Human nonverbal behavior multi-sourced ontological annotation

Boris Knyazev
Bauman Moscow State Technical University
5, 2-nd Baumanskaya Street
Moscow 105005, Russia
+7 (916) 027-6136, +7 (499) 263-6739
bknyazev@bmstu.ru

ABSTRACT
In this paper we introduce the current results of an ongoing three-year research and development project on automatic annotation of human nonverbal behavior. The present output of the project is a tool that provides algorithms and graphical user interface for the generation of ground-truth data about the subset of facial and body activities. These data are essential for the experts who are committed to unraveling the complexity of the linkage between the psychophysiological state and the nonverbal behavior of a human. Our work relied on a Kinect sensor, which computes depth maps together with the coordinates of the body joints and facial points. Local binary patterns are then extracted from the regions of interests of a facial video, which are either spatio-temporally aligned with the depth maps or calculated using the Active Shape Model. Another key idea of the proposed tool is that the extracted feature vector is semantically associated with ontological concepts in perspective providing annotations for most of the nonverbal activities.

Categories and Subject Descriptors
D.1.3 [**Programming Techniques**]: Concurrent Programming – *Parallel programming*. I.2.4 [**Artificial intelligence**]: Knowledge Representation Formalisms and Methods – *Semantic networks*. I.2.4 [**Artificial intelligence**]: Vision and Scene Understanding – *3D/stereo scene analysis*, *Video analysis*.

General Terms
Algorithms, Performance, Design, Experimentation, Verification

Keywords
Nonverbal behavior annotation, Kinect, ontology, LBP

1. INTRODUCTION
Experts in many highly demanding fields, including security, robotics, medicine and psychology require a tool that would provide them with comprehensive statistics of human nonverbal behavior (NVB), which includes, but is not limited to, kinesics (facial and body activities) and proxemics (spatio-temporal characteristics). Coupled with respective psychophysiological (inner) states these statistics could assist them in finding a linkage between these inner states and the nonverbal behavior of a human, a "black box" with essential data (the block ***L***, fig.1). However, there are two serious, closely related problems of unraveling the complexity of this block to be aware of. (1) How to measure and objectify the psychophysiological state – if such methods as measuring the facial movements of a person, measuring the parameters of his or her autonomic nervous system, just questioning a person or others (e.g., analyzed in [1]) are conclusive. (2) The diversity of theories (e.g., presented in the classic works [2-4]) of whether and how the inner state and the NVB of humans are linked, what is the origin of this linkage, how to measure it and many other continuing and sometimes ill-defined controversies.

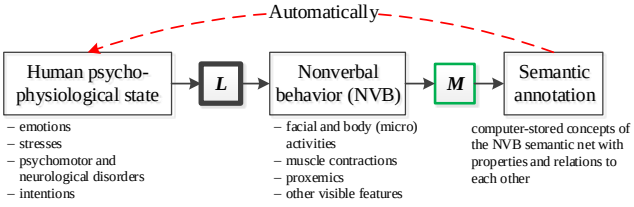

Figure 1. Fundamental problems that our work concerns. The green block (*M*) is a purpose of this work. The prospective possibility of building the black block in bold (*L*) and constructing the red dashed arrow are the motivation behind this project.

These fundamental problems are beyond the scope of this work in which important is the fact that the NVB is informative and the results received here can be applied to resolve them in the future. Facial and body activities are an innate necessity of humans and the features of these activities depend on age, gender, environment circumstances, biorhythms and many other factors. Besides, they reflect the state of physical health, the level of motor, psychological and intellectual development [3, 4]. Altogether, the facial and body activities and nonverbal behavior as a whole are an integral individual characteristic.

For instance, given a frequency and intensity of hand and finger movements (features of nonverbal behavior) it can be inferred if the person suffers from a neurological disorder. The similar could be concluded about a human who might be healthy in general, but endures an emotional stress or is violence-prone, which could be concealed to the naked eye, but is theoretically apparent for a detailed NVB analysis. One of the ways to make it practically

Conference'10, Month 1–2, 2010, City, State, Country.

apparent is to build a linkage between NVB and its semantic annotation (the block ***M***) in the first place. Then, assuming that an objective measure of human psychophysiological states exists (see the first problem above), by collecting substantive statistics between these inner states and respective semantic annotations we will be able to build the block ***L***. In the end, by providing just annotations of the human nonverbal behavior, his or her psychophysiological state would be classified automatically.

This work presents a tool for automatically building the block ***M*** based on three pillars: (1) an interface familiar to prospective users; (2) capacity to receive and extract information about the body and facial features from various sources (sensors); (3) annotations and statistics associated with them should be as informative as possible. This tool provides algorithms and graphical user interface (GUI) for the generation of ground-truth data about the subset of facial and body activities, which includes dynamic and static characteristics of eyes, eyebrows, lips, hands, elbows, shoulders, head, trunk, knees, feet and ankles. These ground-truth data represent semantic annotations of sequential groups of video frames, called *segments*.

