Enhanced Computer Vision with Microsoft Kinect Sensor: A Review

Jungong Han, *Member, IEEE,* Ling Shao, *Senior Member, IEEE,* Dong Xu, *Member, IEEE,* and Jamie Shotton, *Member, IEEE*

***Abstract*—With the invention of the low-cost Microsoft Kinect sensor, high-resolution depth and visual (RGB) sensing has become available for widespread use. The complementary nature of the depth and visual information provided by the Kinect sensor opens up new opportunities to solve fundamental problems in computer vision. This paper presents a comprehensive review of recent Kinect-based computer vision algorithms and applications. The reviewed approaches are classified according to the type of vision problems that can be addressed or enhanced by means of the Kinect sensor. The covered topics include preprocessing, object tracking and recognition, human activity analysis, hand gesture analysis, and indoor 3-D mapping. For each category of methods, we outline their main algorithmic contributions and summarize their advantages/differences compared to their RGB counterparts. Finally, we give an overview of the challenges in this field and future research trends. This paper is expected to serve as a tutorial and source of references for Kinect-based computer vision researchers.**

***Index Terms*—Computer vision, depth image, information fusion, Kinect sensor.**

I. Introduction

KINECT is an RGB-D sensor providing synchronized color and depth images. It was initially used as an input device by Microsoft for the Xbox game console [1]. With a 3-D human motion capturing algorithm, it enables interactions between users and a game without the need to touch a controller. Recently, the computer vision society discovered that the depth sensing technology of Kinect could be extended far beyond gaming and at a much lower cost than traditional 3-D cameras (such as stereo cameras [2] and time-of-flight (TOF) cameras [3]). Additionally, the complementary nature of the depth and visual (RGB) information provided by Kinect bootstraps potential new solutions for classical problems in computer vision. In just two years after Kinect was released, a large number of scientific papers as well as technical demonstrations have already appeared in diverse vision conferences/journals.

In this paper, we review the recent developments of Kinect technologies from the perspective of computer vision. The criteria for topic selection are that the new algorithms are far beyond the algorithmic modules provided by Kinect development tools, and meanwhile, these topics are relatively more popular with a substantial number of publications. Fig. 1 illustrates a tree-structured taxonomy that our review follows, indicating the type of vision problems that can be addressed or enhanced by means of the Kinect sensor. More specifically, the reviewed topics include object tracking and recognition, human activity analysis, hand gesture recognition, and indoor 3-D mapping. The broad diversity of topics clearly shows the potential impact of Kinect in the computer vision field. We do not contemplate details of particular algorithms or results of comparative experiments but summarize main paths that most approaches follow and point out their contributions.

Until now, we have only found one other survey-like paper to introduce Kinect-related research [4]. The objective of that paper is to unravel the intelligent technologies encoded in Kinect, such as sensor calibration, human skeletal tracking and facial-expression tracking. It also demonstrates a prototype system that employs multiple Kinects in an immersive teleconferencing application. The major difference between our paper and [4] is that [4] tries to answer what is inside Kinect, while our paper intends to give insights on how researchers exploit and improve computer vision algorithms using Kinect.

The rest of the paper is organized as follows. First, we discuss the mechanism of the Kinect sensor taking both hardware and software into account in Section II. The purpose is to answer what signals the Kinect can output, and what advantages the Kinect offers compared to conventional cameras in the context of several classical vision problems. In Section III, we introduce two preprocessing steps: Kinect recalibration and depth data filtering. From Section IV to Section VII, we give technical overviews for object tracking and recognition, human activity analysis, hand gesture recognition and indoor 3-D mapping, respectively. Section VIII summarizes the corresponding challenges of each topic, and reports the major trends in this exciting domain.

Manuscript received November 12, 2012; revised April 4, 2013; accepted May 13, 2013. Date of publication June 25, 2013; date of current version September 11, 2013. This work was supported by the Multiplatform Game Innovation Centre (MAGIC), Nanyang Technological University, Singapore. Recommended by Associate Editor D. Goldgof. *(Corresponding author: L. Shao)*

J. Han is with Civolution Technology, Eindhoven 5656AE, The Netherlands (e-mail: jungonghan77@gmail.com).

L. Shao is with the College of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, Jiangsu, China, and also with the Department of Electronic and Electrical Engineering, The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK. (e-mail: ling.shao@sheffield.ac.uk).

D. Xu is with the School of Computer Engineering, Nanyang Technological University, 639798, Singapore (e-mail: DongXu@ntu.edu.sg).

J. Shotton is with Microsoft Research, Cambridge CB1 2FB, U.K. (e-mail: jamiesho@microsoft.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2013.2265378

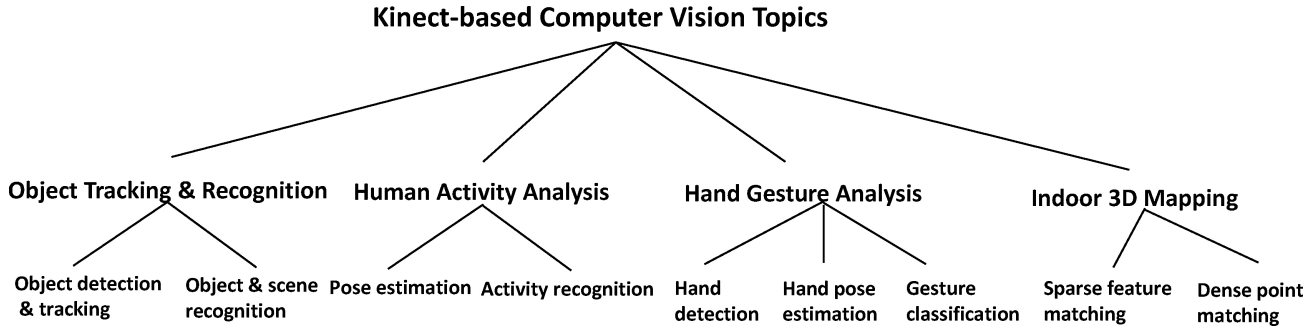


Fig. 1. Tree-structured taxonomy of this review.

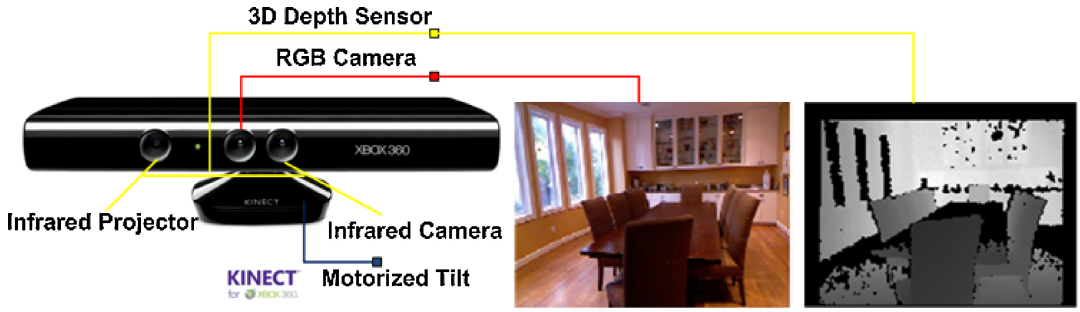


Fig. 2. Hardware configuration of Kinect, on which we point out the location of each sensor. Additionally, two image samples captured by the RGB camera and the depth camera are provided.

II. Kinect Mechanism

Kinect, in this paper, refers to both the advanced RGB/depth sensing hardware and the software-based technology that interprets the RGB/depth signals. The hardware contains a normal RGB camera, a depth sensor and a four-microphone array, which are able to provide depth signals, RGB images, and audio signals simultaneously. With respect to the software, several tools are available, allowing users to develop products for various applications. These tools provide facilities to synchronize image signals, capture human 3-D motion, identify human faces, and recognize human voice, and others. Here, recognizing human voice is achieved by a distant speech recognition technique, thanks to the recent progresses on the surround sound echo cancelation and the microphone array processing. More details about Kinect audio processing can be found in [5] and [6]. In this paper, we focus on techniques relevant to computer vision, and so leave out the discussion of the audio component.

A. Kinect Sensing Hardware

Fig. 2 shows the arrangement of a Kinect sensor, consisting of an infrared (IR) projector, an IR camera, and a color camera. The depth sensor comprises the IR projector and the IR camera. The IR projector casts an IR speckle dot pattern into the 3-D scene while the IR camera captures the reflected IR speckles. Kinect is therefore an instance of a structured light depth sensor. The geometric relation between the IR projector and the IR camera is obtained through an off-line calibration procedure. The IR projector projects a known light speckle pattern into the 3-D scene. The speckle is invisible to the color camera but can be viewed by the IR camera. Since each local pattern of projected dots is unique, matching between observed local dot patterns in the image with the calibrated projector dot patterns is feasible. The depth of a point can be deduced by the relative left-right translation of the dot pattern. This translation changes, dependent on the distance of the object to the camera-projector plane. Such a procedure is illustrated in Fig. 3. More details concerning the structured light 3-D imaging technology can be found in [7].

Each component of the Kinect hardware is described below.

1) *RGB Camera*: It delivers three basic color components of the video. The camera operates at 30 *Hz*, and can offer images at 640×480 pixels with 8-bit per channel. Kinect also has the option to produce higher resolution images, running at 10 *frames/s* at the resolution of 1280×1024 pixels.
2) *3-D Depth Sensor*: It consists of an IR laser projector and an IR camera. Together, the projector and the camera create a depth map, which provides the distance information between an object and the camera. The sensor has a practical ranging limit of 0.8m – 3.5m distance, and outputs video at a frame rate of 30 *frames/s* with the resolution of 640×480 pixels. The angular field of view is 57° horizontally and 43° vertically.
3) *The Motorized Tilt*: It is a pivot for sensor adjustment. The sensor can be tilted up to 27° either up or down.

B. Kinect Software Tools

Kinect software refers to the Kinect development library (tool) as well as the algorithmic components included in the

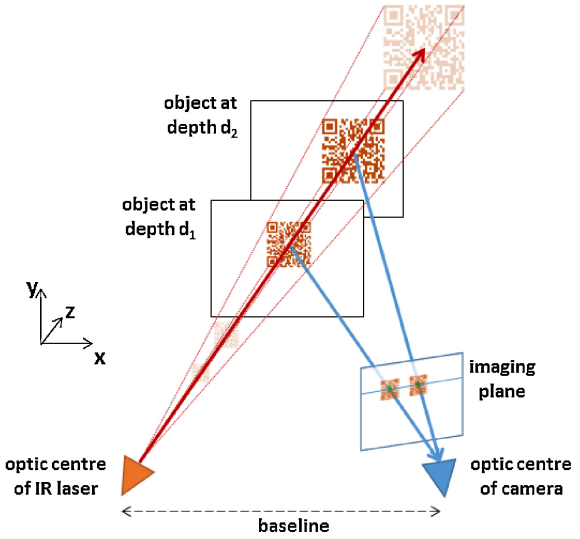


Fig. 3. Illustration of Kinect depth measurement.

library. Currently, there are several available tools including OpenNI [8], Microsoft Kinect SDK [9] and OpenKinect (LibFreeNect) [10]. OpenNI always works together with a Compliant middleware called NITE, and its highest version until March 2013 is 2.0. Microsoft Kinect SDK is released by Microsoft, and its current version is 1.7. OpenKinect is a free, open source library maintained by an open community of Kinect people. Since the majority of users are using the first two libraries, we provide details concerning OpenNI and Microsoft SDK. The Microsoft SDK (version 1.7) is only available for Windows whereas OpenNI (version 2.0) is a multiplatform and open-source tool. Table I gives a comparison between these two tools in terms of their algorithmic components.

