M5AIE: A Method for Body Part Detection, Tracking and Pose Classification using RGB-D Images

André L. Brandão, Leandro A. F. Fernandes (Co-advisor), Esteban W. G. Clua (Advisor)
Instituto de Computação
Universidade Federal Fluminense
Niterói, Brazil
Web page: http://www.ic.uff.br/~medialab/Andre/m5aie.html
Email: brandao@daad-alumni.de, {laffernandes,esteban}@ic.uff.br

Abstract—**The automatic detection and tracking of human body parts in color images is highly sensitive to appearance features such as illumination, skin color and clothes. As a result, the use of depth images has been shown to be an attractive alternative over color images due to its invariance to lighting conditions. However, body part detection and tracking is still a challenging problem, mainly because the shape and depth of the imaged body can change depending on the perspective. We present the M5AIE[1] that uses both color and depth information to perform body part detection, tracking and pose classification. The M5AIE method makes use of Accumulative Geodesic Extrema (AGEX), Affine-SIFT (ASIFT). Three different classifiers were applied in our study to analyze which would be the best for human pose classification. This method can be integrated to computer games that intend to use the Natural User Interface (NUI) paradigm.**

Keywords-**Body part detection; body part tracking; pose recognition; pose classification; background subtraction; classification algorithms.**

I. Introduction

Since 2010, important advances have been achieved in Computer Vision research, especially in gesture recognition. Those advances have created many new possibilities of applications of Human-Computer Interaction, health-care and digital games [1]. We developed the Jecripe [2] game that is designed for children with Down syndrome. The Jecripe game became a successful application and received different awards[2], being translated to five different languages[3]. This game consists of a set of different activities that stimulate different cognitive abilities. The stimulation of the imitation cognitive ability and the launch of low cost devices for Natural User Interface (NUI) motivated this study.

Low-cost capture devices of depth images promoted facilities in gesture recognition research. In 2010, Microsoft Kinect was launched, and it is described in [1]. Shotton et al. present how the device works and its applicability to digital games. The interaction with Kinect characterizes the NUI paradigm.

The context of body part detection and tracking requires information filtering to address only the necessary information. To filter the information, it is necessary to accomplish some tasks. The first task is to remove the most basic useless information, which is the background. Without the background, we can then handle the human body pixels. However, body part detection does not need all of the human body pixels. We decided to apply the Medial Axis transformation to filter an even larger amount of pixels. The Medial Axis provides the number of pixels that enable detection of the five main body parts. Then, a tracking method must be developed. We used a feature extraction and matching method to track each of the body parts from one image to the next image in a sequence.

Body part detection and tracking in image sequences is challenging because this task requires information filtering to bring about the use of less information. The resulting information must be structured because it will provide the detection of the body parts. The body parts are tracked with an algorithm, frame by frame, to store time sequence information. We use a feature extraction and matching algorithm as part of a tracking method because it compares two input images. Once there are human poses to be identified, we use the position of each body part in each image in a sequence to define the human poses that can be applied to classification algorithms for prediction purposes.

The contributions of this work[4] are as follows: (a) A comparison among different background subtraction algorithms; (b) The combination of the AGEX and ASIFT methods using aligned RGB and depth images for labeling five major defined

[1] The name M5AIE is an acronym for each of the used concepts in our approach: *M*edial *A*xis transformation, for data filtering; *A*dapted *A*GEX, for the body part detection; *A*SIFT, for the body parts tracking, *A*ligned *I*mages (RGB-D), and *E*stimation, also for tracking.

[2] Until December 2013, the Jecripe game received the Award from the Cultural Secretariat of Rio de Janeiro State and Best Accessibility Project Award from Guarulhos City. Jecripe was highlighted by the press in several communication vehicles such as radio and television programs, newspapers and websites.

[3] Jecripe is available at *www.jecripe.com* in Portuguese, Spanish, English, German and Turkish.

[4] This text is related to the first author's Ph.D. Thesis.

