18]. In this work however, we will be looking at monitoring the occupants in order to improve the vehicle's cabin layout and the occupant experience.

As no previous works have attempted to solve this problem per se (see Section V for an explanation on how we adapt other methods for comparison), we start by reviewing the suitability of sensors typically available in vehicle cabins for this task. We then look at computer vision systems for observing a driver's position in a vehicle. Next, we cover previous methods specifically designed to track a driver's hands, moving onto shape-change based gesture recognition methods for use in low light conditions and ways of representing these complex hand shapes. As the tracking procedure in the proposed method isolates hand shapes and positions, global feature based gesture recognition techniques are not considered here.

An alternative to a computer vision based approach might be to monitor the weight sensors in the driver and front passenger seats, typically used to indicate whether these seats are occupied [1]. It is possible that weight sensors applied to our task could work for simple cases where just the driver or passenger is interacting with a control. However, more complicated cases where both occupants are interacting with controls simultaneously would be problematic—there would be no way of deciding which occupant is interacting with which control. Additionally, as cameras could eventually replace weight sensors for the task of occupant detection and classification [16]–[18], using a camera for this task could make it more suitable for inclusion in cabin designs of the future.

Zhao *et al.* [2] classified the driver's posture into one of four categories (hands on the steering wheel, changing gear, eating, and on the phone) and relied on head and hand detection and relative positions. The aim of this work was to enhance driver assistance systems, whereby the driver could be prompted to pay attention to the road when eating, or warned that using a mobile phone is prohibited and dangerous. Tran and Trivedi [19] used two cameras to fit a simplified torso model using the locations of the driver's head and hands, again with the aim of providing more information to a driver assistance system. The hand detection parts in these methods, along with many others [20], [21], rely on color information as the first stage in the detection process. One obvious failure case is when the subject is wearing gloves. Some skin colors can also be more difficult to detect, but the greatest limitation which makes these methods unsuitable for our use is that they struggle, or else completely fail, during night time operation. Similarly, depth based methods for hand tracking using time of flight cameras, such as that by Oikonomidis *et al.* [3], which looked at interlocking hands with a frontal view in an indoor environment, are not practical in a vehicle due to the direct sunlight they may encounter.

Perhaps the most relevant and robust work is that by McAllister *et al.* [5], which looked at tracking a driver's hands, using a greyscale camera, in the vicinity of the steering wheel. A background subtraction method was used, followed by a simple circle fitting, which can give a rough approximation to the hands' positions, but this lacks the fidelity required for our task. Crespo *et al.* [4] used background subtraction with prioritizing of certain locations, such as around the steering wheel and on the gear stick, for the same task. They used a NIR camera with NIR illumination to ensure successful night time operation, but again this tracking alone is not accurate enough for our problem.

Cheng *et al.* [6] investigated monitoring controls by taking histogram-of-orientated-gradients over an image patch containing the control panel, with the aim of alleviating driver distraction. However, they did not attempt to determine which occupant was interacting with which control, only giving an indication whether the occupants' hands were in the vicinity of the control panel.

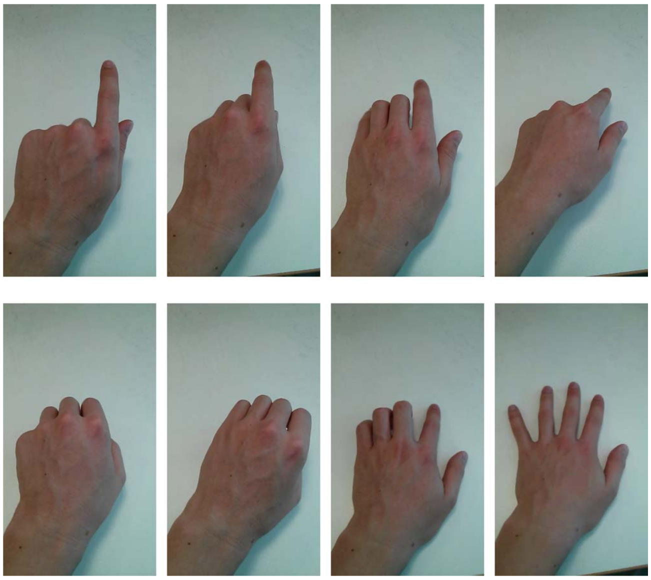


Fig. 3. Some examples from our hand shape training set, which consists of 900 images.

In addition to a hand tracking stage, we will also need a way to model which part of the hand interacts with controls. This is because just taking the closest hand from the hand tracking stage provides insufficient accuracy when two hands are interacting with controls close together. Manifolds provide a way to represent nonlinear data in a reduced space, and are thus ideal candidates for modeling human hand attributes. Key works on gesture recognition in low light conditions are those of Lee and his co-workers [7], [8] which looked at recognizing grasping gestures. An infra red camera was used, and thus they operated on the hand outline. Manifolds were pre-specified, one for each gesture, and the path of a testing sample through one of these manifolds represents the gesture being performed. Choi *et al.* [22] used a manifold generated by the kernel isomap method to classify strokes as gestures, but only looked at paths through space, not how the hand shape changes. Etyngier *et al.* [23] showed how a single manifold generated by the diffusion maps process [24], [25] can be used to organize and model a set of shapes with a large degree of variation, and applied this approach as part of a level set segmentation framework.

III. Dataset

Only a simple set of hand shape training images is needed to construct the hand shape manifold in our method (detailed in Section IV-C). We perform this using 900 hand images taken from *one subject only* with a webcam from a similar angle to the main experiments (see Fig. 3).

To evaluate the proposed method, two datasets are introduced. First, for training and testing, control interactions with the dashboard of a Range Rover Sport were filmed (see Fig. 4). To verify that the proposed method can be used in other vehicles and with different control panels, a second dataset was collected just for testing. This consists of footage from two additional Sports Utility Vehicles (SUVs) and a Saloon car with a lower ceiling height and more vertically mounted control panel (see Fig. 5). In both cases the camera was mounted at ceiling height, just back from and in between the two front seat headrests. Fig. 4(a) gives a profile view of the camera position within the cabin. This camera position was chosen because it allows the camera to capture the hands before they start using controls towards the edge of the control panel, which provides more information to the method in Section IV-E (increasing gesture classification accuracy), and ensures that the occupants' hands enter the frame from their respective sides. A further benefit is that the camera can be incorporated into an area not commonly

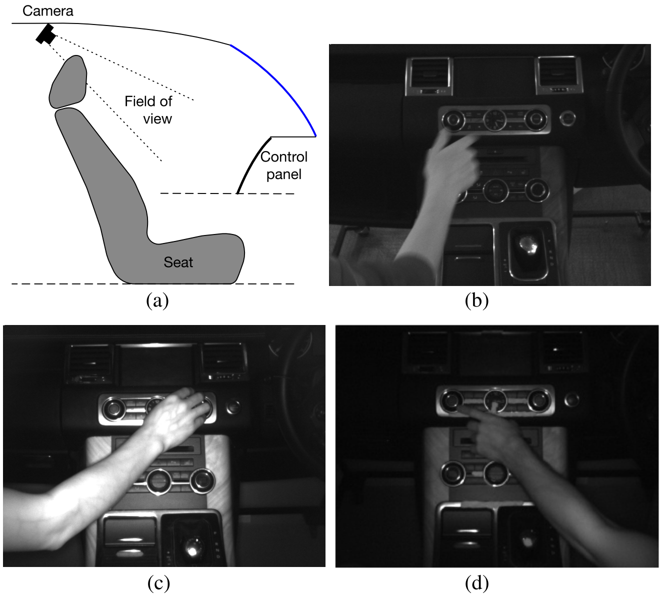


Fig. 4. Camera setup examples. (a) Cabin layout. The camera is mounted on the center line of the vehicle; thus, its view is not obscured by the front seats. (b) Day, no NIR filter or illumination. (c) Day, NIR illumination and filter. (d) Night, NIR illumination and filter.

