G3D: A Gaming Action Dataset and Real Time Action Recognition Evaluation Framework

Victoria Bloom
Kingston University
London. UK
Victoria.Bloom@kingston.ac.uk

Dimitrios Makris
Kingston University
London. UK
D.Makris@kingston.ac.uk

Vasileios Argyriou
Kingston University
London. UK
Vasileios.Argyriou@kingston.ac.uk

Abstract

In this paper a novel evaluation framework for measuring the performance of real-time action recognition methods is presented. The evaluation framework will extend the time-based event detection metric to model multiple distinct action classes. The proposed metric provides more accurate indications of the performance of action recognition algorithms for games and other similar applications since it takes into consideration restrictions related to time and consecutive repetitions. Furthermore, a new dataset, G3D for real-time action recognition in gaming containing synchronised video, depth and skeleton data is provided. Our results indicate the need of an advanced metric especially designed for games and other similar real-time applications.

1. Introduction

The gaming industry in recent years has attracted an increasing large and diverse group of people. A new generation of games based on full body play such as dance and sports games have increased the appeal of gaming to family members of all ages.

Full body play relies on detecting players movements using sensors to provide a controller free gaming experience. The Kinect originally developed for the Xbox 360 games console has a wide range of released titles but they are limited to a small set of actions.

Currently action recognition in gaming ranges from heuristic based techniques to machine learning algorithms. The approach taken depends on the number and complexity of gestures to be performed in the game. For example, a bowling game only requires a few simple gestures and an algorithm can be hardcoded for each gesture. However, this approach may not work well for a greater number of complex gestures where machine learning algorithms are more suited. Various machine learning techniques can be applied to more complex games for example AdaBoost with a boxing game and exemplar matching with a tennis game [1].

The benefit of machine learning algorithms is that they can be trained to recognise a wide range of actions including sporting, driving and action-adventure actions such as walking, running, jumping, dropping, firing, changing weapon, throwing and defending. This approach can increase the complexity and appeal of games that can be developed to include action-adventure games (e.g. Lara Croft).

State of the art action recognition approaches are currently appearance based. The algorithms use input features such as colour, dense optical flow and spatio-temporal gradients extracted from a RGB image. The context of the environment can be used to further improve accuracy as intuitively certain actions will only happen in specific scenes [2]. For example, performing a golf swing in a real golf game would require a golf club and will occur outdoors, probably on a green field. However, performing a golf swing in a Kinect game the user has no golf club and is performing the action indoors. The restricted environment associated with gaming, typically the user's lounge, poses the challenge of missing context. The normal scene and objects usually associated with a given action are missing. This lack of contextual information may mean that appearance-based action recognition approaches may under perform.

An alternative is a pose based approach where joint positions, velocities and angles are extracted from a human articulated pose. This approach was previously disregarded due to the complexity of estimating the human pose. Now it is possible to obtain real-time skeleton data for multiple subjects in an indoor environment using Kinect [3] so pose based approaches are being revisited by researchers. Yao et al. [4] experiments showed that pose based features outperform low-level appearance features in a home monitoring scenario.

To compare the performance of the appearance, pose based and combined approaches for gaming, a common dataset and evaluation framework for measuring performance is required. G3D is a new public gaming action dataset containing synchronised colour, depth and skeleton data that has been captured for this paper. A novel evaluation framework for measuring the performance of real-time action recognition algorithms will also be proposed to provide a common base for comparison.

978-1-4673-1612-5/12/$31.00 ©2012 IEEE

2. Related work

2.1. Action recognition

There is a vast wealth of research on human action recognition in computer vision (for a comprehensive review see Aggarwal and Ryoo [5]). The majority of the state-of-the art algorithms in activity recognition are appearance based as low level features can easily be extracted from video sequences.

Due to recent technological developments in depth camera technology it is now possible and economical to capture real-time sequences of depth images. This has resulted in depth and posed based approaches being developed.