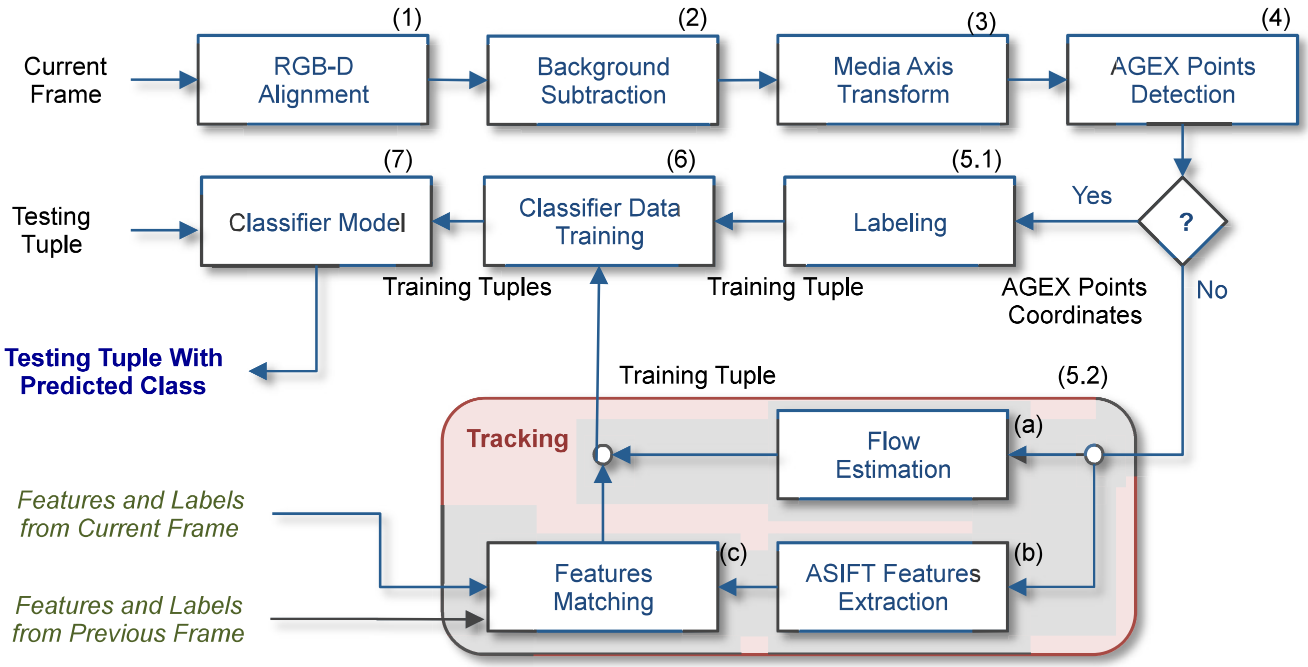

Fig. 1. Task Flow of the M5AIE method.

body parts (hands, feet and head); (c) Tracking each of the body parts using an adapted ASIFT matching algorithm; (d) Description of how different classification algorithms can be used in human pose classification in the digital games context; and (e) A comparative analysis of three classification algorithms in human pose classification.

Due to space limitation, this paper does not present a detailed description of the proposed method or results achieved. The full thesis [3] and the list of related publications, courses, pending submissions are available at *http://www.ic.uff.br/˜medialab/Andre/m5aie.html*.

II. Background Subtraction

The task of background subtraction is facilitated with the depth information that is available. It is, therefore, quite surprising to see that only a few studies on this subject can be found that use depth information for a background subtraction task. Before we make the background subtraction, we first align the depth and color information to have the correspondence of depth information of each colored point in the image (Figure I – (1)). In the background subtraction task (Figure I – (2)), we compared four background algorithms in different situations and chose the Minimum Background Subtraction Algorithm [4].

III. Body Part Detection and Tracking

The task of human body part detection and tracking is not trivial. Addressing the human body is challenging because its shape can be very different from one person to another. Additionally, humans have different skin colors, and clothes can vary in both their colors and shapes. These reasons, among others, make body part detection a complex task. Because body part detection can be used for body part tracking, certain aspects must be considered, such as the human skeleton and medial axis (Figure I – (3)). Even knowing the human skeletons profile, we must assume that there are many degrees of freedom [5].

Reliable results on body part detection and tracking tasks have been achieved by using depth information. Depth information outperforms intensity images in the sense that they intrinsically remove appearance features, such as the color of the skin, the color of the clothes and different background appearances, which can vary for different objects and colors [5]. Additionally, depth images provide extra information about the imaged objects, i.e., their actual geometry.