In general, most corresponding components provided by these two libraries are functionally comparable. Here, we mention a few differences between them. For example, OpenNI's skeletal tracker requires a user to hold a predefined calibration pose until the tracker identifies enough joints. The calibration time varies greatly depending on environment conditions and processing power. On the contrary, Microsoft SDK does not need a specific pose initialization. However, it is more prone to false positives than OpenNI, especially when the initial pose of a human is too complicated. Moreover, the newest version of the Microsoft SDK is capable of tracking a user's upper body (ten joints) in case the lower body is not visible. This is particularly useful when analyzing human postures with a sitting position. Furthermore, OpenNI focuses on hand detection and hand-skeletal tracking whereas Microsoft SDK realizes simple gesture recognition, such as "grip" and "push" recognition.

It is worth highlighting that the new version of OpenNI (2.0) allows users to install Microsoft Kinect SDK on the same machine and run both packages using the Microsoft Kinect driver, which means that the OpenNI is now compatible with the Kinect driver. By doing so, switching between two drivers is not necessary anymore even when users want to benefit from both packages.

TABLE I
COMPARISONS OF THE OPENNI LIBRARY AND THE MICROSOFT SDK

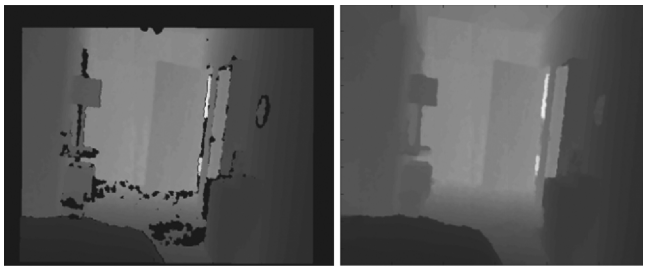
	OpenNI	Microsoft SDK
Camera calibration	√	√
Automatic body calibration	×	√
Standing skeleton	√ (15 joints)	√ (20 joints)
Seated skeleton	×	√
Body gesture recognition	√	√
Hand gesture analysis	√	√
Facial tracking	√	√
Scene analyzer	√	√
3-D scanning	√	√
Motor control	√	√

C. Kinect Performance Evaluation

There are a few papers that evaluate the performance of Kinect from either the hardware or the software perspective. These evaluations help us to understand both the advantages and limitations of the Kinect sensor and thus to better design our own system for a given application.

In [11], the authors experimentally investigate the depth measurement of Kinect in terms of its resolution and precision. Moreover, they make a quantitative comparison of the 3-D measurement capability for three different cameras, including a Kinect camera, a stereo camera, and a TOF camera. The experimental results reveal that Kinect is superior in accuracy to the TOF camera and close to a medium-resolution stereo camera. In another paper, Stoyanov *et al.* [12] compare the Kinect sensor with two other TOF 3-D ranging cameras. The ground truth data is produced by a laser range sensor with high accuracy, and the test is performed in an uncontrolled indoor environment. The experiments yield these conclusions. 1) the performance of the Kinect sensor is very close to that of the laser for short range environments (distance$<$ 3.5 meters); 2) the two TOF cameras have slightly worse performance in the short range test; and 3) no sensor achieves performance comparable to the laser sensor at the full distance range. This implicitly suggests that Kinect might be a better choice (over the TOF cameras) if the application only needs to deal with short range environments, since TOF cameras are usually more expensive than the Kinect sensor. Instead of comparing Kinect with other available depth cameras, Khoshelham *et al.* [13] provide an insight into the geometric quality of Kinect depth data based on analyzing the accuracy and resolution of the depth signal. Experimental results show that the random error of depth measurement increases when the distance between the scene and the sensor increases, ranging from a few millimeters at close range to about 4 cm at the maximum range of the sensor.

Another cluster of papers focus on studying the software capability of Kinect, especially the performance of skeletal tracking algorithm. It is indeed important when applying Kinect to human posture analysis in a context other than gaming, where the posture may be more arbitrary. In [14], the 3-D motion capturing capability offered by Kinect is tested in order to know if the Kinect sensor has comparable accuracy of existing marker-based motion acquiring systems. The result



(a) (b)

Fig. 4. Example for hole-filling based on the bilateral filter [25]. (a) Raw depth image. (b) Depth image after filtering.

turns out that Kinect is able to capture the relative 3-D coordinates of markers with minor errors (< 1cm) in case the sensor is positioned in an ideal range (1m to 3m) and with an effective field of view. In [15], authors examine the accuracy of joint localization and the robustness of pose estimation with respect to more realistic setups. In the experiment, six exercises are conducted, in which the subject is either seated or positioned next to a chair. The exercise is generally challenging for human pose recognition since the self-occlusion appears frequently and the capturing view angle is changed over time. The acquired 3-D location of each joint is then compared to the data generated by a marker-based motion capture system, which can be considered as ground truth data. According to the results, Kinect has a significant potential as a low-cost alternative for real-time motion capturing and body tracking in healthcare applications. The accuracy of the Kinect joint estimation is comparable to marker-based motion capture in a more controlled body pose (e.g., standing and exercising arms). However, in general poses, the typical error of Kinect skeletal tracking is about 10 cm. Moreover, the current Kinect algorithm frequently fails due to occlusions, nondistinguishing depth (limbs close to the body) or clutter (other objects in the scene).

III. Preprocessing

The data obtained with Kinect normally cannot be directly fed into the designed computer vision algorithms. Most of the algorithms take advantage of rich information (RGB and depth) attached to a point. In order to correctly combine the RGB image with the depth data, it is necessary to spatially align the RGB camera output and the depth camera output. In addition, the raw depth data are very noisy and many pixels in the image may have no depth due to multiple reflections, transparent objects or scattering in certain surfaces (such as human tissue and hair). Those inaccurate/missing depth data (holes) need to be recovered prior to being used. Therefore, many systems based on Kinect start with a preprocessing module, which conducts application-specific camera recalibration and/or depth data filtering.

A. Kinect Recalibration

In fact, Kinect has been calibrated during manufacturing. The camera parameters are stored in the device's memory, which can be used to fuse the RGB and depth information. This calibration information is adequate for casual usage, such as object tracking. However, it is not accurate enough for reconstructing a 3-D map, for which a highly precise cloud of 3-D points should be obtained. Moreover, the manufacturer's calibration does not correct the depth distortion, and is thus incapable of recovering the missing depth.

Zhang *et al.* [16] and Herrera *et al.* [17] develop a calibration board based technique, which is derived from Zhang's camera calibration technique used for the RGB camera [18]. In this method, 3-D coordinates of the feature points on the calibration card are obtained from the RGB camera's coordinate system. Feature-point matching between the RGB image and the depth image is able to spatially correlate those feature points between two different images. This spatial mapping helps feature points to get their true depth values based on the RGB camera's coordinate system. Meanwhile, the depth camera measures 3-D coordinates of those feature points in the IR camera's coordinate system. It assumes that the obtained depth values by the depth camera can be transformed to the true depth values by an affine model. As a result, the key is to estimate the parameters of the affine model, which can be done by minimizing the distances between the two point sets. This technique combined with a calibration card allows users to recalibrate the Kinect sensor in case the initial calibration is not accurate enough for certain applications. The weakness of this method is that it does not specifically pay attention to the depth distortion. Correcting the depth distortion may become unavoidable for most 3-D mapping scenarios.

There are a few publications that discuss solutions for Kinect depth distortion correction. Smisek *et al.* [11] discover that the Kinect device has shown radially symmetric distortions. In order to correct this distortion, a spatially varying offset to the calculated depth is applied. The offset at a given pixel position is calculated as the mean difference between measured depth and expected depth in metric coordinates. In [19], a disparity distortion correction method is proposed based on the observation that a more accurate calibration can be made by correcting the distortion directly in disparity units.

An interesting paper [20] deals with more practical issues, which investigates a possible influence of thermal and environmental conditions when calibrating Kinect. The experiment turns out that variations of the temperature and air draft have a notable influence on Kinect's images and range measurements. Based on the findings, temperature-related rules have been established in the paper, which reduce errors in the calibration and measurement process of the Kinect.

B. Depth Data Filtering

Another preprocessing step is depth data filtering, which can be used for depth image denoising or missing depth (hole) recovering. A naive approach considers the depth data as a monochromatic image and thus applies existing image filters on it, such as a Gaussian filter. This simple method works only for regions where the signal statistics is in favor of the underlying filter. A more sophisticated algorithm [21] investigates the specific characteristics of a depth map created by Kinect, and finds out that there are two types of occlusions/holes caused by different reasons. The algorithm automatically separates

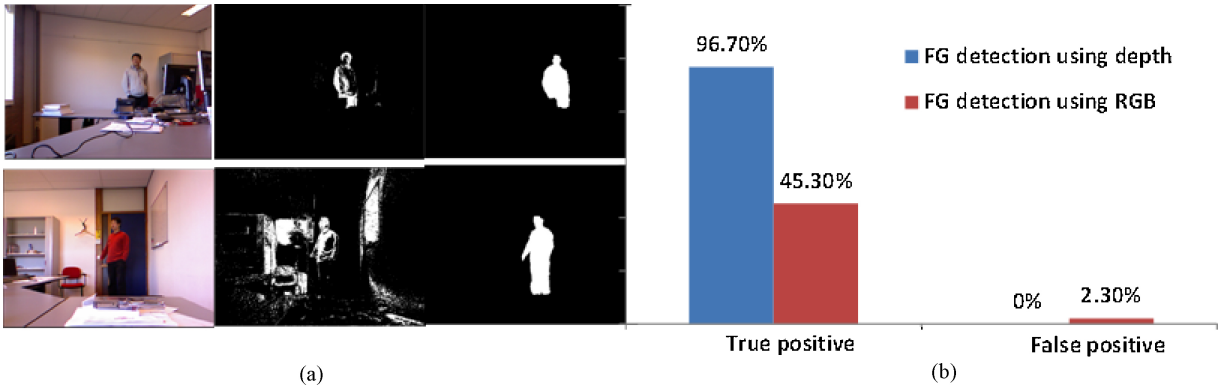


Fig. 5. Foreground (FG) detection results [36]. (a) From the left to the right, RGB images, FG mask using RGB data and FG mask using depth data, respectively. The example on the top shows that it is hard to distinguish clothing from the background, and the one at the bottom reports the results when the person suddenly turns on the lighting. (b) True positive and false positive of FG detection by using depth data and RGB data respectively. Here, the lighting is stable but the foreground and background are similar.

these occlusion cases, and develops different filling schemes accordingly. Although this sort of adaptive filtering scheme is much better than a simple filtering algorithm, it still fails to take advantage of all available information in this application: depth, color and the temporal relation between video frames.

In [22], the relation between the RGB image and the depth image is taken into account when filling the holes on the depth map. This approach first finds the spatial correspondence between two channels by means of matching object borders. Then, objects are located on the color image via segmentation, and the locations of the segmented objects will be transferred to the depth image. A missing depth value within a segment can be interpolated with the assumption that the depth signal is approximately smooth within each segment. Unfortunately, this approach may not generate a uniform depth field when the object surface is extremely colorful. Schmeing *et al.* [23] make use of the combination of depth and RGB information provided by video-plus-depth sensors to enhance corrupted edges on depth maps. In their method, the correct edge information is found on the RGB image via a superpixel segmentation algorithm. This information helps to compute a new representative depth map with robust edges, which is used to enhance the source depth map.

