Kim *et al. Hum. Cent. Comput. Inf. Sci. (2017) 7:25*
DOI 10.1186/s13673-017-0106-5

Human-centric Computing and Information Sciences

RESEARCH

Open Access

A climbing motion recognition method using anatomical information for screen climbing games

Jungsoo Kim[1], Daniel Chung[2] and Ilju Ko[2*]

*Correspondence: andy@ssu.ac.kr
[2] Department of ICMC Convergence Technology, Soongsil University, 369 Sangdo-ro, Dongjak-gu, Seoul, South Korea
Full list of author information is available at the end of the article

Abstract
Screen climbing games have made a new category of gaming experience between a human climber and a virtual game projected onto an artificial climbing wall. Here, climbing motion recognition is required to interact with the game. In existing climbing games, motion recognition is based on a simple calculation using the depth difference between the climber's body area and the climbing wall. However, using the body area in this way is devoid of anatomical information; thus the gaming system cannot recognize which part, or parts, of the climber's body is in contact with the artificial climbing wall. In this paper, we present a climbing motion recognition method using anatomical information obtained by parsing a climber's body area into its constituent anatomical parts. In ensuring that game events consider anatomical information, a climbing game can provide a more immersive experience for gamers.

Keywords: Body area, Body parts, Kinect, Motion recognition, Screen climbing

Introduction

Recently, games integrating information technology and real world sports have hit the market in response to an increased demand for these activities. In such games, the gaming experience is created via engaging content and a human–computer interface (HCI) [1]. In particular, screen sports have utilized a combination of artificial environments and active human motion to create an immersive experience of familiar sports such as golf, baseball, and horseback racing [2, 3]. Similar human–computer interaction technologies have been applied to trampolining, climbing, and mixed martial arts, among others [4].

Screen climbing games, a new category of sports gaming experience, engage climbers with game content projected onto an artificial climbing wall. Raine Kajastila suggested Whack-a-Bat, Spark, and Climball [5, 6], and Jungsoo Kim suggested Ancient Cave Exploration [7]. These games use motion recognition technology native to the Microsoft Kinect to generate depth map image. Here, the body area of a climber is obtained by calculating the difference between the background of the artificial climbing wall environment and the foreground for scaling the climbing wall. This approach has a lower misrecognition rate than that using only skeletal information. In particular, the hand-or-foot recognition accuracy is higher.

Touch events can be handled using the difference between depth map images and an event, if the depth difference is less than a specified difference between a climber's body area and the artificial climbing wall. This information is used to decide whether a climber has obtained a game item or collided with an obstacle. However, anatomical information is not used during this process, so this system cannot recognize the location of a motion on the climber's body. This limits the variety of game events that can be created in screen climbing games [8].

In indoor sports climbing, the hands and feet are the nearest body parts to the climbing wall, as they make contact with the climbing holds. The location of hands and feet can be derived from the position of various appendages. Using the location information of climbing holds installed on a climbing wall, the climbing hold location in contact with a hand or foot can be ascertained. However, to parse a climber's body area into recognizable parts, both depth map difference and the skeletal system information are required. Doing so enables designers to define game items or obstacles that respond to movements from specific body parts. This creates more variety and interactivity in screen climbing games. Therefore, we propose a climbing motion recognition method using anatomical information obtained by classifying body parts using the climber's body area and skeletal information.

We describe different climbing games in "Related work", and the methods we use to parse of climber's body area into constituent parts are described in "Parsing a climber's body area into body parts". In "Climbing motion recognition", a novel motion recognition technique is presented, and we describe our experiments in "Discussion and results". Finally we conclude our paper in "Conclusion".

Related work

Animated Saw

Animated Saw [5] is a screen climbing game developed by Raine Kajastila. Here, a climber must avoid a chain saw that moves along a defined path, as illustrated in Fig. 1. The game is over when a climber touches the chain saw. The chain saw moves linearly, and also rotates, and reacting to more chain saws increases the difficulty of the game as climbers advance in the levels.

The motion recognition technology used in this game is based on skeletal information derived from the Kinect device. In response, a game event occurs when an appendage collides with a game object. Raine Kajastila explains that a customized climber tracker is required to recognize the climber's motion, because skeletal information is not trustworthy in an indoor climbing environment. In particular, the skeleton is unstable when a hand or foot is touching the climbing wall.

Spark

Spark [6] is a climbing game in which a climber has to avoid the electric lines, as illustrated in Fig. 2. The game starts when the climber touches the play button inside the area surrounded by electric lines, which move toward and rotate slowly around the location of the stop button. The climber moves by adjusting his or her climbing posture to avoid touching the electric lines. The game is over when the climber's body touches an electric

