Depth Motion Detection—A Novel RS-Trigger Temporal Logic based Method

Can Wang, *Student Member, IEEE*, Hong Liu, *Member, IEEE*, and Liqian Ma

***Abstract*—Recently, depth data is widely used in computer vision applications such as detection and tracking, which shows great promises in complicated environments due to its complementary natures to RGB data. However, previous works mostly use depth as an auxiliary cue of RGB data and overlook its inherent advantage on motion detection. Intrinsically different from RGB data, points in depth map essentially represents 3-D positions in the world, so depth video represents the variation of these "positions," which is motion. Motivated by this, we proposed a novel motion detection scheme based on RS-Trigger temporal logic which best fits nature of depth data on motion detection. The proposed algorithm can fast detect motion regions in the scene without statistics of background and prior knowledge of objects to detect. In following refinement modules, a depth-invariant density-constant projection is proposed which contributes to a fast spatial clustering and accurate segmentation, for it transforms dense 3-D points cloud to depth-invariant 2-D map with density-constance, not only it overcomes depth-dependent sampling of depth sensor, but also overcomes the common 'scale problem' in 2-D image analysis, which makes it easy to set system parameters to de-noise and pop-out motion regions. Experimental results validate its effectiveness and efficiency.**

***Index Terms*—Depth data, motion detection.**

I. Introduction

COMBINING RGB and depth data for computer vision applications becomes more and more popular especially after Microsoft Kinect sensor is widely used [1]–[6]. However, due to so many sophisticated methods for RGB data exists, in these applications, depth data only plays an auxiliary role to RGB data, which is also because existing depth sensors are not reliable enough, due to low resolution or extensive unstable regions and holes in depth videos.

When it comes to motion detection and objects segmentation, some researchers combine both depth data and intensity data together for detection [7]–[12], but it brings relatively higher computing resource consumption, and require range sensors can obtain both depth and intensity data simultaneously, which is impossible for some kinds of sensors, such as TOF sensors [13] and previous structure-light sensors without calibrated RGB sensor. Therefore, only using depth data can expand the method's applied scope and can reduce the manufacturing cost. From another angle, if detection methods only using depth can achieve good performance, this undoubtedly will be a firm foundation for methods combining more RGB cues and will be competent in higher level applications.

There are indeed some previous works which only use depth data for motion detection [14]–[17], foreground segmentation [4][13] or multiple objects tracking [18]. Work in [4] adopted temporal difference scheme to obtain salient 3-D motion points and perform 3-D clusterings for refinement. But temporal difference scheme will introduce background noise and can hardly get whole motion regions. Motion history image (MHI) method is used in both motion detection module of work [14] and [15]. However, MHI is actually an accumulation of temporal difference and is simply inherited from traditional intensity-based method [19]. Depth data is adopted in work [13] for objects segmentation which is a strong cue to separate foreground from background and another work [18] also try to use depth data to separate different objects in the scene according to their depth layers. However, this is not a generative way for objects segmentation which are only suitable for ideal cases that objects' depth layers are separable.

In this work, we focus on designing a novel motion detection framework which is suitable for depth videos and can fully exploit potential of depth data on motion description. Unlike previous works obtaining motion regions via threshold-based absolute-difference scheme (background subtraction and temporal difference), our main contributions consist of proposing a novel pixel-level temporal logic scheme for motion detection specifically designed for depth data, and a coarse-to-fine scheme which fully utilizes depth information achieving depth-invariance for accurate localization and segmentation, which overcomes hardware drawbacks of depth sensors, such as holes and unstable regions [20] in depth videos. Details of the proposed method are elaborated in following sections.