In [24] and [25], the missing depth values are obtained by iteratively applying a joint-bilateral filter to depth pixels (Fig. 4). A bilateral filter is, in principle, an edge-preserving and noise reducing smoothing filter. The depth value at each pixel in an image is replaced by a weighted average of depth values from nearby pixels. Here, the filter weights are determined by considering RGB data, depth information and a temporal consistency map. Alternatively, Qi *et al.* [26] treat the hole-filling as an inpainting problem, and adapt a non-local inpainting algorithm used for RGB images to fill the holes on the depth map. However, the adaptation is nontrivial, because the object boundary terminating the filling procedure cannot be easily found on the textureless depth map. To solve this problem, the algorithm seeks the object boundary on the color image provided by Kinect and estimates its corresponding position based on the calibration parameters.

IV. Object Tracking and Recognition

The first two sections of this paper have discussed the field of Kinect at a fairly non-technical level. They mainly address the questions "what is the Kinect sensor?" and "what can Kinect offer?" In this section, we begin our exploration of the detailed, technical principles. Throughout the rest of the paper, we will overview what Kinect can do and how people change and enhance Kinect-related techniques in order to address particular vision problems.

A. Object Detection and Tracking

Object detection and tracking are hot topics in RGB-based image and video analysis applications. A widely used approach is background subtraction, when the camera is fixed. In this approach, the background is assumed to be static over time, while the foreground can be extracted by subtracting the background model from the input image. However, in practice, detecting objects in images or videos using background subtraction techniques is not that straightforward due to the high variety of possible configurations of the scenario, such as changes of illumination conditions and subtle movements of the background. For such cases, the background is not static at the signal level. Many approaches try to model the variations and design new background models to cope with the variations. Unfortunately, handling multiple variations with one background model remains challenging in this field.

With the availability of the low-cost Kinect depth camera, researchers immediately noticed that the nature of the depth signal can help to establish a more stable background model, which is resistant to changes in illumination or lack of contrast. To investigate this, we feed depth images and normal RGB images into a Gaussian mixture model (GMM)-based background subtraction algorithm respectively. We test the reliability of these two different signals in two challenging situations (the total length of testing sequences is longer than 10 minutes). One is that the foreground and the background are quite similar (case 1), and the other is that the illumination in the room suddenly changed (case 2). Fig. 5 illustrates the results, where (a) shows image samples of case 1 (top) and case 2 (bottom), and (b) reports quantitative comparison of

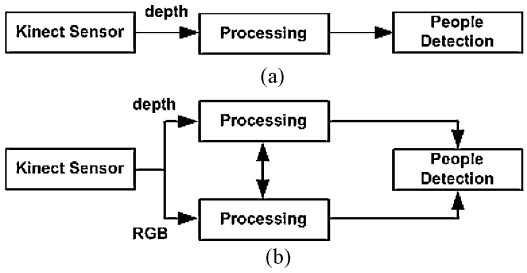


Fig. 6. Two schemes of using Kinect for people detection. Scheme *A*: Using depth images only. Scheme *B*: Combining depth images and RGB images.

case 2. In Fig. 5(a), the images (from the left to right) are the original image, the foreground mask obtained by using RGB images, and the foreground mask obtained by using depth images, respectively. Obviously, the quality of the segmented foreground object from the depth images is much better than that from the RGB images.

Most existing works make use of this unique property of depth images. However, there is still a trade-off between using depth images only and using both depth and RGB images. In Fig. 6, we show the basic ideas of these two schemes. Here, scheme *A* employs the depth images only while scheme *B* takes the advantage of complementary data emanating from the two vision sensors of Kinect.

The algorithm presented in [27] aims at detecting people based on depth information obtained by Kinect in indoor environments. A 2-D chamfer distance matching scans across the whole image and gives the possible regions that may contain people. Each region is further verified by using a 3-D head model, resulting in the final estimation. Afterwards, a region growing algorithm is applied to find the entire body of the people and thus the whole body contour is extracted. Rougier *et al.* [28] and Zhang *et al.* [29] explore the Kinect sensor for the application of detecting falls in the elderly. Here, the core technique is the person segmentation and localization from the depth images. Both systems adopt the similar background subtraction algorithm. A depth background image is obtained from a number of training background images. The mean value and standard deviation are computed for each pixel of the image, and used for calculating the distance of a given pixel to the background pixels. Eventually, the foreground image is cleaned with morphological filtering and the depth silhouette can be obtained by combining the depth image with the foreground silhouette. The work reported in [30] investigates whether existing human detectors on the RGB image can be directly extended to detect humans on depth images. To this end, histograms of oriented gradients (HOG) [31], pyramid HOG [32], local binary patterns (LBP) [33], Local ternary patterns (LTP) [34], and census transform histograms (CENTRIST) [35] are compared based on the same depth image dataset. The results reveal that the LTP-based detector outperforms other detectors, and is more suitable for human detection on the depth map.

Instead of using the depth information only, another group of papers exploit the complementary nature of (synchronized) color and depth images. In principle, the depth signal is more robust against changes in illumination or lack of contrast than the conventional RGB signal. However, the depth is less discriminative for representing an object due to its limited dimension. In [36], Han *et al.* present a human detection and tracking system for a smart environment application, which selectively feeds the complementary data emanating from the two vision sensors to different algorithmic modules. More specifically, it segments the human out of the scene on depth images as it is invariant to various environment changes. Once the object has been located in the image, visual features are extracted from the RGB image and are then used for tracking the object in successive frames. Spinello *et al.* introduce their RGB-D based people detection approach in [37] and [38]. In their system, a local depth-change detector employing HOD is formed, which is conceptually similar to HOG in RGB data. On top of that, a probabilistic model combining HOD and HOG detects the people from the RDB-D data. In another paper [39], a multihypothesis human tracking algorithm is presented, which describes the target appearance with three types of RGB-D features and feeds them to an online learning framework. More recently, they turn their research to investigate how to optimally combine RGB and depth images for the task of object detection [40]. The basic concept is the adaptive fusion that changes the weight of each modality in terms of the measurements of missing information and cross-cue data consistency.

Rather than fully relying on one single attribute of human (both HOG and HOD represent human's shape information), the work reported in [41] adopts three different types of attributes to sense human in the image, including biometrical attributes (e.g., height information), appearance attributes (e.g., clothing color) and motion attributes (e.g., posture). Though the aforementioned human attributes can be deduced with the aid of Kinect (e.g., the depth and the length of a person in the image may help to estimate his/her real height), the way of combing those features retains to be problematic in case that inaccurate detection happens. Choi *et al.* [42] apply this sort of ensemble detection to track people over time, assuming the initial location of people is available. The cues they used contain human upper-body shape, human face, human skin, as well as human motion. These multihypothesis detections are fused into a coherent framework built upon a sampling technique (Markov chain Monte Carlo particle filtering), thus enabling a tracking-by-detection formulation.

B. Object and Scene Recognition

Object recognition differs from object detection in the sense that it does not provide the location of the object but only predicts whether a whole image contains the object or not. Scene recognition is a straightforward extension of object recognition, aiming to densely label everything in a scene. Conventional algorithms on this topic are mainly based on RGB images, where the color, texture, motion, or the combination of them are used to represent the object. A set of features describing a given object is used to learn a classifier, and in turn, the trained classifier is responsible for recognizing the (distorted) object by matching the extracted features. The general conclusion is that the more information

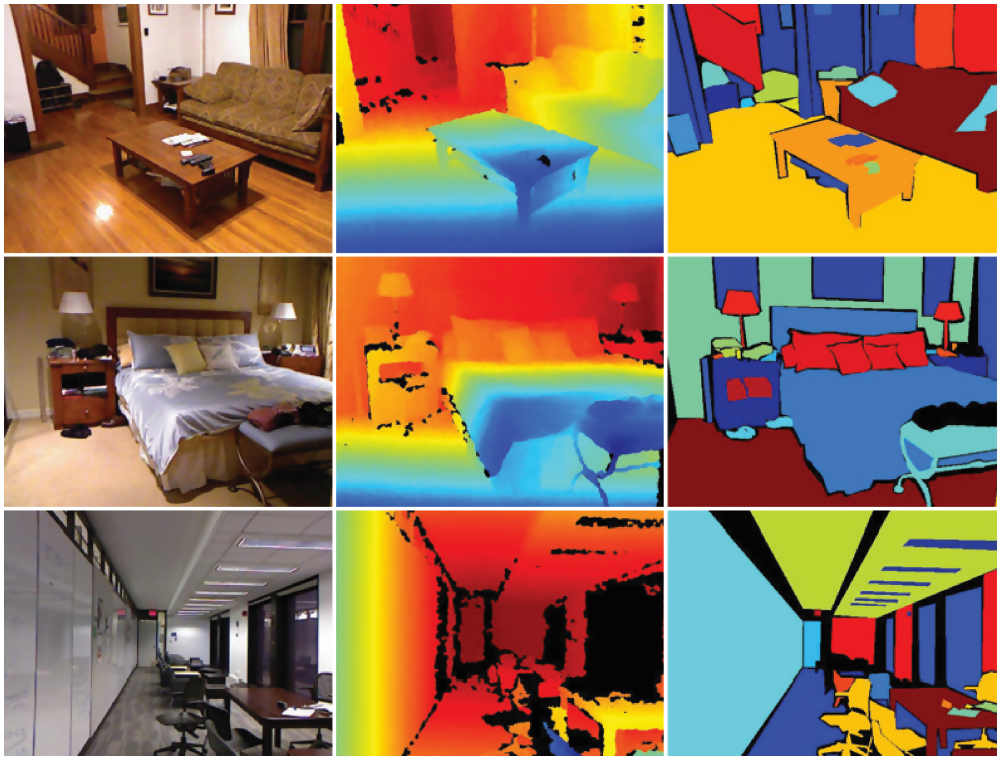


Fig. 7. Image samples from the dataset in [44]. From the left to the right: RGB images, depth images and annotated object regions with labels.

the algorithm uses the better discriminative capability of the algorithm. In other words, increasing the feature dimension helps the object recognition. The availability of Kinect offers a cheaper way to combine RGB features with depth features for object recognition.

To boost research in this area, authors of [43] and [44] dedicate to establish large-scale RGB-D object datasets. The datasets together with their annotation software have been made publicly available to the research community. The former dataset [43] contains multiple views of a set of objects and the objects are organized into a hierarchical category structure, while the later one [44] offers registered RGBD images, detailed object labels, and annotated physical relations between objects in the scene (Fig. 7), thus facilitating the application of indoor scene analysis. In addition to describing the properties of their dataset, the work presented in [43] also extends a conventional RGB image-based object recognition to the RGB-D images, in which the same feature extraction is performed to both RGB and depth images. The best performance is attained by combining image and depth features.