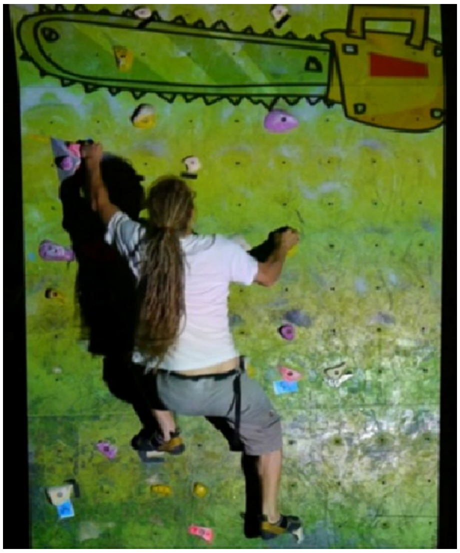


Fig. 1 Animated Saw game

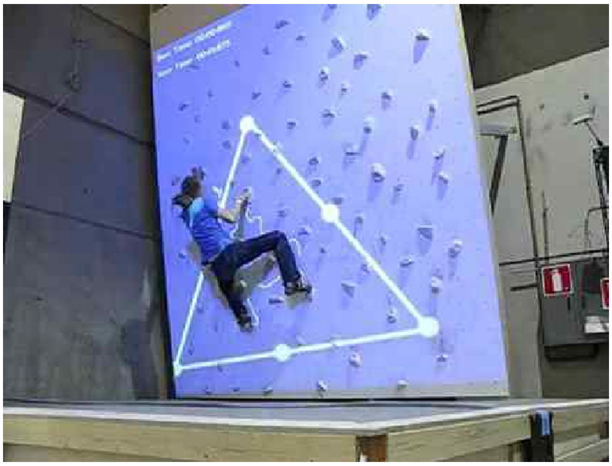


Fig. 2 Spark game

line, at which point the outline of the climber's body is displayed on the projected screen on the climbing wall.

The climber survives if he or she touches the stop button without touching one of the electric lines. When the game is finished, the player can move to the next stage or retry the current stage. The climbing motion recognition technology uses a depth map from the Kinect device to solve the inconsistent motion recognition caused by unstable anatomical information. Accordingly, a one-second averaged background depth map image is obtained to reduce random variance in depth values for the same pixel. The background is defined as the portion of a depth map image where the climber does not appear inside the climbing wall. The depth map difference between the foreground image and background image is used to separate the climber's body area from the background image, and a touch event occurs when the depth difference between the climber's body area and the background is between 2 and 8 cm.

Ancient Cave Exploration

Ancient Cave Exploration [7] is a screen climbing game based on exploring a natural cave, as illustrated in Fig. 3. The game starts when the climber touches the start button. Subsequently, multiple stalactites fall from the top of the cave, as indicated by a falling sound. The target location (the cave entrance location) appears opposite from the starting location. The climber moves toward the target location to clear the stage, while avoiding obstacles and utilizing the effects of game items. The game is over when the climber collides with an obstacle, or does not move for a long time.

The game consists of six stages. A mission is considered successful when the climber moves to the target location (the treasure box location). The game objects are divided into obstacles and game items. Obstacles (stalactites, bats, and spiders) are objects that cause the game to terminate when the climber collides with them; whereas, a game item is an object that benefits the climber when he or she touches it. The target location, lantern, and treasure box are the game objects.

The motion recognition technology in this game is based on the depth difference between the background and foreground images, similar to the motion recognition used in the Spark game. The depth difference image is binarized to obtain the candidate area of the climber's body area. Then, the climber's body area is obtained using a morphology operation.

Parsing a climber's body area into body parts

In this paper, we propose a method of parsing a climber's body area into body parts for climbing motion recognition. The purpose of the proposed method is to trigger game events in response to human motion so that an interactive game can accurately respond to the climber's actions. Figure 4 shows the overall process of the proposed method. Here, depth map information and anatomical recognition are continuous data streams provided by a Kinect device. The stages of the proposed method are body area detection, correction for hand and foot joints, and appendage classification.