In this paper, in section 2 we present a brief overview of related works on media annotation. Next, in section 3 we explain the tool structure and design, and which models, algorithms, libraries and hardware have been used to develop our tool. In section 4 we deliver the performance and the error rates of recognition of the subset of facial and body activities. Finally, in section 5 we discuss these results, giving further suggestions on their improvements.

2. RELATED WORK

The numerous media annotation tools that have appeared over the past decade are comprehensively surveyed in [5, 6]. These tools provide different understandings of how to annotate one or more of the following media formats: video, image and audio. ELAN[1] and The Video Annotation Research Tool[2] (ANVIL) [5] are professional instruments for the manual creation of video and audio annotations which offer multi-layered hierarchies of object types. The principal disadvantage of the non-automatic media annotation has always been known – it is a laborious, expensive and not flawless job. The Semantic Video Annotation Suite[3] (SVAS) and The Video Image Annotation Tool[4] (VIA) are one of the first steps towards its automation using MPEG-7 descriptors and user-loaded ontologies respectively [6] (table 1). These four tools also support export/import functions in one of the XML/RDF/MPEG-7-based formats. A number of crowdsourcing projects on image and video annotation have significantly reshaped the annotation process making it more effective and low-cost. The Video Annotation Tool from Irvine, California[5] (VATIC) focuses on an online interactive interface to annotate context-independent dense video scenes using adaptive object tracking [7].

[1] http://tla.mpi.nl/tools/tla-tools/elan/

[2] http://www.anvil-software.org/

[3] http://www.joanneum.at/en/digital/products-solutions/semantic-video-annotation.html

[4] http://mklab.iti.gr/project/via

[5] http://web.mit.edu/vondrick/vatic/

Table 1. Some of the capabilities covered by the existing tools.

OWL or other semantics	Supported media formats			Shared work	Automatic mode
	Video	Image	Audio		
VIA					VIA (segmentation)
SVAS					SVAS (SIFT descriptor)
	ANVIL, ELAN		ANVIL, ELAN		
	VATIC			VATIC (Amazon Mechanical Turk)	VATIC (HOG descriptor)

Meanwhile, in our work the stress is on an efficient solution for a specific, limited range of object categories – nonverbal activities of one human. We do not claim to develop a system capable of the automatic annotations generation of any video and in any context – a rather impossible task today. However, having limited the context, automatic annotation becomes possible, for example, using three-dimensional sensors in conjunction with holistic and local feature extraction algorithms.

We could still exploit the export/import functions and develop custom statistics, visualization, recognition and other tools (fig. 2). Despite that, there are a couple of important bottlenecks that hardly could be resolved using this kind of approach. First, we have a great deal more information to keep in the export/import project file than those tools are supposed to read and render in their interface. This includes, for example, the fact that besides video and audio annotation, experts need to annotate other media types, like collections of still images, depth maps and, perhaps, some others in the future; or the fact that they would like the annotated segments to be associated with the concepts of a human ontology and fully benefit from this. Second, we would like to integrate and further develop our custom functionalities, like action units charts, shared annotations creation and editing, specific tables, statistics calculations and diagrams, directly to the interface of our annotation tool. We believe this to be the key factor to make an overall user experience more pleasant and effective. Furthermore, we want our tool to be a part of another more scalable system which we are designing for human verbal and nonverbal behaviors research.

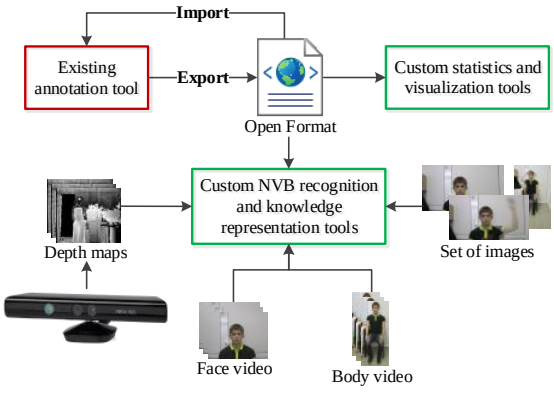

Figure 2. A possible, though impractical, solution using one of the available annotation tools.

For all these reasons, and from table 1, it is clear that existing multipurpose annotation tools, even though altogether they cover most of the functions we need, are difficult to use solely to build the block ***M*** (fig. 1). We have developed a tool for the automatic annotation of a limited range of human activities with a specific GUI, an export/import package format, implemented computer vision algorithms and developed a human ontology. Thus, our work is a further contribution towards automation of the media annotation process which results in the generation of semantically rich ground truth data about the subset of facial and body activities as well as about other human nonverbal behaviors in the future.

3. TOOL

3.1 Structure and interface

The nonverbal behavior recognition, annotation and analysis tool has two secondary and one primary modules:

- the module for dataset recording;
- the central shared storage module;
- the expert software (fig 3).

3.1.1 Dataset recording

The first one is used to collect the datasets in which each sample is associated with a person and consists of standard two-megapixel videos of his or her face and body (in the sitting position), and an XML file with three-dimensional coordinates of the twenty body joints and six facial action unit points (only in new datasets) computed by Kinect. Both face and body videos are used for better user experience, but in addition, the former can also be used to extract facial points (e.g., when facial points from Kinect are not available) and the latter one – for the prospective motion extraction. Although Kinect allows recording RGB video frames as well, the number of frames per second was volatile, and in the end we found it reasonable to keep and visualize videos only from video cameras while preserving only depth maps from Kinect.