Li et al. [6] directly sample 3D representative points from a depth map as features for their action graph method. Their results show recognition errors were halved when using 3D depth data in comparison with 2D silhouettes.

Yao et al. [4] posed the question "Does Human Action Recognition Benefit from Pose Estimation?" Their experiments compared appearance based, pose based and a combined approach in a home monitoring scenario using the same classifier and same dataset. The appearance based features used were colour, dense optical flow and spatio-temporal gradients. The pose based features were qualitative geometric features [7]. Yao et al. [4] results showed that the optimum approach was pose based. This significantly outperformed the appearance based approach and was even slightly better than the combined approach.

Both studies show promising results and could benefit from a direct comparison with state-of-the-art appearance based methods using a common dataset.

2.2. Action recognition datasets

There are many existing public datasets available containing video sequences. For a comprehensive review of these also see Aggarwal and Ryoo [5]. The datasets can be categorized into three groups, as follows. The first are simple action recognition datasets such as the KTH [8] and Weizmann [9] datasets, where each video contains one simple action performed by one actor in a controlled environment. The second type are surveillance datasets such as PETS [10] and i-Lids [11] obtained in realistic environments such as airports. The third type are movie datasets with challenging videos with frequently moving camera viewpoints obtained from real movie scenes such as Hollywood2 [2].

For gaming, the existing action recognition datasets are insufficient as during a game multiple different actions will be performed by the player. Public datasets containing both video and skeleton data for sports and locomotion actions exist [12] [13] [14] but as mentioned previously there is a difference between performing a real action and a gaming action. Even simple actions such as walking are different in the gaming environment as the player will walk on the spot. The MPI HDM05 Motion Capture Database [13] database does include locomotion on the spot but not a full range of gaming actions. Microsoft research specifically developed a gaming action database, MSR Action3D Database [6] which initially consisted of a sequence of depth maps and was later extended by a third party to include skeleton data. Nevertheless, no corresponding video data is available. As far as we are aware there is no publicly available action recognition database for gaming that contains all three types of subject data (video, depth and skeleton). This paper will introduce a publicly available dataset, G3D for real-time action recognition in gaming containing synchronised video, depth and skeleton data.

2.3. Action recognition performance metrics

A common performance measure used in general action recognition is classification accuracy and confusion matrices are frequently used to breakdown the number of correct classifications by class [5].

Applying the performance metric depends on the nature of the input sequence. In the simple case where the video is segmented to contain only one action e.g. KTH [8] the classification process is straightforward. An action label is predicted for each frame in the sequence and a majority decision over all frames is taken to decide the action label for the complete sequence [15]. In the more complex case, with multiple actions within a sequence e.g. the new G3D dataset, the classification process outlined above is not viable.

The simple approach of applying the performance measure to the entire sequence does not incorporate timing or continuity constraints that are required to make action recognition methods applicable for a range of real-world applications. Real time performance is critical for a game to appear responsive to the player's actions. A single action performed by the player must be detected as soon as possible, as one continuous action consisting of multiple sequential frames and not multiple actions to prevent poor gameplay.

In surveillance systems, continuous detection in real-time is required. The i-LIDS event detection metric [11] incorporates the continuous detection of events in real-time by summing occurrences of true positives, false negatives and false positives along an event timeline to produce an F1 score. The limitation of this metric is that it only recognises a single event type e.g. abandon package.

A new real-time action recognition evaluation framework will be proposed that extends the event time-line detection metric to model multiple action classes.