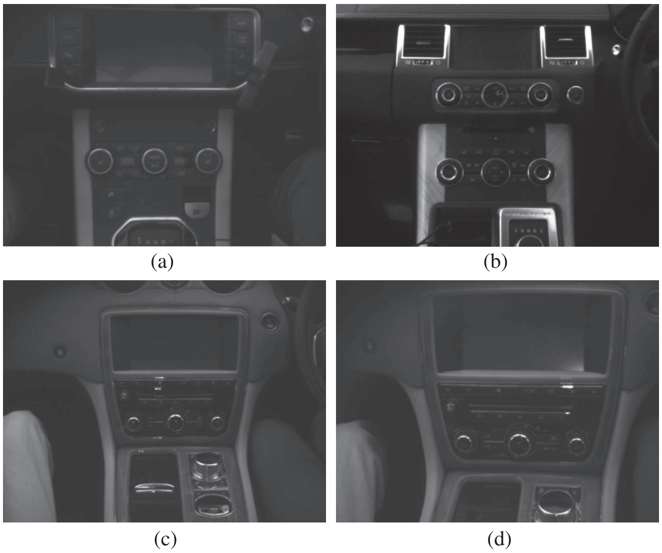


Fig. 5. Examples of other cabin layouts used to verify that the proposed method works in different vehicles. (a) SUV 1. (b) SUV 2. (c) Saloon, normal camera position—mounted between seats. (d) Saloon, camera mounted near the rear-view mirror. See Section V for a discussion of how this affects the performance of the proposed method.

taken up by a sunroof. A consumer grade monochrome camera with a 60° field of view lens, a NIR pass filter and NIR illumination was used to capture footage in daytime and night time, and the same camera was used without these additions for daytime examples (see Fig. 4). Using NIR illumination and filtering provides a number of benefits in an automotive environment. The illumination allows for a shorter exposure time as more light is available to the camera, which reduces the amount of motion blur, and during night time operation it allows the scene to be lit without creating a possible distraction for the driver. There is also more consistency between night and day time footage as a significant proportion of the scene lighting is under control. We had 10 adult volunteers and one child with a wide range of skin tones, jewelery and clothing.

As controls can be approached from any angle, we do not need motion information to be included in our gesture model. As such, we are only interested in how the hand shape changes independent of its position, and this makes our method unsuitable for comparison with standard gesture recognition datasets, such as [26], [27]. In these works, the best performance is obtained by taking global features over the whole image—clearly not a suitable approach for the task addressed here. Additionally, we found that hand shape change over time was a much more reliable indicator of the gestures our test subjects performed than positional trajectories. If we were to incorporate hand trajectories, we would need a much larger quantity of training gestures (which could also reduce accuracy compared to the more consistent hand shape change information). These would be necessary to handle the larger variation between different test subjects, as well as in multiple runs by the same subject. Reference trajectories from every control to every other control would be needed, further increasing the volume of training data required. Another potential issue is that the system would be less portable, as additional training trajectories would be needed for every control that is added or moved.

IV. Proposed Method

We proceed by first using background subtraction and a superpixel-based method to obtain a cleaned foreground mask, as explained in Section IV-A. Next, in Section IV-B, hand contours are extracted from the foreground mask. Then in Section IV-C, training hand shapes are used to generate a manifold that takes into account both overall hand shape and the parts of the hand which interact with controls. The procedure for embedding new hands into this manifold is in Section IV-D and a method for obtaining gesture information from manifold embeddings is then given in Section IV-E. Finally, the interaction confidence score, used to determine which hand is interacting with a specific control, is presented in Section IV-F. Fig. 2 gives an overview of the entire process.

A. Background Subtraction and Superpixel Cleaning

In the first stage of the proposed approach, we use the Pixel-based Adaptive Segmenter [28] for background subtraction. It was chosen as it allows small foreground objects (in our case these are likely to be noise) to decay into the background model quickly whilst larger objects persist. It consists of a background model derived from previous frames, and dynamically updates per pixel decision thresholds and learning rates. This results in a foreground mask—see Fig. 6(b) for an example.

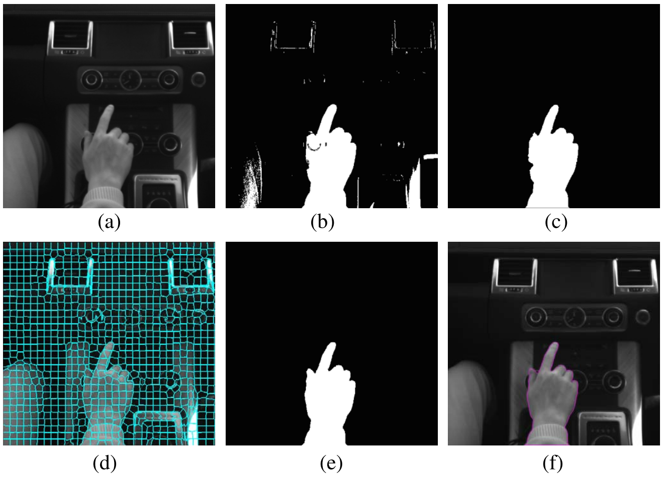


Fig. 6. Background subtraction process. We do not use a median filter, instead using SLIC superpixel voting to obtain a better foreground mask. (a) Original image. (b) Foreground mask. (c) Median filter applied to the foreground mask. (d) SLIC superpixel segmentation. (e) Superpixel voting applied to the foreground mask. (f) Result.

The foreground mask generated by the background subtraction process next needs to be cleaned up to remove noise. One possible approach would be to apply a median filter to the foreground regions [an example is given in Fig. 6(c)]. However, a disadvantage of this is that edge information is not necessarily preserved. A more recent approach by Schick *et al.* [29] introduced the idea of *probabilistic superpixels*.

Before background subtraction, the image is over-segmented using the Simple Linear Iterative Clustering (SLIC) algorithm [30] (we use the gSLIC implementation [31]). See Fig. 6(d) for an example. This is essentially a k-means clustering of pixels in the combined image and intensity space. We apply a high intensity weighting in order to capture more edge information. In [29], each superpixel is assigned a probability of being in the foreground based on the number of pixels in the foreground mask. A Markov Random Field then minimizes an energy functional consisting of this probability and the color similarity to neighboring superpixels. In our case, because we are just looking for large foreground regions and have no guarantee of intensity similarity between neighboring superpixels (e.g., due to a textured background and clothing patterns), it proves sufficient to just take those superpixels with a foreground probability above a certain threshold. This has an additional benefit of reducing the frame processing time.

