On Multi-modal Fusion for Freehand Gesture Recognition

Monika Schak and Alexander Gepperth

Fulda University of Applied Sciences, 36037 Fulda, Germany
{monika.schak,alexander.gepperth}@cs.hs-fulda.de

Abstract. We present a study of multi-modal freehand gesture recognition relying on three sensory modalities. The modalities are RGB images, depth data, and acceleration data from an IMD attached to the hand. Based on a new self-recorded dataset, we initially establish the ability of a deep Long Short-Term Memory (LSTM) network to correctly classify individual data streams from each modality. Notably, classifying the IMD stream alone generates very good results already. In addition, we investigate two different strategies of multi-modal fusion, since there is no agreement in the literature as to which strategy is preferable. Combining the modalities leads to better recognition performance. Most importantly, fusion considerably improves ahead-of-time classification, i.e., gesture class estimates before sequences are completed, for classes that are difficult to classify on their own.

Keywords: LSTM · Freehand gesture recognition · Multi-modal Fusion · Ahead-of-time classification

1 Introduction

Freehand gesture recognition The scientific context of this article is multi-modal freehand gesture recognition, see Fig. 1. This means classifying hand gestures that are extended in time (in contrast to hand poses, which are taken at a single point in time) into several distinct categories or classes.

Typical sensors To this effect, the hand is observed by one or more sensors, typical choices of which are RGB cameras, infrared cameras, depth sensors (using either stereo, time-of-flight, or structured-light technologies). Less commonly used types of sensors are IMDs (inertial measurement devices), which record accelerations, and by integration: velocities. IMDs are small, cheap, and reliable, but need to be physically attached to the hand, which is not always feasible in scenarios for hand gesture recognition.

Multi-modal processing Each sensor gives rise to a separate data stream, which is generally termed a *sensory modality*. Since all sensors measure physically distinct quantities, each sensory modality may contain unique and independent information. On the other hand, since all sensors observe the same thing

Fig. 1: Freehand gesture recognition as treated in this article. Left: experimental setup using an Orbbec Astra (RGB camera and depth sensor) and an IMD (inertial measurement device). The latter is attached to the hand using red tape and transmits its measurements via a serial cable. It will later be replaced by a more compact wireless version. Right: layout of the data used for training machine learning models. Each gesture represents a single data sample, which is composed of several successive frames, each of which is represented by a one-dimensional tensor (or feature vector), thus making the whole database of gestures a 3D structure.

(i.e. the hand in our case), sensory modalities contain at least partially correlated information.
The goal of multi-modal information processing is, in general, to exploit the (partial) independency and complementarity of sensory modalities to obtain more precise and reliable observations.

Sequence classification terminology The gesture recognition problem is a so-called sequence classification problem, where the basic entities to be classified are streams, or *sequences*, of sensory measurements or quantities derived from them. In the context of a database of sequences for training machine learning algorithms, we often denote a single sequence by the term *sample*. Sequence elements are generally called *frames*, which in our setting are one-dimensional tensors (or feature vectors) derived from sensory measurements. Please regard Fig. 1 (right) for a diagram of the basic data layout we use in this article.

Sequence classification Particular properties of sequence classification problems are that the individual frames become available one after the other (in contrast to image classification where all pixels are available at once) and that sequences need not have a common length. This requires machine learning strategies that are adapted to these requirements. A prominent example of this are LSTM neural networks [9], which are used for sequence classification in this study.

Overview In our case, we try to obtain an estimate of the gesture's category by evaluating three sensory modalities: IMD (acceleration) data, RGB images, and depth images. Since each modality represents a temporally extended sequence of individual measurements, we use deep Long Short-Term Memory (LSTM) networks for inferring the class of the hand gesture that is performed. This article first investigates the individual (uni-modal) accuracy achievable when relying on a single modality only. Then, we examine multiple possibilities for combining the three modalities in order to obtain higher robustness or classification accuracy. Lastly, we perform two different types of look-ahead classification and the effect multi-modal fusion can have on it.
Additionally, we present a dataset of freehand gestures consisting of pre-processed data from three different sensors: an RGB camera, a depth camera, and a 6-axis acceleration sensor attached to the back of the user's hand. The aforementioned dataset is used for all the experiments described in this article.