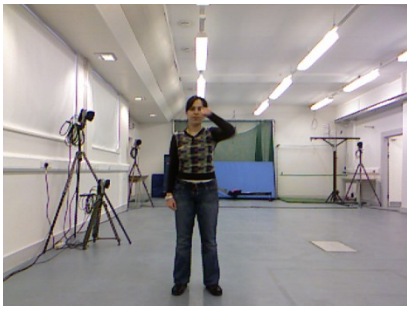


Figure 1 Colour image

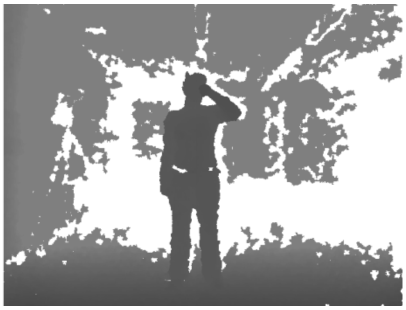


Figure 2 Depth map

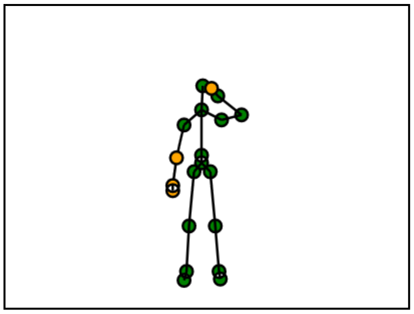


Figure 3 Skeleton data

3. G3D dataset

A new dataset, G3D for real-time action recognition in gaming containing synchronised video, depth and skeleton data has been captured. This dataset is publicly available at http://dipersec.king.ac.uk/G3D/ to allow researchers to develop new action recognition algorithms for video games and benchmark their performance. Due to the formats selected it is possible to view all the recorded data and tags without any special software tools.

The Microsoft Kinect device and Windows SDK [3] enables easy capture of synchronised video, depth and skeleton data. The three streams were recorded at 30fps in a mirrored view so Figures 1-4 are actually a right punch. The PNG image format was selected for storing both the depth and colour images as it is a lossless format, suitable for online access and is open source. The resolution used to store both the depth and colour images was 640x480. The raw depth information contains the depth of each pixel in millimetres and was stored in 16-bit greyscale (see Figure 2) and the raw colour in 24-bit RGB (see Figure 1).

The 16-bits of depth data contains 13 bits for depth data and 3 bits to identify the player. The player index can be used to segment the depth maps by user (see Figure 4). The depth information was also mapped to the colour coordinate space and stored in a 16-bit greyscale. Combining the colour image with the mapped depth data allows the user to also be segmented in the colour image.

The XML text format was selected for storing the skeleton information as it is human readable and again suited for online access. The root node the XML file is an array of skeletons. Each skeleton contains the player's position and pose. The pose comprises of 20 joints as defined by Microsoft [3]. The player and joint positions are given in X,Y and Z co-ordinates in meters. These positions are also mapped into the depth (see Figure 4) and colour co-ordinates spaces. The skeleton data includes a joint tracking state, displayed in Figure 3 as tracked (green), inferred (yellow) and not tracked (red). In many cases the inferred joints will be accurate as in Figure 4 but certain situations where limbs are occluded the inferred joints may be inaccurate as in Figure 5. Consequently, pose data may need to be combined with colour or depth data to improve accuracy.

