AutoBAP: Automatic Coding of Body Action and Posture Units from Wearable Sensors

Eduardo Velloso
Lancaster University
Lancaster, UK
e.velloso@lancaster.ac.uk

Andreas Bulling
Max Planck Institute for Informatics
Saarbrücken, Germany
andreas.bulling@acm.org

Hans Gellersen
Lancaster University
Lancaster, UK
hwg@comp.lancs.ac.uk

***Abstract*—Manual annotation of human body movement is an integral part of research on non-verbal communication and computational behaviour analysis but also a very time-consuming and tedious task. In this paper we present AutoBAP, a system that automates the coding of bodily expressions according to the body action and posture (BAP) coding scheme. Our system takes continuous body motion and gaze behaviour data as its input. The data is recorded using a full body motion tracking suit and a wearable eye tracker. From the data our system automatically generates a labelled XML file that can be visualised and edited with off-the-shelf video annotation tools. We evaluate our system in a laboratory-based user study with six participants performing scripted sequences of 184 actions. Results from the user study show that our prototype system is able to annotate 172 out of the 274 labels of the full BAP coding scheme with good agreement with a manual annotator (Cohen's kappa > 0.6).**

I. Introduction

Facial expressions and speech are rich sources of information and powerful modalities for automatic recognition of basis affective states. They have consequently been investigated for a long time in affective computing research [1], [2]. With the availability and decreasing cost of ambient and on-body sensing systems, there has also been increasing interest in using bodily motion as well as gaze behaviour for the same purpose [3], [4]. Researchers have for example tried to identify correlations of low-level movement features to affective states, such as the velocity of different body parts [5].

A key requirement for developing computational methods for affect recognition from speech, physical and visual behaviour is the availability of extensive and fully annotated datasets. Such annotation is currently performed manually using video annotation tools, such as Anvil or Élan, according to a specific coding system. One of the most well-known coding systems for facial expressions is the Facial Action Coding System (FACS) [6], [7] and a similar system has recently been proposed for body actions and postures (BAP) [8]. High-quality manual annotation requires appropriate training of expert coders, making it a cumbersome and costly task. For example, it took Dael et al. on average 15 minutes to code each 2.5 seconds portrayal in the Geneva Multimodal Emotion Portrayals (GEMEP) corpus using the BAP coding system [9]. Moreover, the output is susceptible to subjective interpretation, mistakes and omissions.

While attempts to automate this task for mature coding systems, such as the Facial Action Coding System, have been made [1], the same does not apply to annotating body expression. As the interest in affective body expressions increases, so does the demand for tools and methods to support research in the topic. However, to the best of our knowledge, there is currently no software tool available to annotate affective body expressions automatically.

We aim to fill this gap by presenting AutoBAP, a prototype system that automatically annotates body and eye motion data according to the Body Action and Posture coding scheme using data from wearable sensors. AutoBAP uses hardcoded rules that implement the coding guidelines as well as decision trees trained with machine learning algorithms on data collected in a user study. The decision trees were trained during the system development so that our prototype doesn't have to be trained for new users. Results from a user study demonstrate that our system is able to automatically extract 172 behaviour variables from wearable motion and gaze tracking data with good correspondence to manual annotation.

II. Related Work

The most widely researched modality in emotion research is facial expression. The Facial Action Coding System is a coding system that deconstructs facial expressions into action units (contractions and relaxation of facial muscles) and their temporal segments [6], [7]. Automatic implementations of FACS include a system trained to automatically detect action units in order to differentiate fake from real expressions of pain [10] and to analyse expressions of neuropsychiatric patients [11]. Techniques to achieve this include analysing permanent and transient facial features in frontal face image sequences [12], using independent component analysis and support vector machines [13] and using Gabor wavelets with neutral face average difference [14].

An early notation system for body motion was Labanotation [15], which was originally developed to describe dance movements and is part of Laban Movement Analysis, which breaks movements down to Body, Effort, Shape and Space. DMAR [16] offers a graphical interface for dance experts to annotate dance concerts or clips, but it does not do it automatically. Birdwhistell's coding system is based on linguistic principles [17]. It defines kinemes (analogous to phonemes in Linguistics), which are groups of movements which are not identical, but communicate the same meaning. This notation has been used to categorise the emotions in emoticons [18]. Attempts to facilitate the transcription of body movements include animating a 3D skeleton to annotate arms' gestures

[19], but this system still requires the annotator to match the animation to the video recording.