The objects geometry is given with the point distances between these objects and the sensor that forms a point cloud. The points of the point cloud can be used for body part detection, as vertices of a graph, and they can be connected with weighted edges. The weight of each edge is the Euclidian distance between the connected points. The generated graph can be used to detect body parts in the extremes of a graph. This approach is used in a method that is described by Plageman et al [5], for which the interesting points are called the AGEX points (Figure I – (4)).

The proposed solution is based on the key observation that once the body parts are detected in one frame, the same body parts can be used by matching methods for tracking each of them in the next frame. For this task, we describe the M5AIE Method, which detects and tracks the body parts. In the beginning of every image sequence that we use as input for our method, the person must stand in front of the sensor in the T-pose. We describe the T-pose a person with open arms and feet together on the floor. Then, the AGEX points are detected (Figure I – (4)) with the head and arms over the centroid and feet under the same centroid. Until the starting configuration stands, we can label the detected points as head, arms and feet (Figure I – (5.1)). If the starting configuration changes, then the detected points are tracked. The tracking method (Figure I – (5.2)) is composed by three stages: (a) flow estimation (b) ASIFT features extraction and (c) features matching. The results of the mentioned stages are combined to label the detected body parts.

IV. Human Pose Classification

Once we have detected and labeled body parts, the method is able to classify which pose the person is in a given moment. A classification task is composed by two stages: the construction of a classifier data training (Figure I – (6)) and the classifier model (Figure I – (7)). The method constructs the classifier data training (Figure I – (6)) with a number of training tuples as input. In our case, the training tuples are built by a sequence of coordinates of each of the body parts according to a grid. We manually classified the training tuples using possible movements that can be done in the Jecripe game. The classifier model (Figure I – (7)) receive as input a set of training tuples from the classifier data training. The classifier model constructs a model which is used to predict the class of a testing tuple. A testing tuple is given as input to the classifier model and the output of this model is the same training tuple, however, with its predicted class. It was not clear which would be the best classification algorithm to be used in our context.

The literature provided only a few studies that compared classification algorithms in the context of games. To the best of our knowledge, no work in the literature has made a comparison among the classification algorithms in human pose recognition in the context of games. In this task, we propose an analysis of classification algorithms that use the M5AIE method: the C4.5 Gain Ratio Decision Tree, Nave Bayes Classifier and K-Nearest Neighbor (KNN) Classifier. As a consequence of this study, the results can help researchers to choose among the selected algorithms for use in human pose classification in a digital games context.

In this work, the algorithms receive as input the labels and the locations of the body parts according to an $N \times N$ grid that is defined inside the bounding box that contains the whole body of the imaged subject. A bounding box was used to identify the cell number of the body parts. The bounding box provides the relative positions according to the detected human body.

V. Experiments and Results

The described approach was implemented in Python and was evaluated on real image sequences. The ASIFT algorithm was implemented in C++. We used the reference implementation provided by Morel and Yu [6]. To perform the distance transformation, we used OpenCV adaptive thresholding and other basic image processing procedures. The image sequences were collected using a Kinect sensor.

The classification algorithms were evaluated using the data mining tool WEKA 3.6.8. To adopt the traditional classifiers C4.5 Gain Ratio Decision Tree, Naïve Bayes and KNN, we used the J48, Naïve Bayes and Ibk implementations that are available in the WEKA tool, respectively.

We used k-fold cross-validation in our test. In this approach, the dataset is randomly partitioned into k subsets. Only one subset is used as validation data for testing the model. The other $k-1$ subsets are used for training the classification model. The cross-validation process is repeated k times. Each of the k subsets is used only once for validation. The final result is the average of the results obtained at each round. In our experiments, we used $k = 10$.

We previously collected sequences with human poses that were inspired in the Jecripe game [2]. The poses define the classes, which are: *T-pose*, *dancing*, *play guitar*, and *play drums*. Three other movements, which were not related to the game, were also included: *punch*, *kick* and *kick* + *punch*.