After the image is segmented into n superpixels, each pixel p is assigned to the superpixel P_i containing it. Given a foreground mask F, its cleaned form $\hat{F}$ is then

$$\hat{F} = \bigcup_{i \in Q} P_i \quad \text{where} \quad Q = \left\{ i : \sum_{p \in P_i} \frac{F(p)}{\|P_i\|} > \lambda \right\}. \tag{1}$$

Here $\lambda = 0.5$ is a threshold determined empirically. As our application features large foreground objects with clearly defined

TABLE I
GESTURE CLASSIFICATION SUCCESS RATES WITH A RANDOM FOREST CLASSIFIER WHEN USING DIFFERENT FOREGROUND MASK CLEANING PROCESSES

Method	Successful classification rate
Median filter	77.3%
Erosion and dilation	73.4%
Superpixel-based	80.1%

edges, slight variation in this value has no adverse effects. Eq. (1) selects superpixels to make up the new foreground mask $\hat{F}$ if they satisfy the criteria of containing a minimum density of pixels in the original foreground mask.

Table I compares the successful gesture classification rates, using the method in Section IV-E with a Random Forest classifier applied to 5 previous frames. Results are given for the superpixel-based method presented here, along with a median filter and an erosion and dilation approach. Fig. 6 shows an example of the superpixel-based method in operation. It is worth noting that a background subtraction method may fail if the hand is the same color and texture as the background—further explanation and examples are given in Section V.

B. Hand Contour Extraction

When there is no occlusion, e.g., starting when a hand first enters the frame at location e, contours can be extracted from the cleaned foreground mask. Given the set of pixels in an arm contour, A, the hand location and size can be taken as the circle $C(c, r)$ with center c and radius r satisfying

$$\arg\max_{c,r} \left(\alpha r^2 + d_E(e, c)\right) \Bigg|\begin{matrix} r_{\min} < r < r_{\max} \\ \text{and } C(c,r) \subset A \end{matrix} . \qquad (2)$$

In (2), α is a weighting constant, d_E is the Euclidean distance and $r_{\min}$ and $r_{\max}$ are the predefined minimum and maximum radii, chosen to be just below the smallest and above the largest expected palm sizes. This avoids accidentally selecting a finger or shoulder as a hand center. The hand contour is taken as the outline of A enclosed within a bounding box around C, resized relative to r and orientated with respect to the arm angle. Fig. 7 shows examples in two arm contours. This is similar to the method in [5], which attempts to maximize the hand circle radius and fit of a straight line to the arm, whereas Eq. (2) attempts to maximize the hand circle radius and distance to arm entry point. However, neither the proposed method for hand localization nor the method in [5] provide the accuracy we require when two hands are part of the same contour. In the event of such occlusions, an enhanced approach is needed, so optical flow is used to infer the movement of the occluded hand. First, sparse Lucas–Kanade optical flow [32] is calculated for each superpixel. This information is used to assign each superpixel to the contour it was in before the occlusion. If the hand is visible, its outline is extracted, resized and rotated as above. If not (i.e. when two separate arm contours overlap), then r is assumed constant and the positional change in c is computed from the flow within the contour containing it. The hand shape change is predicted using the embedding path described in Section IV-E.

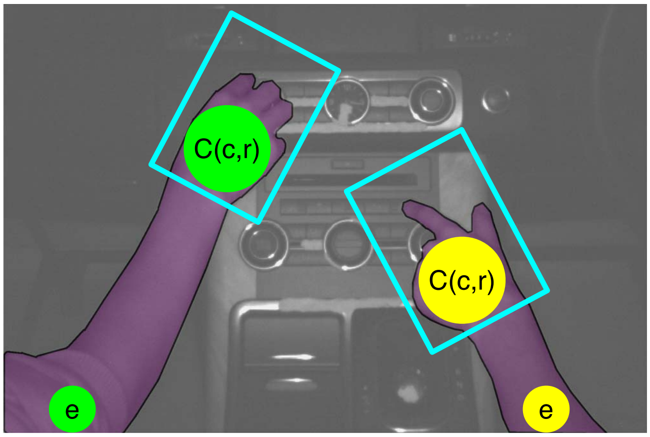
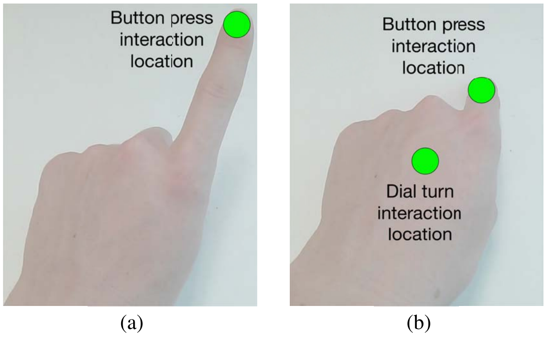


Fig. 7. Hand localization. Circles $C(c, r)$, bounding boxes (blue) and entry points e in two arm contours [purple—denoted A in (2)].


Button press interaction location
Button press interaction location
Dial turn interaction location

(a) (b)

Fig. 8. Hand interaction locations. (a) Hand with one interaction location. (b) Hand with two interaction locations.

C. Manifold Generation

Each hand shape can have one [see Fig. 8(a)] or more [see Fig. 8(b)] interacting areas, depending on the available control types. Before any hand contours are used, they are resized by finding the largest circle in their outline as in Eq. (2), and scaling appropriately. As right and left hands are symmetrical, we can generate one manifold with just left handed shapes (those of the driver in a right hand drive vehicle), and vertically flip any right hand (belonging to the passenger) prior to its embedding.

We choose to proceed with a shape manifold based method, rather than a global feature based approach such as [26], because the shape of the hand is linked to the area with which it interacts. With a properly constructed manifold, we are able to organize hand outlines such that those nearby will be both close in shape and have similar interacting locations. The works by Lee and his co-workers [7], [8] demonstrate how a well chosen manifold is able to represent a specific gesture. Cylindrical manifolds are used to represent grasping gestures, although other gestures, such as pointing, are not investigated. A disadvantage to requiring a separate manifold for each gesture type is that they need to be specified in advance, and this can be difficult if the underlying hand shape change is not as obvious as a grasp/ungrasp. Also, in our case, people can perform the same gesture in different ways, which would require either constructing many manifolds per gesture or a complicated embedding operation.

Here, we construct a single manifold to represent all hand shapes no matter which control types they interact with (e.g., pressing a button or turning a dial), and rely on a classifier to determine which gesture the path through the manifold represents. Note how we do not use a separate manifold for children (although we still include child test subjects in Section V). Given a suitable embedding scheme, knowledge of the occupants' ages does not add any useful information when determining which part of the hand is interacting with a control or the gesture being performed. Including this extra manifold in our pipeline would introduce the problem of deciding which side of the adult/child cutoff a hand is, particularly in borderline cases. It would also increase the amount of training data needed in both the manifold construction and shape change stages, in addition to the issues raised above.

We use the manifold learning method of [23] to create the hand shape manifold from our training hand set via the diffusion maps process. Briefly, this relies on a measure of difference between training shapes, which are arranged such that when this difference is low, shapes are close in the manifold, and vice versa. The manifold construction technique in [23] has not previously been applied to the task of modeling hands or gestures, and we introduce a difference measure designed to organize hand shape embeddings with respect to how they interact with controls.