Two videos and a depth map need to have a global timeline to manage annotations, because it is difficult – if not impossible – to record videos and data from Kinect synchronically. The dataset recording module has a special manual synchronization mechanism which allows writing time offsets of all media sources to the export/import package file.

3.1.2 Central shared storage

The second module generates an export/import package consisting of one dataset sample and the main XML file describing the contents of the package. This module can also add audio files, some sets of still images, Stromotion frames (2-3 frames unified in one) and sometimes a data file with two-dimensional facial points detected in advance from a face video. The package-describing file also stores nonverbal behavior annotations linked to the concepts, properties and relationships of the nonverbal behavior ontology which is stored in a separate .owl file. The central shared storage module and the expert software exchange with packages using the offline crowdsourcing approach. The idea is that every expert has its own priority, which is usually granted depending on the expert's qualification, and can annotate the chunk of video independently. The results are then sent to the central storage where they are merged depending on these priorities. It means that the results of an expert with lower priority may be overwritten with the results of the ones with higher priorities.

3.1.3 Expert software

3.1.3.1 Interface

The expert software is developed using Microsoft .NET Framework 4.0 with all pros and cons of its Windows Presentation Foundation (WPF) subsystem. Behind its relatively simple appearance, it has non-trivial interface styles and business logic (fig. 3). The main panels of the expert software's layout are a media player, a control to work with various types of video frames, a panel with colored segment tracks and a table which duplicates the currently active (selected) track, providing more details about it. Each panel is flexible in size, and can be shown in a separate window, and then docked to its assigned column and row of the layout's main grid.

3.1.3.2 Automatic annotation

The principle purposes of the expert software, a primary module of our tool, are to load a package created in the central shared storage module, to run an automatic annotation process on the selected global timespan, and to manage annotations. The other functions, e.g., to write statistical reports, to plot audio and action units charts, to analyze speech, etc., are not a subject of this work.

Each colored segment track corresponds to only one static (e.g., head position, gaze direction) or dynamic (e.g., eye blinking) nonverbal behavior type. One such track represents a collection of elementary human behaviors of a respective type – *segments*. For instance, a track 'Vertical gaze direction' could render segments like 'up, 'straight', 'down' or 'undefined' (for closed eyes). The overall number of the static and dynamic features is continuing to grow because of the new psychophysiological and ontological works that better reflect the human inner states on a computer, and new object recognition methods that allow us to detect and track human patterns more efficaciously. A partial description of these behavioral types is provided in the next chapter.

3.2 Nonverbal behavior ontology

To make automatically generated annotations more informative and logically consistent, and to increase overall performance of the tool we have considered existing knowledge representation approaches.

Nonverbal behavior represents a multithreaded temporal process of actions, mimics, poses, gestures, etc., so that there can be several hundred static and dynamic types, and several dozen their attributes changing in time and space simultaneously. For better knowledge creation, retrieval and reasoning semantic networks, and in particular ontologies, proved to be effective and should be developed [8]. Ontologies can also boost the accuracy rates of recognition of particular human activities [9]. They can be implemented as a descriptive logic, an RDF/OWL based document or other knowledge representation model. For instance, ontologies developed using the Web Ontology Language (OWL) were successfully used in [6, 10, 11].

The Behavior Markup Language[6] (BML) standard, part of the SAIBA (stands for Situation, Agent, Intention, Behavior, Animation) Framework, was specifically designed to control the verbal and nonverbal behavior of a human, or more generally, of an embodied conversational agent (ECA). This is an XML-based