Application context Such a system that allows the user to interact using freehand gestures, can, for example, be found in the scope of 3D object manipulation or augmented reality. To improve the ability of the before-mentioned system to correctly classify freehand gestures even as they vary due to different angles of the user towards the camera or entirely different users, we suggest the use of a classification system that combines the classification of multiple sensor streams into one joint classification result.

1.1 Related Work

Multi-sensory integration or fusion is a common concept in neurophysiological and psychological research [1, 2] and has been confirmed by many experiments. An interesting fact is that, under certain circumstances, human brains perform the combination of several sensory modalities in a probabilistically well-founded fashion [7]: when assuming that each modality is corrupted by Gaussian noise of distinctive variance, each modality is apparently weighted by a factor that is inversely proportional to its noise variance, which implements the theoretically optimal (MAP) estimate in this situation. An overview of the probabilistic foundations of such set-ups is given in [8].
On the computational side, there are many studies of multi-modal gesture or activity recognition. There is no agreement on the best way to perform multi-modal fusion, rather a variety of possibilities whose advantages and difficulties depend on the dataset and particular task at hand. In [18], late fusion is applied on a dataset for human fall analysis consisting of four modalities (RGB, depth, skeleton, and acceleration). Here, the final score is produced by searching the maximum output score provided by single modality based classifiers.
Another approach [5] uses a collaborative representation classifier to combine the classification outcomes for two modalities (depth and acceleration) using Dempster-Shafer theory to improve human action recognition. A different fusion method at the classifier level is the non-linear combination method of using a Random Decision Forest [17].

A very commonly used technique for late fusion is softmax score fusion [11], where the outputs of multiple classifiers are transformed to probability scores by a softmax layer and then combined using sum rule, product rule, or max rule. An alternative technique is feature fusion [10], where features from fully-connected layers are combined and then fed to any classifier, e.g. linear support vector machines or kernel extreme learning machines.
Early fusion happens at the data level where incoming data from sensors is combined without further preprocessing [12]. Feature level fusion is another kind of early fusion, which first extracts features from raw data and then proceeds to fuse those features before classification [5] by using a collaborative representation classifier.

1.2 Goals and contributions

Since there does not seem to be a consensus in the scientific literature about the best way to conduct multi-modal recognition tasks, this article makes the following novel and relevant contributions:

- We investigate the usefulness of wearable IMDs for gesture recognition, which are cheap and reliable.
- We conduct a computational case study of multi-modal (sequence) recognition, investigating and comparing several strategies for combining sensory modalities.
- We present a new result on ahead-of-time classification, showing that multi-modal fusion markedly improves the ability to correctly identify gestures before they are even completed.
- We publish a new publicly available benchmark for multi-modal freehand gesture recognition that contains 2.660 gestures from 7 classes that are observed with IMD, RGB and depth sensors.

2 Methods

2.1 Gesture classes

We choose seven different free-hand gestures for our dataset. All of them can be performed with only one hand. The seven gestures are described as follows:

- **Agree:** Move the thumb up.
- **Disagree:** Move the thumb down.
- **Pinch Out:** Move the thumb and index finger together as if zooming out on a touch device.
- **Pinch In:** Move the thumb and index finger apart as if zooming in on a touch device.
- **Select:** Move the palm towards the camera.
- **Swipe Left:** Swipe from right to left with the whole hand.
- **Swipe Right:** Swipe from left to right with the whole hand.

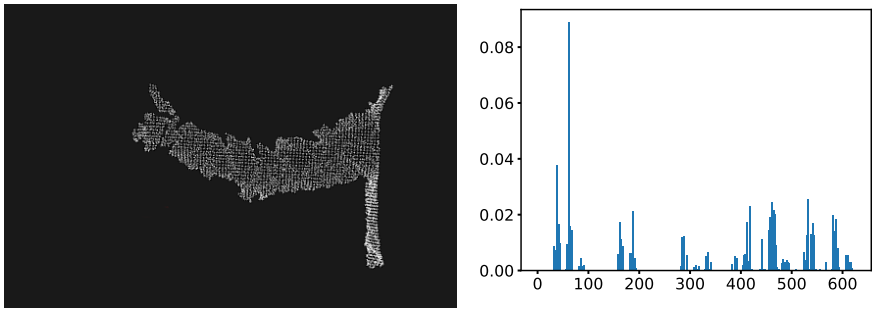

Fig. 2: Example of a point feature histogram (right) corresponding to one frame of a "Thumbs Up" gesture (left: point cloud from which the histogram was computed).