In [23], the difference measure between two contours U and V is taken as the Sobolev $W^{1,2}$ norm, d_W, between their signed distance functions, $\mathcal{D}_U$ and $\mathcal{D}_V$ (which can be quickly computed from the original contours using the Fast Marching method [33]). This is given by [23] as

$$d_W(U,V)^2 = \|\mathcal{D}_U - \mathcal{D}_V\|^2 + \|\nabla\mathcal{D}_U - \nabla\mathcal{D}_V\|^2. \quad (3)$$

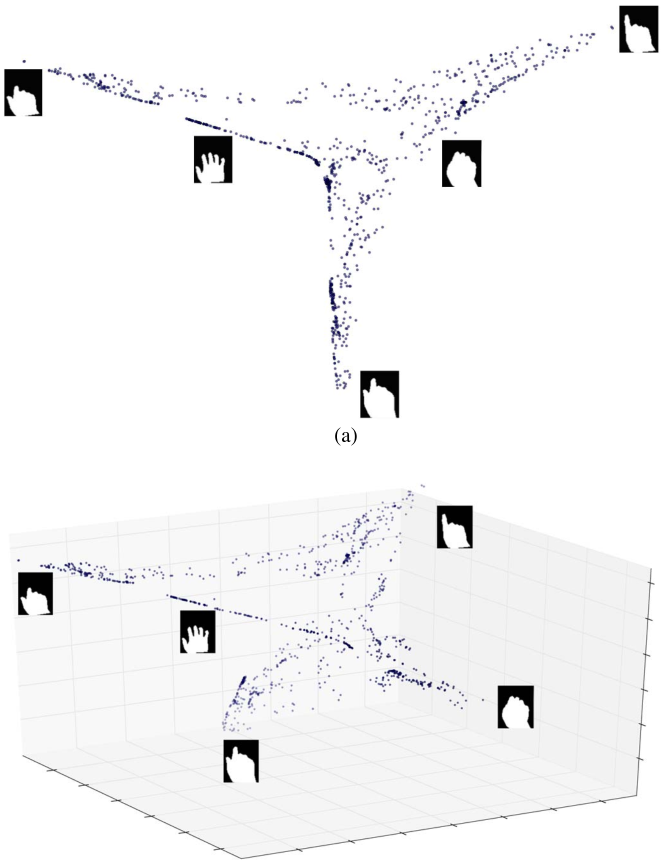
We wish to also take into account the distance between the hands' interaction locations, so the difference measure d_I is introduced as

$$d_I(U,V) = \sum_{t=1}^{\omega} \begin{cases} d_{E_t}(U,V) & \text{if both interaction types } t \text{ are annotated} \\ \Theta_t & \text{otherwise} \end{cases} \quad (4)$$

where d_{E_t} is the Euclidean distance between interaction locations of type t, Θ_t is the maximum distance between hand interaction locations of type t, and ω is the number of interaction types. We then combine d_W and d_I to give d_{WI}, which is the difference measure used to construct the manifold

$$d_{WI}(U,V) = d_W(U,V) + \beta d_I(U,V) \quad (5)$$

where β is a weighting constant. Eq. (3) essentially places similarly shaped contours close to each other on the manifold. If the manifold is generated with just Eq. (3), then embedding methods that rely on combining information from multiple training samples can result in the interaction location being calculated as outside of the hand contour. Eq. (4) is introduced to ensure contours with similar interaction locations have close embeddings, and thus alleviates this concern.


(a)

(b)

Fig. 9. Training sample embeddings in manifolds generated by the diffusion maps process with the difference measure d_{WI}. (a) Two-dimensional manifold. (b) Three-dimensional manifold.

Now we have a difference measure, the manifold is constructed using Eq. (5). Given the training set Ω containing μ samples, we construct the difference matrix M using a Gaussian kernel equipped with d_{WI} with σ being approximated by the median difference between all μ samples in T

$$M_{ij} = \exp\left[-\frac{d^2_{WI}(\Omega_i,\Omega_j)}{2\sigma^2}\right]. \quad (6)$$

As a way of denoising the manifold, only the largest $\mu/10$ entries in each row of M are retained. This is then made symmetrical by adding M to its transpose, then normalized via the Beltrami normalization process. We call this normalized difference matrix $\bar{M}$. Eigen decomposition is then performed on $\bar{M}$, with ν-dimensional manifold being taken as the ν eigenvectors corresponding to the ν largest eigenvalues.

Fig. 9 shows example 2D and 3D manifolds generated by this process. In our experiments, we found that using a manifold with a dimension higher than 3 provided negligible improvements.

D. Sample Embedding

Previous approaches to embedding a sample into a shape manifold (outlined in [23]) have either been a nearest-neighbor

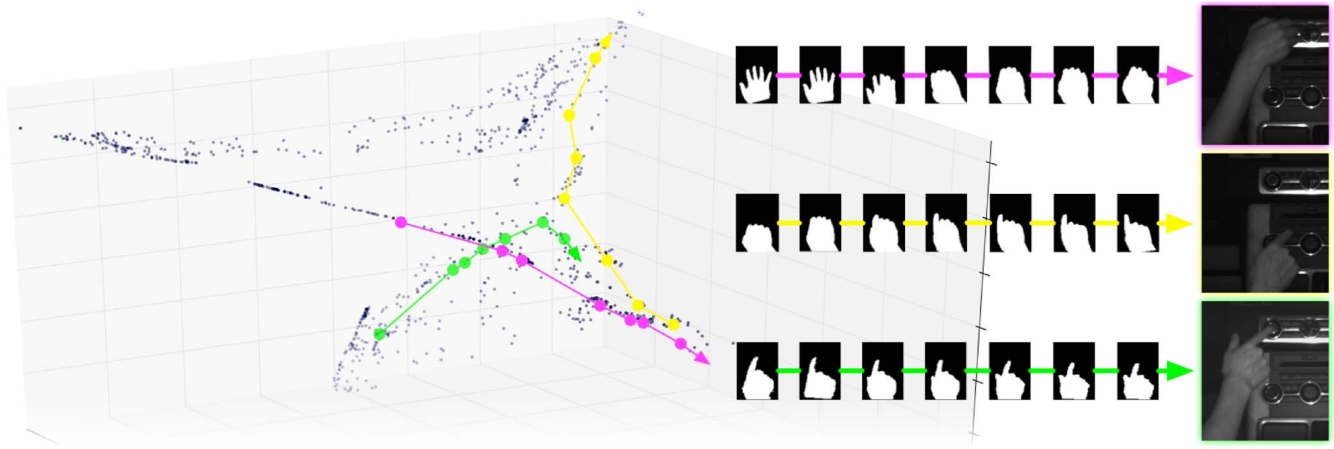


Fig. 10. Some sample paths through a 3-D manifold. The top row of images correspond to a clockwise dial turn. The middle row corresponds to a button press with the index finger, and the bottom row shows how finer details such as a thumb extending can be determined.

embedding, which can lack accuracy, or via the Nyström extensions method. This requires calculating the difference between the sample to be embedded and all the training samples, and is thus unsuitable for a real time application. We require a method that is more robust to noise than the nearest-neighbor embedding, yet still able to run in real time, and so we use a k-nearest neighbor embedding.

