Asia-Pacific Journal of Science and Technology: Volume: 22. Issue: 03. Article ID.: APST-22-03-08. Research Article

APST

Asia-Pacific Journal of Science and Technology

https://www.tci-thaijo.org/index.php/APST/index

Published by the Faculty of Engineering, Khon Kaen University, Thailand

Rectangular layer model for profile-based human action recognition using multi-view depth information

Pongsagorn Chalearnnetkul,[1] * Nikom Suvonvorn[1]

[1]Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University
*Corresponding author: pongsagorn.ch@gmail.com

Abstract

Human action recognition is a fundamental step for understanding complex activities or behaviors, especially for video surveillance and health-care applications. In this paper, we introduce a profile-based human action recognition model based on data from multi-view cameras and using RGB-D information, which we call the Rectangular Layer Model (RLM). Our model was able to improve performance over the single-view approach by ameliorating perspective distortion and providing more information. This fusion model was tested for five basic actions: walking and standing, sitting, bending, and lying down, at different perspective viewpoints. The system was able to perform at 28.99 fps, and its overall precision is significant at about 92.25%.

Keywords: action recognition, multi-view, depth information, rectangular layer model

1. Introduction

Intelligent video analytics play an important role in the detection and investigation of abnormal events and can contribute to the saving of lives and protection of property, especially in the areas of security and health-care. Action analysis is a fundamental step in understanding human behavior. However, most of the research conducted regarding action analysis uses information from a single viewpoint, which leads to some serious problems such as occlusion, the presence of obstacles, and lack of information. These problems can be solved intuitively with a multi-view approach that uses information from different viewpoints in the same Area of Interest. Additionally, a fusion technique is necessary for combining information from the multi-view cameras. Recently, many interesting fusion methods have been proposed, using either 2D or 3D techniques. Techniques using 2D include the bag-of-visual-words model (which uses a spatio-temporal interest point for human modeling and classification) [1], invariance multi-view action masks and movement representation vectors [2], R-transform feature sets from multiple simultaneous views [3], silhouette feature space with PCA [4], etc. The 3D human models, on the other hand, are reconstructed by fusing the features between views. Thus, calibration between camera viewpoints is necessary, which may increase the complexity and time consumption of the process. Some examples of 3D techniques are the tracking of 3D joints of a skeleton [5], human modeling with 3D circular volume [6], 3D optical flows in 3D motion context [7], and temporal shape similarity in 3D video [8]. In our previous work we focused mainly on 2D-3D techniques, which emphasize single-viewpoint profile-based human action recognition using 3D vector modeling [9], interior 2D part movement [10], deformable triangulation for skeleton extraction using string matching [11], the motion feature of depth map with hierarchical growing neural gas learning (GNG) [12] and the local histogram of optical-flow descripted by rectangular using Adaboost classifier [13].

In this paper, we introduce a method for human action recognition based on data from multi-view cameras and using a feature-level fusion technique called the Rectangular Layer Model (RLM). The RGB-D information of a scene is obtained from two cameras at different viewpoints observing a specific area of Interest. The RLM combines features from these views to construct a human model and classify the actions being displayed using conventional classification methods. The system overview is illustrated in Figure 1.

Typo:

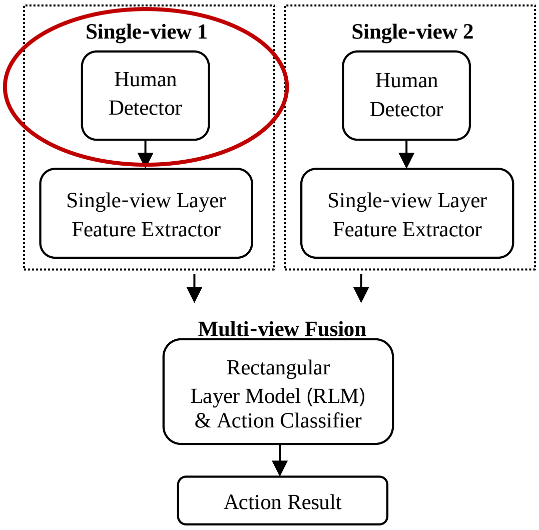

Single-view 1
Human Detector
Single-view Layer Feature Extractor
Single-view 2
Human Detector
Single-view Layer Feature Extractor
Multi-view Fusion
Rectangular Layer Model (RLM) & Action Classifier
Action Result

Figure 1. Overview of the human action recognition system using the Rectangular Layer Model.