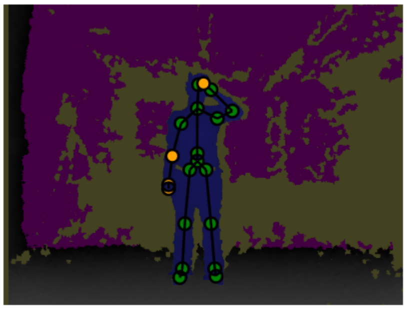


Figure 4 Correctly inferred joints

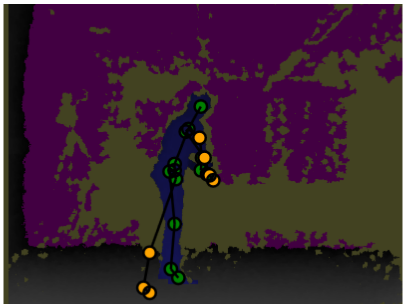


Figure 5 Incorrectly inferred joints

This dataset contains 10 subjects performing 20 gaming actions : *punch right, punch left, kick right, kick left, defend, golf swing, tennis swing forehand, tennis swing backhand, tennis serve, throw bowling ball, aim and fire gun, walk, run, jump, climb, crouch, steer a car, wave, flap and clap.* Most sequences contain multiple actions in a controlled indoor environment with a fixed camera, a typical setup for gesture based gaming. The subjects were given basic instructions as to how to perform the action, similar to those issued in a Kinect game. Nevertheless, the subjects were free to perform the gesture with either hand or in the case of a side facing action stand with either foot forward to create a diverse dataset. Each sequence is repeated three times by each subject. This resulted in over 80,000 frames of video, depth and skeleton data.

All the frames in the dataset that contain actions were manually labelled in a separate file with an appropriate tag. Each tag represents a single action and contains the action class e.g. PunchRight, first frame number and last frame number. The XML tags are also publicly available.

4. Evaluation Framework

The existing performance metric for action recognition, classification accuracy determined for each sequence is inadequate for gaming as it does not incorporate time and continuity constraints critical for real-time action detection. The existing metric is also restricted to sequences containing one action. To address these limitations a new evaluation framework is proposed.

To incorporate multiple actions a frame based evaluation framework could be used. Matches between each predicted frame and ground truth frame could be accumulated in a confusion matrix and average classification accuracy determined. Although this would work for multiple actions in a sequence it still does not incorporate timing and continuity constraints.

To overcome all of the existing limitations an action based evaluation framework is needed. However, matching each predicted action with a ground truth action poses a challenge as both the predicted and ground truth actions represent a sequence of frames rather than just a single frame. It is no longer possible to accumulate matches in a confusion matrix as there is a new case. The new case is where the predicted action class matches a ground truth action type but is a false positive; e.g. in a duplicated action (see the second predicted defend action in Figure 6). As the predicted and ground truth action classes match, the only place to represent this information in a confusion matrix is along the diagonal axis, which represents the true positives. However, in this is clearly wrong as the second defend action is a false positive.

The i-LIDS event detection metric [11] resolves this problem by using an event time-line to determine the correct number of true positives, false negatives and false positives. For example, a true positive is detected if a predicted event occurs within 10 seconds of an actual event. However, this metric only recognises a single event type e.g. an alarm.

Extending the time based comparison of alarm events by the i-LIDS event detection metric [11] for multiple actions results in a new action based evaluation framework. The new framework can evaluate the performance of any real time action recognition algorithm by selecting an appropriate time constraint. A latency of 4 frames is by selecting an acceptable in gaming as it should not be perceived by the user. In Microsoft's experiments [1] an actual latency of 4 frames resulted in a perceived latency of 0 to 1 frames.

The new action matching process is illustrated in Figure 6. For each sequence the start time of any predicted actions of a particular class should be compared to the start time of the ground truth action to evaluate the number of true positive, false positive and false negative actions. In the proposed framework a true positive arises when a predicted event starts within T (T=4 frames) of an actual event. A false positive is a predicted event that does not start within T frames of an actual event; this can be caused by two different situations. Firstly, when the correct action is detected but not within the T time period, for example the predicted punch left action on Figure 6. Secondly, when the incorrect action is predicted, for example the first predicted kick left action on Figure 6. Finally, a false negative is an actual event that remains unmatched with any predicted event. The main difference between this evaluation framework and the i-LIDS event detection is the addition of multiple distinct action classes to detect incorrect actions.

The output of the matching process is the total number of true positives, false positive and false negative actions for each action class. This division by class is extremely helpful for improving action recognition algorithms but not so effective for ranking different algorithms. The performance metrics of precision, recall and F1 are then computed for each action class and a final single average F1 figure calculated.