[6] http://www.mindmakers.org/projects/bml-1-0/wiki

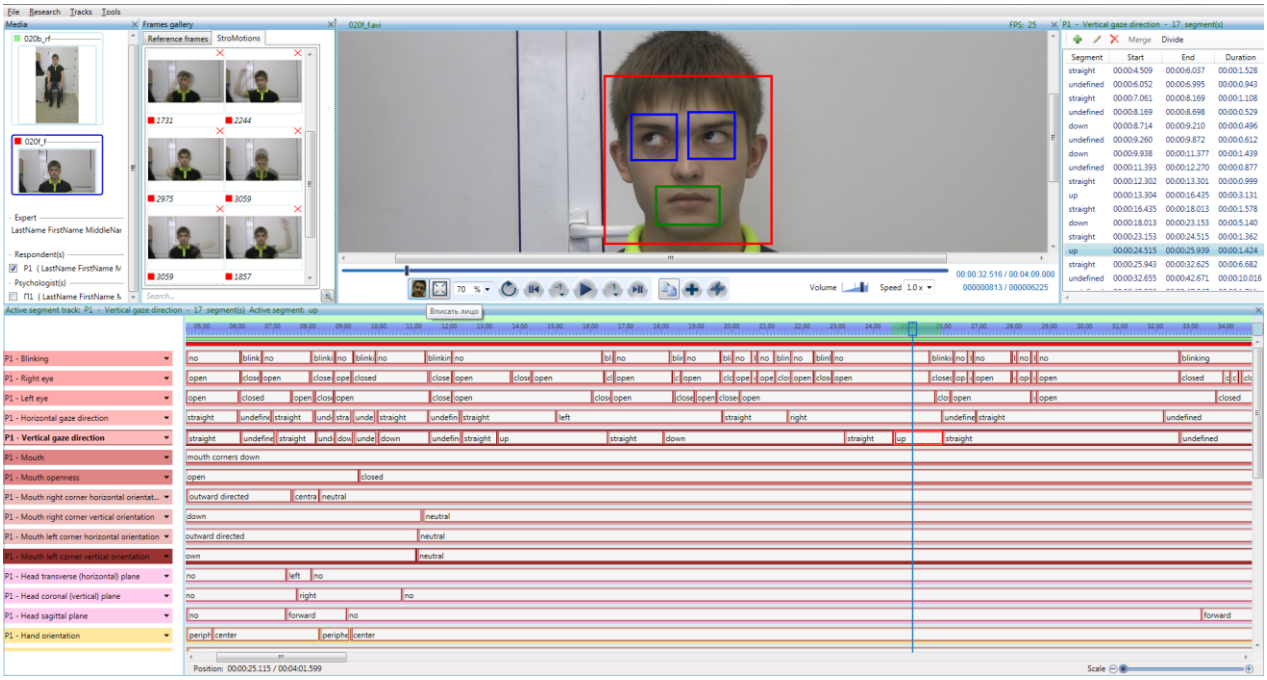

Figure 3. Interface of the expert software of our human nonverbal behavior annotation tool. It allows users to manage automatically generated annotations and send the results to the central storage where they are merged depending on the experts' priorities. The selected (active) segment track is 'Vertical gaze direction', the selected (active) segment is 'up'.

standard which provides virtually all the elements and attributes necessary to describe multimodal behaviors in time. In addition, if required, it can be complemented with custom behaviors designed as new XML elements and attributes.

Open Biological and Biomedical Ontologies[7] Foundry (OBO Foundry) attempts to collect and propagate shared ontologies, and it provides several solutions related to human anatomy (Foundational Model of Anatomy), actions (Neuro Behavior Ontology), diseases (Human disease ontology, The Human Phenotype Ontology) and others. The Virtual Humans Ontology[8] (VHO) developed by AIM@SHAPE provides a detailed vocabulary for human body modeling and analysis. The human nonverbal behavior ontology certainly should be based on some of these OBO/OWL works as they are very comprehensive. In our work, however, at this stage we required a simpler ontology which could easily integrate into the existing environment where other human related ontologies exist in order to sense how we could benefit from it.

According to [8] OWL-DL is the most expressive decidable sub-language of OWL. Among many ontology designers, Protégé[9] has proved to be the leading one, being one of the most user-friendly and powerful instruments [12]. It also includes DIG (Description Logic standard) compliant reasoners FaCT++ and Pellet, supports other useful plugins and OWL extensions (e.g., HermiT), time properties ('duration', 'before', 'overlaps', etc.) and the rules and class expression editors.

In the nonverbal behavior ontology, which we have developed using Protégé, there are four base body features: trunk, joint, limb and bone, and nine facial features: cheek, eye, eyebrow, eyelid, eye pupil, forehead, jaw, lip, mouth corner and nose. These concepts are defined using a basic OWL construction: <owl:Class rdf:about="#BaseFeature"/>, where 'owl' is the 'http://www.w3.org/2002/07/owl#' namespace (fig. 4).

Ten derived body features: head, arm, elbow, hand, leg, knee, hip, foot, ankle and finger are multiple-derived from the base body features using the 'rdfs:subClassOf' structure. Some facial features individuals are constrained with the custom object property 'partOf', for example as following:

<NamedIndividual rdf:about="#EyelidLeft">
    <rdf:type rdf:resource="Eyelid"/>
    <NVB:partOf rdf:resource="#EyeLeft"/>
</NamedIndividual>,

where 'NVB' is the namespace of our nonverbal ontology.
In addition, several dozen static and dynamic descriptive concepts and properties are also defined, like orientations, directions, states, gestures and facial motions, and others, which are indispensable to create nonverbal instances, such as hand gestures, poses, eye blinking, emotional states, etc. Most of the concepts are equivalent to the collections of other simpler concepts restricted using one or more of the 'owl:unionOf', 'owl:intersectionOf' and 'owl:oneOf' properties. For instance, a head represents a collection of certain facial features; a body – a collection of the body features, and so on. Base facial features concepts also could be collections of other lower-level concepts.

The components of the feature vector are the input of the ontology module, whereas, the ontology instance(s) satisfying the query conditions is the output. Extraction of the feature vector and its components will be discussed in the next section in detail.