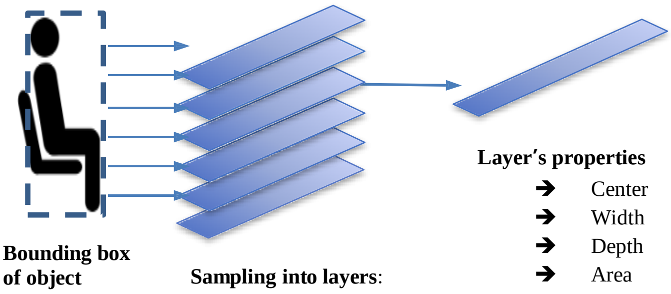


Figure 2. Rectangle layer features.

2. Materials and Methods

2.1 Single-view processing

Single-view processing consists of two main functions: human detection and feature extraction. Preprocessing and human detection consists of five steps: motion detection, noise reduction, filling depth in motion object, object localization, and arm rejection. For motion detection, an adaptive background subtraction, the Gaussian mixture- based background/foreground segmentation algorithm [14], is applied in order to separate the human object from the depth background. The detected movement, or motion object, will be recognized as human. After that, the noise of the motion object is reduced by using opening and closing morphological operators. This is followed by filling in the depth information using an AND operation. The information regarding the motion object and depth information is known as the motion-depth object. Then, we trace the boundary of the object using contour finding to locate the minimum rectangle or bounding box of the object, which will later be used in feature extraction. Additionally, a simple technique for arm rejection is applied in order to decrease the number of false alarms caused by protruding parts that lead to false classification. The technique is based on the horizontal and vertical image projection, adapted from a character segmentation algorithm that is well known in the field of Optical Character Recognition (OCR).

For the feature extraction step, each motion-depth object will be proceeded. We divided the features into two types: regular and penalty. The regular features are represented in a layer format (Figure 2). This concept is derived from the fact that the specific horizontal portions of the human object (also displayed as layers) can represent the characteristics of different actions while allowing for a reduction in the number of dimensions of the feature data.

In the layer sampling step, the motion-depth object surrounded by the bounding box is sampled horizontally $\mathcal{N}$ rows for $\mathcal{N}$ layers. A sampling row ($\mathscr{R}$) of layer could be demonstrated by equation (1). Where $\mathscr{r}$ is the number of object's rows and $\mathcal{N}$ is the number of layers.

$$\mathscr{R}(i) = \left(\frac{\mathscr{r}}{\mathcal{N} + 1} \times i\right) \mid i = 1,2,\dots,\mathcal{N} \qquad (1)$$

After acquiring the row of each layer, its properties will be determined, for instance the width of layer ($\mathcal{W}$) and the depth of layer ($\mathcal{D}$). In each layer i, the $\mathcal{W}$ is calculated from the projection of bright-intensity pixel ($\mathscr{B}$)

where its depth value is greater than the depth Threshold ($\mathcal{T}$), as described in equation (2). The $\mathcal{D}$ is determined by the average of bright-intensity pixel ($\mathcal{B}$) of the motion-depth object, defined by equation (3). Note that c represents the number of the object's columns.

$$\mathcal{W}(i) = \sum_{x=1}^{c} \mathcal{B}\big(x, \mathcal{R}(i)\big) \tag{2}$$

$$\mathcal{D}(i) = \frac{\sum_{x=1}^{c} f\big(\mathcal{B}(x, \mathcal{R}(i))\big)}{\sum_{x=1}^{c} \mathcal{B}(x, \mathcal{R}(i))} \tag{3}$$

In order to correct the depth values that are non-linear due to perspective distortion, depth of layer ($\mathcal{D}$) must be converted to real-depth ($\mathcal{D}_r$) in centimeters with respect to the intensity depth. We use the intensity-depth to real-depth conversion equation proposed by P. Chawalitsittikul [15], which can be derived using polynomial regression power sixth as in following equation (4).

$$\mathcal{D}_r = 1.2512174874918e^{-10}\mathcal{D}^6 - 1.0370379397852e^{-7}\mathcal{D}^5 + 3.5014810721037e^{-5}\mathcal{D}^4 - 0.0061006393631\mathcal{D}^3 + 0.5775953878726\mathcal{D}^2 - 27.63426816635553\mathcal{D} + 5.6759940397611e^2 \tag{4}$$