2.2 Dataset

The dataset contains 380 recordings of each of the 7 gesture classes, all coming from a single person, totaling to 2660 recordings. This is consistent with a scenario where a device is predominantly used by a single (or a few) individual(s) to whom it has been specifically adapted. This approach ensures that the conducted gestures are performed correctly and consistently across the dataset, and also that the variability of each gesture class is not excessive. Each gesture sample, irrespectively of its class, lasts for two seconds.

RGB data To record RGB data, we use one of the two streams provided by an Orbbec Astra 3D sensor. It simultaneously sends a stream of 800x600 RGB images and a stream of 640x480 depth images which are converted to point clouds. After cropping the images to obtain only the part of the image in which the hand is visible, we reduce the size of the images to 72×48 pixels. Afterwards, we calculate the histogram of oriented gradients (HOG) descriptor [13, 19, 6] for each image, using the OpenCV implementation [3]. We use the default parameters except for cell size which is set to be 8×8 pixels, and block size which is 16×16 pixels, giving a descriptor of 1440 entries. We set the frame rate such as to receive twelve images per gesture. Thus, a gesture is characterized by twelve HOG descriptors, each having a fixed size of 1440 values.

3D data To record 3D data, we use the stream of depth images/point clouds provided by an Orbbec Astra 3D sensor. During the two-second window for each gesture, we receive a total of six point clouds. Each of these point clouds is passed through the five steps of preprocessing:

– *Downsampling:* In the first step, we reduce the size of the point clouds to lower the computational costs. Therefore, we create a 3D-voxel grid over the

input point cloud data. For every voxel, we calculate the centroid of all its points and use this to represent the voxel.
- *VoI Filtering:* In the second step, we use conditional removal to crop the point cloud to a defined volume of interest. Thus, removing background data and leaving just the area in which the hand is present.
- *Removing NaN:* Afterwards, we remove measurement errors by deleting all points whose x-, y-, or z-value is equal to NaN.
- *Computing Normals:* In the fourth step, we use approximations to infer the surface normals for all points in the point cloud.
- *Creating a Point Feature Histogram:* To get a descriptor of fixed length [15, 16], regardless of the size of the point clouds, that can be fed to the deep LSTM model, we decided on a representation with point feature histograms (PFH). These descriptors characterize the phenomenology of hand, palm, and fingers in a precise manner while remaining computationally feasible at the same time. PFH is based on the surface normals computed in step 4. In this step, we repeatedly select two points and compute their descriptor [14, 4] which provides four values based on the length and relative orientation of the surface normals. Each of the four values is subdivided into five intervals, giving a total of 625 discrete possibilities. The result, therefore, is a 625-dimensional histogram for each point cloud. Lastly, we normalize the histogram so that all the 625 values total to 1.

We receive six frames for every gesture. Each frame consists of 625 values. Figure 2 shows the point feature histogram of one frame of one gesture (right) and the corresponding point cloud (left).

IMD data To record the acceleration data, we use a 6-axis acceleration sensor (JoyWarrior56FR1-WP) attached to the back of the user's hand. It can record 3-axis acceleration data and 3-axis yaw rates at a frequency of 500 Hz, generating a 6-tuple at every measurement. To clean the rather noisy signals, we gather all $N = 100$ 6-tuples from each consecutive 200-millisecond window into a block and calculate statistical values for each entry of the 6-tuples: variance $\mathrm{Var}(x)$, mean $\bar{x}$, and standard deviation $S(x)$, as shown in equations 1-3.

$$\bar{x} = \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right) = \frac{x_1 + x_2 + \cdots + x_N}{N} \tag{1}$$

$$\mathrm{Var}(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 \tag{2}$$

$$S(x) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{3}$$

Thus, we receive ten frames for every gesture ($200ms \cdot 10$ frames $= 2s$). Each frame consists of 18 values: three statistical values for each of the six axes.