[7] http://www.obofoundry.org/

[8] http://www.aimatshape.net/resources/aas-ontologies/virtualhumansontology.owl

[9] http://protege.stanford.edu/

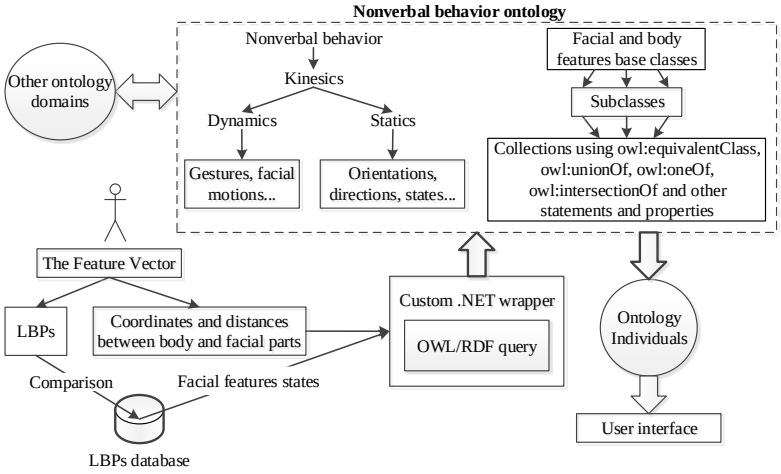

Figure 4. The ontology module of our nonverbal behavior annotation tool. The input of this module is the components of the feature vector, the output – ontology instance(s) satisfying the query conditions.

Our custom ontology API wrapper for .NET Framework, based on the publicly available libraries[10], processes queries in one of the available formats (currently it is SPARQL). In our work queries contain coordinates and distances between the body and facial parts or already computed linguistic statements reflecting spatio-temporal properties of these parts, but, ideally, this query is supposed to contain as many visual and contextual properties of an object as possible: texture, orientation, shape, colors, context, etc. The output should be the most relevant instance or a list of the instances with respective measures of relevance. We are also working on nonverbal behavior time properties, but currently our tool does not support querying with temporal conditions.

3.3 Feature vector extraction

3.3.1 Skeleton points

To automate the human nonverbal behavior annotation process, we rely on three media sources contained in a package: the depth data from the Kinect sensing device (stored in the XML format) and videos of a face and a body (in any format supported by Windows Media Player).

Kinect has increasingly become more powerful while remaining affordable. Previously it has proved to yield human skeleton recognition results as accurate as 85-90% and higher in particular conditions (illumination, the distance from the object to the device, etc.) [11, 13]. These results are similar to the ones produced by the state-of-the-art motion-capture-based methods, which may include video background subtraction in the case of a non-complex scene followed by the detection of skeletal and joint points using complex differential, statistical and contextual detectors (see [14] for details). Without Kinect our automatic annotation software would be significantly more resource-consuming to be built.

Let I be the input signal and S be the target feature vector of size N calculated over the input 3-D cube for a given time token t:

$$S = f(I, dx, dy, dz, t)\,. \tag{1}$$

To construct this vector we can use both holistic and local feature extraction:

$$S = \{S_{Holistic}, S_{Local}\}\,. \tag{2}$$

Holistic methods imply globally computed filters (F) and transformations (T), such as the Gabor and Gaussian wavelets, the Fourier, Haar or discrete cosine transforms of input data I.

$$S_{Holistic} = (F * I) * T\,, \tag{3}$$

where * - is some operator. Transformations F and T should be designed in a way to derive and maintain spatiotemporal information, for example, like the 3-D Fourier transform in [15] or the motion field, the extension of the optical flow, in [16]. Recognition of crowd activities and dense scenes are of particular research interest today, but in this work we are only concerned about activities of one person at a time.

In this work we assume that the F and T transformations are successfully done by Kinect and its software development kit (SDK), however, in future studies, in order to increase skeleton recognition rates, it would be reasonable to update available or develop our own transformation functions.

3.3.2 Facial points

In order to build the local features vector S_{Local}, we have to locate and describe the regions of interests (ROIs) in the first place. In our tool the ROIs are: eyes, eyebrows and corners of the mouth. To locate the characteristic points of these regions, Kinect could be used again. Alternatively, the Active Shape Model, implemented for example in the Stasm[11] library, might provide competitive results. Our experience has shown that in some situations (depending on illumination, orientation and scale) these points are better computed by one method and in others – by the second one. Therefore, currently we are working on averaging the facial points' coordinates preliminarily centering them by the eye pupils and translating them into world coordinates.

Various sorts of visual descriptors can be exploited to describe and classify these regions of interest: local binary patterns (LBP), histograms of oriented gradients (HOG), histogram of optical flow (HOF), scale-invariant feature transform (SIFT) and their manifold extensions. For instance, in [9] the HOG for local regions of interest in conjunction with the support vector machine

[10] For example, https://bitbucket.org/dotnetrdf/dotnetrdf

[11] http://www.milbo.users.sonic.net/stasm/

(SVM) classifier was used to predict the object's path. Note, however, that the size of the HOG depends on the size of the image, cells, and the number of bins. If the image size is 32×16, then the HOG size can reach 3780 elements and more. On the other hand, the LBP and its extensions provide fairly high results for emotions recognition [17] and face recognition with the rate of up to 80%, even for more realistic facial expressions and under face orientation challenges. The size of the uniform LBP is $P = P(P-1) + 2 = \{P = 8\} = 58$, where P – the number of neighboring image pixels to compute one binary code. In addition, there is one label for the remaining non-uniform patterns, so that the overall size becomes equal 59. We are working on the implementation of different extensions of the available visual description algorithms with more discriminative power, but at present, on a GPU we implemented only a classic version of the LBP operator with (P, R) = (8, 1), where R – radius of a circle in pixels. Despite its computational simplicity, this version is rarely outperformed by the other cognate algorithms, and there are extensions, for example, the center-symmetric local binary pattern (CS-LBP), which proved to be competitive using these values of radius and this number of neighboring image pixels [18].