II. Pixel-Level RS-Trigger Based Motion Detection

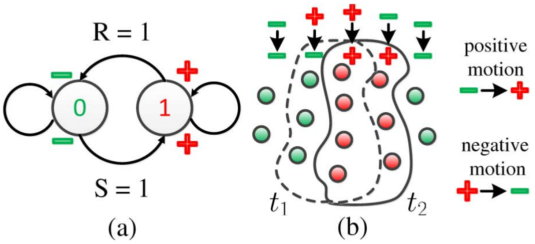
In digital electronics, RS-Trigger (RST) is a temporal logic with two stable states, as shown in Fig. 1(a). The states switch between 0 and 1 according to the input signals R and S. This is

Manuscript received December 04, 2013; revised March 05, 2014; accepted March 10, 2014. Date of publication March 24, 2014; date of current version April 03, 2014. This work was supported by National Natural Science Foundation of China (NSFC) under Grants 61340046, 60875050, and 60675025, the National High Technology Research and Development Program of China (863 Program) under Grant 2006AA04Z247, the Scientific and Technical Innovation Commission of Shenzhen Municipality under Grants JCYJ20120614152234873, CXC201104210010A, JCYJ20130331144631730, and JCYJ20130331144716089, and by the Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20130001110011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Petros T. Boufounos.

The authors are with the Engineering Lab on Intelligent Perception for Internet of Things (ELIP) and Key Laboratory of Machine Perception (Ministry of Education), Shenzhen Graduate School, Peking University, China (e-mail: canwang@pku.edu.cn; hongliu@pku.edu.cn; maliqianmemory@gmail.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/LSP.2014.2313345



(a) (b)

Fig. 1. (a) The logic graph of RS-Trigger. (b) A typical example of how pixels' states change when an object move from time t_1 to time t_2.

quite similar to the motion detection problem which also has two states for each pixel: on a moving object or not. Based on this intuition, a pixel-level motion detection approach only using depth data is proposed:

First, given a depth frame $D(t)$ at time t, let each pixel p_i with coordinates (u_i, v_i) be denoted as a 4-tuple:

$$p_i(t) := \{u_i, v_i, d_i(t), s_i(t)\} \tag{1}$$

where $d_i(t)$ is the depth value of p_i at time t and $s_i(t)$ is the binary variable indicating the current state of p_i, whether on moving objects $(s_i(t) = 1)$ or not $(s_i(t) = 0)$, corresponding to two stable states 0 and 1 of RST.

Then, given consecutive depth frames $D(t-1)$ and $D(t)$, for each point p_i, two variables $\mathcal{J}_{pm}$ and $\mathcal{J}_{nm}$ corresponding to two control signals S and R of RST are defined as:

$$\mathcal{J}^i_{pm}(t) = \begin{cases} 1 & \text{if } d_i(t) - d_i(t-1) > \tau_i(t) \\ 0 & \text{otherwise} \end{cases}$$
$$\mathcal{J}^i_{nm}(t) = \begin{cases} 1 & \text{if } d_i(t) - d_i(t-1) < -\tau_i(t) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where the subscript 'pm' and 'nm' indicate 'positive motion' and 'negative motion' respectively, which represent significant motion on the pixel-level. The term 'positive motion' can be intuitively interpreted as the point skips closer to the sensor and 'negative motion' the point skips far away from the sensor. Here $\tau_i(t)$ is an adaptive threshold indicating whether there is significant motion on pixel p_i at time t, formulated as:

$$\tau_i(t) = \min(\tau_{pm}, \alpha_0 \cdot (d_{max} - d_i(t-1))) \tag{3}$$

where d_{max} is the maximum pixel value on depth map being used (here is 255), τ_{pm} is set to limit the maximum threshold and α_0 is a scaling factor. This adaptive threshold can suppress commonly hopping in far distance away from depth sensor due to hardware drawbacks, and also can increase sensitivity of motion detection in short distance.

Third, the state of each pixel $s_i(t)$ is updated by variables $\mathcal{J}_{pm}$ and $\mathcal{J}_{nm}$ following the RST temporal logic in Algorithm 1. Any pixel with state 1 is regarded as a point on moving objects. Fig. 1(b) shows how pixel states change when an object moving in the scene according to RST scheme. Thus, points set on moving objects at each frame t can be given as:

$$M(t) := \{p_i(t) | s_i(t) = 1, p_i(t) \in D(t)\} \tag{4}$$

Fig. 2 gives the visualization of $M(t)$ in temporal sequence. It can be seen that the RST temporal logic based motion detection is a cumulative process and points on objects are gradually detected while they are moving in the scene.