Along with an ν-dimensional training set embedding, we also store the possible hand interaction locations for each sample in the training set. Now, given a sample S, we find the $\nu + 1$ nearest neighbors of the sample in the training set, K_S, and the reciprocals of their distances from S, denoted R_{K_S} via the difference measure used to create the manifold [d_{WI} in Eq. (5)]. We also find the embeddings of the training samples K_S, denoted Φ_{K_S}, and their interaction locations for each control type t, denoted $\Psi^t_{K_S}$. The sample embedding Φ_S and sample interaction location Ψ^t_S of S are then

$$\Phi_S = \hat{R}_{K_S} \cdot \Phi_{K_S} \quad \text{and} \quad \Psi^t_S = \hat{R}_{K_S} \cdot \Psi^t_{K_S} \tag{7}$$

where $\hat{R}_{K_S}$ is the normalized form of R_{K_S}. To enable this process to run in real time with a large training shape set, a vantage-point (or metric) tree [34] is constructed during the initialization stage using d_{WI} and the training samples. This then provides a fast k-nearest neighbor search when queried.

E. Path Classification

At a given moment, it is necessary to determine which action is being performed. We would like to use the path through the manifold to make this decision. To build our path classifier, we first gather some example gestures, for example pressing a button with the index finger or turning a dial clockwise.

Given a sample gesture g that consists of f frames, the manifold embedding g_i for each frame i is calculated. As gestures can occur at different rates, we use the training gestures to generate many training samples of different lengths with which to compare testing samples. In each generated sample, some embeddings are randomly left out to allow for noisy hand shape data being fed into the manifold, for example by a bad hand location, dropped frame or failed segmentation. The set of generated samples G from gesture g is defined as

$$G_g = \{P(k, j, b) | l_1 \leq j \leq l_2, \; 0 \leq k \leq f - j\} \tag{8}$$

where l_1 and l_2 are the minimum and maximum sample lengths respectively in frames to be classified. $P(k, j, b)$ denotes the path from g_k to g_{k+j}, with each coordinate having probability b of being removed and replaced by a Hermite interpolation of its neighbors. This step prevents the classifier over-relying on a single coordinate. In our case, we found $b = 0.2$ to be appropriate. In the event of an occlusion, the change in hand shape—and hence its interaction location—can be estimated by finding the nearest neighbor (with the gesture type as indicated by the classifier) in the generated set to the testing sample. The sample that this nearest neighbor was generated from can then be used, stretched as necessary. Fig. 10 gives three example hand shape change embedding paths.

We evaluate the classification success rates of nearest-neighbor, radial basis function support vector machine (SVM), decision tree, and random forest classifiers, as well as a comparison against dynamic time warping (DTW) on the non-generated example gestures. Fig. 11(a) shows these classifications on whole single interactions, from the time the hand enters the frame until the control is pressed. Fig. 12 shows two video examples. The embedding path is subsampled, and the number of sample points varied. Here, the nearest-neighbor is the best choice as it performs well at both low and high sample rates.

Fig. 11(b) shows the results of just classifying the previous few frames at a random interruption before the control is activated to simulate an occlusion. This interruption is restricted to occurring between the halfway point in the sequence and when the control is used. As can be seen in Fig. 12, the earlier this interruption is made the more difficult it is to make a correct classification. When looking at a small number of previous frames (less than 10 frames at 60 fps, i.e. 0.17 s), the random forest classifier performs best, but it is reliably outperformed by the nearest-neighbor and SVM classifiers when looking further back (greater than 10 frames). In both cases, 5-fold cross validation was performed 100 times on a verification set

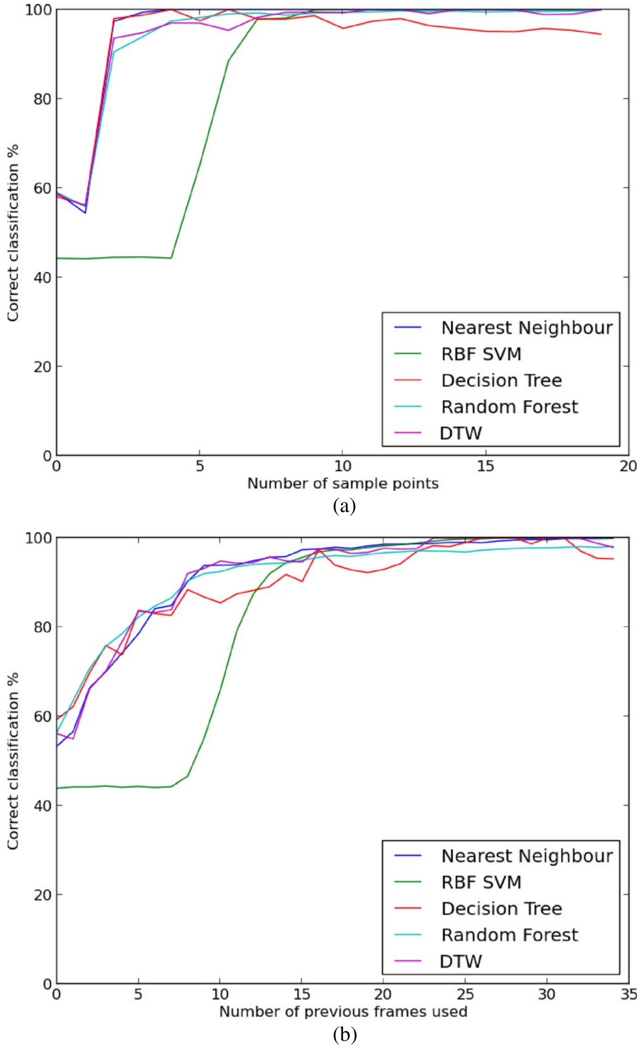


Fig. 11. Comparison of classifiers on the whole interaction and a set number of previous frames. Footage was taken at 60 frames/s. (a) The whole interaction (hand entering the frame until control activation) is classified, with a variable number of sampling points. (b) The interaction is classified at a random interruption after the hand has entered the frame. The number of previous frames used before the interruption is varied.

containing 25 button presses, dial turn right, and dial turn left interactions from one subject.

F. Interaction Confidence Score

Now, when a control is used, each hand can be scored based on how well the gesture agrees with the control type and the distance between the hand's interacting point and the control. Given a control activation at location x, this score is defined as

$$\text{Interaction score} = \underbrace{\tau^{\gamma}}_{a} \underbrace{\left(\frac{k}{f-j}\right)^{\delta}}_{b} \underbrace{\frac{1}{d_E(x,l)}}_{c}. \tag{9}$$

In (9), τ is the classifier confidence score that the current gesture being performed is used to interact with the type of control being activated. The Euclidean distance is given by d_E, and l is the interaction location of current hand shape with the queried control type from Eq. (7). The weighting constants γ and δ are learned via a grid search from a validation set. For this search, the validation set is split randomly in half, with the condition enforced that all control types must appear in each half. The first half is taken for training and the second half for testing, with the mean taken over five runs. Whichever hand obtains the highest interaction score is determined as the hand interacting with the control at location x. Part a of Eq. (9) is the weighting of the gesture contribution, and prioritizes a hand which is performing a gesture likely to be interacting the control type being used. Part b corresponds to how far through the gesture path the current hand shape embedding is (or the embedding of a predicted hand shape change in the event of an occlusion). As all hand shape change training samples end with a control interaction, this prioritizes a hand that is more likely to be currently using a control. Part c prioritizes a hand with its interaction location close to the current control being activated. The form of Eq. (9) was initially chosen as it provides a simple way of combining the interaction location and hand shape change information, where only two weighting parameters need to be found. It is less susceptible to overfitting with a small validation set and outperformed an SVM applied to the same problem.