We characterize the classes as the following: The *T-pose* constitutes a person with both arms and hands at the same level as the shoulders. In the *dancing* class, one of the hands is on the head; the other hand is on the hip, and one or both feet are on the ground. As a consequence, we have six combinations of poses for the class *dancing*: (i) left hand on the head and feet on the ground; (ii) left hand on the head and moving left foot; (iii) left hand on the head and moving right foot; (iv) right hand on the head and feet on the ground; (v) right hand on the head and moving right foot; and (vi) right hand on the head and moving left foot. All of the six poses have the same class, which is *dancing*.

In the *playing guitar* class, the user imitates the moves of playing an instrument, shaking the right hand while the left hand stays at the same level as his/her shoulders. The *playing drums* class is when the user shakes his/her hands up and down alternately. There are two possible poses for the *punch* class, both of which have feet on the ground: (I) right hand and (II) left hand. Similar to the *punch*, the *kick* class can be made with: (a) right foot and (b) left foot, with both hands below the centroid. The *kick* + *punch* class can be made in four different poses: (A) kick with left foot and punch with left hand; (B) kick with left foot and punch with right hand; (C) kick with right foot and punch with right hand; and (D) kick with right foot and punch with left hand.

We used three different volunteers in our experiments: A, B and C. For each user, we collected a different number of sequences. Volunteer A is male, 1.76 meters tall, and has dark hair. Table I shows the collected sequences with Volunteer A. We collected 17 sequences with all of the classes.

Volunteer B is male, 1.90 meters tall and has blond hair. Volunteer B made 14 different sequences in four classes, all of them without self-occlusion. All of the possible poses for each of the four classes were collected. Table II details each of the collected poses from Volunteer B.

TABLE I
IMAGE SEQUENCE EVALUATION FOR VOLUNTEER A.

Sequence Number	Movement	Number of Images	Tracking Until The End
Sequence A1	dancing (i)	140	yes
Sequence A2	dancing (i)	116	yes
Sequence A3	dancing (ii)	100	yes
Sequence A4	playing guitar	140	yes*
Sequence A5	playing drums	190	yes*
Sequence A6	playing drums	130	yes*
Sequence A7	playing drums	130	yes*
Sequence A8	punch (I)	84	yes
Sequence A9	punch (I)	81	yes**
Sequence A10	kick (a)	66	yes
Sequence A11	dancing (iii)	58	yes
Sequence A12	dancing (ii)	68	yes
Sequence A13	kick + punch (A)	57	yes
Sequence A14	dancing (iv)	104	yes
Sequence A15	dancing (v)	152	yes
Sequence A16	dancing (vi)	98	yes
Sequence A17	kick + punch (D)	55	yes

*Tracked until the end of the sequence but it had a problem in the presence of self-occlusion.

**Problem caused by movement velocity.

TABLE II
IMAGE SEQUENCE EVALUATION FOR VOLUNTEER B.

Sequence Number	Movement	Number of Images	Tracking Until The End
Sequence B1	dancing (i)	99	yes
Sequence B2	dancing (iv)	84	yes
Sequence B3	dancing (iii)	84	yes
Sequence B4	dancing (ii)	62	yes
Sequence B5	dancing (v)	72	yes
Sequence B6	dancing (vi)	79	yes
Sequence B7	punch (I)	65	yes
Sequence B8	punch (II)	75	yes
Sequence B9	kick (b)	70	yes
Sequence B10	kick (a)	79	yes
Sequence B11	kick + punch (C)	73	yes
Sequence B12	kick + punch (D)	74	yes
Sequence B13	kick + punch (B)	99	yes
Sequence B14	kick + punch (A)	97	yes

TABLE III
IMAGE SEQUENCE EVALUATION FOR VOLUNTEER C.

Sequence Number	Movement	Number of Images	Tracking Until The End
Sequence C1	dancing (i)	48	yes
Sequence C2	dancing (iv)	69	yes
Sequence C3	dancing (iii)	45	yes
Sequence C4	dancing (ii)	54	yes
Sequence C5	dancing (v)	54	yes
Sequence C6	dancing (vi)	45	yes
Sequence C7	punch (I)	90	yes
Sequence C8	punch (II)	88	yes
Sequence C9	kick (b)	49	yes
Sequence C10	kick (a)	54	yes
Sequence C11	kick + punch (C)	90	yes
Sequence C12	kick + punch (D)	100	yes
Sequence C13	kick + punch (B)	85	yes

Volunteer C is female, 1.66 meters tall and has dark hair. Similar to Volunteer B, we collected sequences of four different classes with Volunteer C. Additionally, no problem was detected during the collection of the poses, which shows that the M5AIE method works well in sequences that do not have self-occlusions. We collected 13 sequences with Volunteer C because we wanted to test fewer training tuples with the pose *kick + punch* (A).