Finally, both properties need to be normalized using their max values, as shown in equations (5) and (6).

$$\mathcal{W}_n(i) = \frac{\mathcal{W}(i)}{max(\mathcal{W})} \tag{5}$$

$$\mathcal{D}_n(i) = \frac{\mathcal{D}_r(i)}{max(\mathcal{D}_r)} \tag{6}$$

In addition, we establish a penalty feature that aims to support the classification of lying down, as it is quite different from other actions. The feature is defined as width-height ratio ($\propto$) of the bounding box of the object as shown in the following equation (7):

$$\propto = \frac{c}{r} \tag{7}$$

The features mentioned above will be applied to the multi-view fusion using the models detailed in the following section.

2.2 Multi-view feature fusion

In single-view processing, features might be insufficient to recognize actions affected by perspective distortion or lack of information. Thus the fusion of features ascertained from several single-views is necessary. Our proposed multi-view fusion method is based on the rectangular layer feature. Two models are established: the Rectangular Layer Model (RLM) and Inverse Rectangular Layer Model (InvRLM).

The Rectangular Layer Model (RLM) is a simple model that will use only the width of the layer ($\mathcal{W}_n$) to estimate the rectangular area the feature ($\mathcal{RA}$) with a maximum width-height ratio ($\propto_{max}$). This concept is demonstrated in Figure 3.

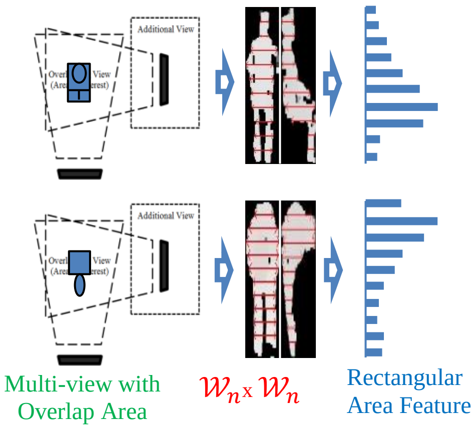

Additional View
Additional View
Multi-view with Overlap Area
$\mathcal{W}_n$ x $\mathcal{W}_n$
Rectangular Area Feature

Figure 3. Demonstration of the rectangular layer model to analyze sitting and bending actions.

The Rectangular Area feature ($\mathcal{RA}$) in each layer is determined by the multiplication of the widths of the layers ($\mathcal{W}_n$) by those of their corresponding layers, as shown in equation (8).

$$\mathcal{RA}(i) = \mathcal{W}_n(i)_{\text{view1}} \times \mathcal{W}_n(i)_{\text{view2}} \tag{8}$$

This will be concatenated into a feature vector that is appropriate to the training technique used in the relevant classification method. The feature vector is defined as follows:

$$\vec{F}_{\mathcal{RA}} \in \{\, \mathcal{RA}(1), \mathcal{RA}(2), \mathcal{RA}(3), \ldots, \mathcal{RA}(\mathcal{N}), \propto_{max} \,\} \tag{9}$$

The Inverse Rectangular Layer Model (InvRLM) is derived from the fact that when cameras are set up to view a scene horizontally in the manner of an overhead view, the depth value will inversely indicate the hidden volume of the object. For example, in the case of a front view of someone sitting, the depth value of the upper legs will be smaller than those of the other parts of body, while having a width of layer ($\mathcal{W}_n$) value nearly equal to the other parts, as shown in Figure 4. Therefore, we created an Inverse-Depth Weight Kernel ($\mathcal{IDK}$), as described in equation (10), to adjust the rectangular area feature ($\mathcal{RA}$). This will enhance the pattern of features that lead to better classifications of some actions. Thus, in the improved feature, the $\mathcal{W}_n$ will be multiplied by $\mathcal{IDK}$ in order to induce the corresponding actions, as a weighted $\mathcal{W}_n$ ($\mathcal{W}_w$). This is shown in equation (11).