More recently, Dael et al. proposed a coding system for the description of body movement on anatomical, form and functional levels, more suitable for coding nonverbal emotion expression [8]. Some advantages of this coding system are that it minimises observer bias by being supported by a reliable observation protocol; because it is not based on linguistic principles it is independent from other modalities such as speech; and it is generic enough to be used outside of emotion research. Presently, the authors perform the coding using the Anvil software, which is a manual annotation tool [20]. As of yet, there is no system that extracts the BAP coding automatically from body motion.

III. The Body Action and Posture Coding System

BAP separates its behaviour variables into 12 categories: head orientation, action and posture; trunk orientation, action and posture; arms action and posture; whole body posture; gaze; action functions and other. In BAP, *orientation* labels are coded using an external frame of reference, namely the interlocutor. *Posture* labels can be of three types: (1) *posture units* (PU), which are broken down into (2) *posture transition phase* (PT) and (3) *posture configuration phase* (PC). Posture transition refers to the period of time to reach the end position and posture configuration is the period of time in which the posture is maintained. The direction of postures is coded according to the three orthogonal planes that cross the centre of mass of the body in the standard anatomical position. *Actions* change more frequently than postures, so they are coded differently as *action units*, which can be broken down into its different steps (*action subunits*).

Even though the coding system attempts to be as objective as possible, it offers challenges to its automatic implementation. The first challenge is about how the data is segmented. The coding guidelines are very specific about the definition of onset and offset points for the segments, so it is important to define precisely the frames where the label begins and ends. However, when implementing it automatically, there will always be issues of noise and synchronisation between different sensors due to different sample rates. Second, even if the segmentation is correct, its labels depend on its context, so the same data segment may have different labels depending on the previous and next segment. Let's take the example of a right head turn. If after turning the head the user holds the head to the right, it is labelled as a posture transition, but if it is followed by a left head turn, it is labelled as an action sub-unit. Moreover, if the user's head posture was already annotated as being turned to the right before turning the head to the right, the head turn is considered part of the configuration phase and not labelled at all.

IV. System Overview

AutoBAP is the algorithmic layer that sits between a wearable sensing system and a graphical user interface. Figure 1 shows an overview of our prototype. In the bottom layer of our prototype lies the sensing system, which includes a wearable inertial sensors-based motion capture suit and a computer vision-based wearable eye tracker. We opted for wearable

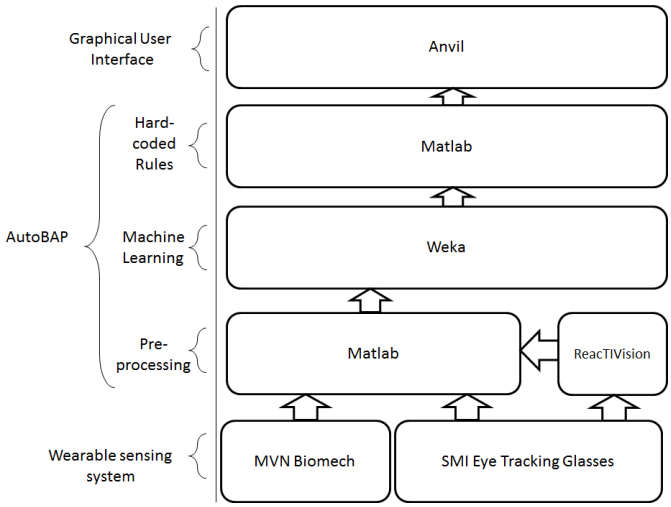

Fig. 1. System Overview. Motion and gaze tracking data are captured with their corresponding tracking system and preprocessed in Matlab. Additionally, we use a computer vision toolkit to track a fiducial marker simulating an interlocutor. We then use the Weka machine learning library to classify the data into initial categories and annotate it using hard-coded rules in Matlab. The system outputs an XML (.anvil) file with the annotation, which can be visualised and edited using Anvil.