We observed that the M5AIE method had problems with poses that had self-occlusions. The problems were detected in the *playing guitar* and *playing drums* poses. This problem detection was crucial for the collection of the other users sequences; as a result, we avoided collecting these poses. However, we kept the results to make the tuples and test the classification algorithms. In only one sequence, the tracking method had problems that were caused by the movement velocity, but the pose classification was not affected.

The dataset that was used for both the training and testing comprises the grid-coordinates that body parts assume at each frame of a set of image sequences that were produced for this work and the manual classification of the pose in each frame. We varied the number of cells of the grid in each frame, as follows: 8×8 (Table IV), 16×16 (Table V), 32×32 (Table VI) and 64×64 (Table VII).

The set of k values for the KNN algorithm is $\{1, 3, 5, 7, 9, 11\}$, and different distances were used in our experiments. We combined the set of k values with the Manhattan, Chebyshev and Euclidean distances. For each of the N values of the grids $N \times N$, we made a data set that had all of the tuples from the three different users that made the described poses and 2128 tuples.

Table IV, where $N = 8$, shows that the Naïve Bayes Classifier gave the highest number of incorrectly classified instances (21.22%). For all of the other classifiers, the percentage of instances that were correctly classified were above 93%. The C4.5 Gain Ratio Decision Tree had similar results as the KNN algorithm when $k >= 3$. As the k value increased, the percentage of correctly classified instances decreased. Nevertheless, the Manhattan distance had the best results for every k value. The best of all of the results in Table IV were with $K = 1$, primarily from using the Manhattan distance, with a 98.24% correct. Most of the errors made by the classifier were from confusing *dancing* with *punch* and *kick + punch* classes.

Considering $N = 16$ (Table V), once more, the Naïve Bayes Classifier gave the highest percentage of incorrectly classified instances, with 28.74%. For all the other classifications, the incorrectly classified instances were less than 8%. The C4.5 Gain Ratio Decision Tree had only similar results with $k >= 7$ considering the Manhattan and Euclidean distances. If we consider only the values with the same value k, the Chebyshev distance gave the worst results. On the other hand, the Manhattan distance gave the best results. We could observe that the best results were obtained again with $k = 1$ and the Manhattan distance. Again, increasing the value of k, the results become worse for all of the used distances. The best percentage of correctness with $N = 16$ (98.84%) was slightly better than with $N = 8$ (98.24%), when both used $k = 1$ and the Manhattan distance. The *dancing* class was confused with the *playing guitar*, *punch* and *kick + punch* classes.

With $N = 32$, similar to with $N = 8$ and $N = 16$,

TABLE IV
RESULTS FOR $N = 8$.

Classification with Grid 8×8	**Correct***	**Incorrect****
C4.5 Gain Ratio Decision Tree	97.26%	2.74%
Naïve Bayes	78.78%	21.22%
KNN with K=1 and Manhattan Distance	98.24%	1.76%
KNN with K=1 and Chebyshev Distance	98.07%	1.93%
KNN with K=1 and Euclidean Distance	98.24%	1.76%
KNN with K=3 and Manhattan Distance	97.79%	2.21%
KNN with K=3 and Chebyshev Distance	97.01%	2.99%
KNN with K=3 and Euclidean Distance	97.66%	2.34%
KNN with K=5 and Manhattan Distance	96.89%	3.11%
KNN with K=5 and Chebyshev Distance	95.94%	4.06%
KNN with K=5 and Euclidean Distance	96.60%	3.40%
KNN with K=7 and Manhattan Distance	96.23%	3.77%
KNN with K=7 and Chebyshev Distance	94.22%	5.78%
KNN with K=7 and Euclidean Distance	95.99%	4.01%
KNN with K=9 and Manhattan Distance	95.94%	4.06%
KNN with K=9 and Chebyshev Distance	93.32%	6.68%
KNN with K=9 and Euclidean Distance	95.86%	4.14%
KNN with K=11 and Manhattan Distance	96.31%	3.69%
KNN with K=11 and Chebyshev Distance	93.20%	6.80%
KNN with K=11 and Euclidean Distance	96.15%	3.85%

*Correctly Classified Instances

**Incorrectly Classified Instances

TABLE V
RESULTS FOR $N = 16$.