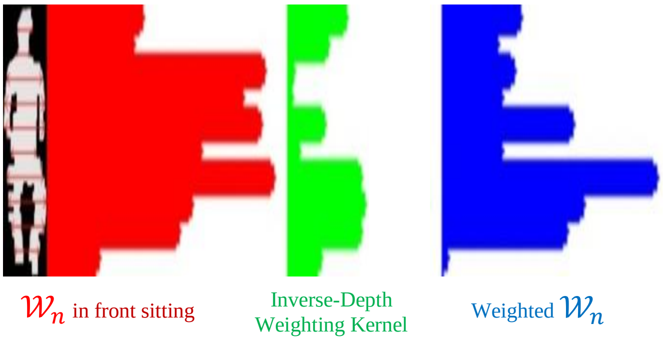

$\mathcal{W}_n$ in front sitting
Inverse-Depth Weighting Kernel
Weighted $\mathcal{W}_n$

Figure 4. Example of Inverse Depth Weighting Kernel.

$$\mathcal{IDK}(i) = \frac{(\,\mathcal{D}_n(i) - max(\mathcal{D}_n)\,)}{(\,min(\mathcal{D}_n) - max(\mathcal{D}_n)\,)} \tag{10}$$

$$\mathcal{W}_w(i) = \mathcal{W}_n(i) \times \mathcal{IDK}(i) \tag{11}$$

However, we still believe in both width of layer ($\mathcal{W}_n$) and weighted width of layer ($\mathcal{W}_w$) features. Thus, the adjustment of the weight of these features is balanced by using $\bar{\alpha}$ learning rate, which will provide a completely weighted width of layer ($\mathcal{W}_{\bowtie}$), as shown in equation (12). The weighted rectangular area feature ($\mathcal{RA}_w$) is then estimated by equation (13). Finally, the $\mathcal{RA}_w$ is concatenated with the maximum width-height ratio ($\propto_{max}$) from all views into the feature vector, as determined by equation (14), which is suitable for classification.

$$\mathcal{W}_{\bowtie}(i) = (\,(1 - \bar{\alpha}) \times \mathcal{W}_n) + (\,\bar{\alpha} \times \mathcal{W}_w\,) \tag{12}$$

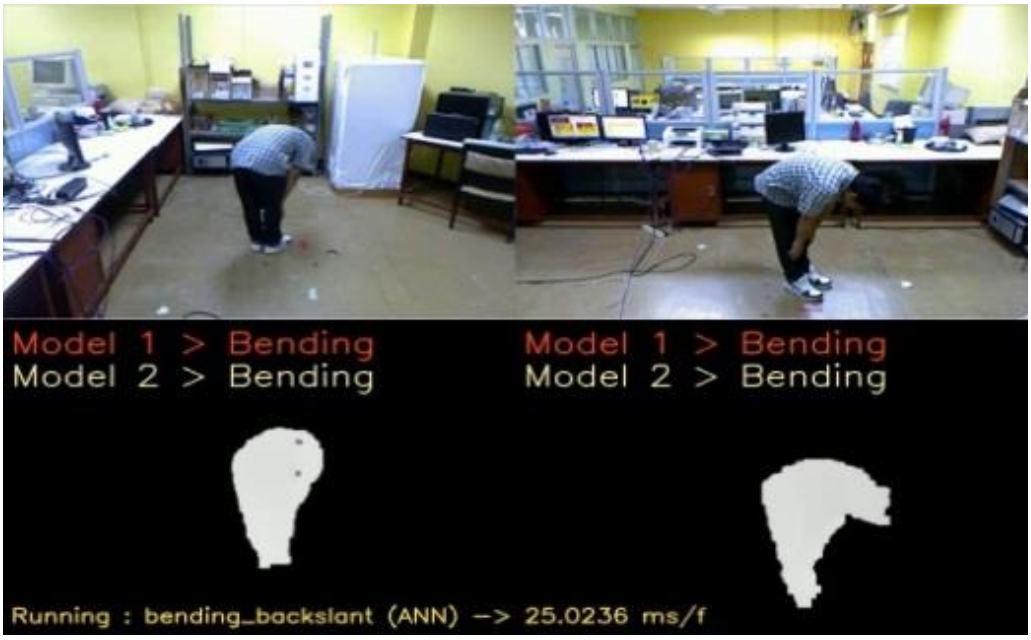
$$\mathcal{RA}_w(i) = \mathcal{W}_{\bowtie}(i)_{\text{view1}} \times \mathcal{W}_{\bowtie}(i)_{\text{view2}} \tag{13}$$

$$\vec{F}_{\mathcal{RA}_w} \in \{\, \mathcal{RA}_w(1), \mathcal{RA}_w(2), \mathcal{RA}_w(3), \ldots, \mathcal{RA}_w(\mathcal{N}), \propto_{max} \,\} \tag{14}$$