2.3 Fusion methods

Since the three sensory modalities we are considering here have different numerical formats and arrive at different frequencies, we discard "early fusion" types of approaches (see Sec.1.1) and focus on an analysis of the LSTM readout layer activities $\vec{r}^m$ for the last frame in each modality $m \in \mathcal{M} = \{\mathrm{RGB}, \mathrm{3D}, \mathrm{IMD}\}$. Since these are standard linear softmax layers (the same as the final layer of a standard DNN), they have **a)** as many entries as there are gesture classes, **b)** entries $r_i^m \in [0, 1]$ and **c)** entries normalized to 1: $\sum_i r_i^m = 1 \forall m \in \mathcal{M}$.

When denoting the decision for a certain class, in modality $m \in \mathcal{M}$ based on the last readout layer activity $\vec{r}^m$, by $\mathcal{C}^m$, the following fusion strategies to obtain the fused class estimate $\mathcal{C}$ seem natural:

max-conf: use the most certain uni-modal class decision to obtain

$$x = \mathrm{argmax}_{m \in \mathcal{M}} \left(\max_i r_i^m \right) \tag{4}$$

$$\mathcal{C} = \mathcal{C}^x \tag{5}$$

prob: treat the output layer activities as independent conditional probability distributions for a class i given the unimodal input sequence $\vec{x}^m$. The probabilities are then given by $r_i^m = p^m(\mathcal{C}^m = i|\vec{x}^m)$, and fusion is achieved by multiplication and renormalization using a constant C (this step can be skipped if we just want to infer the class of maximal probability):

$$\mathcal{C} = \mathrm{argmax}_i \left(\frac{1}{C} \prod_{m \in \mathcal{M}} p(\mathcal{C}^m = i|\vec{x}^m) \right) \tag{6}$$

$$= \mathrm{argmax}_i \left(\prod_{m \in \mathcal{M}} r_i^m \right) \tag{7}$$

3 Experiments

3.1 Uni-Modal Baselines

To establish the initial ability of a deep LSTM network to correctly classify each of the modalities, we train LSTM classifiers separately for each modality. To make sure results are not affected by network architecture, we run multiple experiments per modality, varying different hyper-parameters: batch size, number of layers, and number of LSTM cells (neurons) per layer. Each training run is repeated five times with the same hyper-parameters. The results show test accuracies averaged over all 5 training runs. The recorded gestures from the database are randomly split into training and test groups with a proportion of 80:20 prior to training and subsequently used as training and test data in all uni-modal and multi-modal experiments.

Table 1: Classification results of IMD data: Averaged accuracy (in percent) over five experiments.

BS \ (L, S)	$(2, 150)$	$(2, 200)$	$(2, 250)$	$(3, 150)$	$(3, 200)$	$(3, 250)$	$(5, 150)$	$(5, 200)$	$(5, 250)$
150	98.12	**98.61**	97.26	97.28	97.97	97.89	96.84	97.29	97.33
250	97.89	97.74	97.63	98.27	98.04	98.01	97.63	97.82	98.12
500	98.38	98.35	98.42	97.82	97.97	94.89	98.12	97.14	98.57
1000	98.23	98.04	98.20	97.97	98.46	98.12	97.37	97.74	97.97

IMD data We vary the batch size $b \in \{150, 250, 500, 1000\}$, the number of hidden layers $L = \{2, 3, 5\}$ and their size $S = \{150, 200, 250\}$. We use a fixed learning rate of $\epsilon = 0.001$ and a fixed number of iterations $I = 1.000$. Table 1 shows the results for the freehand gesture classification on acceleration data only. We observe that this modality can, just by itself, perform a reliable gesture classification.

The average accuracy obtained across all experiments is 97.83%. As can be seen, an LSTM architecture with two hidden layers and 200 cells per layer $(L, S) = (2, 200)$ and a batch size $b = 150$ achieves the best results (98.61%). Table 4a shows the confusion matrix for the classifiers trained only on IMD data.