In [45], the authors integrate the RGB and depth information of an object and recognize it at both the category and the instance levels. Category level recognition tells which type the object belongs to while instance level recognition identifies whether an object is physically the same object as one that has previously been seen. The key of such recognition systems is how to efficiently combine heterogeneous features, such as RGB images and depth images. Here, local distance learning is adopted, which can model complex decision boundaries by combining elementary distances. To make the algorithm invariant to the change of the camera view, its model learns a distance function jointly for all views of a particular object. Instead of using a supervised learning [45], Bo *et al.* [46] alternatively investigate the possibility of using an unsupervised learning method for RGB-D object recognition. The system based on their previous work [47] adapts a hierarchical matching pursuit (HMP) algorithm to be suitable for color and depth images captured by Kinect. HMP automatically learns dictionaries over image and depth patches, and represents observations as sparse combinations of codewords. The success of the algorithm proves that current recognition systems can be significantly improved without resorting to carefully designed, handcrafted features. In their recent work [48], a contextual modeling is put on the top of object recognition in order to label an indoor scene. The object recognition part of this paper may not be that interesting since the similar ideas are presented in their previous papers. For contextual modeling, it combines segmentation tree and superpixel Markov random fields (MRFs), in which the segmentation tree technique is slightly modified for RGB-D data.

Unlike the above works that improve the decision making of the systems, the following papers aim to extract better features from RGB-D data. In [49], a new feature, namely histogram of oriented normal vectors (HONV), is designed to capture local 3-D geometric characteristics on depth images. The motivation of inventing this feature is that an object can be recognizable by looking at its 3-D surface without texture. To characterize the object surface, the local distribution of its tangent plane orientations is calculated, which can also be formed as a concatenation of the local histograms of azimuthal angles and zenith angles. Comparison results show that this new descriptor significantly outperforms HOG (HOG

on RGB images) and HOGD (HOG on depth images) in both object detection and classification. Blum *et al.* [50] present an algorithm that automatically learns feature responses from the image. In order to cope with the high dimensionality of the RGB-D signal, the algorithm only describes interest points. This new feature descriptor encodes color and depth into a concise representation. Alternatively, the work in Bo *et al.* [51] introduces five depth kernel descriptors that capture different recognition cues including size, shape and edges. The usage of kernel descriptors turns pixel attributes into patch-level features, enabling to generate rich features from a variety of recognition cues. Apart from adapting RGB image gradients and local binary patterns to depth images, the size descriptor, the kernel principal component analysis (PCA) descriptor and the spin kernel descriptor are also designed for depth images. The combination of these feature descriptors significantly improves the accuracy of object recognition.

C. Summary

The algorithms introduced in this section can be generally divided into two categories: object detection and object recognition. The former contains topics like object segmentation and human detection, while the latter covers topics such as object recognition and scene labeling. For the former one, the reported results from various methods as well as our own experiments clearly show that the algorithms based on analyzing the depth signal significantly increase the robustness of the system. This is mainly due to that the depth signal is inherently resistant to lighting condition changes and environment clutters, which are difficult problems for conventional algorithms. We believe that the algorithms specifically designed for depth cameras (e.g., Kinect) will become a standard processing module in future indoor video surveillance applications.

Regarding object recognition, the classification accuracy can be also further improved when using RGB-D information. Such results are anticipated because adding depth information into the object descriptor theoretically enhances the discriminative power of the descriptor. However, it is also clear that the benefit comes with the cost as it increases the load of data processing. And, the achieved gain seems to be limited according to the experimental results provided in [44] and [48]. Therefore, it may not be practical yet to use proposed algorithms in a real-time oriented application.

V. Human Activity Analysis

Analyzing human activities from video is an area with increasingly important consequences from security/surveillance to entertainment. In recent years, this topic has caught the attention of researchers from academia, industry, consumer agencies and security agencies. As a result, in the last 10 years a huge number of papers were published, among which the majority of papers uses RGB video as an input. In this paper, we only deal with Kinect-related techniques, and our intention is to give insights to the recent developments of human activity analysis based on the Kinect sensor.

The research devoted to this particular field can be grouped into two broad categories. In the first category, researchers investigate skeletal tracking. We will refer to this as pose estimation, because its goal is to achieve either a faster or a more accurate skeletal joints approximation. The research in the second category is called activity recognition, since it steps forward to recognize the semantic activity of a human in the context of various applications. Briefly speaking, pose estimation, in this scenario, provides the position of skeletal joints in the 3-D space, while the activity recognition tells what the human is doing through analyzing temporal patterns in these joint positions.

A. Pose Estimation

Besides reliably providing depth images in a low-cost way, another innovation behind Kinect is an advanced skeletal tracker, which opens up new opportunities to address human activity analysis problems. A core part of the algorithm is described in [52]. This paper introduces a per-pixel body part classification, followed by estimating hypotheses of body joint positions by finding a local centroids of the body part probability mass using mean shift mode detection. This algorithm runs per frame, and uses no temporal information, allowing the system as a whole to be robust to loss of track. The main contributions of [52] are the use of body part recognition as an intermediate representation for human pose estimation, and the demonstration that the classifier can be made invariant to human body shape and pose by training from a large corpus of synthetic data using a parallelized multicore implementation [53]. The body part recognition algorithm considerably improves the accuracy compared to related work, and even more importantly, it runs at least 10 times faster. Finally, using an unpublished, proprietary algorithm, a skeleton model is fitted to the hypothesized joint positions. This algorithm exploits temporal and kinematic constraints to result in a smooth output skeleton that can handle occlusions.

The representatives of pose estimation algorithms are [54]–[57]. The first two papers present enhanced algorithms from the inventors of the Kinect skeletal-tracking algorithm. In [54], an offset vote regression approach is proposed, in which pixels vote for the positions of the different body joints, instead of predicting their own body part labels. The new algorithm is capable of estimating the locations of joints whose surrounding body parts are not visible in the image due to occlusion of field or limited view of the sensor. In [55], authors further extend the original machine learning approach by learning to predict direct correspondences between image pixels and a 3-D mesh model. An energy minimization is then used to efficiently optimize the pose of the 3-D mesh model without requiring the standard iterated closest point (ICP) iteration between the discrete correspondence optimization and the continuous pose optimization. Ye *et al.* [56] and Shen *et al.* [57] aim to achieve a highly accurate and robust pose estimation from a single depth image in the case that the algorithm complexity can be overlooked. Fig. 8 illustrates their processing pipeline, consisting of pose estimation and pose refinement/correction steps. More specifically, their pose estimation algorithm finds the best matching pose from a precaptured motion database, given an input point cloud obtained from the depth image. The initial estimation is then refined by directly fitting the body

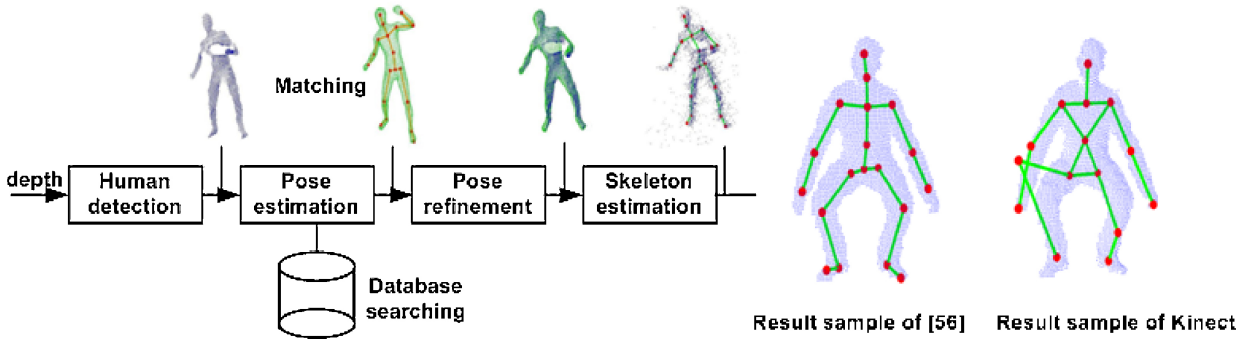


Fig. 8. Basic processing pipeline of algorithms reported in [56] and [57], where the image examples are from [56].

configuration with the observation (the input depth image). In fact, the initial estimated pose usually yields poor results due to personal shape variations and small pose differences, so that the pose refinement step is vital. In [56], authors treat pose refinement as a non-rigid registration problem, and the coherent drift point (CPD) algorithm is exploited to establish point correspondences. The work in [57] strengthens the pose refinement/correction step by jointly optimizing temporal motion consistency and the systematic bias. Generally, these systematic biases are associated with complex data manifolds, which can be learned by leveraging the exemplar information within a specific human action domain. The comparison with Kinect's embedded skeletal-tracking shows its superiority when dealing with human actions that have certain regularity, e.g. golf or tennis.

B. Activity Recognition

Another cluster of papers aim to recognize semantic human activities in the context of specific applications, given the skeletal joints at each frame. The systems presented in [58]–[60] attempt to find compact and discriminative feature descriptors to represent the configuration of a set of human joints. In [58] and [59], the spatial position differences between detected joints as well as the temporal differences between corresponding joints are used to be the visual feature for activity recognition. This simple but efficient descriptor combines action information including static posture, motion, and offset. Alternatively, a compact representation of human postures based on 3-D joint locations (HOJ3-D) is examined in [60]. In this representation, the 3-D space is partitioned into n bins using a spherical coordinate system. Human joints casted into certain bins are accumulated, thus constructing a histogram. The collection of HOJ3-D vectors are first reprojected using linear discriminant analysis (LDA) and then clustered into a k posture vocabulary. According to the reported results, this representation makes the recognition system invariant to the camera view changes. Considering that the skeletal joints are not always detected accurately, Liu and Shao [109] learn discriminative representations directly from the raw RGB-D video data for activity recognition. The feature learning is optimized using a restricted graph-based genetic programming (RGGP) approach, in which a group of primitive 3-D operators are first randomly assembled as graph-based combinations and then evolved by evaluating on a set of RGB-D video samples. Finally the best-performed combination is selected as the (near-)optimal representation for activity recognition. Their results show that the RGGP feature learning approach outperforms state-of-the-art hand-crafted and machine-learned descriptors.

Several publications [61]–[63] focus on the inference part of the recognition system, assuming the locations of skeletal joints are available. To encode the sequential changes of the features, they all utilize the hidden Markov model (HMM), a popular graphical model in RGB camera-based human activity recognition. The major difference between these approaches is that the work [61] adopts a single layer structure while the methods in [62], [63] exploit a two-layer hierarchical structure. The benefit of using a two-layer structure is that a human activity can be naturally considered as a combination of a set of sub-activities over time. Rayes *et al.* [64] represent a human model based on a feature vector formed by 15 joints on a 3-D human skeleton model. Furthermore, they use dynamic time warping (DTW) with automatic feature weighing on each joint to achieve real-time action recognition. Wang *et al.* [65] contribute to both feature generation and activity inference. In this paper, a new feature called local occupancy pattern is proposed to represent the "depth appearance," which is designed to capture the relations between the human body parts and the environmental objects in the interaction. Moreover, it defines an actionlet as a particular conjunction of features for a subset of the joints, indicating a structure of the features. Based on it, one human action can be interpreted as an actionlet ensemble that is a linear combination of the actionlets. This actionlet ensemble coupled with a learning technique yields a robust way to identify human activities. The work in [66] presents a *low-latency* online action recognition system, which runs in real-time and processes a video stream without given temporal segmentation. This kind of systems is highly demanded for the applications such as interactive gaming and touch-based user interfaces. The major contribution of this paper is the concept of "action points" that serve as natural temporal anchors of simple human actions. An action point is defined as a specific time instance at which the presence of the action is clear and it can be easily identified for all instances of the action. Detecting an action point only incorporates information from the past, so that the presence of an action can be immediately confirmed by detecting an action point.