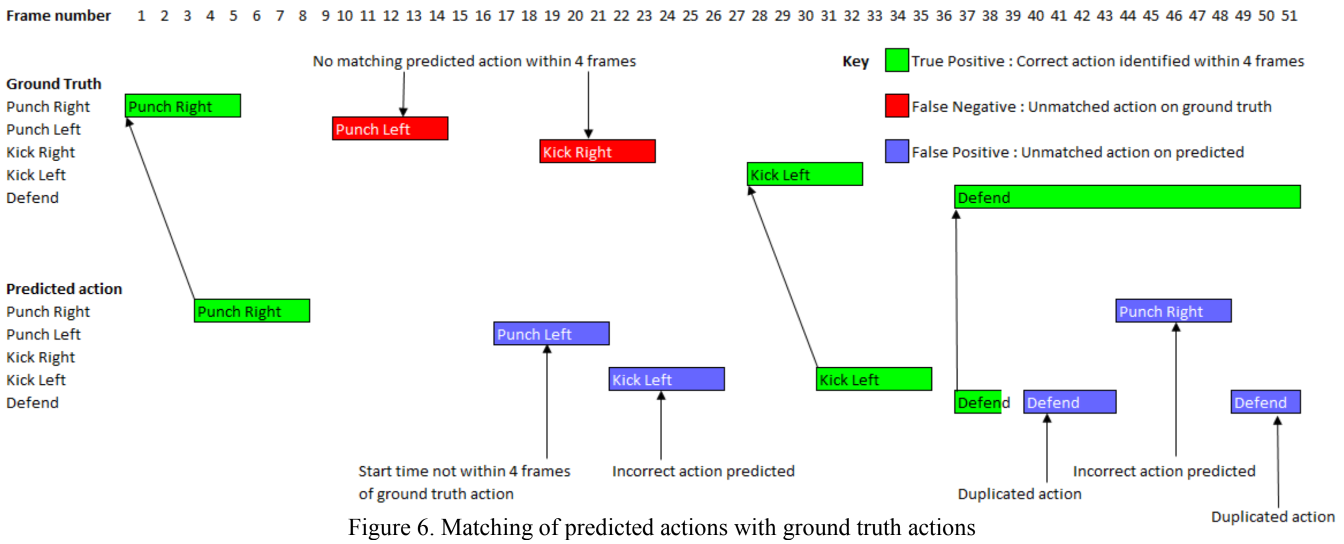

Frame number 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
No matching predicted action within 4 frames
Key
True Positive : Correct action identified within 4 frames
False Negative : Unmatched action on ground truth
False Positive : Unmatched action on predicted
Ground Truth
Punch Right
Punch Left
Kick Right
Kick Left
Defend
Punch Right
Punch Left
Kick Right
Kick Left
Defend
Predicted action
Punch Right
Punch Left
Kick Right
Kick Left
Defend
Punch Right
Punch Left
Kick Left
Kick Left
Punch Right
Defend
Defend
Defend
Start time not within 4 frames of ground truth action
Incorrect action predicted
Duplicated action
Incorrect action predicted
Duplicated action

Figure 6. Matching of predicted actions with ground truth actions

5. Action recognition algorithm

To test our new evaluation framework we used Adaptive Boosting (AdaBoost) [16] a real-time action recognition machine learning algorithm proved successful for gaming gestures [1].

Microsoft [1] recommended pose based input features such as position difference, position velocity, position velocity magnitude, angle velocity and joint angles. A description of the features is not available so the following is our own definition.

Let $p_{j_i,t} \in \mathbb{R}^3$ be the 3D location $(x_{j_i,t}, y_{j_i,t}, z_{j_i,t})$ and of joint j_i at time t. The position difference features are defined as the difference between two distinct joints in a single pose:

$$\boldsymbol{F^{pdx}}\ (\boldsymbol{j_1, j_2; t_1}) = \boldsymbol{x_{j_1,t_1} - x_{j_2,t_1}} \qquad (1)$$

$$\boldsymbol{F^{pdy}}\ (\boldsymbol{j_1, j_2; t_1}) = \boldsymbol{y_{j_1,t_1} - y_{j_2,t_1}} \qquad (2)$$

$$\boldsymbol{F^{pdz}}\ (\boldsymbol{j_1, j_2; t_1}) = \boldsymbol{z_{j_1,t_1} - z_{j_2,t_1}} \qquad (3)$$