To describe the regions of interests they are divided into 2-4 areas W, for which the local binary pattern is extracted. Thus, the local features vector:

$$S_{Local} = S_{LBPs} = \{LBP_{Eyes}, LBP_{Eyebrows}, LBP_{Lips}\}. \quad (4)$$

Consequently, the resulting vector S is following:

$$S = \{S_{Kinect}, S_{LBPs}\}. \quad (5)$$

In this work we use all twenty skeletal points calculated by Kinect, we also have eight LBPs for eyes (each eye is divided into four areas W), six LBPs for eyebrows (we also considered the glabella) with two areas W for each part, and four patterns for two areas W of each corner of the mouth. Thus, considering the size of each LBP equals 59, the overall size of our feature vector S is $20 + 4 \cdot 2 \cdot 59 + 2 \cdot 3 \cdot 59 + 2 \cdot 2 \cdot 59 = 1082$.

Comparison of extracted LBPs with the LBP database (fig. 4) can be done using the Euclidean distance, the Hamming distance, the Kullback–Leibler distance or the Fisher's linear discriminant. For simplicity and low computational cost in our tool currently LBPs are compared using the Euclidean distance.

3.3.3 Performance considerations

The size of an area W for the LBP computation is decided to be 32×16 pixels, because the maximum size of thread blocks in a graphics processor G84 (and in many other GPUs), which was used for computations, is 512; the warp size is 32 threads. Thus, each thread block of the GPU is able to calculate the LBP in a whole and independently of other blocks, and without any warp divergence, which is essential for GPU computing. In [19] the speedup was at least 30 times compared to the CPU version of the regular algorithm and up to 100 times for its variations.

In this work we have not specifically measured our performance increase yet, however, as presented below, the formulation of the complexity Q of this algorithm for the area W of size X×Y together with experimental observations are calling for use of the GPU version:

$$Q = \frac{X \cdot Y \cdot P}{M}, \quad (5)$$

where M – the number of threads executing in parallel, which would ideally equal W. In that case Q would converge to P.

Algorithm 1. Pseudo code for LBP computation on a GPU

Require: *TILE_W* = 32, *LBP_W* = 3, *indices* = {0,1,2,5,8,7,6,3}, *input*, *output*, *length*
Result: uniform decimal codes for the input image

```
1:  __global__ function declaration
2:  begin
3:      size = LBP_W ^2
4:      col, row, t = current thread X,Y and linear positions
5:      if t > length then
6:          return
7:      __shared__ partialLBP[32x16] = read from the input
8:      synchronize threads calling __syncthreads()
9:      lbp_circle[size] = read current circle from partialLBP
10:     threshold = central value of lbp_circle
11:     if lbp_circle[0] >= threshold // first loop of cycle
12:       dec_code += 0x80
13:     for i = 1:size-1 // main loop
14:         if lbp_circle[indices[i]] >= threshold then
15:             dec_code += 0x80 >> i
16:         if (bit value at i != bit value at (i-1)) then
17:             ++transitions
18:         if transitions > 2 then
19:             dec_code = 0
20:             break
21:     synchronize threads calling __syncthreads()
22:     write the dec_code value to the output
23: end
```

3.4 Kinect challenges

Kinect is a perfect device that can greatly lessen our work aimed to automatically annotate human activities by giving skeleton and facial points. In spite of that, there is a nuance of using this sensor: temporal occasional loss and misdetection (with occlusions like chair legs) of the human's lower body points (knees, feet and ankles) in the sitting position.

Meanwhile, for psychophysiological experts, arrangement of the knees of a person is highly informative, so to classify their positions more accurately it was reasonable to train a classifier. To promptly know how we could benefit from it and because of the extremely limited learning and testing databases, a simple perceptron neural network (ANN) with one hidden layer was trained. This ANN was learned by a specifically collected dataset of humans in the sitting position, which consisted only of sixteen one minute video samples. Each of the four training subjects was sitting on a chair for one minute with one of the four knees states (crossed legs, knees more and less than shoulder-width apart, and knees together) making slight natural movements. Obviously this is not enough to build a robust classifier and one of the further steps in this research project should be resorting to the active classification methods and learning algorithms for which very few examples would be enough.

Our resulted ANN had eleven hidden neurons, six input states (distances between knees, feet and hip joints) and four output

states corresponding to the aforementioned knees positions. To examine this classifier a different dataset described below was used. It also turned out that migrating to the new Kinect SDK and setting up its smoothing parameters also had a positive effect on the classification rates.