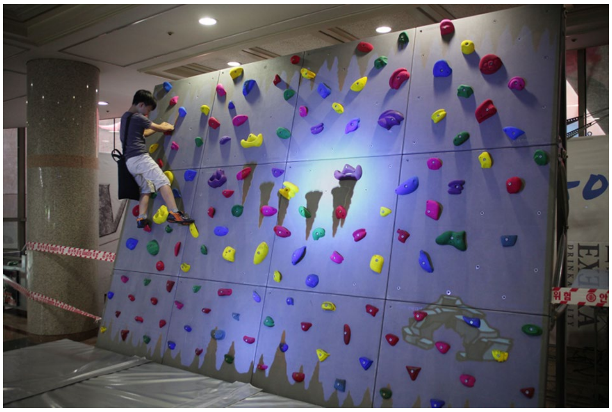


Fig. 3 Ancient Cave Exploration game

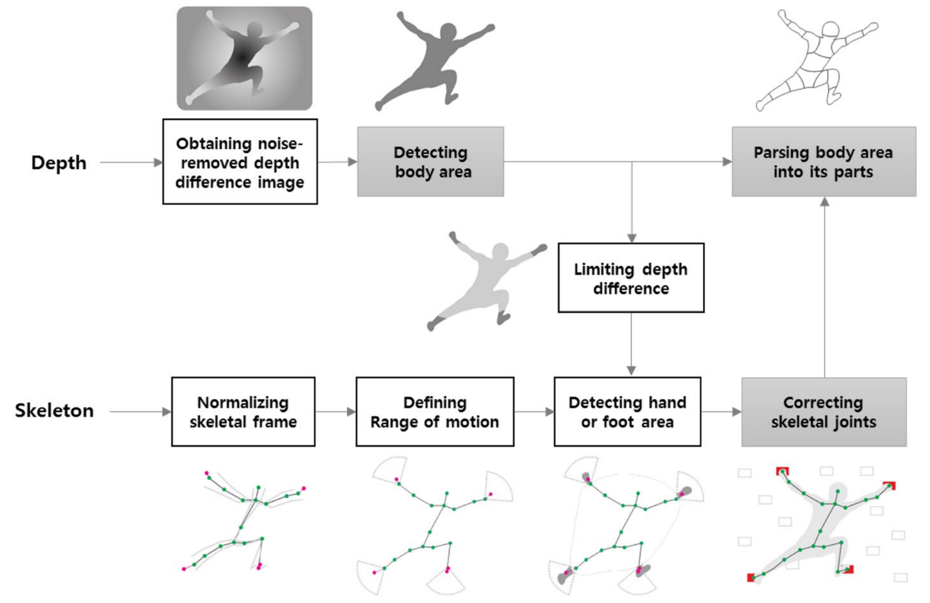

Depth
Obtaining noise-removed depth difference image
Detecting body area
Parsing body area into its parts
Limiting depth difference
Skeleton
Normalizing skeletal frame
Defining Range of motion
Detecting hand or foot area
Correcting skeletal joints

Fig. 4 Overall process for parsing a climber's body area into body parts

Body area detection

The climber's body area is detected using the depth difference between the background and foreground images. The background is an image of the environment that is captured when installing the artificial climbing wall, while the foreground is an image of a climber scaling the climbing wall. The depth difference is bigger within the climber's body area than in the rest of the foreground image because the latter is the same as its corresponding part in the background image; thus, only the body area can be detected using depth difference [9].

However, the depth values vary for each depth map frame because of the noise around the boundaries of the climbing holds [10]. To reduce this noise, we calculate the following depth map frames: the initial depth difference, the averaged noise frame, and the final depth difference. Before calculating the initial depth difference, the averaged background image is obtained by averaging multiple background images from different depth map frames. The initial depth difference is calculated by subtracting the averaged background image from a foreground image. The averaged noise frame is obtained by averaging the depth difference between the averaged background image and a specific background image. The final depth difference is obtained by subtracting the averaged noise frame from the initial depth difference. Figure 5 shows the entire process of detecting the climber's body area.

Correction for hand and foot joints

Figure 6 shows the process of correcting for hand and foot joints. First, in a process called *skeletal frame normalization*, we correct all skeletal joints using the most recent skeletal frames to obtain reliable skeletal system information. A correction weight is assigned to each skeletal joint in the skeletal frames. This value is bigger if a joint is more reliable and if a frame is more recent. The reliability of a skeletal joint is divided into the following three states sorted by reliability: tracked, inferred, and not tracked. The detailed correction process for a skeletal joint *j* is shown in Fig. 7.