solutions as in the future we would like to use our approach to automatically annotate "in-the-wild", i.e. out of the laboratory, behavioural data. In our prototype, we track motion using an Xsens MVN Biomech full body motion tracking system. This is an ambulatory 3D human kinematic measurement system that comprises 17 inertial measurement units (10 in the upper body and 7 in the lower body) and outputs 3D orientation and position of 23 body segments, 22 joints, body centre of mass and raw data from inertial sensors at a sample rate of 120Hz. It transmits its data wirelessly to MVN Studio (version 3.4) which synchronises it to the corresponding frames from an Allied Vision Technologies Prosilica GS650C Ethernet video reference camera (25Hz). Gaze tracking is performed with SMI Eye Tracking Glasses. This is a non-invasive video-based glasses-type binocular eye tracker with automatic parallax compensation at a sample rate of 30Hz, a spatial resolution of 0.1 degrees and a gaze position accuracy of 0.5 degrees over all distances. The glasses are connected with a USB cable to a Windows 8 laptop running the iViewETG software, which streams it wirelessly to another Windows laptop running a custom-built application that records and processes the data.

AutoBAP is the layer above the sensing and is comprised of three components. First, it preprocesses the sensor data. This involves synchronising different sample rates, merging data from different sensors and extracting derived features for the machine learning algorithms. Also, in this prototype we simulated an interlocutor with a fiducial marker placed next to the reference camera and used a computer vision algorithm to extract its position from the eye tracker's scene camera. We do this in the preprocessing stage using the reacTIVision toolkit [21]. The second component is the decision tree algorithms. These trees were trained by machine learning algorithms from the Weka library [22] using user data. The data collection procedure is described in the next section. The output of the decision trees is then analysed by the third component, which implements the guidelines described in BAP's coding

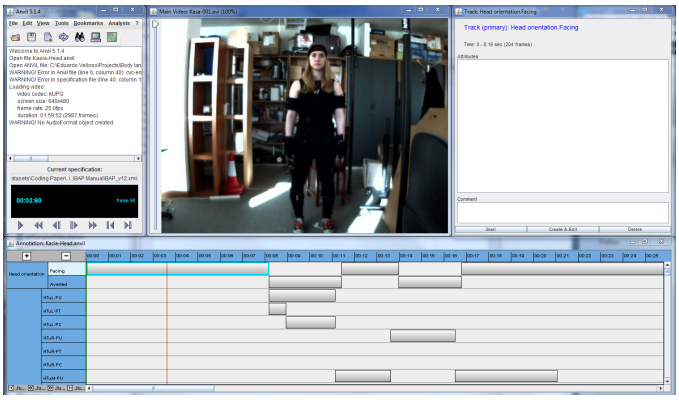

Fig. 2. Anvil user interface. The user can see all labels generated by AutoBAP on a timeline as well as the video recording.

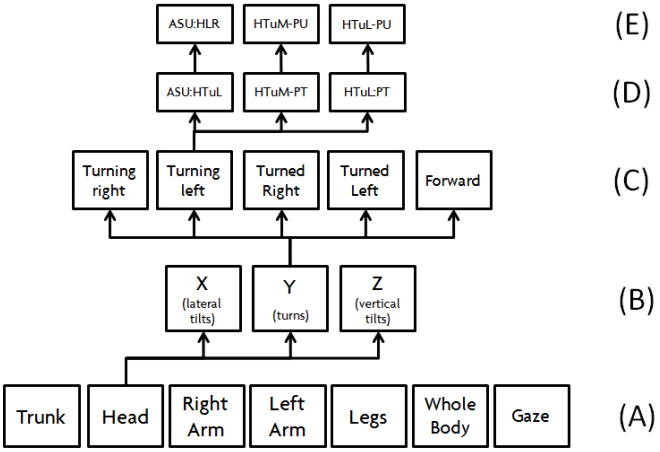

Fig. 3. Example annotation extraction for a left head turn. Input is the data recorded using the motion capture system (A). We then analyse each component of the movement independently (B) and use machine learning to identify the direction of movement or the orientation of the posture (C). Using hard-coded rules that follow BAP's coding guidelines, we analyse the temporal context of the segment and assign the appropriate label (D). Depending on the context, we also combine segments into parts of larger actions such as head shakes and/or extract other labels such as posture units (E).

guidelines document and manual. We implemented the rules in Matlab. Finally, after annotating the data, this component downsamples it back to the camera's sample rate and creates an XML file with the annotation data. Section V provides an overview of the annotation extraction procedure.