3D Data We vary the batch size $b \in \{150, 250, 500, 1000\}$, the number of hidden layers $L = \{2, 3, 5\}$ and their size $S = \{150, 200, 250\}$ in the same way as for IMD data. We use a fixed learning rate of $\epsilon = 0.001$ and a fixed number of iterations $I = 5.000$. Table 2 shows the results for freehand gesture classification on 3D data alone. Preliminary experiments have shown that a smaller amount of fewer than 3.000 iterations leads to less accurate classification results.
The average accuracy obtained across all experiments is 86.92%. As can be seen, an LSTM architecture with two hidden layers and 200 cells per layer $(L, S) = (2, 200)$ and a batch size $b = 250$ can achieve the best results (93.61%). Table 4b shows the confusion matrix for the classifiers trained only on 3D data.

Table 2: Classification results of 3D data: Averaged accuracy (in percent) over five experiments.

BS \ (L, S)	$(2, 150)$	$(2, 200)$	$(2, 250)$	$(3, 150)$	$(3, 200)$	$(3, 250)$	$(5, 150)$	$(5, 200)$	$(5, 250)$
150	91.17	93.23	92.37	89.62	89.70	90.26	78.76	78.61	81.69
250	92.78	**93.61**	91.84	88.68	89.96	90.00	77.67	79.44	80.34
500	90.34	91.77	91.62	89.14	87.33	90.30	78.87	79.02	80.49
1000	90.60	91.54	92.14	89.55	88.95	90.34	77.93	79.21	80.41

Table 3: Classification results of RGB data: Averaged accuracy (in percent) over five experiments.

BS \ (L, S)	$(2, 150)$	$(2, 200)$	$(2, 250)$	$(3, 150)$	$(3, 200)$	$(3, 250)$	$(5, 150)$	$(5, 200)$	$(5, 250)$
150	98.87	99.00	**99.06**	98.93	98.74	98.75	97.49	98.50	98.56
250	98.37	98.81	99.00	97.25	97.81	98.05	96.55	97.75	96.30
500	97.93	98.31	98.31	96.37	97.56	97.37	98.31	94.55	95.49
1000	97.74	97.74	98.87	96.24	97.74	97.18	96.99	96.53	98.31

RGB Data We vary the batch size $b \in \{150, 250, 500, 1000\}$, the number of hidden layers $L = \{2, 3, 5\}$ and their size $S = \{150, 200, 250\}$, as we did for the IMD data and the 3D data. We use a fixed learning rate of $\epsilon = 0.001$ and a fixed number of iterations $I = 1.000$. Table 3 shows the results for the freehand gesture classification on the RGB data alone.
The average accuracy obtained across all experiments is 97.76%. As can be seen, an LSTM architecture with two hidden layers and 250 cells per layer $(L, S) = (2, 250)$ and a batch size $b = 150$ can achieve the best results (99.06%). Table 4c shows the confusion matrix for the classifiers trained only on RGB data.

3.2 Multi-modal Fusion

Having established the uni-modal baselines, we pick the best architecture for each modality and compute class prediction accuracies on test data using the two fusion strategies outlined in 2.3.

Fusion strategy: max-conf As a first step, we convert the predictions by using a softmax function to receive probabilities for each of the classes for every gesture. Then, using Eq. 4 and Eq. 5, we pick the most certain uni-modal class

Table 4: Confusion matrix for the uni-modal classifications. Rows display the actual gesture and columns show the predicted gesture.

(a) IMD Data

71	0	0	0	0	0	0
0	67	0	0	0	0	0
1	0	85	0	0	0	1
0	0	0	75	2	0	0
1	0	1	0	71	0	0
0	0	1	0	0	69	0
0	0	0	0	0	0	89

(b) 3D Data

60	4	1	0	1	3	2
1	64	2	0	0	0	0
1	1	82	1	0	0	0
1	0	7	68	1	0	0
1	0	5	1	66	0	0
2	0	1	0	5	62	0
1	0	0	0	0	2	86

(c) RGB Data

71	0	0	0	0	0	0
0	66	0	0	1	0	0
0	0	83	2	0	0	0
1	0	2	74	0	0	0
0	0	0	0	73	0	0
0	0	0	0	0	70	0
0	0	0	0	0	0	89

decision as our final prediction. Using this fusion strategy, we can improve our classification to achieving an accuracy of 100% by correctly classifying 532 out of 532 gestures.