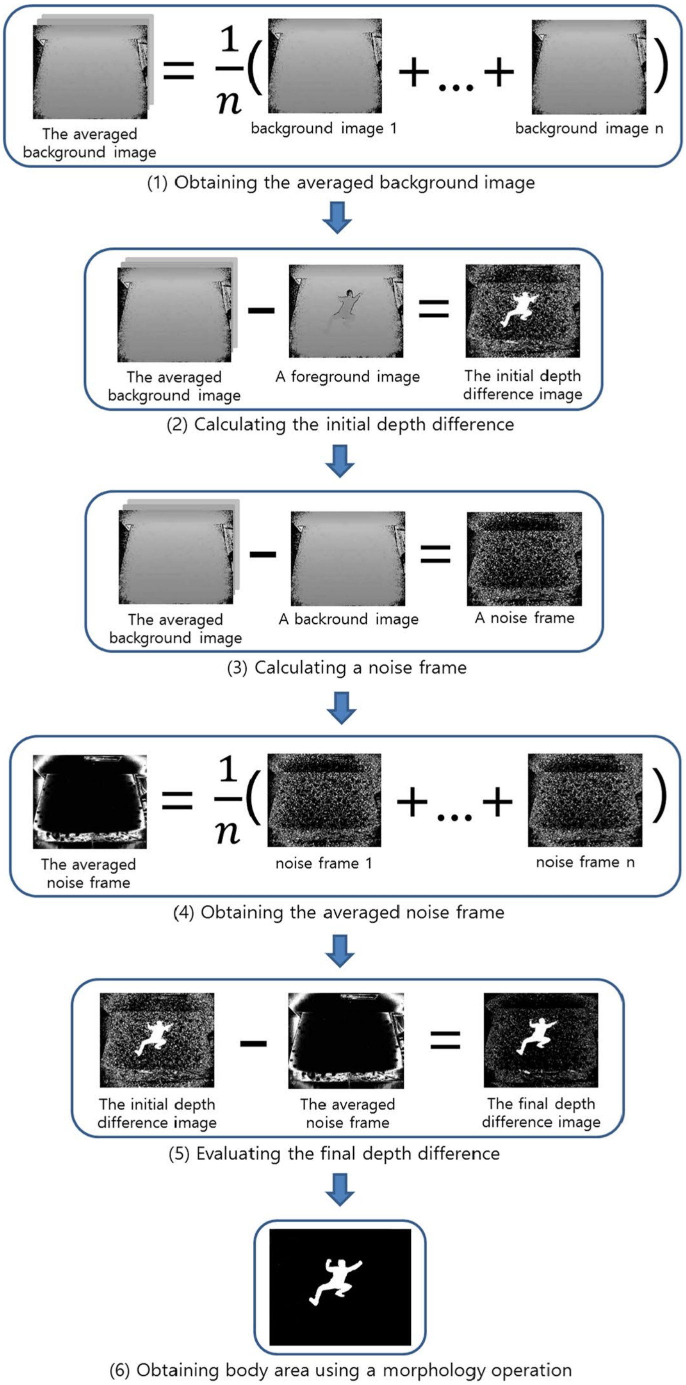

The averaged background image
$= \frac{1}{n}($ background image 1 $+ \ldots +$ background image n $)$
(1) Obtaining the averaged background image
The averaged background image
A foreground image
The initial depth difference image
(2) Calculating the initial depth difference
The averaged background image
A backround image
A noise frame
(3) Calculating a noise frame
The averaged noise frame
$= \frac{1}{n}($ noise frame 1 $+ \ldots +$ noise frame n $)$
(4) Obtaining the averaged noise frame
The initial depth difference image
The averaged noise frame
The final depth difference image
(5) Evaluating the final depth difference
(6) Obtaining body area using a morphology operation

Fig. 5 The process of detecting the climber's body area

The next stage of skeletal correction is defining the range of motion for each hand and foot joint, then finding the candidate area for each hand and foot, as illustrated in Fig. 6b. The range of motion is estimated using the angles between the elbow and hand, and the knee and foot. By using a range of motion information, we can find the candidate area for each hand or foot.

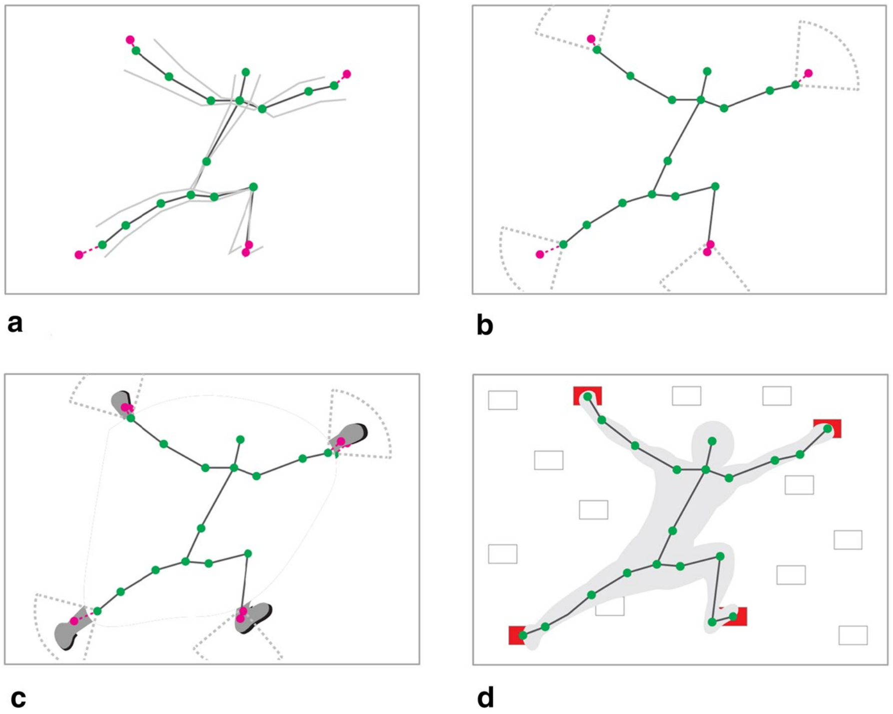


Fig. 6 The process of skeletal joint correction for the hand-or-foot area. **a** Skeletal frame normalization, **b** range of motion definition in hand-or-foot area, **c** hand-or-foot detection, **d** skeletal joint correction of hand-or-foot area

$$LOC(\bar{j}, f_t) = \frac{\sum_{i=0}^{n-1}[\, w(j,f_{t-i})LOC(j,f_{t-i})\,]}{\sum_{i=0}^{n-1} w(j,f_{t-i})}$$

where

$$w(j, f_{t-i}) = w_0(j)\, e^i\, s(j, f_{t-i}) \text{ and}$$

$$s(j,f) \begin{cases} 1: \mathit{tracked} \\ \alpha: \mathit{inferred} \quad , \; 0 \le \alpha \le 1 \\ 0: \mathit{not\ tracked} \end{cases}$$

$\bar{j}$ is the corrected joint j by skeletal normalization
f_t is the current depth map frame at time t
f_{t-i} is the i-th previous depth map frame at time t
$LOC(j, f)$ is the location of j in the frame f
$w_0(j)$ is the initial correction weight for j
$w(j, f)$ is the correction weight for j in the frame f
$s(j, f)$ is the tracked state of j in the frame f
α is the weight reflecting the traked state

Fig. 7 Calculating a joint location for skeletal normalization

Third, we use the body area information to find the candidate area for each hand or foot, as illustrated in Fig. 6c. When either is close to the artificial climbing wall, the skeletal joint in the hand-or-foot area is unreliable; thus, we need to detect the smallest previous area of depth difference in the body area. If the detected area is in the range of motion area for the hand or foot, the detected area is considered as a hand or foot;

otherwise, it is considered as a hand-or-foot candidate area. The distance from the hand or foot is used if the non-detected hand or foot previously existed.