The top layer is the graphical user interface for visualising and editing the extracted annotation. In order to leverage the capabilities and familiar user interface of a widely used tool, we chose Anvil [20] for this purpose. In Anvil, the user can make any desired changes supported by the platform, such as adding, editing and removing labels (see Figure 2).

V. Annotation Extraction

In this section, we describe our approach to classifying actions and postures, which includes decision trees and hard-coded rules. We describe the collection and manual annotation of the data and the selection of features to train the decision trees. We then describe how we adjust the output of these classifiers to adhere to the coding guidelines using hard-coded

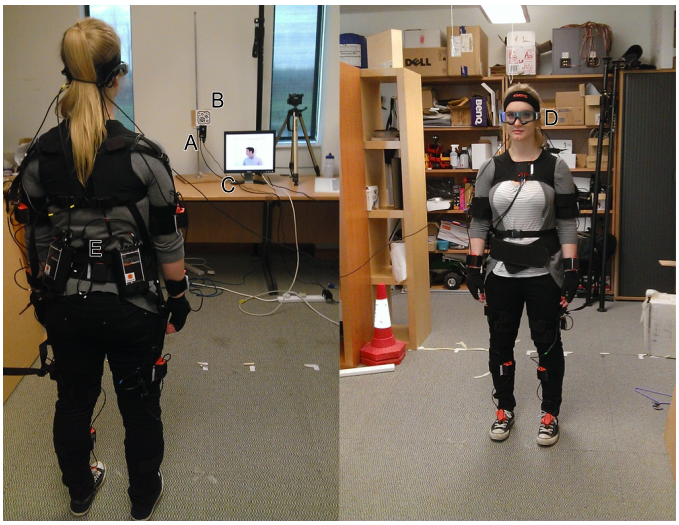

Fig. 4. Sensing setup. Participants were recorded by a Prosilica video camera (A), positioned next to a fiducial marker (B). They performed actions displayed on an LCD screen (C) whilst being tracked by an SMI eye tracker (D) and an Xsens Biomech motion capture suit (E).

rules. We exemplify our approach with the example of a right head turn.

A. Data Collection

The first step in our annotation extraction procedure is to use decision trees training with a machine learning algorithm to classify the data. In order to train our classifier and to evaluate it subsequently, we collected a motion tracking dataset that could cover all labels in the coding system. We collected data from 6 participants, aged 18-31 (mean 24.7), of which 4 were male and 2 were female. They had different body builds, ranging from 1.65m to 1.82m of height (mean 1.74m) and from 59kg to 90kg (mean 73.7kg) of weight. Each data collection session involved only one participant and one researcher and took place at a quiet laboratory environment. The sensing setup consisted of the Xsens MVN Biomech full body motion tracking system and the eye tracker (see Figure 4). Participants stood 2m away from the camera, which was mounted on a stand 1.12m above the floor, next to a 17" LCD display. Motion and eye tracking data were recorded separately. The average length of the recording for each participant was 9 minutes and 45 seconds.

When participants arrived, they filled in a consent form and a personal details questionnaire. We then took measurements of each participant's height, foot size, arm span, ankle height, hip height, hip width, knee height, shoulder width and shoe sole height. These are data that can be input in MVN Studio to improve the accuracy of the motion capture. We then assisted each participant in mounting straps with the motion sensors. Each participant was then asked to follow a script of actions so that each behaviour variable in the code appeared at least once in the data. This script was displayed on the LCD screen in a slide presentation showing the instruction and a photo of a person performing the desired posture or action. For example, in the case of the head, participants were instructed to turn left/right and hold the posture; turn left/right without holding

the posture and turn left/right repeatedly. The same was done for head lateral and vertical tilts.