Fusion strategy: prob Same as for the max-conf fusion strategy, we convert the predictions by using a softmax function. Then, we use Eq. 7 to obtain a multi-modal probability, in which the highest probability determines the predicted class. Using this fusion strategy, we can improve our classification to an accuracy of 100% by correctly classifying all of the 532 gestures.

3.3 Ahead-of-time classification and multi-modal fusion

A feature of deep LSTM network classifiers is the possibility to read out class estimates ahead of time, i.e., before the sequence is complete. In reality, this enables technical systems to, e.g., anticipate a user's gesture command, thus realizing a more efficient and natural interaction. We compute the uni-modal ahead-of-time classification accuracies by simply making the LSTM output layer at a certain frame $f < F$ the basis of gesture class estimation, instead of using the last frame F. The same strategy is applicable for fusion: instead of fusing three output layer activities from the last sequence frame F according to 2.3, we fuse output layer activities evaluated at frames $f_m, m \in \mathcal{M} = \{\mathrm{RGB}, \mathrm{3D}, \mathrm{IMD}\}$. This is complicated by the fact that the three modalities have slightly unequal sequence lengths (12, 6 and 10 for RGB, 3D and IMD data). We therefore investigate the benefit of fusion for look-ahead classification for strong look-ahead ($f_{\mathrm{RGB}} = 5, f_{\mathrm{3D}} = 3$ and $f_{\mathrm{acc}} = 4$) and moderate look-ahead ($f_{\mathrm{RGB}} = 8, f_{\mathrm{3D}} = 4$ and $f_{\mathrm{IMD}} = 7$). The results are shown in Tab.5. Since the results for uni-modal classification on IMD data are already very high, we investigate the effects of multi-modal fusion on look-ahead classification for $f_m, m \in \mathcal{M}_1 = \{\mathrm{RGB}, \mathrm{3D}, \mathrm{IMD}\}$ and $f_m, m \in \mathcal{M}_2 = \{\mathrm{RGB}, \mathrm{3D}\}$.

As expected, the lower results of RGB and 3D data classification have a negative impact on the high results of IMD data classification. Fusion does not help in this case. On the other hand, looking at the effects of fusion on $\mathcal{M}_2$ shows that multi-modal fusion using the **prob** strategy improves the results for a moderate look-ahead. Using the **max-conf** strategy improves the results for a strong look-ahead. Therefore, a positive benefit of multi-sensory fusion can be observed on look-ahead classification for modalities that are more difficult to classify.

Table 5: Effects of multi-modal fusion on look-ahead classification. Shown is the accuracy in % for uni-modal classification as well as fusing RGB and 3D data[1] and all three modalities[2] respectively.

Method / Look-ahead	RGB	3D	IMD	prob		max-conf	
strong: 50%	30.45	17.11	52.26	27.26[1]	27.26[2]	31.77[1]	42.48[2]
moderate: 75%	62.97	37.78	93.05	70.49[1]	70.49[2]	60.71[1]	87.41[2]

4 Discussion and conclusion

Data used in this study One might argue that the classification problem is a simple one since all recorded gestures come from the same person. On the other hand, this ensures that gestures are performed expertly and that any variability in the gestures is an intrinsic, structured one and not just caused by inexpert users performing gestures incorrectly. In the latter case, gesture classification may be hard for a human observer as well.

Assessment of results The presented results show, unsurprisingly, that fusing results from multiple modalities increases classification performance. This is however less spectacular since the results are already very satisfactory even without fusion. What makes multi-modal fusion worthwhile in this context is its impact on ahead-of-time classification accuracy which is significantly increased for moderate look-ahead where classification is attempted even though only 66% of a gesture has been observed.

Comparison of fusion schemes The two presented fusion schemes are admittedly simplistic, but on the other hand, they are real-time capable since they do not incur a measurable computational overhead. Furthermore, fusing only the readout layer activities in the last frame for each modality is not as restrictive as it seems since LSTM networks retain information about states from past frames. One may, therefore, conclude, that by fusing only at the last frame, one takes into account information from the whole sequence of multi-modal measurements. Based on this study, we can state that both investigated fusion schemes are equivalent.