Lastly, we correct the skeletal system of the hand-or-foot area using both its detected area and the climbing hold area information, as illustrated in Fig. 6d. If the detected hand-or-foot area overlaps with the climbing hold area, we can consider the center of the climbing hold area as the location of a hand or foot; otherwise, the location is the center of the detected hand-or-foot area.

Appendage classification

The corrected skeletal system information for the hands and feet is used to parse the climber's body area into constituent parts. To do so, we overlap the corrected skeletal system information with the climber's body area in the same depth map coordinates. We then use the recognition area of each joint from its joint location in the skeletal system. Figure 8 shows a body area parsed into body parts. The recognition area of each joint is expanded simultaneously at the same expansion rate, as shown in Fig. 8a, and can be expanded even if the corresponding joint location is outside the body area. In this instance, the recognition area is restricted to the expanded region in the body area, and the expansion is finished when all pixels of the climber's body area are parsed into body parts, as shown in Fig. 8b.

The process shown in Fig. 8 can be translated to find the nearest skeleton joint from a given pixel in the detected body area. Here, the recognition area of a specific joint ***j*** is the set of pixels satisfying the following condition: the nearest joint from a pixel is ***j***. The algorithm for the classification process is shown in Fig. 9.

Figure 10 shows body parts classified using the method described above. This information is used for appendage-specific event.

Climbing motion recognition

Climbing motion is recognized using motion recognition events, for which anatomical information is used to detect motion events in response to game objects, the artificial climbing wall, and climbing holds. A motion recognition event is divided into a body part recognition event and a tactile event initiated by a hand or foot.

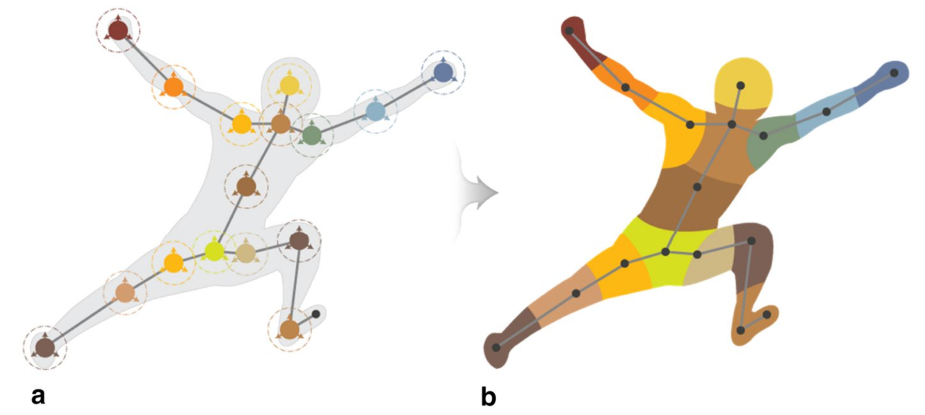


Fig. 8 The process of body area parsing into body parts. **a** Starting body area classification by expanding recognition area from each skeletal joint. **b** Finished expanding the recognition area

```
For a pixel p in the body area
  Set minDist to the largest distance of the depth map coordinate
  For a joint j in the skeleton
    Get the joint location LOC(j)
    Find the distance dist(j,p) between LOC(j) and p
    If dist(j,p) < minDist then
      Set nearestJoint(p) to j
      Set minDist to dist(j,p)
    End if
  End for|
  Classify p to bodyPart(x) where x = nearestJoint(p)
End for
```

Fig. 9 Body area classification algorithm

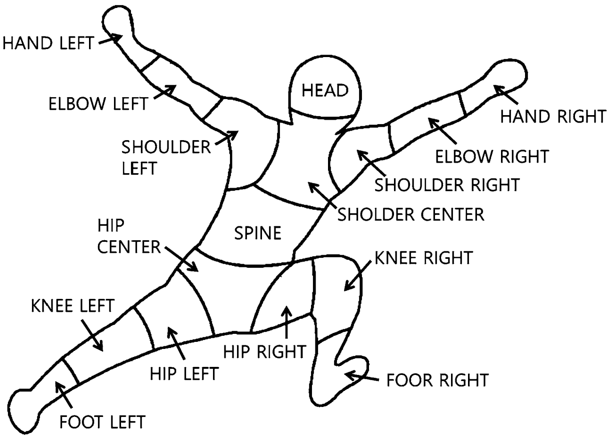


Fig. 10 Body parts classified using the proposed method

A body part recognition event occurs as a climber scales the wall and some body parts overlap with a recognition object. The depth difference between the overlapped body parts and the climbing wall should be less than 1 m. A tactile event occurs when a hand-or-foot area approaches the climbing wall. Here, the depth difference should be between 5 and 20 cm. The object touched is deciphered using climbing hold information. If the touched location matches one of the climbing holds, the touched object is classified as such; otherwise, it is considered to be part of the climbing wall. Figure 11 shows a recognized object and an event occurrence.