We exported the data from MVN Studio to a XML file that contained the timestamped position, orientation, velocity and angular velocity of each segment in the global frame and the angle on each joint in their own reference frame. We used the SMI BeGaze software to export the data from the SMI glasses to a log file containing the timestamped gaze position in the scene camera reference frame.

B. Manual Data Annotation

We annotated each recording twice: once to use as input when training the decision trees and once to evaluate the final output of the system. We did not use BAP labels to train our classifiers because some behaviours may be assigned to completely different movements and impact training. For example, the transition phase for head turn towards the lateral middle position might be a left or right movement, as long as they end in the middle. To simplify the training, we annotated the data separately using labels that describe the movement or posture independently of the sequence of behaviours. In the same example, instead of annotating a transition phase to the middle, we annotated a right or left turn accordingly. This way, the machine learning could learn how to classify the direction of the movement and the orientation of the posture and leave the annotation of what it means in the sequence of behaviours to the hardcoded rules.

We also manually annotated the video recording from the reference camera using Anvil, based on the BAP specification file for this platform, which can be obtained from its authors' website. Even though we had no formal training or practice with this particular coding system, since it had just been published, we followed the manual and additional guidelines carefully. We exported the annotation data using Anvil's "Export Annotation Frame-by-Frame" feature. This creates a tab-separated text file with a table in which each row represents a frame and each column, a label containing a boolean value representing the presence or absence of the label in that frame. We then used the timestamps to synchronise the annotation data with the sensor data. All the annotation was performed by the same person (first author). This second annotation was used to evaluate the final output of the system.

C. Feature Selection

Due to the complexity of human movement, using motion capture data to detect actions and postures also becomes a complex problem. For example, our tracking system can output for each sample up to 794 attributes (4D orientation, 3D position, 3D velocity, 3D acceleration, 3D angular velocity and 3D angular acceleration for each of the 23 segments; 3D acceleration, 3D angular velocity, 3D magnetic field and 4D orientation for each of the 17 sensors; 3D ZXY and 3D XZY angles for each of the 22 joints, the 3D position of the centre of mass and the timestamp), not counting other features that may be derived from those. At a sample rate of 120Hz, this quickly becomes an enormous amount of data, so selecting relevant features increases the speed of training and classification.

Moreover, several behaviours are completely independent of one another. For example, a user may turn his head to any direction independently to his arms configurations. Therefore, classifying behaviours of the head, whilst taking into account the features related to the arm, may improve recognition performance on a training set, but cause erroneous classification on testing data. Therefore, selecting a relevant feature set, also reduces overfitting and increases the recognition performance for further datasets.

Considering that the data were labelled according to the direction of movements and orientation of postures, we treated the annotation of each axis as a 5-class classification problem. For example, in the case of head turns the classes were: right turn, left turn, facing forward, facing to the right and facing to the left. We then used the angular speed to discriminate between movements and the joint angle to discriminate between postures, so that the output of this step would then be used in the subsequent classification procedure.

Some arm postures such as crossed arms, however, involve a specific configuration of more than one axis and segment, so for these cases we used multiple features. Other labels that the coding system is less specific about also require multiple features, but in a less restrictive way. For example, lower limb movements are only coded regarding leg movement or knee bend, so we need to analyse the movement from all segments in either leg to look for movement. For gaze labels we use as features the distance vector between the gaze point and the fiducial marker as extracted by the computer vision algorithm.

D. Decision Trees and Hardcoded Rules

Extracting a BAP annotation file is a problem that is reduced to filling in a matrix with 274 columns representing each behaviour and one row for each frame in the recording. Each cell in this matrix is a boolean variable that represents the presence or absence of the behaviour in the frame. A naïve approach would be to train 274 classifiers, but this would not take into account the relationship between behaviours. Therefore, we reduce the problem even further by grouping sets of exclusive labels. For example, the head cannot be turned to the left and to the right at the same time. Moreover, if the head is moving, it is not, in principle, in any posture (although it can happen, as we discuss in the next session). For each of these sets of behaviours we train a separate classifier using the appropriate feature set, effectively reducing our problem to 28 classifiers. In the case of the head, this leaves us with a separate classifier for head turns, vertical tilts and lateral tilts, with the output being one of five possibilities: the head is either turning to one direction or another, or is being held facing one the middle, one direction or another. We used the J48 decision tree training algorithm available in the Weka machine learning library using the simpler annotation as described previously.