The position velocity features are similar except they encode the differences between a single joint separated by time, where $\boldsymbol{t_1 \neq t_2}$:

$$\boldsymbol{F^{pvx}}\ (\boldsymbol{j_1; t_1, t_2}) = \boldsymbol{x_{j_1,t_1} - x_{j_1,t_2}} \qquad (4)$$

$$\boldsymbol{F^{pvy}}\ (\boldsymbol{j_1; t_1, t_2}) = \boldsymbol{y_{j_1,t_1} - y_{j_1,t_2}} \qquad (5)$$

$$\boldsymbol{F^{pvz}}\ (\boldsymbol{j_1; t_1, t_2}) = \boldsymbol{z_{j_1,t_1} - z_{j_1,t_2}} \qquad (6)$$

The position velocity magnitude feature is defined as the Euclidean distance between a single joint separated by time, where $\boldsymbol{t_1 \neq t_2}$:

$$\boldsymbol{F^{pvd}}\ (\boldsymbol{j_1; t_1, t_2}) = \ \| \boldsymbol{p_{j_1,t_1} - p_{j_1,t_2}} \| \qquad (7)$$

The angle velocity features are defined as the angle between a single joint separated by time projected in the x-y plane and the x-z plane:

$$\boldsymbol{F^{avxy}}\ (\boldsymbol{j_1; t_1, t_2}) = \mathbf{arctan}\left(\frac{\boldsymbol{y_{j_1,t_1} - y_{j_1,t_2}}}{\boldsymbol{x_{j_1,t_1} - x_{j_1,t_2}}}\right) \qquad (8)$$

$$\boldsymbol{F^{avxz}}\ (\boldsymbol{j_1; t_1, t_2}) = \mathbf{arctan}\left(\frac{\boldsymbol{z_{j_1,t_1} - z_{j_1,t_2}}}{\boldsymbol{x_{j_1,t_1} - x_{j_1,t_2}}}\right) \qquad (9)$$

The joint angle feature is defined as the 3D angle between three distinct joints in a single pose, let vector $\boldsymbol{v_1}$ be $(j_2 - j_1)$ and vector $\boldsymbol{v_2}$ be $(j_2 - j_3)$:

$$\boldsymbol{F^{jf}}\ (\boldsymbol{v_1, v_2; t_1}) = \mathbf{arctan}\left(\boldsymbol{norm}\Big(\frac{\boldsymbol{v_1 \times v_2}}{\boldsymbol{v_1 \,.\, v_2}}\Big)\right) \qquad (10)$$

We have implemented 170 features based on the Microsoft features described by equations (1)-(10). Instead of obtaining thresholds through experimentation [7] to produce Boolean features we quantised the continuous output of the features into 16 discrete outputs.

We used the skeleton data from the G3D dataset which was split by subjects where the first 5 subjects were used for training and the remaining 5 subjects for testing.

6. Results

The output from the AdaBoost algorithm is per frame confidence results (see Figure 8). Microsoft [1] recommend filtering the result by using a sliding window to sum up the per frame results. A sliding window of size 3 is being used to filter our results (see Figure 9). The filtering is important to link broken actions into a single continuous block. However, the sliding window has the negative side effect of delaying the detection of the action. The length of the delay is directly related to the size of the sliding window so there is an inherent tradeoff between detecting continuous actions and the latency in detecting the action.