In our experiment, we used two traditional action classification methods: artificial neural network (ANN) and support vector machine (SVM), applied to both fusion models. We trained the ANN with a back-propagation algorithm using 10 and 20 hidden nodes for the Models 1 and 2 respectively. Also, we trained the SVM using radial basis and C-SVC functions.

3. Results and Discussion

There were standard benchmarks or datasets established that focused on action analysis using in-depth data and multiple views. We, thus, built our own datasets. The action range is about 3 to 5.5 meters from cameras that could ascertain an image of the full human body. The training dataset included 8700 action frames in a clear room, with a perpendicular angle between the cameras. An example of the scene is shown in Figure 5. In addition, the testing dataset was built from a different scene and included 7800 action frames with three different angles (45°, 60°, and 90°). An example of the scene is shown in Figure 6. Our system was tested with an Intel Core i5 4590 3.30GHz processor using OpenCV for the computer vision library, OpenMP for parallel processing, and CLNUI for acquiring the depth information from the Kinect camera. We divide the single-view processing into threads which were run simultaneously using parallel processing.

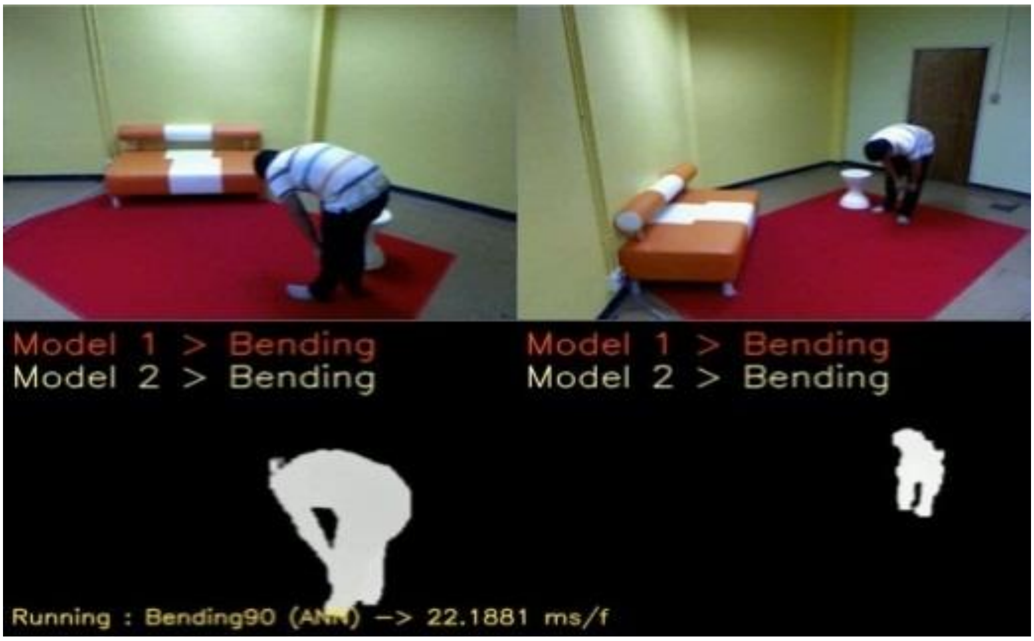


Model 1 > Bending
Model 2 > Bending

Model 1 > Bending
Model 2 > Bending

Running : bending_backslant (ANN) -> 25.0236 ms/f

Figure 5. Example of the scene used in the experiment in the training dataset.



Model 1 > Bending
Model 2 > Bending

Model 1 > Bending
Model 2 > Bending

Running : Bending90 (ANN) -> 22.1881 ms/f

Figure 6. Example of the scene used in the experiment in the test dataset.

Table 1. Average time consumption of system.