Discussion and results

This study consisted of finding a way to parse a climber's body area into constituent parts and recognize their motion. The experimental environment consisted of an artificial climbing wall, beam projector, Kinect, and client, as shown in Fig. 12. The area of the climbing wall was 4 × 3 m (width × height), and the beam projector was used to display the virtual environment onto the climbing wall. We used a Microsoft Kinect v2

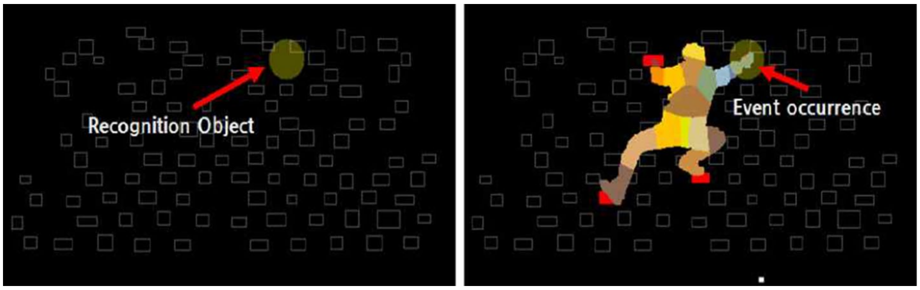


Fig. 11 A recognized object and an event occurrence

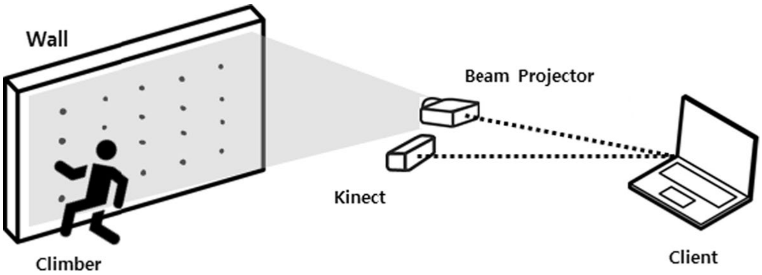


Fig. 12 The experimental environment

for Windows to detect motion. The Kinect box was located in front of the climbing wall in order to record its entire area. The climbing motion recognition program for the proposed method was installed in the client.

Validating the quality of appendage classification

In order to guarantee the quality of appendage classification, the system needed to check the amount of noise in the difference between the foreground and background images, as well as the trustworthiness of the skeletal system information obtained from skeletal frame normalization. We checked the first criterion by comparing the white pixel count between the body area detection methods. In the same depth map frame, the amount of noise depended on the white pixel count, since the final difference image was binarized and the white pixel count of the body area was similar in both images. Figure 13 shows this comparison result. Here, A is the naïve method, B shows the Raine Kajastila's method, and C illustrates the proposed method.

The naïve method used the basic difference between a foreground and background image. The Raine Kajastila's method assessed the difference between a foreground image and one-second averaged background images, as described in "Related work". The proposed method involved subtracting the averaged noise frame from the initial depth difference as described in "Parsing a climber's body area into body parts". The proposed method (C) had the smallest amount of noise, indicating that we could easily remove the noise.

We checked the second criterion by evaluating variations in the location of a specific skeletal joint, measured as distance similarity. If the variance was small and stable, valid skeletal system information could be obtained. The distance similarity of joint location converged to 0 if the variation in the joint location between the skeletal frames was large;

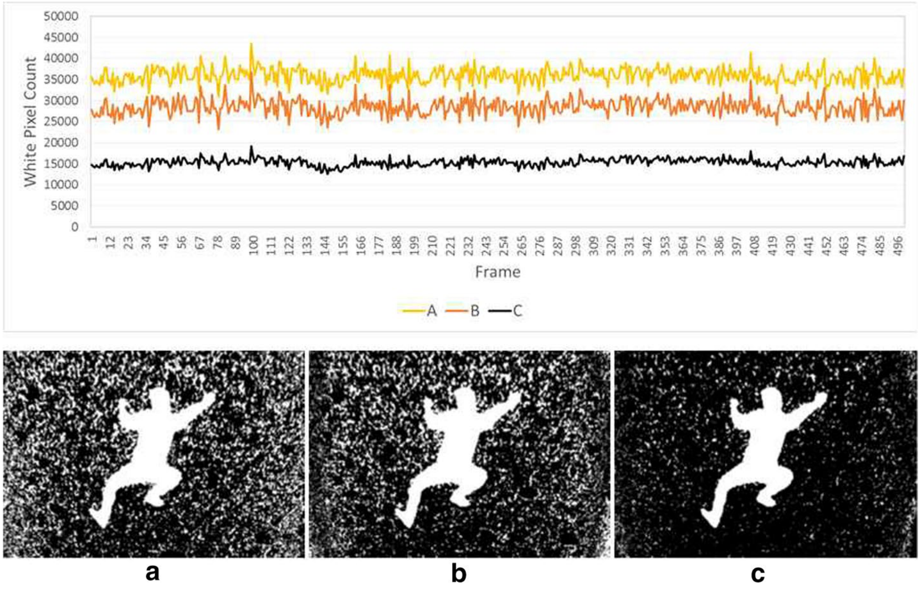


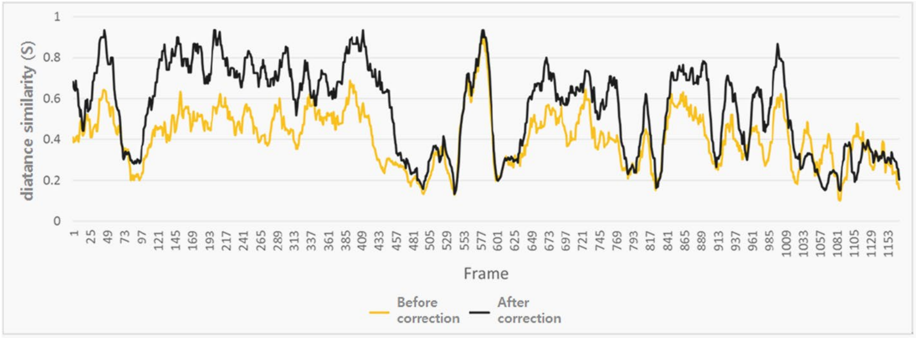
Fig. 13 A white pixel count comparison between body area detection methods. *Top* graph of white pixel count comparison; **a** naïve method, **b** Raine Kajastila's method, **c** proposed method. *Bottom* visual appearance comparison between binarized images; **a** naïve method, **b** Raine Kajastila's method, **c** proposed method