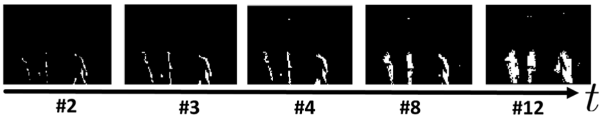


Fig. 2. The temporal accumulation of the RST motion detection procedure.

Algorithm 1 RS-Trigger temporal logic for updating $s_i(t)$

```
Input: $p_i = \{p_i(1), p_i(2), \cdots, p_i(T)\}$
Output: $s_i = \{s_i(1), s_i(2), \cdots, s_i(T)\}$;
initial $s_i(0)$ to 0;
for $t = 1$ to $T$ do
    if $\mathcal{J}^i_{pm}(t) = 1$ then
        $s_i(t) = 1$;
    else if $\mathcal{J}^i_{nm}(t) = 1$ then
        $s_i(t) = 0$;
    else
        $s_i(t) = s_i(t-1)$
    end if
end for
```

III. Motion Segmentation Refinement

The RST motion detection scheme can obtain a set of points cloud $\mathcal{M}(t)$ in each depth frame. However, motion segmentation in point clouds is harder than in color images alone due to irregular sampling and the high noise levels of depth sensors [21]. Moreover, because of ubiquitous unstable regions and holes in the depth video [14][20], the pixel-level motion detection approach is inevitably affected, so the obtained coarse motion detection $\mathcal{M}(t)$ needs refinement.

A. Frequent-Hopping Filtering

First, a frequent-hopping filtering (FHF) approach is proposed to remove the adverse effects of unstable regions:

$$\mathcal{N}(t) := \left\{ p_i(t) \middle| \sum_{k=T-T_0}^{T} (|\mathcal{J}^i_{pm}(k)| + |\mathcal{J}^i_{nm}(k)|) > \frac{T_0}{\alpha_f} \right\} \tag{5}$$

where α_f is used to control the threshold of frequency. Then, the refined motion points set can be given as:

$$\hat{\mathcal{M}}(t) = \mathcal{M}(t) - \mathcal{N}(t) \tag{6}$$

The FHF actually acts as a de-noise process based on the observation that unstable regions and holes in depth videos exhibits frequent hopping, but real motion points do not have such feature for real moving objects seldom disappear and appear frequently and constantly at the same point.

B. Depth-Invariant Density-Constant Projection

Pixel-level motion detection scheme intrinsically suffers from the lack of spatial constraints, especially for depth video where have amount of noise points and separated regions in the coarse detection results. To handle this, previous works usually

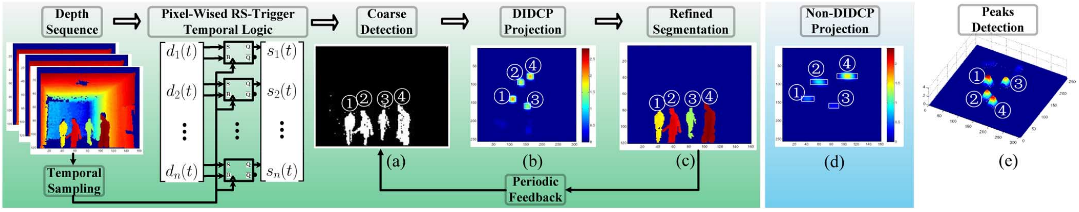


Fig. 3. A brief illustration of the proposed method, where (d) is specifically given for comparison with (c).

adopt noise filtering and morphology processing techniques on the 2-D grayscale map of coarse detection. But for 3-D point clouds, noise filtering and morphology processing on three-dimensional space is relatively computation-consuming. On the other side, if depth map is simply treated as 2-D grayscale map and processed using classic 2-D techniques, there is no doubt that depth information is lost.

In order to handle this dilemma, we proposed a depth-invariant density-constant projection (DIDCP) method, which transforms the original 3-D coordinates of points cloud to new 2-D image coordinates which have stable length metric proportional to the absolute length metric of the real world. The pixel value of the projection map is the density of projection on that point. The DIDCP is formulated as follows:

Given a point in original 3-D coordinates $p_i = (u_i, v_i, d_i)$ and a its corresponding point $p'_j = (u'_j, d'_j, w'_j)$ in the new coordinate, the DIDCP projection can be denoted as:

$$\mathcal{P} : \hat{\mathcal{M}} \rightarrow \mathcal{W} \Rightarrow \{p_i \xrightarrow{\mathcal{P}} p'_j\} \tag{7}$$

where $\hat{\mathcal{M}}$ is the 3-D motion points set in Eq. (6) and $\mathcal{W}$ is the projection map.