Classifier	Time Consumption (ms)		Frame Rate (fps)	
	Model 1	Model 2	Model 1	Model 2
ANN	34.26	34.49	29.19	28.99
SVM	35.53	34.82	28.15	28.72

We measured the time consumption using an OpenMP wall-clock and OpenCV tick-frequency timer, which do not affect other measurements of time consumption for models and classifiers. We found that the time consumption, or execution time, of each frame was around 35 milliseconds, or 29 fps, as shown in Table 1.

Table 1 shows that the average time consumption of the different models were very similar. The minimum processing time is 34.26 ms in model 1 while using an ANN Classifier. The processing frame rate was around 28-30 fps, which was about the same as that of the Kinect camera (30 fps). This shows that this system can be implemented in real-time.

A precision measurement was estimated from frames of both views in which a human was detected. A given result was counted as a true positive only if the target actions are correctly classified. In contrast, a false positive was when the actions were recognized incorrectly. True and false negatives were not evaluated. To evaluate the robustness of the fusion model in detecting each action, we defined a test dataset where the angle between cameras was different. In addition, we tested the performance of both artificial neural network and support vector machine classification methods, as shown in sections A and B, respectively. The testing dataset concerns the actions at normal speed and view of the human from every perspective. The recorded videos used in our experimental Artificial Neural Network (ANN) model are available at https://www.youtube.com/watch?v=WRAq0A-0vhk (Used with the Test Dataset) and https://www.youtube.com/watch?v=uTPiyOlnlFA (Used with the Training Dataset).

3.1 Experimentation using the RLM

We tested the RLM using only rectangle-shaped areas, as these would produce least complex models. We tested both ANN and SVM. The average of the ANN results was 90.37% and that of the SVM results was 83.97%, suggesting that ANN performs better than SVM (Table 2). The majority of the RLM results were significant with respect to actions and angles between cameras. However, the performance of this model with regard to detecting when a person is sitting was not adequate at a narrow angle. The features of the sitting action combined with a slanted view were particularly similar to the walking & standing action. When the angle between the cameras was very narrow, the precision fell accordingly.

3.2 Experimentation using the InvRLM

The experimental results are detailed in Table 3. In summary, the overall accuracy percentages of the InvRLM model using ANN and SVM were 92.25% and 85.57%, respectively. In this experiment, the weight adjustment parameters ᾱ were specified at 0.5 and 0.7. We found that the precision was 82.50% when ᾱ was 0.5 and 92.25% when ᾱ was 0.7. The optimal value of ᾱ was, thus, 0.7. We found that the InvRLM model, which is the enhancement of RLM by using properties of real-depth to weight the pattern of the rectangle area, was able to improve the precision in detecting when a person is sitting, which is one of the problems when using the RLM. However, it had lower precision in detecting the lying-down action due to the lack of visible body parts, which normally requires the full-part of human object. Most errors in detecting lying down are the result of a lack of visible humas body parts, which causes the depth value to be close to background, making it difficult to distinguish the motion.

Table 2. Precision of RLM.

Action	Angle Between Camera					
	Precision of ANN (%)			*Precision of SVM (%)*		
	45°	*60°*	*90°*	*45°*	*60°*	*90°*
Walking & Standing	96.75	99.09	99.35	90.94	99.64	96.12
Sitting	69.17	75.90	87.67	50.78	52.87	63.85
Bending	92.35	87.16	83.76	88.71	86.53	80.13
Lying Down	98.82	97.47	96.97	98.63	100.00	99.42
Average by Angle	89.27	89.91	91.94	82.26	84.76	84.88
Overall	90.37			83.97		

Table 3. Precision of InvRLM.

Action	Angle between camera					
	Precision of ANN (%)			*Precision of SVM (%)*		
	45°	*60°*	*90°*	*45°*	*60°*	*90°*
Walking & Standing	95.90	98.18	93.53	91.11	95.09	84.48
Sitting	97.91	99.79	99.43	99.79	100.00	99.72
Bending	93.62	89.05	87.61	93.81	85.68	83.33
Lying down	81.37	84.44	86.17	73.73	62.45	57.64
Average by Angle	92.20	92.86	91.68	89.61	85.81	81.29
Overall	92.25			85.57		

Table 4. Precision comparison.