Classification with Grid 16×16	**Correct***	**Incorrect****
C4.5 Gain Ratio Decision Tree	97.39%	2.61%
Naïve Bayes	71.26%	28.74%
KNN with K=1 and Manhattan Distance	98.84%	1.16%
KNN with K=1 and Chebyshev Distance	98.31%	1.69%
KNN with K=1 and Euclidean Distance	98.79%	1.21%
KNN with K=3 and Manhattan Distance	98.36%	1.64%
KNN with K=3 and Chebyshev Distance	96.33%	3.67%
KNN with K=3 and Euclidean Distance	97.97%	2.03%
KNN with K=5 and Manhattan Distance	98.02%	1.98%
KNN with K=5 and Chebyshev Distance	95.60%	4.40%
KNN with K=5 and Euclidean Distance	97.58%	2.42%
KNN with K=7 and Manhattan Distance	97.39%	2.61%
KNN with K=7 and Chebyshev Distance	95.22%	4.78%
KNN with K=7 and Euclidean Distance	97.29%	2.71%
KNN with K=9 and Manhattan Distance	97.20%	2.80%
KNN with K=9 and Chebyshev Distance	94.30%	5.70%
KNN with K=9 and Euclidean Distance	96.86%	3.14%
KNN with K=11 and Manhattan Distance	96.47%	3.53%
KNN with K=11 and Chebyshev Distance	92.90%	7.10%
KNN with K=11 and Euclidean Distance	96.18%	3.82%

*Correctly Classified Instances

**Incorrectly Classified Instances

TABLE VI
RESULTS FOR $N = 32$.

Classification with Grid 32×32	**Correct***	**Incorrect****
C4.5 Gain Ratio Decision Tree	98.59%	1.41%
Naïve Bayes	76.17%	23.83%
KNN with K=1 and Manhattan Distance	99.72%	0.28%
KNN with K=1 and Chebyshev Distance	99.58%	0.42%
KNN with K=1 and Euclidean Distance	99.77%	0.24%
KNN with K=3 and Manhattan Distance	99.39%	0.61%
KNN with K=3 and Chebyshev Distance	98.45%	1.55%
KNN with K=3 and Euclidean Distance	99.34%	0.66%
KNN with K=5 and Manhattan Distance	99.34%	0.66%
KNN with K=5 and Chebyshev Distance	97.93%	2.07%
KNN with K=5 and Euclidean Distance	98.83%	1.17%
KNN with K=7 and Manhattan Distance	99.15%	0.85%
KNN with K=7 and Chebyshev Distance	97.32%	2.68%
KNN with K=7 and Euclidean Distance	98.64%	1.36%
KNN with K=9 and Manhattan Distance	98.73%	1.27%
KNN with K=9 and Chebyshev Distance	95.82%	4.18%
KNN with K=9 and Euclidean Distance	98.03%	1.97%
KNN with K=11 and Manhattan Distance	98.26%	1.74%
KNN with K=11 and Chebyshev Distance	95.21%	4.79%
KNN with K=11 and Euclidean Distance	97.37%	2.63%

*Correctly Classified Instances

**Incorrectly Classified Instances

TABLE VII
RESULTS FOR $N = 64$.