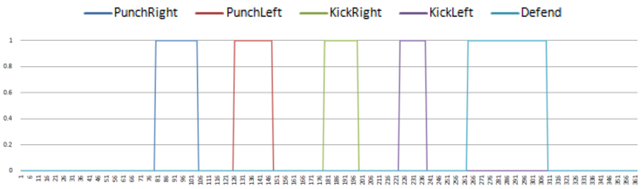


Figure 7. Ground truth (frame based) for trial 24

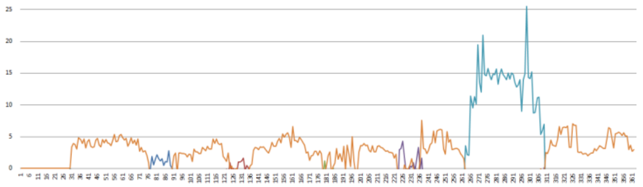


Figure 8. Confidence results (frame based) for trial 24

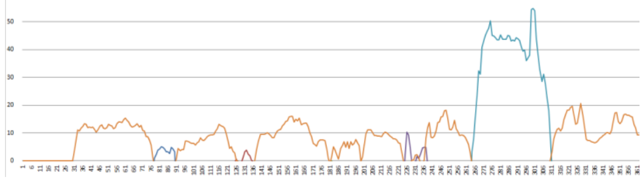


Figure 9. Filtered confidence results (frame based) for trial 24

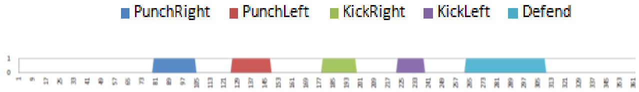


Figure 10. Ground truth (action based) for trial 24

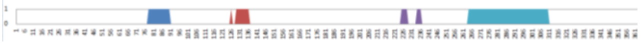
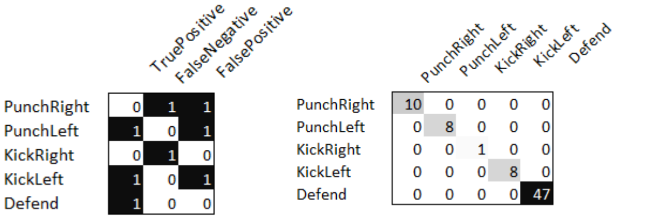


Figure 11. Highest confidence results (action based) for trial 24

The filtered frame based results were then converted to action based results by detecting the class with the highest confidence at each frame. If "other" was the class with highest confidence then no action was predicted. Otherwise, the action predicted was simply the action with the highest confidence. Once an action was predicted sequential frames of the same class where linked to produce a single start and end frame number for each action (see Figure 11).

The predicted actions were then matched against the ground truth actions (see Figure 10) using the new matching process illustrated in Figure 6. Since a confusion matrix is not feasible for the action based approach, a matrix with the estimated true positives, false negatives and false positives is provided in Figure 12 (left). The action-based F1 value for each class was computed from this matrix and then averaged to produce an F1 value for the fighting sequence, of 46.66%. For comparison a frame based confusion matrix was also computed (see Figure 12 right) and the frame-based average F1 value for the same fighting sequence is 61.17%.



	TruePositive	FalseNegative	FalsePositive
PunchRight	0	1	1
PunchLeft	1	0	1
KickRight	0	1	0
KickLeft	1	0	1
Defend	1	0	0

	PunchRight	PunchLeft	KickRight	KickLeft	Defend
PunchRight	10	0	0	0	0
PunchLeft	0	8	0	0	0
KickRight	0	0	1	0	0
KickLeft	0	0	0	8	0
Defend	0	0	0	0	47

Figure 12. Fighting matrices action based (left), frame based (right)

This process was repeated for all sequences in the G3D dataset and Table 1 shows a comparison of the frame based and action based results. The results highlight a dramatic difference in the frame based F1 values and the action based F1 values. Whilst the algorithm appears to be performing well at the frame level it is severely underperforming at the action level. The difference between the frame based and action-based evaluation framework is significant. Therefore, in real-time game applications, a performance measure that does not take into consideration timing and continuity may lead to a wrong conclusion and therefore should be avoided.