Depth-invariant indicates that length (measured in pixels) of the new 2-D coordinates corresponds to constant length (measured in meters) in the real world. Given any 3-D point p_i, the 2-D coordinates (u'_j, d'_j) of p'_j on the projection map $\mathcal{W}$ can be given as:

$$\begin{aligned} u'_j &= \alpha_u \cdot \frac{u_i - u_0}{f_u} \cdot d_i \\ d'_j &= \alpha_d \cdot d_i \end{aligned} \tag{8}$$

where α_u and α_d are both scale factors, and f_u and u_0 are intrinsic parameters of the RGB-D sensor. This projection means a lot, for example, on 2-D map projection of noise regions closer to the sensor are larger than motion regions farther from the sensor due to the 'scale problem'. But with depth-invariance, the 'scale problem' is overcome and system parameters are no longer scene-dependent compared to 2-D computer vision.

Density-constant indicates that projection density is only related to real volume of the projected object in the world, but no longer depends on sampling density of the sensor. Let the binary variable $l_{\mathcal{P}}(p_i, p'_j)$ denote whether p_i is projected to p'_j through the projection $\mathcal{P}$, and given

$$l_{\mathcal{P}}(p_i, p'_j) = 1 \quad \text{if} \quad p_i \xrightarrow{\mathcal{P}} p'_j \tag{9}$$

then, the pixel value ω'_j of p'_j on projection map $\mathcal{W}$ can be given as:

$$\omega'_j = d_i \sum_i l_{\mathcal{P}}(p_i, p'_j) \tag{10}$$

where ω'_j denotes the projection density on the point p'_j. Depth value d_i here is to make sure the projection density is also depth-invariant (constant). This operation is necessary because the density of points cloud captured by depth sensor varies a lot in different distance. Previous work in [22] also tries to get constant density of 3-D points cloud before the detection, which down-samples the original points cloud with real length 0.06 m, so points density no longer depends on their distances from the sensor. In this framework, the density-constance is achieved implicitly with DIDCP projection but more fast.

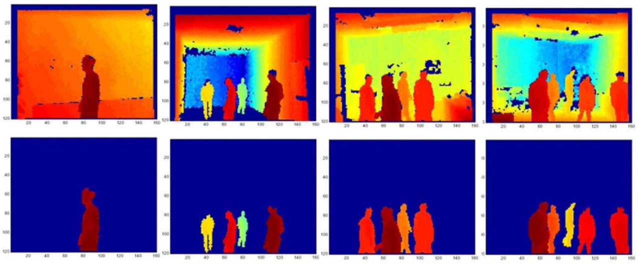


Fig. 4. Dataset samples and corresponding ground truth.

C. Localization and Segmentation

In the last module, based on the projection map $\mathcal{W}$, motion objects segmentation are achieved by following steps: First an isotropic Gaussian spatial filter is used to smooth the projection map $\mathcal{W}$ (Fig. 3(b)), to combine more spatial constraints, which is necessary to the pixel-level detection scheme. Secondly, a peaks detection operation is done to detect all salient peaks on map $\mathcal{W}$. Around each peak, an rectangle area b_k is obtained according to connectivity on the map $\mathcal{W}$. For example, as shown in Fig. 3(b) and (e), four rectangle areas around four peaks are obtained. Thirdly, 3-D points cloud projected to each rectangle b_k are segmented from original depth map $D(t)$, formulated as:

$$f_k = \{p_i | p'_j \in b_k, p_i \in D(t), l_{\mathcal{P}}(p_i, p'_j) = 1\} \tag{11}$$

The final motion segmentation is the set of $\{f_k\}$ corresponding to peaks areas $\{b_k\}$. An example of final segmentation result is shown in Fig. 3(c) corresponding to four rectangles in Fig. 3(b). At last, refined segmentation $\mathcal{F}$ is used to update coarse detection map ($\mathcal{M}(t) = \mathcal{F}$) periodically, which provides accurate base for the following cumulative RST detection, as well as inhibits accumulation of noise regions on $\mathcal{M}$ alone time (as shown in Fig. 3(a) and (c)).