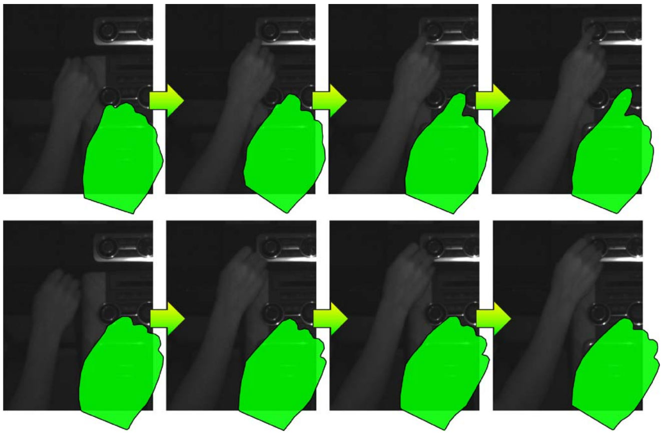


Fig. 12. Frames taken from two of the video sequences used to generate Fig. 11(b) with the hand shapes enlarged for clarity. The top row is part of a button press with the index finger, and the bottom row is part of a clockwise dial turn. The gesture classification is made at a random point during the sequence.

V. Experiments

To construct the hand shape manifold, 900 images from one subject are used, as introduced in Section III with examples shown in Fig. 3. Due to the diffusion maps process, more could be added with little effort to include additional poses as required. For the gesture and interaction training and verification sets, 25 interactions with each control type (button, dial and touchscreen) from the same participant are used. For the validation set required by Eq. (9), 50 complex interactions are used. The main testing dataset consists of 1544 control interactions by 10 adults and one child with a wide variety of skin tones, clothing and jewelery. Table II gives the number of times items of clothing and jewelery occur in the test set. Of these, 596 are daytime and 603 night time with NIR illumination and filtering, and 345 are daytime at normal greyscale. Our data is captured at 60 fps, so when testing at 30 fps one out of every two frames

TABLE II
NUMBER OF TIMES ITEMS OF CLOTHING AND JEWELERY APPEAR ON A HAND INTERACTING WITH A CONTROL IN THE TEST SET

Clothing/jewellery	Count
Watch	127
Ring	151
Glove	21
Sleeve	186
Rolled up sleeve	253

TABLE III
SUCCESSFUL DECISION PERCENTAGE ON OUR DATASET AT 30 FRAMES/S. *C.H.* DENOTES THAT THE METHOD JUST USES THE CLOSEST HAND WHEN MAKING THE DECISION. DAY + NIR REFERS TO THE DAYTIME FOOTAGE WITH NIR ILLUMINATION AND FILTERING, NIGHT + NIR REFERS TO THE NIGHTTIME FOOTAGE WITH NIR ILLUMINATION AND FILTERING, AND DAY − NIR REFERS TO DAYTIME FOOTAGE WITH NO ILLUMINATION OR FILTERING. THE NUMBER OF INTERACTIONS FOR EACH CAMERA SETUP IS IN BRACKETS

	Day + NIR	Night + NIR	Day - NIR	Average
Method	(596)	(603)	(345)	(1544)
Proposed	98.5%	96.8%	95.0%	97.1%
Proposed *C.H.*	79.3%	88.9%	87.8%	85.0%
[5] *C.H.*	61.1%	68.5%	68.4%	65.6%

is ignored and three out of every four ignored when testing at 15 fps.

We compare the effectiveness of the proposed method with and without knowledge of hand interaction locations. When making a decision without this information, we choose the hand that is closest to the control when it is activated (marked as *C.H.* for Closest Hand in the results table). We also use this same criteria applied to McAllister's driver hand tracking method [5]. Table III shows these results, clearly indicating the substantial improvement obtained by including interaction location information over the naive closest hand approach. This improvement is consistent across the day time footage with infra-red illumination (98.5% compared to 79.3%), night time footage with infra-red illumination (96.8% compared to 88.9%) and the unilluminated day time footage (95.0% compared to 87.8%). On average, across all test footage, the driver or passenger is chosen correctly 85.0% of the time when just looking for the closest hand, and this increases to 97.1% when including the interaction location and gesture confidence in the decision making process.

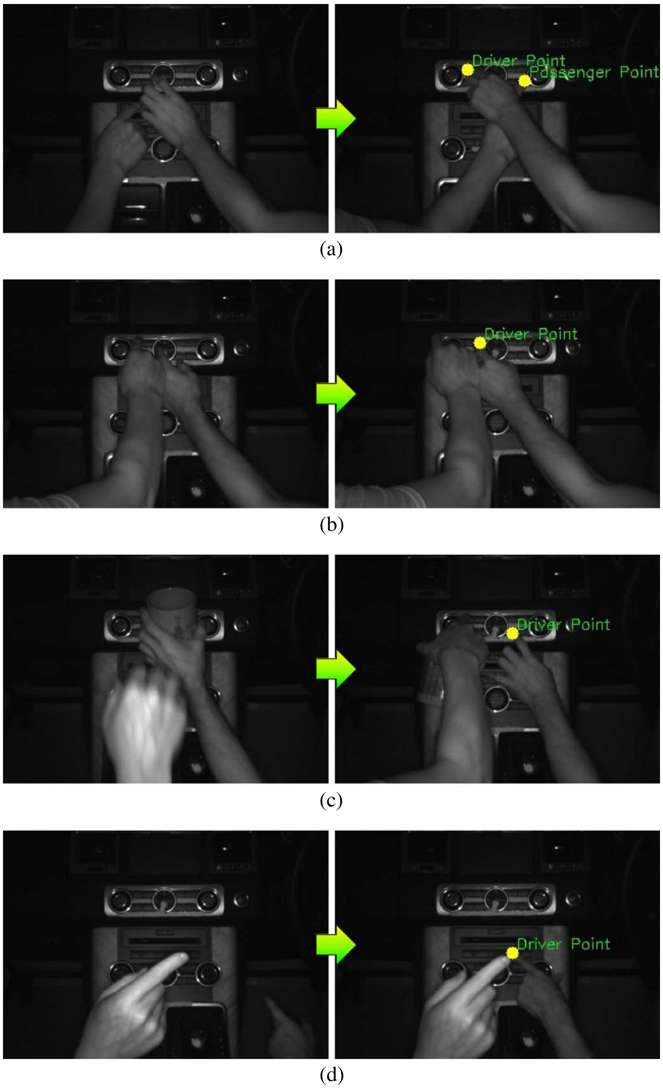
Table III also highlights improvements in the hand segmentation and locating components of the proposed method over [5]. For example, when just taking the closest hand to make a decision, improved results are observed across all footage types. [5] makes the correct decision for 65.6% of the interactions in the test set, whereas the proposed method scores 85.0%.

There are two main reasons for this improvement. The first is better hand segmentation due to the use of a more advanced background subtraction method and the preserving of edge information. The second is how the two approaches handle occlusions. Using the motion of the visible part of the arm to adjust the position of an occluded hand results in a more accurate hand location than the distance transform approach applied to the foreground mask in [5].