Method	Precision rates of action (%)					
	Standing	*Walking*	*Sitting*	*Bending*	*Laying*	*Average*
Our method	95.87	95.87	99.04	90.09	83.99	92.97
P. Chawalitsittikul [9]	98.00	98.00	93.00	94.10	98.00	96.22
N. Noorit [10]	99.41	80.65	89.26	94.35	100.0	92.73
Chi-Hung [11]	-	92.40	97.60	95.40	-	95.80
G. I. Parisi [12]	96.67	90.00	83.33	-	86.67	89.17
N. Sawant [13]	91.85	96.14	85.03	-	-	91.01

3.3 Precision Comparison

Table 4 shows the comparison results. We compared the performance of our methods with the single-view action recognition system, proposed by P. Chawalitsittikul [9] and N. Noorit [10]. The InvRLM experiment was performed using an ANN classifier (average values from all angles between cameras). We found that the method tested here demonstrated approximately the same precision as other methods. However, our method was able to perform well at various orientations, and its execution was faster at 28.99 fps.

4. Conclusion

In this paper, we proposed the multi-view fusion model for profile-based human action recognition using depth information. Our fusion models were able achieve action recognition at every camera viewpoint. The overall average precision was 92.25% using InvRLM with an ANN classifier. In addition, the task was able to be performed in real-time at 28.99 fps.

Going forward, we will apply these new advanced methods to better action classification. We will also consider human models that are able to recognize more complex actions. Additionally, the features of future projects might be partially in 3D for structural movement modeling.

5. Acknowledgement

This work was supported by a Graduate School Thesis Grant awarded by Prince of Songkla University in 2015.

6. References

[1] Liu, J., Shah, M., Kuipers, B., Savarese, S., 2011. Cross-view action recognition via view knowledge transfer, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3209-3216.

[2] Gkalelis, N., Nikolaidis, N., Pitas, I., 2009. View independent human movement recognition from multi-view video exploiting a circular invariant posture representation, in: IEEE International Conference on Multimedia and Expo (ICME)., 394-397.

[3] Souvenir, R., Babbs, J., 2008. Learning the viewpoint manifold for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1-7.

[4] Ahmad, M., Lee, S.W., 2006. Hmm-based human action recognition using multi-view image sequences, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 1, 263-266.

[5] Tran, C., Trivedi, M.M., 2008. Human body modeling and tracking using volumetric representation Selected recent studies and possibilities for extensions, in: ACM/IEEE International Conference on Distributed Smart Cameras, 1-9.

[6] Pehlivan, S., Duyugulu, P., 2010. A new pose-based representation for recognizing actions from multiple cameras. Journal of Computer Vision and Image Understanding ACM 115, 140–151.

[7] Holte, M.B., Moeslund, T.B., Nikolaidis, N., Pitas, I., 2011. 3D human action recognition for multi-view camera systems. Proc 3D Imaging, Modeling, Processing, Visualization and Transmission IEEE., 342-349.

[8] Huang, P., Hilton, A., Starck, J., 2010. Shape similarity for 3D video sequences of people. Journal of Computer Vision Springer 89, 362-381.

[9] Chawalitsittikul, P., Suvonvorn, N., 2012. Profile-based Human Action Recognition using Depth Information. Proc Advances Computer Science and Engineering ACTA Press, 376–380.

[10] Noorit, N., Suvonvorn, N., Karnchanadecha, M., 2010. Model-based Human Action Recognition. Proc Digital Image Processing SPIE 7546.

[11] Chuang, C.H., Hsieh, J.W., Tsai, L.W., Fan, K.C., 2008. Human Action Recognition Using Star Templates and Delaunay Triangulation. Proc Intelligent Information Hiding and Multimedia Signal Processing, 179-182.

[12] Parisi, G.I., Weber, C., Wermter, S., 2014. Human action recognition with hierarchical growing neural gas learning. Proc Artificial Neural Networks Springer 8681, 89-96.

[13] Sawant, N., Biswas, K.K., 2009. Human Action Recognition Based on Spatio-temporal Features. Proc Pattern Recognition and Machine Intelligence Springer 5909, 357-362.

[14] KaewTraKuPong, P., Bowden, R., 2001. An improved adaptive background mixture model for real-time tracking with shadow detection. Proc Advanced Video-Based Surveillance Systems Springer, 135-144.

[15] Chawalitsittikul, P., 2013. Side-View Based Human Action Recognition Using Stereo Vision [Thesis]. Songkhla: Prince of Songkla University, Thai.