IV. Experiments and Discussions

A. Evaluation

The experimental comparison is little bit harder for this work because few previous works focus on designing a specific motion detection method for depth data. As mentioned in the Section I, several related works do use RGB-D or depth data for motion detection, but are intrinsically different from motivation of this work. When it refers to evaluation metric,

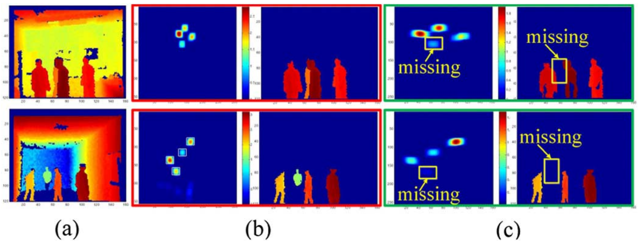


Fig. 5. Qualitative comparison of DIDCP and non-DIDCP (a) Ori. (b) DIDCP (c) non-DIDCP.

TABLE I
Average Accuracy in All Depth Videos Recorded in Four Scenes in Our Datasets According to the F_1 Metric in Eq. (12)

Methods	F_1 metric in several datasets			
	Scene 1	Scene 2	Scene 3	Scene 4
FHF + DIDCP	**0.930**	**0.901**	**0.880**	**0.895**
non-FHF + DIDCP	0.892	0.884	0.867	0.883
FHF + non-DIDCP	0.864	0.765	0.796	0.802
non-FHF + non-DIDCP	0.821	0.787	0.702	0.835
MOG [23]	0.720	0.502	0.462	0.486
CB1D [24]	0.782	0.523	0.507	0.563
FD + 3D Clustering [4]	0.624	0.650	0.623	0.583

previous motion detection evaluation prefers to use qualitative evaluation. In this paper a quantitative metric is designed: ground truth here is obtained by manually setting motion area in the scene, thus all 3-D points cloud inside the motion area are segmented without introducing background noise. Given ground truth $\mathcal{G}$ and final motion segmentation $\mathcal{F}$, we adopt the widely used metric F_1 which combines $precision$ and $recall$ to evaluate the quality of the segmentation. The F_1 measure is defined as follows:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{12}$$

B. Datasets and Settings

Experiments have been conducted on 20 depth videos captured by the Kinect sensor in 4 different scenes. The moving objects may move slow, fast or keep static occasionally, and may change speed suddenly. There is also much diversity in scenes like cluttered background, illumination changes, semi-occluded and full-occluded cases. Datasets are recorded by Microsoft Kinect sensor. In all experiments, system parameters are set as follows: $\tau_{pm} = 20$, $\alpha_0 = 0.15$ in Eq. (3), $\alpha_f = 5$ in Eq. (5), $\alpha_u = 0.25$, $\alpha_d = 0.5$ in Eq. (8). Pixel value of depth map is quantified to $0 - 255$ and points closer to sensor have bigger pixel value, and $d_{max} = 255$ in Eq. (3).

C. Comparison and Discussion

As there are few counterparts for comparison, in order to evaluate the performance of the proposed method, we organized the experimental comparison in two aspects:

On one hand, we emphasis on self-comparison by evaluating the performance of several modules in the framework. The RST motion detection results, with or without the following refinement modules (FHF and DIDCP modules) are all evaluated. The qualitative and quantitative results are given in Fig. 5 and Table I respectively. It can be seen from row 1-4 in Table I that FHF noise filtering module contributes to the accuracy in both DIDCP and non-DIDCP settings, and contributes more with non-DIDCP setting (row 3-4). This is because frequent-hopping regions in depth video brings amount of noise detection, but with DIDCP refinement, these noise can be partially removed. In order to give more intuitive observation of advantages using DIDCP, the qualitative comparision of DIDCP and non-DIDCP are given in Fig. 5. It can be observed that through DIDCP, both size and density of the projections got depth-invariance. In practice, without depth-invariance and density-constance, objects far away from the sensor usually are treated as background noise due to small size and low density on projection map (as the missing detection in Fig. 5(c)). On the contrary, noise regions near the sensor can be treated as moving objects. So without DIDCP, it is hard to choose suitable parameters to separate noise and true detection. But in our framework, all parameters can be set to fixed values in all scenes owe to DIDCP.