Table IV details the effectiveness of the proposed method at different frame rates. Across all test sequences, the proposed method makes the correct decision 95.9% of the time on footage provided at 15 fps, and this rises to 97.1% at 30 fps and 97.7% at 60 fps. Increasing the frame rate results in a higher successful decision score as the gesture classifier performs better with more hand shape samples, as illustrated in Fig. 11.

TABLE IV
SUCCESSFUL DECISION PERCENTAGE OF THE PROPOSED METHOD AT DIFFERENT FRAME RATES

	Day + NIR	Night + NIR	Day - NIR	Average
Frame rate	(596)	(603)	(345)	(1544)
15 fps	93.8%	97.2%	97.4%	95.9%
30 fps	98.5%	96.8%	95.0%	97.1%
60 fps	99.2%	98.7%	93.6%	97.7%



(a)

(b)

(c)

(d)

Fig. 13. Proposed method working on examples from our dataset. In each case, the right image is captured a few frames after the left. (a) Occlusions. (b) Collisions. (c) Foreign objects. (d) Pointing at other cues.

Fig. 13 demonstrates the proposed method handling complex situations such as hand crossing and occlusion [see Fig. 13(a)], the driver and passenger contesting over a control with contact occurring [see Fig. 13(b)], objects being passed around [see Fig. 13(c)] and distracting hands [see Fig. 13(d)].[1]

[1] See the supplementary material for video examples.

TABLE V
SUCCESSFUL DECISION PERCENTAGE IN DIFFERENT CABINS, AS INTRODUCED IN FIG. 5. FOR THE SALOON CAR IN THIS EXPERIMENT, THE NORMAL CAMERA POSITION WAS CHOSEN [SEE FIG. 5(C)]

Method	**SUV 1** (176)	**SUV 2** (178)	**Saloon** (246)	**Average** (600)
Proposed	94.9%	94.9%	91.1%	93.3%
Proposed *C.H.*	91.6%	92.0%	85.4%	89.2%
[5] *C.H.*	80.3%	73.9%	82.9%	79.5%

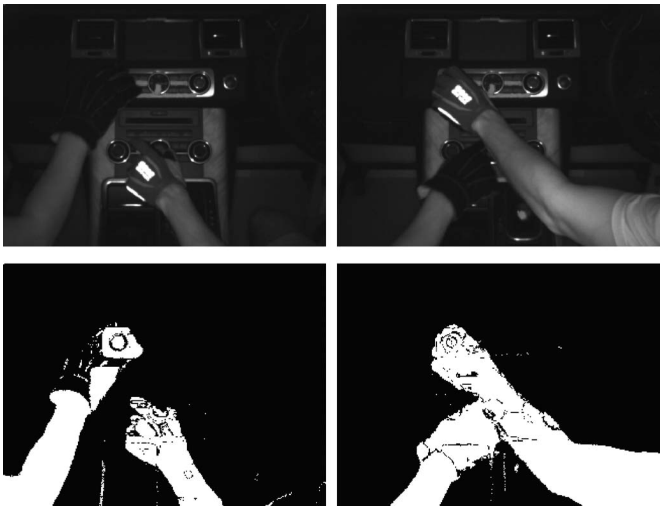


Fig. 14. Top row shows nighttime NIR footage with gloves, and the bottom row shows the corresponding foreground masks. Note how the black glove does not show up clearly in the foreground mask when it is over an area of the same color and texture.

Table V shows the results when using the above methods in the additional vehicle cabins introduced in Fig. 5. A similar trend is observed to the main test footage—the proposed method achieves average 93.3% with the interaction confidence score and 89.2% without. The closest hand implementation of [5] scores an average rate of 79.5%. We also trialled a more forward camera mount in the Saloon [see Fig. 5(d)], but results were not as good since the above methods scored 87.4%, 78.8% and 82.3% on 198 test interactions. This can be attributed to a number of factors, including the hand shape manifold and gesture training relying on hand shapes taken from a different viewpoint. Also, the hand appears in the frame later (which provides less gesture information) and with less, if any, of the arm visible (which can cause a less accurate optical flow calculation).

In general, if the hand tracking succeeds, the correct decision is made. However, a failure in the hand segmentation stage can result in an incorrect decision. An example is given in Fig. 14, where a texture-less glove, the same color as the background, is not clear in the foreground mask. This is an inherent disadvantage to using a background subtraction based method. Nevertheless, as in [4] and [5], we found background subtraction to be the preferred choice for hand segmentation. Edge based methods that use training shape examples [35] struggle to reliably segment the large variety of possible hand configurations, particularly against a backdrop of a control panel with well defined edges. Additionally, segmentation techniques that rely on skin color (used in [2], [19], [20]) are clearly not suitable for use with the greyscale images provided by NIR cameras.

There are two basic approaches to background subtraction. The first is to have a static model, learned from example images with different lighting conditions, and new frames are compared against this model. This approach functions well when there is a sudden large change in the scene lighting (when entering or leaving a tunnel, for example), but cannot handle an evolving background. This makes it unsuitable for an automotive environment, as the background will change throughout the life of the vehicle, degrading the performance of the static model over time. The second approach (used here) is to have a constantly evolving model of the background. In the event of a sudden change in lighting, the background model will take a few frames to be updated, resulting in a foreground mask containing errors for this period. This approach is still preferable for our use case, due to the background changing throughout the life of the vehicle. This constantly evolving approach can still handle small or slower changes to illumination, and the performance under these conditions can be improved by maintaining more control over the scene lighting. One example of how this can be achieved is with more powerful illumination, which makes any natural change less significant.

Assuming a reasonable hand segmentation, if just one hand is in view then the decision is trivial and a naive closest hand method is sufficient—no knowledge of interaction location or the hand shape change is necessary. Similarly, this information is not needed if both hands are in view and not close. The interaction location becomes important when the hands get very close (if it is not possible to accurately segment one hand from the other) and when contact is made. The hand shape change information is required when significant occlusion occurs.

VI. CONCLUSION

We presented a method that determines the interaction of the driver and the passenger with the controls on the centers panel of a car—the first attempt in the literature to distinguish between individual controls. This will enable the driver and passenger to interact with both sides of a full view touchscreen, and allow manufacturers to remove duplicate controls.

The proposed method uses a background subtraction algorithm, followed by a novel use of a shape manifold to determine which part of the hand is interacting with a control, as well as providing gesture information. An evaluation was performed on a challenging dataset, and inclusion of information obtained from this manifold provided improved results over a closest hand approach.

We suggest a number of possibilities for future work. One is to investigate if the proposed method is capable of recognizing a wider variety of gestures that incorporate motion as well as hand shape change (e.g. swipe to open the sunroof). Another possibility is to focus on safety. For example, a distracted driver might interact with a control in a different manner to an alert driver, and it would be interesting to see whether the hand shape change considered in this paper can be a reliable indicator for this. Such an indicator can be a trigger to prompt a distracted driver to refocus on the road.

ACKNOWLEDGMENT

Due to a confidentiality agreement with Jaguar Land Rover, the data supporting this article is not openly available.

REFERENCES

[1] T. J. Schousek, "Vehicle occupant restraint with seat pressure sensor," U.S. Patent 5 474 327, Dec. 12, 1995.

[2] C. Zhao, B. Zhang, J. He, and J. Lian, "Recognition of driving postures by contourlet transform and random forests," *IET Intell. Transp. Syst.*, vol. 6, no. 2, p. 161, 2012.