To facilitate the research in this field, Ni *et al.* [67] establish a human activity recognition benchmark database and make it publicly available. In this dataset, 12 different human daily activities within a home environment are recorded by Kinect. Another popular dataset [68] is the MSRC-12 Kinect gesture dataset, consisting of 12 gestures performed by 30 people. The dataset contains tracks of 20 joints estimated by the Kinect pose estimation algorithm and the associated gesture labels. ACT42 [69] is a recently-published indoor human action recognition dataset. It provides synchronized data from 4 views and 2 sources, aiming to facilitate the research of action analysis across multiple views and multiple sources.

Several other papers on human activity recognition investigate how to apply skeletal tracking to different scenarios. They usually define certain heuristic rules in the context of a given application. Those rules, in turn, help to detect predefined activities. For example, the algorithm described in [70] enables humans to interact with the music, generating musical notes based on the motion-related features such as the velocity or acceleration of body parts. Kinect skeletal tracking plays a role in bridging the gap between human motion detection and computer-based music generation. In another application [71], Kinect is used as a motion acquiring tool to help the identification and examination of the dancing gestures. In [72], a Kinect-based construction worker tracking and action recognition algorithm is presented, which acts as a part of an automated performance assessment system in cluttered and dynamic indoor construction environments.

C. Summary

Having been a well-explored topic in computer vision for many years, human activity analysis has recently regained its popularity on RGB-D data provided by Kinect. Compared with conventional methods, which extract either holistic or local features from raw video sequences, the new algorithms are supplied with the extra skeletal joints in addition to the RGB and depth images. An accurate set of coordinated body joints can potentially yield an informative representation of the human body encoding the locations of different body parts and their relative positions. This kind of representation defines an activity as a sequence of articulated poses and its availability is accredited to Kinect, because body parts are very difficult to be obtained from normal RGB video data. Using such a representation is able to significantly simplify the learning of the activity recognition, since the relevant high-level information has already been extracted. Moreover, 3-D skeleton poses are viewpoint and appearance invariant, enabling potentially more robust arbitrary-view dynamic pose estimation and action recognition.

Naturally, most Kinect-based pose estimation and activity recognition approaches attempt to take advantage of the skeletal joints and design models on top of them. However, current methods are still in their infant stage and have been proposed through intuition and heuristics. A predictable trend is to automatically learn optimal representations of the body joints via advanced machine learning techniques. On the other hand, detected body joints tend to be noisy and unreliable, especially for cluttered and occluded scenes. Therefore, representations that are robust to inaccurate and missing joints will be highly desirable. In addition, as the original RGB-D images contain the most information, effectively fusing features from the raw RGB-D video and the high-level joint representation would be another research direction of human activity analysis.

VI. Hand Gesture Analysis

Hand gesture analysis based on Kinect is an emerging topic, enabling users to have better interactions with machines. In contrast to human activity analysis relying on whole body parts, hand gesture analysis is in some cases more efficient, as only data around the hand area need to be processed. Additionally, hand detection, which is a key component, can be enhanced by considering the skin-tone color information of the hand. However, the gain of using the color image for human body detection and skeletal tracking may not be significant because of the color variation of human clothing in different body parts.[1] Overall, recognizing hand gestures is a challenging task and requires solving several sub-problems including automatic hand detection and tracking, 3-D hand pose estimation, and gesture classification.

A. Hand Detection

Similar to the object detection introduced before, hand detection can be carried out either on depth images only or by fusing the RGB and depth information. The former aims to obtain a fast algorithm, whereas the latter targets an accurate system. Several proposals detect hands on depth images. For instance, authors in [73] treat hand segmentation as a depth clustering problem, where the pixels are grouped at different depth levels. Here, the critical part is to determine a threshold, indicating at which depth level the hand is located. In this paper, the threshold is estimated by analyzing the human posture dimension, assuming that the human posture is known. The algorithm seems to be rather heuristic, and is thus restricted to specific applications. Lee *et al.* [74] not only detect hands but also locate the fingertips on depth images. Hand detection is accomplished by the k-means clustering algorithm with a predefined threshold. The detection of fingertips requires convex hull analysis for the hand contour.

Two more elegant algorithms are reported in [75] and [76]. In [75], Liang *et al.* detect hands on the depth images through a clustering algorithm followed by morphological constraints. Afterwards, a distance transform to the segmented hand contour is performed to estimate the palm and its center. Instead of starting by hand detection, Hackenberg *et al.* [76] directly seeks pipe- or tip-like objects on a depth image. Those objects are initially selected to be the candidates of palm and fingers. They, as a whole, are further verified based on the spatial configuration checking since the palm and fingers together form a unique shape. In [77], Caputo *et al.* begin with a human-body skeleton generated by Kinect. On this skeleton map, the positions of both hands can be easily extracted with the aid of the hypothesis. Given the 3-D position of a hand, the

[1]This may also partially explain why most publications introduced in the previous section do not involve the RGB images in the analysis loop.

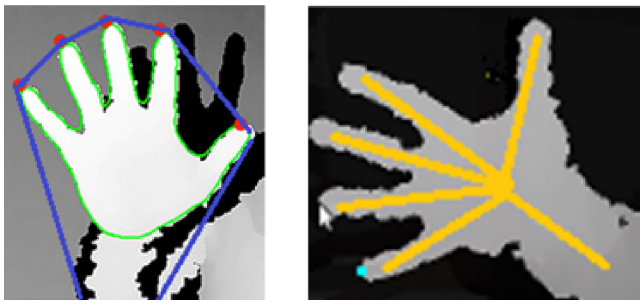


Fig. 9. Illustrations for the shape-based pose estimation and the hand skeleton-based pose estimation. Left: Hand contour (green line), convex hull (blue line) and positions of fingers (red dots). Right: Hand skeleton.

algorithm finds the corresponding hand size in a look-up table, which stores the sizes of a standard human hand at different depth levels. With the size of the hand, their method is able to roughly locate the hand region on the depth image.

Some other works further enhance the accuracy of hand detection by integrating the color information into the framework. For instance, in [78] and [79], they treat the hand detection as a kind of pixel labeling problem, where each pixel is labeled as either a hand pixel or a non-hand pixel. The skin color detector on the RGB image and the clustering on the depth image serve as two conditions to qualify a hand pixel. The hand region is the intersection of a skin region and a uniform region (cluster on the depth image) that is closer to the camera.

B. Hand Pose Estimation

Most publications generally fall into three categories: shape-based, 3-D model-based and hand skeleton-based. Shape-based approaches typically match the shape features of the observation to a predefined, finite set of hand shape configurations. In [80], the hand contour is extracted, implying the shape and the boundary of the hand. Both the hand model and the testing hand depth images are normalized in such a way that the feature is invariant to the pose translation. The depth similarity measure between two images is defined as the inverse of their pixel-wise Euclidean distance. Ren *et al.* [81] propose to use a Finger-Earth Mover's distance to measure the dissimilarities between different hand shapes. Yao *et al.* [82] introduces an efficient feature descriptor that combines shape, local curvature and relative position information.

3-D model-based approaches formulate the pose estimation as an optimization problem that minimizes the discrepancy between 3-D hand hypotheses and the actual observations. In [78], the authors adopt a 3-D hand model consisting of a set of assembled geometric primitives. Each hand pose is presented as a vector of 27 parameters. They employ rendering techniques to generate comparable skin and depth maps for a given hand pose hypothesis. Then, a stochastic optimization method, called particle swarm optimization (PSO), helps to estimate the 27 model parameters that minimize the distance between the rendered hypotheses and the inputs. The selection of PSO is due to its proven capability of solving high dimensional optimization problem. The work in [79] further extends their system to estimate the poses of two interacting hands. Additionally, they accelerate their system by using GPU implementation, achieving a near real-time operation.

Skeleton-based approaches deduce the hand pose based on the configuration of the hand skeleton. The skeleton generation is the key to these methods. A pioneering work [83] is the random decision forests (RDF)-based hand skeleton tracking, which can be seen as a hand-specific version of Shotton's human-body skeleton tracking [52]. It performs per-pixel classification by means of the RDF technique and assigns each pixel a hand part. Then, the mean shift algorithm is applied to estimate the centers of hand parts to form a hand skeleton. This algorithm is subsequently enhanced by using a multilayered RDF framework [84], in which a hand shape classification serves as an intermediate layer to bridge the depth pixel and the hand pose estimator. In Fig. 9, we show visual features used by shape-based and skeleton-based pose estimation, respectively.

C. Gesture Classification

Hand gesture classification is the task of understanding the meanings of hand gestures, thus facilitating silent communication applications such as sign language. It usually incorporates hand detection and/or pose estimation techniques and outputs semantic keywords in the context of the application. Tang *et al.* [85] aim at establishing a fast system to identify simple hand gestures (e.g. grasping, pointing). In their work, a person's hand is estimated based on a skeletal tracker [77]. Next, local shape-related features, such as radial histogram and modified speeded-up robust features (SURF) [97], are extracted from the image. A support vector machine(SVM) classifier (with a radial basis function) is used to distinguish the hand gestures. The work reported in [86] organizes three classifiers in a hierarchical way, assuming the positions of all fingers at each frame are available. In the first layer, the system determines the number of fingers. In the second layer, the finger names can be provided. The classifier in the third layer calculates the angles between each pair of fingers, enabling to deduce the hand pose. In [87], Doliotis *et al.* adapt a well-known technique, called DTW, to depth-based gesture recognition. By doing so, the system is invariant to translation and scale changes of the gesture, leading to a system with fewer restrictions. In order to enhance the robustness of gesture recognition, authors in [88] employ two Kinect sensors in the system. The algorithm is designed to assist people who have difficulty in using a standard keyboard and a mouse. Two calibrated Kinects provide a rich point cloud, indicating positions of human hands in 3-D space. Eventually, a majority voting scheme is applied to multiple descriptors computed from the point cloud.

D. Summary

Kinect's depth signal adds considerable value to the visual understanding of human hands. Firstly, it simplifies the problem of robustly detecting and localizing hands in the image, benefiting downstream recognition of hand poses and gestures. Secondly, the depth signal is largely invariant to lighting conditions and skin colors, and gives a very clear

segmentation from the background. This allows the analysis of the hand to work robustly across different users in different environments. Finally, the calibrated depth signal gives you information about the distance of the hand from the sensor, allowing you to build a system that works at different depths.

However, the limited resolution of the Kinect sensor has its drawbacks: when the hand is far from the sensor, too few structure light dots illuminate the finger, and thus fingers tend to drop of the depth image. Despite this limitation, Microsoft has recently released version 1.7 of the Kinect SDK which includes a hand grip-release detector. Having been trained on a wide variety of data, this can work reliably across a wide range of hand shapes and depths, even when the fingers have dropped out.

VII. Indoor 3-D Mapping

Indoor 3-D mapping aims at creating a digital representation of an indoor environment, thus enabling automatic localization in that environment or the reconstruction of an environment. Earlier techniques [89], [90] have mostly relied on expensive sensors, for example, range sensors or laser finders, to generate 3-D point clouds. Spatially aligning 3-D point clouds of consecutive frames helps in building 3-D maps of indoor environments. For most algorithms, matching 3-D point clouds is typically accomplished by the ICP algorithm, where the correspondences are initialized by looking for nearest neighbor points in a 3-D space. Such algorithms are simple and usually have reasonable performance as long as a good initialization is available. However, the main constraint is that it normally requires a full overlap between the two point clouds. Additionally, the lack of RGB information makes the correspondence finding unreliable.