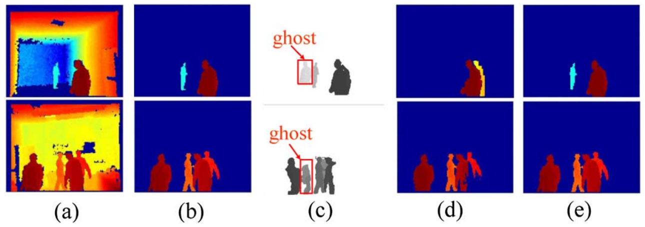


Fig. 6. Qualitative motion segmentation results comparison. (a) Ori. (b) GT (c) MOG [23] (d) FD [4] (e) Ours.

On the other hand, we also compared the proposed method with two classic motion detection schemes, background subtraction and frame difference. For background subtraction (BS), two classic background modeling methods MOG [23] and Codebook [24] are adopted for comparison here. However, this kind of methods perform badly when there are frequent walking and wondering in the scene because it is hard for this kind of methods to model satisfactory background model. If the scene is relatively simple and clean, like scene 1, they performs better than in other scenes. This can be seen from Table I. For frame difference (FD), the comparison comes from a similar application [4] to ours which directly uses FD for depth motion detection and performs 3-D points cloud clustering for refinement. It can be seen from Fig. 6(c) that MOG performs bad because of ‘ghosting problem’, and it is even server in indoor scenes. In Fig. 6(d) it can be seen that method in [4] can hardly get whole motion regions. It introduced many background points to final detection, as FD scheme can only get motion points, which could be on both background and foreground. But RST scheme only gets points on moving objects if not taking noise into account.

V. Conclusions

In this paper, we focus on finding a suitable motion detection scheme for depth data, without directly using classic intensity-based detection scheme or simply treating depth data as an auxiliary cue of RGB data. In general, the RST motion detection scheme is quite simple and fast for depth data, and enjoys good performance in various scenes. The fast processing rate makes it the firm foundation for the follow-up applications, and makes a realtime online system possible. Its limitation is that it cannot be used on the mobile platform, therefore it is more suitable for the indoor surveillance system, and current widely used depth sensors are also suitable for indoor applications. The future works are focused on combining RGB and depth data for more challenging tasks based on RST motion detection, such as multi-tracking and re-identification, treating the proposed framework as a fundamental work.

REFERENCES

[1] W. Choi, C. Pantofaru, and S. Savarese, "Detecting and tracking people using an RGB-D camera via multiple detector fusion," in *IEEE Int. Conf. Computer Vision Workshops, ICCVW 2011*, Nov. 2011, pp. 1076–1083.

[2] M. Luber, L. Spinello, and K. O. Arras, "People tracking in RGB-D data with on-line boosted target models," in *Int. Conf. Intelligent Robots and Systems, IROS2011*, Sep. 2011, pp. 3844–3849.

[3] Y. Park, V. Lepetit, and W. Woo, "Texture-less object tracking with online training using an RGB-D camera," in *IEEE Int. Symp. Mixed and Augmented Reality, ISMAR2011*, Oct. 2011, pp. 121–126.

[4] J. Han, E. J. Pauwels, P. M. Zeeuw, and P. H. N. With, "Employing a 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.

[5] C. Wang and H. Liu, "Salient-motion-heuristic scheme for fast 3D optical flow estimation using RGB-D data," in *IEEE Int. Conf. Acoustics, Speech and Signal Processing, ICASSP*, 2013, pp. 2272–2276.

[6] H. Liu and C. Wang, "Hierarchical data association and depth-invariant appearance model for indoor multiple objects tracking," in *IEEE Int. Conf. Image Processing, ICIP*, 2013, pp. 2635–2639.