[3] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *Proc. CVPR*, 2012, pp. 1862–1869.

[4] R. Crespo, I. Diego, C. Conde, and E. Cabello, "Detection and tracking of drivers hands in real time," *Progr. Pattern Recognit., Image Anal. Comput. Vis. Appl.*, vol. 6419, pp. 212–219, 2010.

[5] G. McAllister, "Tracking drivers hands using computer vision," in *Proc. Int. Conf. Syst. Man Cybern.*, 2000, pp. 1388–1393.

[6] S. Y. Cheng and M. M. Trivedi, "Vision-based infotainment user determination by hand recognition for driver assistance," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 759–764, Sep. 2010.

[7] C. Lee and S. Park, "Tracking hand rotation and grasping from an IR camera using cylindrical manifold embedding," in *Proc. ICPR*, 2010, pp. 2612–2615.

[8] C. Lee, S. Chun, and S. Park, "Tracking hand rotation and various grasping gestures from an IR camera using extended cylindrical manifold embedding," *Comput. Vis. Image Understand.*, vol. 117, no. 12, pp. 1711–1723, 2013.

[9] F. Vicente *et al.*, "Driver gaze tracking and eyes off the road detection system," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2014–2027, Aug. 2015.

[10] R. O. Mbouna, S. G. Kong, and M. G. Chun, "Visual analysis of eye state and head pose for driver alertness monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1462–1469, Sep. 2013.

[11] J. Nuevo, L. M. Bergasa, M. A. Sotelo, and M. Ocana, "Real-time robust face tracking for driver monitoring," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2006, pp. 1346–1351.

[12] A. Dasgupta, A. George, S. L. Happy, and A. Routray, "A vision-based system for monitoring the loss of attention in automotive drivers," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1825–1838, Dec. 2013.

[13] I. Garcia, S. Bronte, L. M. Bergasa, J. Almazan, and J. Yebes, "Vision-based drowsiness detector for real driving conditions," in *Proc. IEEE Intell. Veh. Symp.*, 2012, pp. 618–623.

[14] I. G. Daza *et al.*, "Drowsiness monitoring based on driver and driving data fusion," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 2011, pp. 1199–1204.

[15] L. M. Bergasa and J. Nuevo, "Real-time system for monitoring driver vigilance," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 63–77, Mar. 2006.

[16] S. S. Huang, "Discriminatively trained patch-based model for occupant classification," *IET Intell. Transp. Syst.*, vol. 6, no. 2, pp. 132–138, 2012.

[17] Z. Gao and L. Duan, "Vision detection of vehicle occupant classification with Legendre moments and support vector machine," in *Proc. IEEE Int. Congr. Image Signal Process.*, 2010, pp. 1979–1983.

[18] S. Y. Cheng and M. M. Trivedi, "Human posture estimation using voxel data for 'smart' airbag systems: Issues and framework," in *Proc. IEEE Intell. Veh. Symp.*, 2004, pp. 84–89.

[19] C. Tran and M. Trivedi, "Towards a vision-based system exploring 3D driver posture dynamics for driver assistance: Issues and possibilities," in *Proc. IEEE Intell. Veh. Symp.*, 2010, pp. 179–184.

[20] Z. Feng *et al.*, "Features extraction from hand images based on new detection operators," *Pattern Recognit.*, vol. 44, no. 5, pp. 1089–1105, 2011.

[21] A. Mittal, A. Zisserman, and P. Torr, "Hand detection using multiple proposals," in *Proc. BMVC*, 2011, pp. 1–11.

[22] H. Choi, B. Paulson, and T. Hammond, "Gesture recognition based on manifold learning," in *Proc. Struct. Syntactic Statist. Pattern Recognit. Joint IAPR Int. Workshop*, 2008, pp. 247–256.

[23] P. Etyngier, F. Segonne, and R. Keriven, "Shape priors using manifold learning techniques," in *Proc. ICCV*, 2007, pp. 1–8.

[24] R. Coifman *et al.*, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proc. Nat. Acad. Sci.*, vol. 102, no. 21, pp. 7426–7431, 2005.

[25] S. Lafon and A. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1393–1403, Sep. 2006.

[26] T. Kim, S. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *Proc. IEEE CVPR*, 2007, pp. 1–8.

[27] L. Liu and L. Shao, "Learning discriminative representations from RGB-D video data," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 1493–1500.

[28] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Proc. IEEE CVPR Workshop*, 2012, pp. 38–43.

[29] A. Schick, M. Bauml, and R. Stiefelhagen, "Improving foreground segmentations with probabilistic superpixel Markov random fields," in *Proc. CVPR Workshop*, 2012, pp. 27–31.

[30] R. Achanta, A. Shaji, K. Smith, and A. Lucchi, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2281, Nov. 2012.

[31] C. Ren and I. Reid, "gSLIC: A real-time implementation of slic superpixel segmentation," Tech. Rep., Dept. Eng. Sci., Univ. Oxford, Oxford, U.K., 2011.

[32] S. Baker and I. Matthews, "Lucas–Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, vol. 56, no. 3, pp. 221–255, Feb. 2004.

[33] J. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Nat. Acad. Sci.*, vol. 93, no. 4, pp. 1591–1595, Feb. 1996.

[34] J. Uhlmann, "Satisfying general proximity/similarity queries with metric trees," *Inf. Process. Lett.*, vol. 40, no. 4, pp. 175–179, 1991.

[35] T. Cootes, E. Baldock, and J. Graham, "An introduction to active shape models," *Image Process. Anal.*, pp. 223–248, 2000.

Toby Perrett received the B.Sc. degree in mathematics and the M.Sc. degree in computer science in 2011 and 2012, respectively, from University of Bristol, Bristol, U.K., where he is currently working toward the Ph.D. degree in computer vision under the supervision of Prof. Mirmehdi.

His research interests include computer-vision-based approaches to vehicle occupant monitoring.

Majid Mirmehdi received the B.Sc. and Ph.D. degrees in computer science from the City University London, London, U.K., in 1985 and 1991 respectively.

He is currently a Professor in computer vision with the Department of Computer Science, University of Bristol, Bristol, U.K., where he is the Graduate Dean and Faculty Graduate Education Director in the Faculty of Engineering. His research interests include natural scene analysis and medical imaging, and he has more than 180 refereed conference and journal publications in these and other areas.

Dr. Mirmehdi is a Fellow of the International Association for Pattern Recognition. He is Editor-in-Chief of *IET Computer Vision* and an Associate Editor of *Pattern Analysis and Applications*. He is a member of the IET, and he serves on the Executive Committees of the British Machine Vision Association and the IET Vision and Imaging Network.

Eduardo Dias received the M.S. degree in electronics and telecommunications engineering from University of Aveiro, Aveiro, Portugal, in 2009.

He is a Research Engineer in the human–machine interaction area of the Research and Technology Department, Jaguar Land Rover, Coventry, U.K. From 2009 to 2012, he pursued his interest in the biomedical research area as a Researcher at the Institute of Electronics and Telematics Engineering of Aveiro (IEETA), Aveiro, investigating automated endoscopic capsule analysis techniques and movement quantification during epileptic seizures. In 2012, he decided to shift his career into the automotive industry and applied for a Jaguar Land Rover Research role, where he is currently working on future vehicle technologies.