Alternatively, a stereo-camera setup [91] is adopted in the indoor 3-D mapping filed, in which the point depth needs to be computed by using stereo matching techniques. Although this solution is less costly and takes advantage of rich visual information, it is extremely difficult to obtain a *dense* depth map from a standard stereo-camera setup, especially in indoor environments with very dark or sparsely textured areas.

RGB-D sensor-based indoor 3-D mapping became popular recently because it combines depth sensors with color sensors to overcome the weaknesses of both. As a representative of the RGB-D sensors, Kinect was immediately used in this particular research field after it was released. Basically, existing indoor mapping systems sequentially consist of two main components. 1) Data acquisition and feature correspondence finding. 2) Loop closure detection and global optimization. In Fig. 10, we show the basic diagram of such systems. The loop closure detection and global optimization parts are often similar regardless of the sensing modality. Hence, we only focus on explaining how/what Kinect brings additive benefits to point clouds matching, and the detailed discussions on the entire chain are out of the scope of this paper.

A. Sparse Feature Matching

Several systems base their scene alignment on *sparse* feature points matching, in which a number of distinct points are extracted from successive frames and the geometric relation between them is found. The core idea is to compute a transformation by a fraction of *accurate* point-correspondences.

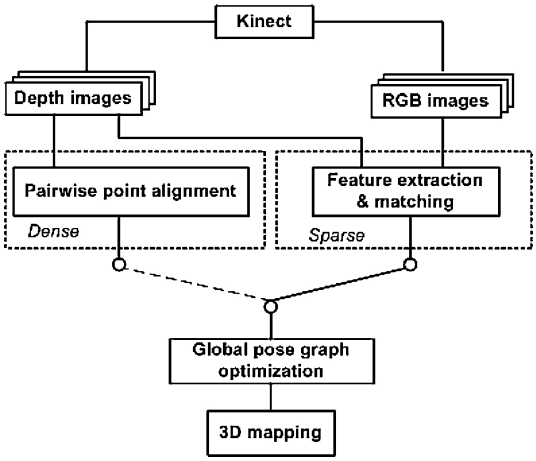


Fig. 10. Basic components of an indoor 3-D mapping system. It is noted that most papers choose either sparse feature alignment or dense point alignment.

A typical sparse feature matching approach is presented in [92], which improves the transformation estimation by using a more accurate feature matching. The novelty of the paper is a joint optimization for both appearance and shape matching via an ICP algorithm, where the shape can be reliably sensed on the depth map. Afterwards, a pose graph optimization ensures the global consistency using RGB feature correspondences. Later, many systems initialize with this algorithm [92] and aim to further improve its performance. Here, we list some of them. In [93] and [94], scale invariant feature transform (SIFT) [95]-based feature extraction and description used in [92] is replaced by some recently developed techniques, such as FAST feature descriptor [96] and SURF descriptor [97], which are relatively cheaper with respect to the computational load. Stuckler *et al.* [98] represent the shape-texture information in a more compact way. In [99], authors notice that the feature matching algorithm used in [92] fails for repetitive patterns such as carpet and wallpaper. To address this problem, they modify the matching algorithm to be a maximum likelihood estimation based on the prior knowledge regarding the continuity of motion between video frames.

B. Dense Point Matching

On the other hand, several systems insist on using the *dense* points tracking between two frames, where all pixels in the image contribute to the scene alignment. Since the point matching is computationally expensive, the dense point-tracking can only be performed on depth images if one targets real-time processing. The pioneering work of dense point-tracking is termed KinectFusion [100], [101]. This system demonstrates that the algorithm, using an efficiently parallelized implementation, can perform fast *dense* frame-to-frame depth tracking. The parallelized program is carried out on high-end graphics cards, thus enabling a real-time 3-D reconstruction. The key to such a system is the frame-to-frame depth tracking, which applies ICP between the new,

noisy depth frame and a clean rendering of the previous frame given the current 3-D reconstruction. With the new camera pose provided by ICP, the algorithm can fuse the new depth information into a 3-D volume using a truncated signed distance function. In [102], the KinectFusion method is evaluated by comparing it to high-end ground truth generation techniques. The objective is to study under which circumstances the 3-D reconstruction of KinectFusion is accurate enough to be used as training data. The results reveal that the system can resolve object details with a minimum size of approximately 10 mm. However, sharp depth edges or highly concave scenes are still problematic for the KinectFusion system. More recently, several systems are presented in order to adapt the KinectFusion system to more realistic situations, such as large lighting variations [103], [104], allowing camera roam free [105], and unbounded environments [106]. Similar to the research of RGB-D object recognition, a testing database as well as a benchmark are established by [107], containing color images, depth maps, and associated ground-truth camera pose information.

C. Summary

Depth information has been well explored in the indoor 3-D mapping area prior to the emergence of Kinect. Compared with other range sensors (e.g., TOF camera) used for 3-D mapping, Kinect enables much cheaper way of generating 3-D point clouds. More importantly, these 3-D point clouds are more accurate and attached with rich visual information, which potentially lead to a better point-cloud matching. In contrast to a stereo-camera based system that only provides depths for texture-rich points in the image (surfaces with little or no texture cannot be matched easily), novel Kinect-based systems can produce much denser depth maps, making applications that require a dense point cloud matching possible.

We have to point out that neither sparse nor dense (feature) point matching is good enough in terms of the system performance. For instance, most (feature) point descriptors used for matching are simple extensions/variations of available RGB feature descriptors without taking the characteristics of RGB-D images into account. A very new paper from Shotton *et al.* proves that advanced machine learning techniques (e.g., a decision forest), given simple RGB-D features, are able to considerably improve both sparse feature matching and whole-image keyframe matching [108].

VIII. Problems, Outlook and Conclusion

A lot of enthusiasm has been generated in the vision community by recent advances in low-cost 3-D depth cameras such as the Microsoft Kinect sensor. As we can see from the selection of work reviewed above, Kinect and its associated tools indeed have helped to overcome several problems that plague RGB camera-based algorithms and thus to enable many interesting applications of computer vision.

However, several important issues remain to be addressed. In this section, we first discuss some of these issues from the global point of view, and then we summarize the future trends for each vision topic mentioned in the paper. Finally, we draw a conclusion.

A. Real-World Applications

We have seen that the success of Kinect in gaming applications is mainly due to its efficient combination of new camera sensors and a highly robust skeletal tracker. However, it should be also noted that several environmental constraints in this scenario simplify the problem somewhat. For example, users in a game usually will not be too far away from the display, since they want to clearly see what happens on the screen. And since the camera is co-located with the display, this tends to avoid the range limitation of the depth sensor. Moreover, it's natural that users stand upright with their bodies near perpendicular to the camera plane when playing games. These valid assumptions make the human-pose estimation slightly easier: it is not necessary to tackle the view-angle change problem and the orientation of the human body can be well approximated. Furthermore, the limited number of users (one or two) and also the few obstacles within the playing area reduce the possibility that users' body parts are occluded. In fact, estimating the human pose when a person is partially or heavily occluded in the scene remains challenging.

Unfortunately, the above assumptions do not hold in more unconstrained computer vision applications, such as surveillance, smart environments and robotic vision. Factors that can severely restrict the applicability of Kinect-based intelligent systems in real world conditions include camera view-range limitation, occlusions, unpredictable human poses, etc. Methods that are robust to these factors merit further investigation.

Most high-level recognition systems, such as RGB-D object recognition and human activity recognition, assume that objects and humans can be reliably detected. However, the fact is that errors in feature extraction step can be propagated to higher levels. For example, if a human tracking algorithm (used for a feature extraction) does not extract the human at the focus of attention, then recognizing the activity being performed becomes much more difficult or impossible. Therefore, designing a high-level recognition system which is able to compensate for such low-level failures is a challenging task.

B. Efficient Integration of Algorithms

Most algorithms discussed above aim to solve one particular vision problem. Each solved problem can be seen as a crucial stepping stone toward the larger goal of designing computer-based intelligent systems. But how can we seamlessly integrate these algorithms? One straightforward way is to integrate them in a sequential order in terms of the requirements of the system. However, this is not efficient enough, because different algorithms may share more or less the same components. A typical example is the automatic indoor robotic system, which consists of RGB-D object recognition and indoor 3-D mapping. Both algorithms incorporate the RGB and depth feature extraction and matching/alignment. If we explore these algorithms separately, they may extract different types of features using the same input (e.g. depth images). In order to obtain a more efficient system, it is necessary to design improved feature extraction and description methods that are multifunctional.

C. Information Fusion

Several Kinect-related computer vision algorithms can be interpreted, at a high level, as performing information fusion of RGB and depth images. There are two common fusion schemes adopted by most current approaches. The first scheme is to selectively feed the complementary data (RGB or depth) to different algorithmic modules. A typical example is the Kinect gaming system, in which depth images are used to extract a player's skeleton while RGB images are the input of a facial feature based human identification algorithm. The second scheme simply feeds all available information (visual features) into an optimization algorithm. Such an example can be found in RGB-D object recognition.

The first scheme enables a fast system because it is not necessary to process all data at each algorithmic module. However, the decision about which data should be fed to which module is set empirically based on simple experiments. There are no proposals for the situation where either depth or RGB information is missing or polluted. The second scheme is potentially more accurate because all information can be taken into account. However, it is also clear that some information is typically redundant for certain algorithmic modules, and so a crude fusion in this case may not improve accuracy and will likely slow the system down. Therefore, the investigation for intelligent information fusion or interactive fusion is highly desirable, and could have a large impact. Fortunately, some works, for example [40], have started research in this direction.

Additionally, the way of extracting useful information (features) from depth images seems not to be sophisticated enough. Most of current approaches just slightly change feature extraction and description algorithms available in the RGB image domain, such as HOG, SIFT and SURF. The suitability of such algorithms for depth images is suspectable, because the characteristics of RGB and depth images are different, e.g. the texture on the depth image is much less than that on the RGB image. Therefore, specific feature extraction algorithms designed for depth images, such as HONV in [49], are encouraged.

D. Outlook for the Future

By analyzing above papers, we believe that there are certainly many future works in this research community. Here, we discuss potential ideas for each of main vision topics separately.

1) *Object tracking and recognition.* Seen from Fig. 5, background subtraction based on depth images can easily solve practical problems that have hindered object tracking and recognition for a long time. It will not be surprising if tiny devices equipped with Kinect-like RGB and depth cameras appear in normal office environments in the near future. However, the limited range of the depth camera may not allow it to be used for standard indoor surveillance applications. To address this problem, the combination of multiple Kinects may be a potential solution. This will of course require the communication between the Kinects and object reidentification across different views.
2) *Human activity analysis.* Achieving a reliable algorithm that can estimate complex human poses (such as gymnastic or acrobatic poses) and the poses of tightly interacting people will definitely be active topics in the future. For activity recognition, further investigations for low-latency systems, such as the system described in [66], may become the trend in this filed, as more and more practical applications demand online recognition.
3) *Hand gesture analysis.* It can be seen that many approaches avoid the problem of detecting hands from a realistic situation by assuming that the hands are the closest objects to the camera. These methods are experimental and their use is limited to laboratory environments. In the future, methods that can handle arbitrary, high degree of freedom hand motions in realistic situations may attract more attention. Moreover, there is a dilemma between shape based and 3-D model based methods. The former allows high speed operation with a loss of generality while the latter provides generality at a higher cost of computational power. Therefore, the balance and trade-off between them will become an active topic.
4) *Indoor 3-D mapping.* According to the evaluation results from [107], most current approaches fail when erroneous edges are created during the mapping. Hence, the methods that are able to detect erroneous edges and repair them autonomously will be very useful in the future [107]. In sparse feature-based approaches, there might be a need to optimize the key point matching scheme, by either adding a feature look-up table or eliminating non-matched features. In dense point-matching approaches, it is worth trying to reconstruct larger scenes such as the interior of a whole building. Here, more memory efficient representations will be needed.