Next steps As ahead-of-time classification has a high potential impact on user interaction, it would be an interesting theoretical study to modify the LSTM loss function such that early correct classifications are rewarded, thus actively enforcing ahead-of-time classification. Another conceptually important point is detecting outlier gestures that belong to no known class. This requires a learned, generative description of sequence data by, e.g., Hidden-Markov models.

References

1. Angelaki, D.E., Gu, Y., DeAngelis, G.C.: Multisensory integration: psychophysics, neurophysiology, and computation. Current opinion in neurobiology **19**(4), 452–458 (2009)
2. Beauchamp, M.S.: See me, hear me, touch me: multisensory integration in lateral occipital-temporal cortex. Current opinion in neurobiology **15**(2), 145–153 (2005)
3. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
4. Caron, L.C., Filliat, D., Gepperth, A.: Neural network fusion of color, depth and location for object instance recognition on a mobile robot. In: European Conference on Computer Vision. pp. 791–805. Springer (2014). https://doi.org/10.1007/978-3-319-16199-0_55

5. Chen, C., Jafari, R., Kehtarnavaz, N.: Improving human action recognition using fusion of depth camera and inertial sensors. IEEE Transactions on Human-Machine Systems **45** (10 2014). https://doi.org/10.1109/THMS.2014.2362520
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 886–893 vol. 1 (2005)
7. Ernst, M.O., Banks, M.S.: Humans integrate visual and haptic information in a statistically optimal fashion. Nature **415**(6870), 429–433 (2002)
8. Gepperth, A.R., Hecht, T., Gogate, M.: A generative learning approach to sensor fusion and change detection. Cognitive Computation **8**(5), 806–817 (2016)
9. Graves, A., Jaitly, N.: Towards end-to-end speech recognition with recurrent neural networks. In: Xing, E.P., Jebara, T. (eds.) Proceedings of the 31st International Conference on Machine Learning. pp. 1764–1772. No. 2 in Proceedings of Machine Learning Research, PMLR, Bejing, China (22–24 Jun 2014), http://proceedings.mlr.press/v32/graves14.html
10. Imran, J., Raman, B.: Evaluating fusion of rgb-d and inertial sensors for multimodal human action recognition. Journal of Ambient Intelligence and Humanized Computing (Feb 2019). https://doi.org/10.1007/s12652-019-01239-9, https://doi.org/10.1007/s12652-019-01239-9
11. Khaire, P., Kumar, P., Imran, J.: Combining cnn streams of rgb-d and skeletal data for human activity recognition. Pattern Recognit. Lett. **115**, 107–116 (2018)
12. Liu, K., Chen, C., Jafari, R., Kehtarnavaz, N.: Fusion of inertial and depth sensor data for robust hand gesture recognition. IEEE Sensors Journal **14**(6), 1898–1903 (2014)
13. McConnell, R.: Method of and apparatus for pattern recognition (Jan 1986)
14. Rusu, R.B., Blodow, N., Marton, Z.C., Beetz, M.: Aligning point cloud views using persistent feature histograms. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3384–3391. IEEE (2008). https://doi.org/10.1109/IROS.2008.4650967
15. Sachara, F., Kopinski, T., Gepperth, A., Handmann, U.: Free-hand gesture recognition with 3d-cnns for in-car infotainment control in real-time. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). pp. 959–964 (Oct 2017). https://doi.org/10.1109/ITSC.2017.8317684
16. Sarkar, A., Gepperth, A., Handmann, U., Kopinski, T.: Dynamic hand gesture recognition for mobile systems using deep lstm. In: Horain, P., Achard, C., Mallem, M. (eds.) Intelligent Human Computer Interaction. pp. 19–31. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-72038-8_3
17. Stein, S., McKenna, S.J.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 729–738. UbiComp '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2493432.2493482, https://doi.org/10.1145/2493432.2493482
18. Tran, T., Le, T., Pham, D., Hoang, V., Khong, V., Tran, Q., Nguyen, T., Pham, C.: A multi-modal multi-view dataset for human fall analysis and preliminary investigation on modality. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 1947–1952 (Aug 2018). https://doi.org/10.1109/ICPR.2018.8546308
19. William T. Freeman, M.R.: Orientation histograms for hand gesture recognition. Tech. Rep. TR94-03, MERL - Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 (Dec 1994), https://www.merl.com/publications/TR94-03/