This procedure outputs a table with the predicted labels for each sample in each column, which can be noisy. We smooth the classification output by trying to estimate the onset and offset of each label from the dataset. We use an adaptation of Velloso et al.'s [23] approach to motion modelling to find these points and assign to the segment the mode of the labels it contain, but instead of looking for characteristic points in multiple periodic repetitions of data, we look for these points in a single instance of the movement.

TABLE I. BEHAVIOURS ANNOTATED WITH A KAPPA OVER 0.6

Category	Labels
Head	Facing, Averted, HTuL, HTuR, HTuM, HTiL, HTiR, HTiM, HVU, HVD, HVM
Trunk	Facing, Averted, TLF, TLB, TLMF, TLL, TLR, TLML, TRL, TRR, TRM
Whole body	BF, BB, BMF, BL, BR, BML
Arms	LA/RA side, LA/RA front, LA/RA back, LH/RH neck, AA crossed, AA front, A hold A front/back, AA sym, AA asym
Gaze	Toward, Upward, Downward, Averted Sideways, Eyes Closed
Head Action form	HTuL, HTuR, HTiL, HTiR, HVU, HVD, HVUD, HLR
Trunk Action form	TLF, TLB, TLL, TLR, TRL, TRR, TLLR, TRLR, TLFB
Arms Action form	AA sym, AA asym, wrist, elbow, shoulder, up, down, forward, backward, left, right, toward, up-down, left-right, forward-backward, circular, retraction
Lower limbs	Knee bend, leg movement

Once we can differentiate directions of movement and orientations of postures, we then take a step back and consider the position of each behaviour in the time series. We hardcoded rules that implement the coding guidelines in the BAP manual. For example, if the segment following a left head turn is a posture held facing forward, we annotate it as a transition phase of a lateral head turn towards a middle position (HTuM-PT), but it is followed by another movement, we label it as an action form sub-unit (Action form.Head-ASU:HTuR). Moreover, if a pattern of repeated action sub-units is detected, we classify it differently (Action form.Head-ASU:HLR, in the case of repeated left and right head turns turning into a head shake). Also, we combine transitions and configurations to extract posture units. Figure 3 exemplifies the classification possibilities for a left head turn.

E. Exporting the Annotation

Our classifiers output a matrix in which each column contains a boolean value for the presence or absence of each label and each row represents a data sample from the motion tracker. We reduce the sample rate of 120Hz to 25Hz in order to match the frame rate of the video recording by taking the 4 or 5 samples corresponding to each frame and creating a data point with the mode of the labels in that interval. We then group intervals with the same label and write it to an XML file according to the Anvil file format. This allows us to visualise and edit the annotation data using Anvil's graphical user interface as if the annotation had been performed manually.

F. Evaluation

We evaluated the system using cross-validation, using the data from five participants for training and one for testing. We then compared the output of the system with manually annotated data by calculating the agreement between manual and automatic annotations using Cohen's kappa [24] based on the presence or absence of a behaviour unit on each frame in the portrayal. This is a measure of inter-annotator agreement that takes into account the agreement occurring by chance. We considered the labelling successful when the kappa was over 0.6 [25]. Table I shows the reliably annotated labels. Posture units shown on the table include all related labels (PU - posture unit, PT - posture transition and PC - posture configuration).

VI. DISCUSSION

The Body Action and Posture coding system is still in its early days at the time of writing. As the coding system matures and increases in adoption, studies in Affective Computing will lead to a better understanding of how these labels correlate to affective states. Hence, the automatic extractions of such labels will make it feasible to implement affect recognition systems that take into account the domain knowledge.