E. Conclusion

The dream of building a computer that can recognize and understand scenes like human has already brought many challenges for computer-vision researchers and engineers. The emergence of Microsoft Kinect (both hardware and software) and subsequent research efforts have brought us closer to this goal. In this review, we summarized the main methods that were explored for addressing various vision problems. The covered topics included object tracking and recognition, human activity analysis, hand gesture analysis, and indoor 3-D mapping. We also suggested several technical and intellectual challenges that need to be studied in the future.

ACKNOWLEDGMENTS

A part of this work was done while J. Han was employed by CWI, the Netherlands. Therefore, J. Han would like to thank Dr. E. Pauwels for granting the opportunity to work on this interesting topic.

REFERENCES

[1] *Kinect camera* [Online]. Available: http://www.xbox.com/en-US/kinect/default.htm

[2] *Stereo camera* [Online]. Available: http://en.wikipedia.org/wiki/Stereo_camera

[3] S. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor system description, issues and solutions," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2004, pp. 35–45.

[4] Z. Zhang, "Microsoft Kinect sensor and its effect," *IEEE Multimedia Mag.*, vol. 19, no. 2, pp. 4–10, Feb. 2012.

[5] I. Tashev, "Recent advances in human-machine interfaces for gaming and entertainment," *Int. J. Inform. Technol. Security*, vol. 3, no. 3, pp. 69–76, 2011.

[6] K. Kumatani, T. Arakawa, K. Yamamoto, J. McDonough, B. Raj, R. Singh, and I. Tashev, "Microphone array processing for distant speech recognition: Towards real-world deployment," in *Proc. APSIPA Annu. Summit Conf.*, 2012, pp. 1–10.

[7] J. Geng, "Structured-light 3-D surface imaging: A tutorial," *Adv. Optics Photonics*, vol. 3, no. 2, pp. 128–160, 2011.

[8] *OpenNI* [Online]. Available: http://www.openni.org/

[9] *Microsoft Kinect SDK* [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/.

[10] *OpenKinect* [Online]. Available: https://github.com/OpenKinect/libfreenect/.

[11] J. Smisek, M. Jancosek, and T. Pajdla, "3-D with Kinect," in *Proc. IEEE ICCV Workshops*, 2011, pp. 1154–1160.

[12] T. Stoyanov, A. Louloudi, H. Andreasson, and A. Lilienthal, "Comparative evaluation of range sensor accuracy in indoor environments," in *Proc. Eur. Conf. Mobile Robots*, 2011, pp. 19–24.

[13] K. Khoshelham and S. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[14] T. Dutta, "Evaluation of the Kinect sensor for 3-D kinematic measurement in the workplace," *Appl. Ergonom.*, vol. 43, no. 4, pp. 645–649, Jul. 2012.

[15] S. Obdrzalek, G. Kurillo, F. Ofli, R. Bajcsy, E. Seto, H. Jimison, and M. Pavel, "Accuracy and robustness of Kinect pose estimation in the context of coaching of elderly population," in *Proc. IEEE EMBC*, 2012, pp. 1188–1193.

[16] C. Zhang and Z. Zhang, "Calibration between depth and color sensors for commodity depth cameras," in *Proc. IEEE ICME*, 2011, pp. 1–6.

[17] D. Herrera, J. Kannala, and J. Heikkila, "Accurate and practical calibration of a depth and color camera pair," in *Proc. CAIP*, 2011, pp. 437–445.

[18] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[19] D. Herrera, J. Kannala, and J. Heikkila, "Joint depth and color camera calibration with distortion correction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 2058–2064, Oct. 2012.

[20] D. Fiedler and H. Muller, "Impact of thermal and environmental conditions on the Kinect sensor," in *Proc. Int. Workshop Depth Image Anal.*, 2012.

[21] Y. Berdnikov and D. Vatolin, "Eal-time depth map occlusion filling and scene background restoration for projected-pattern-based depth camera," in *Proc. Int. Conf. Comput. Graph. Vision* , 2011, pp. 1–4.

[22] S. Milani and G. Calvagno, "Joint denoising and interpolation of depth maps for MS Kinect sensors," in *Proc. ICASSP*, 2012, pp. 797 –800.

[23] M. Schmeing and X. Jiang, "Color Segmentation Based Depth Image Filtering," in *Proc. Int. Workshop Depth Image Anal.*, 2012.

[24] M. Camplani and L. Salgado, "Efficient spatio-temporal hole filling strategy for Kinect depth maps," in *Proc. SPIE Int. Conf. 3-D Image Process. Appl.*, vol. 8290, 2012, pp. 1–10.

[25] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *Proc. Int. Conf. Comput. Vision Workshop 3-D Representation Recognit.*, 2011, pp. 601–608.

[26] F. Qi, J. Han, P. Wang, G. Shi, and F. Li, "Structure guided fusion for depth map inpainting," *Pattern Recognition Letters*, vol. 34, no. 1, pp. 70–76, 2013.

[27] L. Xia, C. Chen, and J. Aggarwal, "Human detection using depth information by Kinect," in *Proc. Int. Workshop HAU3D*, Jun. 2011, pp. 15–22.

[28] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier, "Fall detection from depth map video sequences," in *Proc. ICOST*, 2011, pp. 121–128.

[29] Z. Zhang, W. Liu, V. Metsis, and V. Athitsos, "A viewpoint-independent statistical method for fall detection," in *Proc. ICPR*, Nov. 2012, pp. 3626–3630.

[30] Y. Shen, P. Wang, and S. Ma, "Performance study of feature descriptors for human detection on depthMap," in *Proc. Int. Conf. Syst. Simulat. Sci. Comput.*, 2012.

[31] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2005, pp. 886–893.

[32] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proc. Int. Conf. Image Video Retrieval*, 2007, pp. 401–408.

[33] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[34] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1635–1650, Jun. 2010.

[35] J. Wu and J. Rehg, "CENTRIST: A visual descriptor for scene categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1489–1501, Aug. 2011.

[36] J. Han, E. Pauwels, P. de Zeeuw, and P. de With, "Employing an RGB-D sensor for real-time tracking of humans across multiple re-entries in a smart environment," *IEEE Trans. Consumer Electron.*, vol. 58, no. 2, pp. 255–263, May 2012.

[37] L. Spinello and K. Arras, "People detection in RGB-D data," in *Proc. Int. Conf. IROS*, 2011, pp. 3838–3843.

[38] L. Spinello, C. Stachniss, and W. Burgard, "Scene in the loop: Towards adaptation-by-tracking in RGB-D data," in *Proc. Workshop RGB-D, Adv. Reason. Depth Cameras*, 2012.

[39] M. Luber, L. Spinello, and K. Arras, "People tracking in RGB-D data with On-line boosted target models," in *Proc. Int. Conf. IROS*, 2011.

[40] L. Spinello and K. Arras, "Leveraging RGB-D Data: Adaptive fusion and domain adaptation for object detection," in *Proc. IEEE ICRA*, 2012, pp. 4469–4474.

[41] W. Liu, T. Xia, J. Wan, Y. Zhang, and J. Li, "RGB-D based multi-attribute people search in intelligent visual surveillance," in *Proc. Int. Conf. Adv. Multimedia Modeling*, 2012, pp. 750–760.

[42] W. Choi, C. Pantofaru, and S. Savarese, "Detecting and tracking people using an RGB-D camera via multiple detector fusion," in *Proc. Workshop Challenges Opportunities Robot Perception*, 2011, pp. 1076–1083.

[43] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multiview RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1817–1824.

[44] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vision*, 2012, pp. 746–760.

[45] K. Lai, L. Bo, X. Ren, and D. Fox, "Sparse distance learning for object recognition combining RGB and depth information," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 4007–4013.

[46] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for RGB-D based object recognition," in *Proc. Int. Symp. Experimental Robot.*, 2012.

[47] L. Bo, K. Lai, X. Ren, and D. Fox, "Object recognition with hierarchical kernel descriptors," in *Proc. CVPR*, 2011, pp. 1729–1736.

[48] X. Ren, L. Bo, and D. Fox, "RGB-(D) scene labeling: Features and algorithms," in *Proc. CVPR*, 2012, pp. 2759–2766.

[49] S. Tang, X. Wang, X. Lv, T. Han, J. Keller, Z. He, M. Skubic, and S. Lao, "Histogram of oriented normal vectors for object recognition with a depth sensor," in *Proc. Asian Conf. Comput. Vision*, 2012.

[50] M. Blum, J. Springenberg, J. Wulfing, and M. Riedmiller, "A learned feature descriptor for object recognition in RGB-D data," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 1298–1303.

[51] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *Proc. Int. Conf. IROS*, 2011, pp. 821–826.

[52] 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 a single depth image," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2011, pp. 1297–1304.

[53] M. Budiu, J. Shotton, D. Murray, and M. Finocchio, "Parallelizing the training of the Kinect body parts labeling algorithm," in *Proc. Workshop Big Learn., Algorithms, Syst. Tools Learn. Scale*, 2011, pp. 6–12.

[54] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," in *Proc. ICCV*, 2011, pp. 415–422.

[55] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon, "The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation," in *Proc. CVPR*, 2012, pp. 103–110.

[56] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, "Accurate 3-D pose estimation from A single depth image," in *Proc. ICCV*, 2011, pp. 731–738.

[57] W. Shen, K. Deng, X. Bai, T. Leyvand, B. Guo, and Z. Tu, "Exemplar-based human action pose correction and tagging," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 1784–1791.

[58] X. Yang and Y. Tian, "EigenJoints-based action recognition using naive-bayes-nearest-neighbor," in *Proc. IEEE Workshop CVPR Human Activity Understand. 3-D Data*, 2012, pp. 14–19.

[59] X. Yang and Y. Tian, "Effective 3-D action recognition using eigenJoints," *J. Visual Commun. Image Represent.*, doi: 10.1016/j.jvcir.2013.03.001.

[60] L. Xia, C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3-D joints," in *Proc. IEEE Workshop CVPR Human Activity Understand. 3-D Data*, 2012, pp. 20–27.

[61] A. Jalal, S. Lee, J. Kim, and T. Kim, "Human activity recognition via the features of labeled depth body parts," in *Proc. Int. Conf. Smart Homes Health Telematics*, 2012, pp. 246–249.

[62] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Human activity detection from RGBD images," in *Proc. AAAI Workshop Plan, Activity, Intent Recognit.*, 2011, pp. 47–55.

[63] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructured human activity detection from RGBD images," in *Proc. IEEE ICRA*, 2012, pp. 842–849.

[64] M. Reyes, G. Dominguez, and S. Escalera, "Featureweighting in dynamic timewarping for gesture recognition in depth data," in *Proc. IEEE ICCV Workshop*, 2011, pp. 1182–1188.