4. EXPERIMENTS AND RESULTS

To evaluate the performance and error rates of our tool we collected an experimental dataset consisting of five packages (one package per person) with facial (1920×1080, 25 fps) and body (1920×1080, 30 fps) videos, a Kinect depth map (640×480, close to 30 Hz) and other secondary files (table 2).

Table 2. Experimental dataset. The duration format is {minutes:seconds}.

Package/ Subject	Sex	Face	Body	
		Video duration	Video duration	Kinect depth map duration
1	M	4:09	4:02	4:05
2	F	5:32	5:28	5:09
3	M	6:46	6:46	6:40
4	F	5:54	5:53	5:28
5	M	5:44	5:43	5:22

Each subject was asked to show static and dynamic nonverbal behaviors from the nonverbal behavior ontology (described in section 3.2) using his or her facial and/or body features.

Performance of an automatic annotation process was measured using the relative time complexity (speed) of the algorithms:

$$R = \frac{Dur_A}{Dur_S}, \quad (6)$$

where Dur_A – duration of an automatic annotation process, Dur_S – duration of an analyzed sample, which equals the duration of videos minus their respective time offsets necessary for synchronization (usually equals the minimum duration among facial and body videos and a recorded Kinect's depth map).

Accuracy rates of an automatic annotation process were estimated using false positive (*FP*) and false negative (*FN*) errors. Firstly, automatic annotation of all nonverbal behaviors was run, and then three experts with different priorities annotated the samples manually. After that their results were compared with the ones produced automatically. There are four possible cases: *FP* – false positive when an automatic process detected a unit (facial or body activity) that had not been annotated by an expert; *TP* – true positive when both an automatic process and an expert detected a unit, *FN* – false negative when an automatic process did not detect a unit that had been annotated by an expert, *TN* – true negative when both an automatic process and an expert did not detect a unit. In this way,

$$FP = \frac{T_{FP}}{T_{FP} + T_{TP}}, \quad (7)$$

where T_{FP} – total time duration where *FP* cases are present, T_{TP} – total time duration where *TP* cases are present.

$$FN = \frac{T_{FN}}{T_{FN} + T_{TN}}, \quad (8)$$

where T_{FN} – total time duration where *FN* cases are present, T_{TN} – total time duration where TN cases are present.

In our experiments the true beginning and end points of the nonverbal behaviors were estimated by experts. In the case when two or more experts did not agree, the annotations of the expert with a higher priority were considered to be favorable.

The results for nonverbal behaviors grouped into 10 categories are presented in table 3 with worst cases collected for all static and dynamic activities of the respective facial or body features. For instance, for the body parts positions such groups would include static positions in all three planes of a body (sagittal, coronal, transverse), and may also include relativity to each other and other specific properties. In table 3, *N* is the average number of annotated segments of a particular nonverbal group per package.

Table 3. Performance and error rates for ten NVB groups

Nonverbal behavior group	*R*	*N*	*FP*	*FN*
Eyes closed/opened states	3.92	178	0.35	0.39
Eyes blinking	1.7	115	0.30	**0.42**
Gaze directions	2.13	364	0.37	0.27
Eyebrows states	2.51	134	**0.43**	0.37
Lips and mouth corners states	**4.21**	164	0.24	0.31
Head position	0.87	36	0.40	0.29
Trunk position	0.92	68	0.39	0.30
Arms, hands and elbows position	0.90	74	0.34	0.35
Knees position	1.27	177	**0.15**	**0.25**
Feet and ankles position	**0.83**	29	0.30	0.38

Among all the nonverbal behavior groups the best case corresponds to 0.83 for performance, which means that an analysis was running faster than a video sample was playing, 15% for false positive and 25% for false negative errors, and in the worst case the numbers are respectively 4.21, 43% and 42%.

5. DISCUSSION

Both performance and recognition rates of the tool turned out to be far away from the state-of-the-art results (80-95 % [14]) for human activities recognition and classification. There are several possible reasons for that. First, the collected statistics are not completely reliable, because of the limited training and experimental datasets. To correct this, we are continuing to update our dataset with additional more representative samples. We also should try already existing datasets, like ChaLearn[12] datasets or the Carnegie Mellon University Motion Capture Database[13]. Alternatively, we could resort to the active classification methods and learning algorithms, for which very few examples would be enough.

Second, we heavily rely on the Kinect sensor, which although perfectly suitable for game practice, might not be as effective for gathering ground truth data and building objective human nonverbal statistics. The scene in our research was not dynamic, so we could reach higher accuracy using advanced background subtraction methods followed by human skeleton extraction.

[12] http://gesture.chalearn.org/

[13] http://mocap.cs.cmu.edu/

Third, the ASM model used for the facial points extraction is not as accurate and robust to image irregularities (such as orientation of the object, illuminations and contrast changes) as our objectives require. To improve it, we might need to focus on the other 2-D/3-D facial models and landmarks calculations, or to utilize the Kinect 3-D facial points, as well as the body ones. Additionally, the classification of local binary patterns would be more precise if video frames were preprocessed using a bank of Gabor filters (operator *F* in the formula (3)). Having collected facial landmarks accurately, we could then, perhaps, receive more correct results by implementing the algorithms computing different extensions of the LBP, HOG, SIFT, etc. Vectors comparison should be replaced with the Fisher's linear discriminant instead of the simple Euclidean one.