Action Category	**Frame based F1**	**Action based F1 (T=4)**
Fighting	70.46%	58.54%
Golf	83.37%	11.88%
Tennis	56.44%	14.85%
Bowling	80.78%	31.58%
FPS	53.57%	13.65%
Driving a car	84.24%	2.50%
Misc	78.21%	18.13%

Table 1 Testing results

7. Conclusion

A novel and reliable evaluation framework for real-time action recognition algorithms was suggested in this work. Compared with the existing metric utilised in this area it provides additional restrictions related to time and repetitions. Therefore, correct recognitions but not in time have the same negative effect in games as other real-time applications. Additionally, recognising multiple times the same action while only one occurrence is present will have disastrous effects in a game and thus it has to be taken into consideration during the evaluation process and the corresponding metrics. Experiments were performed using AdaBoost a real-time action recognition method and further indicate the need of the proposed metric that takes into consideration all these new parameters related to games and real-time applications. Finally, a new dataset for gaming action recognition is provided containing synchronised video, depth and skeleton data. This combined dataset will allow further research into the comparison of pose and appearance based methodologies.

8. References

[1] Marais C. (2011). *Kinect Gesture Detection using Machine Learning* [Online]. Available: http://www.microsoft.com/download/en/details.aspx?id=28066.

[2] M. Marszalek, I. Laptev and C. Schmid, "Actions in context," in CVPR, pp. 2929-2936, 2009.

[3] Microsoft. (2012). *Kinect Natural User Interface (NUI) Overview* [Online]. Available: http://msdn.microsoft.com/en-us/library/hh855352.aspx.

[4] A. Yao, J. Gall, G. Fanelli and L. Van Gool, "Does Human Action Recognition Benefit from Pose Estimation?"BMVC. pp. 67.1-67.11, 2011.

[5] J.K. Aggarwal and M.S. Ryoo, "Human activity analysis: A review," CSUR, vol. 43, 2011.

[6] W. Li, Z. Zhang, and Z. Liu, "Action Recognition Based on A Bag of 3D Points", CVPR4HB, pages 9-14, San Francisco, CA, USA, June 18, 2010.

[7] M. Muller, T. Roder and M. Clausen, "Efficient content-based retrieval of motion capture data," ACM Trans.Graph., vol. 24, pp. 677-685, July 2005. 2005.

[8] C. Schuldt, I. Laptev and B. Caputo, "Recognizing human actions: a local SVM approach,", ICPR. pp. 32-36 Vol.3, 2004.

[9] M. Blank, L. Gorelick, E. Shechtman, M. Irani and R. Basri, "Actions as space-time shapes," in ICCV, pp. 1395-1402, 2005.

[10] D. Thirde. (2005). *PETS: Performance Evaluation of Tracking and Surveillance* [Online]. Available: http://www.cvg.rdg.ac.uk/slides/pets.html.

[11] Home Office, "Imagery Library for Intelligent Detection Systems : The i-LIDS User Guide," 2011.

[12] Carnegie Mellon University. (2011). *Motion Capture Database* [Online]. Available: http://mocap.cs.cmu.edu/.

[13] M. Muller, T. Roder, M. Clausen, B. Eberhardt, B. Kruger and A. Weber, "Documentation Mocap Database HDM05," Universitat Bonn., Tech. Rep. No. CG-2007-2, 2007.

[14] L. Sigal, A.O. Balan and M.J. Black, "HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion," Int.J.Comput.Vision, vol. 87, pp. 4-27, March 2010. 2010.

[15] A. Gilbert, J. Illingworth and R. Bowden, "Action Recognition Using Mined Hierarchical Compound Features," TPAMI, vol. 33, pp. 883-897, 2011.

[16] Freund, Y. and Schapire, R. E. (1996b). Experiments with a new boosting algorithm. In Machine Learning: Proceedings of the Thirteenth International Conference 148–156. Morgan Kaufman, San Francisco.