[65] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 1290–1297.

[66] S. Nowozin and J. Shotton, "Action points: A representation for low-latency online human action recognition," Microsoft Research Cambridge, Tech. Rep. MSR-TR-2012-68, 2012.

[67] B. Ni, G. Wang, and P. Moulin, "RGBD-HuDaAct: A color-depth video database for huamn daily activity recognition," in *Proc. IEEE Workshop Consum. Depth Cameras Comput. Vision*, 2011, pp. 1147–1153.

[68] S. Fothergill, H. Mentis, S. Nowozin, and P. Kohli, "Instructing people for training gestural interactive systems," in *Proc. ACM Conf. Human Factors Comput. Syst.*, 2012, pp. 1737–1746.

[69] Z. Cheng, L. Qin, Y. Ye, Q. Huang, and Q. Tian, "Human daily action analysis with multiview and color-depth data," in *Proc. ECCV Workshop Consum. Depth Cameras Comput. Vision*, 2012, pp. 52–61.

[70] T. Berg, D. Chattopadhyay, M. Schedel, and T. Vallier, "Interactive music: Human motion initiated music generation using skeletal tracking by Kinect," in *Proc. Conf. Soc. Electro-Acoustic Music United States*, 2012.

[71] M. Raptis, D. Kirovski, and H. Hoppe, "Real-time classification of dance gestures from skeleton animation," in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, 2011, pp. 147–156.

[72] V. Escorcia, M. Golparvar-Fard, and J. Niebles, "Automated vision-based recognition of construction worker actions for building interior construction operations using RGBD cameras," in *Proc. Construction Res. Congr.*, 2012, pp. 879–888.

[73] R. Tara, P. Santosa, and T. Adji, "Hand segmentation from depth image using anthropometric approach in natural interface development," *Int. J. Sci. Eng. Res.*, vol. 3, no. 5, pp. 1–4, May 2012.

[74] U. Lee and J. Tanaka, "Hand controller: Image manipulation interface using fingertips and palm tracking with Kinect depth data," in *Proc. Asia Pacific Conf. Comput. Human Interact.*, 2012, pp. 705–706.

[75] H. Liang, J. Yuan, and D. Thalmann, "3-D fingertip and palm tracking in depth image sequences," in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 785–788.

[76] G. Hackenberg, R. McCall, and W. Broll, "Lightweight palm and finger tracking for real-time 3-D gesture control," in *Proc. IEEE Conf. Virtual Reality*, 2011, pp. 19–26.

[77] M. Caputo, K. Denker, B. Dums, and G. Umlauf, "3-D hand gesture recognition based on sensor fusion of commodity hardware," in *Proc. Conf. Mensch Comput.*, 2012, pp. 293–302.

[78] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3-D tracking of hand articulations using Kinect," in *Proc. Brit. Mach. Vision Conf.*, 2011, pp. 101.1–101.11.

[79] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 1862–1869.

[80] P. Doliotis, V. Athitsos, D. Kosmopoulos, and S. Perantoni, "Hand shape and 3-D pose estimation using depth data from a single cluttered frame," in *Proc. ISVC*, 2012, pp. 148–158.

[81] Z. Ren, J. Meng, J. Yuan, and Z. Zhang, "Robust hand gesture recognition with Kinect sensor," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 759–760.

[82] Y. Yao and Y. Fu, "Real-time hand pose estimation from RGB-D sensor," in *Proc. IEEE Conf. Multimedia Expo*, 2012, pp. 705–710.

[83] C. Keskin, F. Kira, Y. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in *Proc. ICCV Workshop*, 2011, pp. 1228–1234.

[84] C. Keskin, F. Kira, Y. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multilayered randomized decision forests," in *Proc. Eur. Conf. Comput. Vision*, 2012, pp. 852–863.

[85] M. Tang, "Recognizing hand gestures with microsoft's Kinect," Dept. Electrical Eng., Stanford Univ., CA, USA, Tech. Rep., 2011.

[86] Y. Li, "Hand gesture recognition using Kinect," in *Proc. Int. Conf. Software Eng. Service Sci.*, 2012, pp. 196–199.

[87] P. Doliotis, A. Stefan, C. Mcmurrough, D. Eckhard, and V. Athitsos, "Comparing gesture recognition accuracy using color and depth information," in *Proc. Conf. Pervasive Technol. Related Assistive Environments*, 2011, article 20.

[88] R. Mihail, N. Jacobs, and J. Goldsmith, "Static hand gesture recognition with 2 Kinect sensors," in *Proc. Int. Conf. Image Process., Comput. Vision Pattern Recognit.*, 2012.

[89] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multirobot and 3-D mapping," in *Proc. IEEE Int. Conf. ICRA*, 2000, pp. 321–328.

[90] J. Biswas and M. Veloso, "Depth camera based indoor mobile robot localization and navigation," in *Proc. IEEE Int. Conf. ICRA*, 2012, pp. 1697–1702.

[91] L. Paz, P. Pinies, J. Tardos, and J. Neira, "Large-scale 6-DOF SLAM with stereo-in-hand," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 946–957, Oct. 2008.

[92] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3-D modeling of indoor environments," in *Proc. Int. Symp. Experimental Robot.*, 2010.

[93] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3-D modeling of indoor environments," *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 647–663, Apr. 2012.

[94] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, "Real-time 3-D visual SLAM with A hand-held camera," in *Proc. RGB-D Workshop 3-D Perception Robot. Eur. Robot. Forum*, 2011.

[95] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[96] E. Rosten, R. Porter, and T. Drummond, "Faster and Better: A machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, Jan. 2010.

[97] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Comput. Vision Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[98] J. Stuckler and S. Behnke, "Integrating depth and color cues for dense multiresolution scene mapping using RGB-D cameras," in *Proc. IEEE Int. Conf. Multisensor Fusion Inform. Integr.*, 2012.

[99] Y. Zhang, C. Luo, and J. Liu, "Walk & Sketch: Create floor plans with an RGB-D camera," in *Proc. ACM Int. Conf. Ubiquitous Comput.*, 2012.

[100] R. Newcombe, A. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 127–136.

[101] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: Real-time 3-D reconstruction and interaction using a moving depth camera," in *Proc. ACM Symp. User Interface Software Technol.*, 2011, pp. 559–568.

[102] S. Meister, S. Izadi, P. Kohli, M. Haemmerle, C. Rother, and D. Kondermann, "When can we use KinectFusion for ground truth acquisition?" in *Proc. Workshop Color-Depth Camera Fusion Robot.*, 2012.

[103] M. Meilland, A. Comport, and P. Rives, "Real-time dense visual tracking under large lighting variations," in *Proc. Conf. Brit. Mach. Vision*, pp. 1–11, 2011.

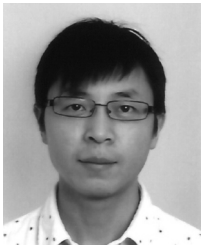
[104] M. Meilland, A. Comport, and P. Rives, "Dense RGB-D mapping for real-time localisation and navigation," in *Proc. Workshop Navigation Positiong Mapping*, 2012.

[105] H. Roth and M. Vona, "Moving volume KinectFusion," in *Proc. Brit. Mach. Vision Conf.*, 2012, pp. 1–11.
[106] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended KinectFusion," in *Proc. Workshop RGB-D, Adv. Reason. Depth Cameras*, 2012, article 4.
[107] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. Int. Conf. Intell. Robot Syst.*, 2012, pp. 573–580.
[108] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proc. CVPR*, accepted by CVPR.
[109] L. Liu and L. Shao, "Learning discriminative representations from RGB-D video data," in *Proc. IJCAI*, accepted by IJCAI.

Jungong Han (M'12) received the Ph.D. degree in telecommunication and information system from Xidian University, Shaanxi, China, in 2004.

While pursuing the Ph.D. degree, he spent one year at the Internet Media Group, Microsoft Research Asia, China. From 2005 to 2010, he was with the Signal Processing Systems Group, Technical University of Eindhoven (TU/e), Eindhoven, The Netherlands. In 2010, he joined the Multiagent and Adaptive Computation Group, Centre for Mathematics and Computer Science, Amsterdam, The Netherlands, as a Research Staff Member. In 2012, he took up a Senior Scientist position with Civolution technology (a combining synergy of Philips Content Identification and Thomson STS), Eindhoven, The Netherlands. Since 2010, he has also been a Guest Researcher at TU/e. He has authored and co-authored over 60 papers. His current research interests include multimedia content identification, multisensor data fusion, and computer vision.

He is an Associate Editor of *Elsevier Neurocomputing*, and has organized several special issues on international journals, such as the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: B, *Pattern Recognition Letters* and *Neurocomputing*.



Ling Shao (M'09–SM'10) received the B.Eng. degree in electronic engineering from the University of Science and Technology of China, Anhui, China, the M.Sc. degree in medical image analysis, and the Ph.D. (D.Phil.) degree in computer vision from the Robotics Research Group from the University of Oxford, Oxford, U.K.

He is currently a Senior Lecturer (Associate Professor) in the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, U.K. and a Guest Professor with Nanjing University of Information Science and Technology, China. Before joining Sheffield University, he was a Senior Scientist in Philips Research, The Netherlands. He has authored/co-authored over 80 academic papers in refereed journals and conference proceedings and has over 10 awarded patents and patent applications. His current research interests include computer vision, pattern recognition, and video processing.

Dr. Shao is an Associate Editor of IEEE TRANSACTIONS ON CYBERNETICS, the *International Journal of Image and Graphics*, the *EURASIP Journal on Advances in Signal Processing*, and *Neurocomputing*, and has edited several special issues for journals of IEEE, Elsevier and Springer. He has organized several workshops with top conferences, such as ICCV, ACM Multimedia and ECCV. He has been serving as a Program Committee member for many international conferences, including ICCV, CVPR, ECCV, ICIP, ICASSP, ICME, ICMR, ACM MM, CIVR, BMVC, and so on. He is also a Fellow of the British Computer Society.

Dong Xu (M'07) received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Anhui, China, in 2001 and 2005, respectively.

While pursuing the Ph.D. degree, he was with Microsoft Research Asia, Beijing, China, and the Chinese University of Hong Kong, Shatin, Hong Kong, for more than two years. He was a Post-Doctoral Research Scientist with Columbia University, New York, NY, USA, for one year. In May 2007, he joined Nanyang Technological University, Singapore, where he is currently an Associate Professor with the School of Computer Engineering. His current research interests include computer vision, statistical learning, and multimedia content analysis.

Dr. Xu was the co-author of a paper that won the Best Student Paper Award in the IEEE International Conference on Computer Vision and Pattern Recognition in 2010.

Jamie Shotton (M'10) received the B.A. degree in computer science and the Ph.D. in computer vision from the University of Cambridge, Cambridge, U.K. He had been a Post-Doctoral Toshiba Fellow at Toshiba's Corporate Research and Development Center, Japan.

Since 2008, he has been a Senior Researcher in the Machine Learning and Perception Group, Microsoft Research, Cambridge, U.K. His recent work has focused on human pose estimation for Microsoft's Kinect camera. His current research interests include gesture and action recognition, object recognition, medical imaging, and machine learning.

Dr. Shotton was a recipient of the Best Paper Award at CVPR 2011, and the Royal Academy of Engineering's MacRobert Award 2011 for his work on human pose estimation for Microsoft's Kinect camera.