Classification with Grid 64×64	**Correct***	**Incorrect****
C4.5 Gain Ratio Decision Tree	98.54%	1.46%
Naïve Bayes	77.02%	22.98%
KNN with K=1 and Manhattan Distance	99.81%	0.19%
KNN with K=1 and Chebyshev Distance	99.62%	0.38%
KNN with K=1 and Euclidean Distance	99.81%	0.19%
KNN with K=3 and Manhattan Distance	99.62%	0.38%
KNN with K=3 and Chebyshev Distance	98.26%	1.74%
KNN with K=3 and Euclidean Distance	99.34%	0.66%
KNN with K=5 and Manhattan Distance	99.44%	0.56%
KNN with K=5 and Chebyshev Distance	97.23%	2.77%
KNN with K=5 and Euclidean Distance	99.20%	0.80%
KNN with K=7 and Manhattan Distance	99.25%	0.75%
KNN with K=7 and Chebyshev Distance	96.76%	3.24%
KNN with K=7 and Euclidean Distance	98.92%	1.08%
KNN with K=9 and Manhattan Distance	99.01%	0.99%
KNN with K=9 and Chebyshev Distance	95.39%	4.61%
KNN with K=9 and Euclidean Distance	98.50%	1.50%
KNN with K=11 and Manhattan Distance	98.50%	1.50%
KNN with K=11 and Chebyshev Distance	94.55%	5.45%
KNN with K=11 and Euclidean Distance	97.84%	2.16%

*Correctly Classified Instances

**Incorrectly Classified Instances

the Naïve Bayes classifier gave the smallest percentage of correctly classified instances (76.17%). All of the other results had more than 95% correctness on instances of classification. If we compare the C4.5 algorithm with KNN (without the Chebyshev distance), we obtain similar results to when $k >= 9$. The best results were with $K = 1$ but with the Euclidean distance (99.77%), which was followed very closely by the Manhattan distance (99.72%). This result is even better than the best result in Table V. Most of the incorrectly classified instances occurred with instances of *dancing*, *punch* and *kick + punch*. Table VI shows the results for $N = 32$.

Table VII shows the results with $N = 64$. As was expected, the Naïve Bayes had 22.98% incorrectly classified instances, followed by KNN with $k = 11$ and the Chebyshev distance, which had 5.45% incorrect. All of the others gave more than 94% correctly classified instances. The C4.5 algorithm had similar results with only KNN when $k = 11$. Similar to the other best results, in Table VII, KNN with $k = 1$ gave the best results with both distances, Manhattan and Euclidean, with exactly the same value, 99.81%. Because the results are very close to 100% using $N = 64$, we could observe a relatively high number of errors using Naïve Bayes, which gave errors in the classes *dancing*, *punch* and *kick + punch*.

Until this point, we exposed the results, showing each table in an isolated way. However, we can observe additional results by comparing the tables with one another. All of the algorithms had similar results while considering the same algorithm with different N values. In all of the cases, the worst results came from the Naïve Bayes Classifier. The C4.5 had similar results with KNN depending on the k value of each Table. Although the results are very similar from one table to another, we can see that the results of the C4.5 algorithm and KNN become better when N becomes higher. Considering the distances, in general, the Manhattan gave the best results if we compare the same k value in every Table. The Euclidean distance gave very similar results to the Manhattan, and only once the results from the Euclidean distance were better than the Manhattan distance. In all of the KNN experiments, the Chebyshev distance gave a percentage of incorrectly classified instances that was higher than for the other two considered distances.

We believe that if we continue to increase the value of N, it could improve the results even more until a certain limit value is obtained. From that limit value for N onward, the results could start to become worse (as mentioned in item 3, above). Perhaps if we normalized the coordinates according to the bounding box instead of a grid divided into cells, we could obtain the best results. We consider the KNN with $k = 1$ and the Manhattan distance as the winning algorithm.

According to the concept of each distance measure, our inference for why we obtained the worst results using the Chebyshev distance is that this distance undervalues the distance between the body parts in each frame and the classifier makes mistakes when making its predictions. The Chebyshev distance gives the longest distance considering all of the axis distances from point A to another point B. Then, the body parts can be closer than they actually are to each other. However, the Manhattan and Euclidean distances can be more realistic for human movements. This last assumption should be the reason for the best results for the Manhattan distance, and the Euclidean distance gives very similar results in comparison to the Manhattan distance.

VI. Conclusions and Future Directions

The focus of this thesis is on Computer Vision and Digital Games research. In fact, the primary motivation for this work is to contribute with research that makes it easier to implement different concepts in Natural User Interfaces. Since the beginning of the development of the Jecripe game [9], [2], there was the intention to stimulate the movements of children with Down syndrome throght NUI.