Nevertheless, we believe that successful application of the state-of-the-art methods would not be enough to solve our problem – an automatic generation of ground truth data about human facial and body features behaviors. To succeed, the human behavior should be modeled using one of the multilevel dynamic models which allow detailed reflection of the reality. Integrated with a human ontology and more powerful visual descriptors this model then could make our tool more reliable.

6. CONCLUSIONS

In this work, the current results of an ongoing research and development project on a tool for automatic annotation of human nonverbal behavior are presented. To develop this tool, various media annotation solutions, computer vision and knowledge representation methods were examined. In our work, the nonverbal behavior ontology was developed and a low dimensional Kinect-LBP-based feature vector computing the human body and facial features was built. The effectiveness of this tool was evaluated using the performance and error rates. Despite relatively poor results, this work introduces a tool for the automatic annotation of the subset of nonverbal behaviors decreasing the need for human resources, increasing the speed of annotation and making an overall user experience more productive. To improve the results, a number of complex updates discussed above should be applied. If applied in perspective this tool could address the yet unsolved fundamental psychophysiological problems as well as solve present daily problems and improve human-computer interaction.

7. ACKNOWLEDGMENTS

This work was supported by the Bauman Moscow State Technical University (Russia) graduate scholarship.

8. REFERENCES

[1] Harrigan, J., Rosenthal, R., and Scherer, K., 2008. New Handbook of Methods in Nonverbal Behavior Research. *Oxford University Press*, 536 pages.

[2] Birdwhistell, Ray L. 1970. Kinesics and Context: Essays on Body Motion Communication. Philadelphia: University of Pennsylvania Press.

[3] Ekman, P., Friesen, W. V. 1969. The Repertoire of Nonverbal Behavior: Categories, Origins, Usage and Coding. *Semiotica*, 1, 49- 98.

[4] Ilin, E. P. 2003. Psychomotor human organization: Textbook, 1st edition, *Spb*.

[5] Rohlfing, K., Loehr, D., Duncan, S., et al. 2006. Comparison of multimodal annotation tools: Workshop report. *Gesprächforschung* , 7, 99-123.

[6] Dasiopoulou, S., Giannakidou, E., et al. 2011. A survey of semantic image and video annotation tools. In *Knowledge-driven multimedia inf. extr. & ontology evolution*, 196-239.

[7] Vondrick, C., Patterson, D., Ramanan, D. 2012. Efficiently Scaling Up Crowdsourced Video Annotation, *International Journal of Computer Vision*. Vol. 101, Issue 1, 184-204.

[8] Staab, S., and Studer, R. 2004. Handbook on Ontologies. *International Handbooks on Information Systems. Springer Berlin Heidelberg*.

[9] Akdemir, U., Turaga, P., and Chellappa, R. 2008. An ontology based approach for activity recognition from video. In *Proceedings of the 16th ACM international conference on Multimedia (MM '08)*. ACM, New York, NY, USA, 709-712. DOI= http://doi.acm.org/10.1145/1459359.1459466

[10] Chen, L., Nugent, C. 2009. Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems*, Vol. 5 Iss. 4, 410 – 430.

[11] Nekhina, A., Knyazev, B., Kashapova, L., Spiridonov, I. 2012. Applying an ontology approach and Kinect SDK to human posture description. *Biomedicine Radioengineering (ISSN 1560-4136)*. No.12, 54–60.

[12] Khondoker, M. R., Mueller, P. 2010. Comparing Ontology Development Tools Based on an Online Survey, *Proceedings of the World Congress on Engineering*. London, U.K.

[13] Shotton, J., Fitzgibbon, A., Cook, M. Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A. 2011. Real-Time Human Pose Recognition in Parts from Single Depth Images. In *CVPR '11 Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, 1297-1304.

[14] Aggarwal, K., and Ryoo, M.S.. 2011. Human activity analysis: A review. *ACM Comput. Surv*. 43, 3, Article 16 (April 2011), 43 p.

[15] Solmaz, B., Assari, S.M., Shah, M. 2012. Classifying web videos using a global video descriptor, *Machine Vision and Applications*, 1-13.

[16] Hu, M., Ali, S., Shah, M. 2008. Learning Motion Patterns in Crowded Scenes Using Motion Flow Field. In *Proc. International Conference on Pattern Recognition (ICPR)*.

[17] Moore, S., Bowden, R. 2011. Local Binary Patterns for Multi-view Facial Expression Recognition. In *Computer Vision and Image Understanding*, 115(4), 541-558.

[18] Heikkilä, M., Pietikäinen, M. and Schmid, C. 2009. Description of interest regions with local binary patterns. Pattern Recogn. 42, 3 (March 2009), 425-436. DOI=http://dx.doi.org/10.1016/j.patcog.2008.08.014.

[19] Liebstein, J., Findt, A., Nel, A. 2010. Texture Classification Using Local Binary Patterns on Modern Graphics Hardware, SATNAC, Spier Estates, Cape Town.