whereas the distance similarity converged to 1 if the variance was small. As shown in Fig. 14, we confirmed that the variation in anatomical location decreased due to skeletal frame normalization. The variation in joint location was the Euclidian distance of change in joint location between skeletal frames.

Figure 15 shows the results of parsing a climber's body area into its parts using the method described in "Parsing a climber's body area into body parts".

Demonstrating motion recognition event

Table 1 shows the results of a motion recognition event. In the event log column, "Frame index" is the frame number of the video used to detect a motion recognition event; "Game object ID" is the identification number of the recognized object; "Body event"



$$S = 1 / (1+d)$$

where **S** is the distance similarity, and **d** is the Euclidian distance of the joint location change between the skeleton fames

Fig. 14 Skeletal frame normalization for the *left* hand

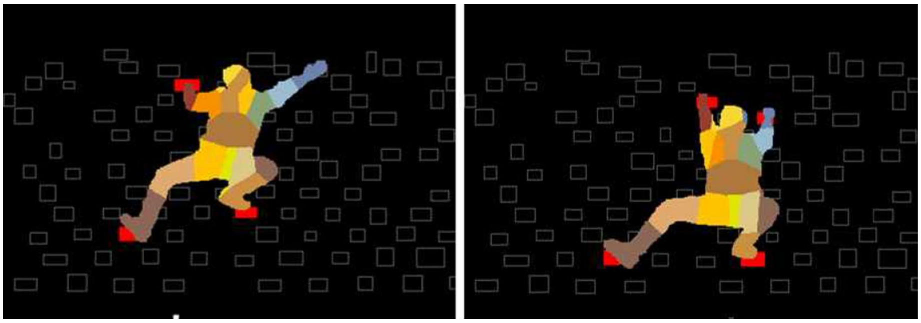


Fig. 15 Results of parsing a climber's body area into its parts

is the body part recognition event; and "Touch event" is the tactile event initiated by a hand or foot.

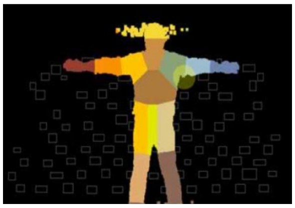
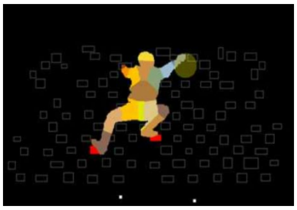
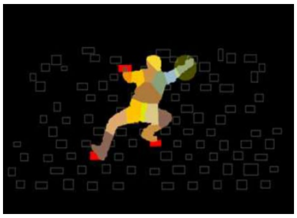
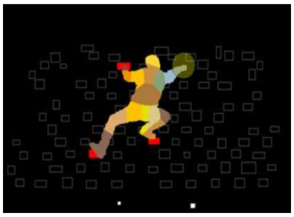
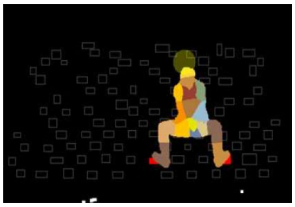
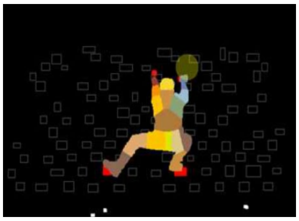
Scene 1 is shows a climber in a T posture to create skeletal system information before starting the climbing game. Although some of the climber's body area overlapped with the recognition object, the depth difference between the body area and recognition object was bigger than the event occurrence condition; therefore, a motion recognition event did not occur. Scene 2 depicts a climber stretching his or her right arm and hand, which overlap with Game object 1. Scene 3 shows multiple motion recognition event occurrences. A motion recognition event occurred for the right hand, and then a separate event occurred for the right elbow as the climber stretched his or her right arm to the right of Game object 1. Scene 4 illustrates a touch event, since the right hand area did not overlap with any climbing hold areas. Scenes 5 and 6 are situations that caused motion recognition events for the head and right hand, respectively.

Conclusion

In this paper, we propose a climbing motion recognition method using anatomical information derived from a climber's body area and skeletal system information. The climber's body area can be found using the depth difference between the background and foreground images in an indoor climbing environment. The skeletal system information is updated based on skeletal frame normalization and hand-or-foot joint correction, instead of using the original information provided by the Kinect SDK. The anatomical information is obtained by parsing the body area into its parts using the climber's body area and skeletal system information. We show that this anatomical information can be used for motion recognition events caused by human interactions with the game objects, climbing wall, and climbing holds.