The first stage of this study was the background subtraction task. In the background subtraction task we had done a study to compare different algorithms considering images collected from Kinect. We adapted some of the algorithms to make all of them in equal conditions of comparison. The results of this study motivated us to use the *Minimum Background* algorithm and we published this work in [4].

This thesis describes the M5AIE method for detecting and tracking five main parts of the human body (head, hands and feet) in sequences of RGB-D images. The proposed approach combines an effective background subtraction method, the discrete medial axis transformation, in the construction of simpler graphs to be used in the detection of AGEX points, heuristics for labeling, and ASIFT-based tracking of labeled structures. The experiments and first results of the M5AIE method were published in [7].

To prove that the M5AIE method is effective, including on the human pose prediction task, we made a comparison among the classification algorithms when applied to human pose recognition in the game context. In this study, we proposed and developed a detailed analysis using the M5AIE with different algorithms. We published the we had done concerning a comparison of different classification algorithms in [8].

We presented the M5AIE method which was implemented as a proof of concept. Recently, the literature presented encouraging studies with real time results for the Medial Axis transformation and the SIFT algorithm. We intend to improve our method and integrate it with computer games. We will present how our research contributed in different computer science research area [10] and another study will give more details on how our research contributed mainly for the Human-Computer Interaction area [11].

Acknowledgment

We would like to thank the German Research Center for Artificial Intelligence (DFKI) and DAAD.

References

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. of CVPR 2011*, Colorado Springs, CO, USA, 2011, pp. 1297–1304.

[2] A. Brandão, L. Brandão, G. Nascimento, B. Moreira, C. N. Vasconcelos, and E. Clua, "Jecripe: stimulating cognitive abilities of children with down syndrome in pre-scholar age using a game approach," in *Proc. of ACE '10*. New York, NY, USA: ACM, 2010, pp. 15–18, qualis B1.

[3] A. L. Brandão, "M5AIE: A method for body part detection, tracking and pose classification using rgb-d images," Ph.D. dissertation, Universidade Federal Fluminense, Niterói, Brasil, 2013.

[4] K. Greff, A. Brandão, S. Krauß, D. Stricker, and E. Clua, "A comparison between background subtraction algorithms using a consumer depth camera," in *Proc. of VISAPP 2012*, vol. 1. Rome, Italy: SciTePress, 2012, pp. 431–436, qualis B3.

[5] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time identification and localization of body parts from depth images," in *Proc. of ICRA 2010*, Anchorage, Alaska, USA, 2010, pp. 3108–3113.

[6] J.-M. Morel and G. Yu, "Asift: A new framework for fully affine invariant image comparison," *SIAM J. Img. Sci.*, vol. 2, no. 2, pp. 438–469, Apr. 2009.

[7] A. L. Brandão, L. A. F. Fernandes, and E. Clua, "M5AIE: A method for body part detection and tracking using rgb-d images," in *Proc. of VISAPP 2014*, vol. 1. Lisbon, Portugal: SciTePress, January 2014, pp. 367–377, qualis B3.

[8] A. Brandão, L. A. F. Fernandes, and E. Clua, "A comparative analysis of classification algorithms applied to M5AIE-extracted human poses," in *Proc. of SBGAMES*, São Paulo, Brasil, 2013, qualis B4.

[9] A. Brandão, D. Trevisan, L. Brandão, B. Moreira, G. Nascimento, P. Mourão, C. N. Vasconcelos, and E. Clua, "Semiotic inspection of a game for children with down syndrome," in *Proc. of SBGAMES*, Florianópolis, Brasil, 2010, pp. 199–210, qualis B4.

[10] A. L. Brandão, L. A. F. Fernandes, D. Trevisan, E. Clua, and D. Stricker, "Jecripe: How a serious game project encouraged studies in different computer science areas," in *Accepted for publication in Proc. of SEGAH'14*, vol. 1, Niterói, Brasil, May 2014, pp. 1–8.

[11] ——, "Jecripe: How a serious game project encouraged studies in different computer science areas," in *Selected for publication on a special issue in Intern. Journal of Medical Informatics*, vol. 1, Niterói, Brasil, May 2014, pp. 1–8, qualis A2.