Our study results show that AutoBAP can encode a wide range of BAP units. Some subtle postures were not picked up by our motion capture system despite the fact that we used state-of-the-art motion and gaze tracking systems. For example, the motion capture suit does not include sensors on the fingers, so in this prototype, we did not attempt to label finger actions. Also, even though Xsens' proprietary algorithms extracts a very accurate model of motion, from the available sensors, the orientation of some segments where no sensors are attached to, such as the neck, are inferred from the orientation of other sensors. This makes the detection of some movements such as neck retractions and extensions more difficult.

In this study, we recorded and annotated very specific and controlled scripted movements, in order to have an unambiguous and comprehensive dataset for training our algorithms. We demonstrated that our approach can classify these datasets accurately but we clearly need to validate our system using other datasets. Also, the training and testing data we used were labelled by the same person. We started from the assumption that the coding system is reliable enough so that two independent raters may end up with a reasonably similar result, as suggested by Dael et al. [8], but we can't make any statements about how the system would perform when compared to third-party annotations. We limited the scope of this study to objective movements, so we did not attempt to classify action functions, such as emblems and illustrators. Due to the wide range of possibilities for such gestures, classifying them becomes a whole challenge on its own.

Our prototype is currently coupled to the chosen sensing system and annotation GUI, but we posit that our approach would be transferable to others. We chose Anvil as the export format as BAP's original specification was published in this format. As it is an XML file and BAP is, in principle, compatible with other annotation tools, it should not be a problem to convert it to other formats. See Schmidt et al. for a description of an effort to convert between annotation formats [26]. Other eye trackers could also be used with few adjustments as long as it provides a scene camera to track the interlocutor or some other means to extract relative orientation and position. Using other motion trackers might be more complicated though. Motion capture systems vary widely in terms of accuracy and which segments they track. While a different implementation would be needed to match the new features to annotation labels, the implementation procedure we described could still be applied.

VII. LIMITATIONS AND FUTURE WORK

The labels we attempted to extract in this work were limited by the capabilities of the tracking system. In the future, we would like to explore additional sensing modalities to detect the remaining labels, such as hand tracking and touch sensors.

For this paper, we recorded a scripted dataset to cover as many labels as possible, but this means that the actions were not natural. Future work will include recording unscripted affective data to improve the training dataset and to evaluate the classifiers in a realistic dataset. This will also allow us to explore the classification of action functions, such as emblems, illustrators and manipulators as well as the possibilities of using automatically extracted labels for affect recognition.

In this first prototype we simulated the interlocutor as a fiducial marker and annotated gaze according to the distance between the gaze point and the fiducial marker as extracted by a computer vision toolkit. In the future, we would like to replace this for a face recognition system, so the system may be used in a real life setting.

In this paper, we attempted to annotate as many labels in BAP as possible. However, the coding system was initially created to code the data in a specific dataset, the Geneva Multimodal Emotion Portrayals (GEMEP) corpus, in which actors portray emotions while standing up and being recorded by face and upper body cameras [9]. Therefore, the scope of the coding system is limited to behaviours expressed in such a way. By capturing body expressions with a tracking system, the coding system could be extended in the future to cover more behaviours such as specific leg movements and sitting down postures.

We started the paper by arguing for a combination of the low-level data provided by motion capture systems with high-level posture and action units annotated manually. In the future, we would like to extend the coding specification to leverage this combination. This way, automatically extracted labels could include additional data such as range of motion and average speed for action units and average orientation angle for posture units.

VIII. Conclusion

In this paper, we presented an approach to extract BAP annotation labels automatically from motion and gaze tracking data. Our prototype extracts 172 out of the 274 labels in the coding system with a Cohen's kappa higher than 0.6. Because manually annotating video data is a highly time-consuming and error-prone activity, it is unsuitable for long recordings. By using our approach, it is possible to annotate bigger datasets, making it possible to apply BAP to new application areas that take into account longer periods of time such as computational behaviour analysis or life logging.

References

[1] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259 – 275, 2003.

[2] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[3] A. Kleinsmith and N. Bianchi-Berthouze, "Affective body expression perception and recognition: A survey," *Affective Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2012.