Screen climbing games utilizing climbers' body parts can be implemented using these events instead of more general data points from a climber's body area. Doing so can make for a more interactive, realistic gaming environment that enables game designers to create a wider variety of experiences. For example, the Spark game described in the "Introduction" ends if any part of a body area makes contact with the electric line displayed on the artificial climbing wall. Using our technique, the game designer could vary the amount of damage depending on which part of the body makes contact with the electric line. Further data can make use of heart rate sensors [11], electromyography sensors, and additional device such as helmets. Moreover, Internet of Things (IoT)

Table 1 Results of motion recognition events

Scene no.	Motion recognition event scene	Event log
1		
2		Frame index: 415 Game object ID: 1 Body event: Handright Depth difference 313 mm
3		Frame index: 430 Game object ID: 1 Body event: Elbowright Depth difference: 331 mm Frame index: 430 Game object ID: 1 Body event: Handright Depth difference: 233 mm
4		Frame index: 462 Game object ID: 1 Touch event: Handright Touch object: Wall Depth difference: 33 mm
5		Frame index: 1135 Game object ID: 1 Body event: Head Depth difference: 737 mm
6		Frame index: 1261 Game object ID: 1 Body event: Handright Depth difference: 91 mm

technology can tap into these sensors to communicate with other people and computer systems [12]. Since the Kinect v2 can identify two or more people and provide corresponding skeletal system information, screen climbing games using interactions between multiple people could be implemented in a similar way to the games described by [6] and [13]. The variety of game events described in this paper should provide more entertainment and a more immersive experience for gamers.

Abbreviations

HCI: human–computer interface; ID: identification; IoT: internet of Things; SDK: Software Development Kit.

Authors' contributions

JK the 1st author, suggested main idea and wrote the draft version. DC the 2nd author, refined the main idea and edited the content of this manuscript. IK the corresponding author, the advisor of the 1st and 2nd authors. All authors read and approved the final manuscript.

Author details

[1] Department of Media, Soongsil University, 369 Sangdo-ro, Dongjak-gu, Seoul, South Korea. [2] Department of ICMC Convergence Technology, Soongsil University, 369 Sangdo-ro, Dongjak-gu, Seoul, South Korea.

Acknowledgements

This work was supported by the BK21 plus Program through NRF grant funded by the Ministry of Education (No. 31Z20150313339).

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

We did not use any data publicly opened. The data is based on the image processing results of the depth map images provided by the Kinect SDK. The depth map images are obtained from the video files recorded by the customized program in the situation of indoor climbing in a specific location. Therefore, the obtained data cannot be shared.

Declarations

This manuscript is submitted to "the special issue of Human-centric Computing and Information Sciences (HCIS)—Springer (SCOPUS)" due to the recommendation from WITC2017.

The title of the recommended paper is "Body-area and skeleton matching for climber motion recognition" by Jungsoo Kim, Daniel Chung, Ilju Ko.

We discussed the content of the manuscript sufficiently and agreed to include the content in the manuscript. Therefore, we declare that all of us agreed to submit this paper and there is no issue to conflict.

Funding

This work was supported by the BK21 plus Program through NRF grant funded by the Ministry of Education (No.31Z20150313339).

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 26 March 2017 Accepted: 29 June 2017
Published online: 04 September 2017

References

1. Ha H, Seo H (2014) Strategy for Gangwon-do winter sports IT convergence service. Korean Manag Sci Rev 31(4):107–116
2. Park K, Lim S (2013) An indoor golf simulator for continuous golf games. Int J Smart Home 7(3):75–84
3. Kim DG, Jin CY, Shin SY (2014) A suggestion of baseball simulation game using high speed camera sensor. J Korea Inst Inf Commun Eng 18(3):535–540
4. Kajastila R, Hämäläinen P (2015) Motion games in real sports environments. Interactions 22(2):44–47
5. Kajastila R, Hämäläinen P (2014) Augmented climbing: interacting with projected graphics on a climbing wall. In: Proceedings of the extended abstracts CHI. ACM, 2014, pp 1279–1284
6. Kajastila R, Holsti L, Hämäläinen P (2016) The augmented climbing wall: high-exertion proximity interaction on a wall-sized interactive surface. In: Proceedings of the 2016 CHI conference on human factors in computing systems. ACM, 2016
7. Kim JS, Chung D, Sung BK, Chon S, Ko IJ (2016) Ancient cave exploration: a screen climbing game for children. J Korea Game Soc 16(3):117–126
8. Chung D, Kim JS, Ko IJ, Sung BK, Park JH (2016) Sensing of locations of climbers' hands and feet during screen-climbing games. Int J Smart Device Appl. 4(2):35–42
9. Piccardi M (2004) Background subtraction techniques: a review. In: 2004 IEEE international conference on systems, man and cybernetics, vol 4, pp 3099–3104
10. Lee GC, Yoo J (2013) Real-time virtual-view image synthesis algorithm using kinect camera. J Korean Inst Commun Inf Sci 38(5):409–419
11. James AP (2015) Heart rate monitoring using human speech spectral features. Hum-centric Comput Inf Sci 5:33
12. Maity S, Park JH (2016) Powering IoT devices: a novel design and analysis technique. J Converg 7(2):1–18
13. Leftheriotis I, Chorianopoulos K, Jaccheri L (2016) Design and implement chord sand personal windows for multi-user collaboration on a large multitouch vertical display. Hum-centric Comput Inf Sci 6:14