[7] P. Wan, Y. Feng, G. Cheung, I. V. Bajic, and O. C. Au, "3-D motion estimation for visual saliency modeling," *IEEE Signal Process. Lett.*, vol. 20, no. 10, pp. 972–975, Oct. 2013.

[8] L. Spinello and K. O. Arras, "Leveraging RGB-D data: Adaptive fusion and domain adaptation for object detection," in *IEEE Int. Conf. Robotics and Automation, ICRA2012*, May 2012, pp. 4469–4474.

[9] L. Spinello and K. O. Arras, "People detection in RGB-D data," in *IEEE Int. Conf. Intelligent Robots and Systems, IROS2011*, Sep 2011, pp. 3838–3843.

[10] A. Bonnin, R. Borras, and J. Vitria, "A cluster-based strategy for active learning of RGB-D object detectors," in *IEEE Int. Conf. Computer Vision Workshops, ICCVW2011*, Nov. 2011, pp. 1215–1220.

[11] L. Spinello, K. O. Arras, R. Triebel, and R. Siegwart, "A layered approach to people detection in 3D range data," in *Proc. Twenty-Fourth AAAI Conf. Artificial Intelligence, AAAI2010*, Jul. 2010, pp. 1625–1630.

[12] R. Crabb, C. Tracey, A. Puranik, and J. Davis, "Real-time foreground segmentation via range and color imaging," in *IEEE Computer Society Conf. Computer Vision and Pattern Recognition Workshops, CVPRW2008*, Jun. 2008, pp. 1–5.

[13] L. Wang, C. Zhang, R. Yang, and C. Zhang, "TofCut: Towards robust real-time foreground extraction using a time-of-flight camera," in *3D Data Processing Visualization and Transmission, 3DPVT 2010*, Paris, France, May 2010.

[14] B. Ni, G. Wang, and P. Moulin, "RGBD-HuDaAct: A color-depth video database for human daily activity recognition," in *IEEE Int. Conf. Computer Vision Workshops, ICCVW2011*, Nov. 2011, pp. 1147–1153.

[15] B. Ni, N. C. Dat, and P. Moulin, "RGBD-camera based get-up event detection for hospital fall prevention," in *IEEE Int. Conf. Acoustics, Speech and Signal Processing, ICASSP2012*, Mar. 2012, pp. 1405–1408.

[16] C. Wang and H. liu, "A reliable moving human detection and tracking method based on accurate objects segmentation using range sensor," in *IEEE Int. Conf. Multi-sensor Fusion and Information Integration, MFI2012*, Hamburg, German, Sep. 2012, pp. 330–335.

[17] C. Wang and H. liu, "Robust visual tracking based on adaptive depth-color-cue integration using range sensor," in *IEEE Int. Conf. Multi-sensor Fusion and Information Integration, MFI2012*, Hamburg, Germany, Sep. 13–15, , pp. 336–343.

[18] E. Parvizi and Q. M. J. Wu, "Multiple object tracking based on adaptive depth," in *Can. Conf. Segmentation,Computer and Robot Vision, CRV2008*, May 2008, pp. 273–277.

[19] A. F. Bobick and J. W. Davis, "The representation and recognition of action using temporal templates," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[20] L. Cruz, "Kinect and RGBD images: Challenges and applications," in *SIBGRAPI Conf. Graphics, Patterns and Images Tutorials, SIBGRAPI2012*, Aug. 2012, pp. 36–49.

[21] H. Evan, R. X. feng, and F. Dieter, "RGB-D flow: Dense 3-D motion estimation using color and depth," in *IEEE Int. Conf. Robotics and Automation, ICRA2013*, May 2013, pp. 2276–2282.

[22] M. Munaro, F. Basso, and E. Menegatti, "Tracking people within groups with RGB-D data," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems, IROS2012*, Oct. 2012, pp. 2101–2107.

[23] Z. Zivkovic and F. V. Heijden, "Efficient adaptive density estimapion per image pixel for the task of background subtraction," *Patt. Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, 2006.

[24] F. Sanchez, L. Rubio, J. Diaz, and E. Ros, "Background subtraction model based on color and depth cues," *Mach. Vis. Applicat.*, pp. 1–15, Oct. 2013.