[4] Y. Zhao, X. Wang, and E. Petriu, "Facial expression anlysis using eye gaze information," in *Computational Intelligence for Measurement Systems and Applications (CIMSA), 2011 IEEE International Conference on*, 2011, pp. 1–4.

[5] H. M. Paterson, F. E. Pollick, and A. J. Sanford, "The Role of Velocity in Affect Discrimination," in *Proc. of the Twenty-Third Annual Conference of the Cognitive Science Society*, 2001, pp. 756–761.

[6] P. Ekman and W. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Palo Alto: Consulting Psychologists Press, 1978.

[7] P. Ekman, W. V. Friesen, and J. C. Hager, *Facial action coding system*. A Human Face Salt Lake City, 2002.

[8] N. Dael, M. Mortillaro, and K. Scherer, "The body action and posture coding system (bap): Development and reliability," *Journal of Nonverbal Behavior*, pp. 1–25, 2012.

[9] T. Bänziger and K. R. Scherer, "Introducing the geneva multimodal emotion portrayal (gemep) corpus," *Blueprint for affective computing: A sourcebook*, pp. 271–294, 2010.

[10] G. Littlewort, M. Bartlett, and K. Lee, "Automatic coding of facial expressions displayed during posed and genuine pain," *Image and Vision Computing*, vol. 27, no. 12, pp. 1797–1803, 2009.

[11] J. Hamm, C. Kohler, R. Gur, and R. Verma, "Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders," *Journal of neuroscience methods*, vol. 200, no. 2, pp. 237–256, 2011.

[12] Y. Tian, T. Kanade, and J. Cohn, "Recognizing action units for facial expression analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 2, pp. 97–115, 2001.

[13] C. Chuang and F. Shih, "Recognizing facial action units using independent component analysis and support vector machine," *Pattern recognition*, vol. 39, no. 9, pp. 1795–1798, 2006.

[14] J. Bazzo and M. Lamar, "Recognizing facial actions using gabor wavelets with neutral face average difference," in *Automatic Face and Gesture Recognition, 2004. Proc.. Sixth IEEE International Conference on*. IEEE, 2004, pp. 505–510.

[15] R. von Laban, *Principles of Dance and Movement Notation*. Macdonald & Evans, 1956.

[16] B. Ramadoss and K. Rajkumar, "Semi-automated annotation and retrieval of dance media objects," *Cybernetics and Systems: An International Journal*, vol. 38, no. 4, pp. 349–379, 2007.

[17] R. Birdwhistell, *Kinesics and context: Essays on body motion communication*. University of Pennsylvania press, 1970, vol. 2.

[18] M. Ptaszynski, J. Maciejewski, P. Dybala, R. Rzepka, and K. Araki, "Cao: A fully automatic emoticon analysis system based on theory of kinesics," *Affective Computing, IEEE Transactions on*, vol. 1, no. 1, pp. 46–59, 2010.

[19] Q. Nguyen and M. Kipp, "Annotation of human gesture using 3d skeleton controls," in *Proc. of the Seventh International Conference on Language Resources and Evaluation, LREC*. Citeseer, 2010.

[20] M. Kipp, "Anvil-a generic annotation tool for multimodal dialogue"," *7th European Conference on Speech Communication and Technology*, 2001.

[21] M. Kaltenbrunner and R. Bencina, "reactivision: a computer-vision framework for table-based tangible interaction," in *Proc. of the 1st international conference on Tangible and embedded interaction*. ACM, 2007, pp. 69–74.

[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[23] E. Velloso, A. Bulling, and H. Gellersen, "MotionMA: Motion modelling and analysis by demonstration," in *Proc. of the 31st SIGCHI International Conference on Human Factors in Computing Systems*, 2013.

[24] J. Cohen *et al.*, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[25] J. L. Fleiss, B. Levin, and M. C. Paik, "The measurement of interrater agreement," *Statistical methods for rates and proportions*, vol. 2, pp. 212–236, 1981.

[26] T. Schmidt, S. Duncan, O. Ehmer, J. Hoyt, M. Kipp, D. Loehr, M. Magnusson, T. Rose, and H. Sloetjes, "An exchange format for multimodal annotations," in *Multimodal corpora*. Springer, 2009, pp. 207–221.