IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 29, 2018, accepted June 26, 2018, date of publication July 9, 2018, date of current version July 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2854262

3D Reconstruction With Time-of-Flight Depth Camera and Multiple Mirrors

TRONG-NGUYEN NGUYEN[1], HUU-HUNG HUYNH[2], AND JEAN MEUNIER[1]
[1]DIRO, University of Montreal, Montreal, QC H3T 1J4, Canada
[2]University of Science and Technology, The University of Danang, Danang 556361, Vietnam

Corresponding author: Trong-Nguyen Nguyen (nguyetn@iro.umontreal.ca)

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN-2015-05671.

ABSTRACT In order to extract more detailed features, many recent practical applications work with 3-D models instead of 2-D images. However, 3-D reconstruction usually requires either multiple cameras or a depth sensor and a turntable. This paper proposes an approach for performing a 3-D reconstruction using only one depth camera together with two or more mirrors. Mirrors are employed as virtual depth cameras placed at different positions. All measured depth data are provided in only one frame at each time. Significant depth distortion behind a mirror, which occurred with a standard time-of-flight depth sensor, is reduced by removing unreliable points and/or re-estimating better positions for these points. The experiments on easy-to-evaluate geometric objects show that the proposed approach could play a basic role in reconstructing intermediate 3-D object models in practical applications using only cheap devices.

INDEX TERMS Depth camera, depth distortion, mirror, reflection, space carving, time-of-flight.

I. INTRODUCTION

Reconstructing 3D models is an important process in a wide variety of fields including computer animation, medical imaging, computer graphics, etc. A typical strategy for that matter is using a depth camera combined with a turntable where the object is placed on (e.g. [1]). An obvious limitation is that such system is not appropriate to work on dynamic objects (e.g. a walking person) as well as requires prior knowledge such as rotation speed of the turntable. Other studies perform the shape-from-silhouette approach with the support of multiple cameras to retrieve the object visual hull. To overcome the main drawback of this method, i.e. missing concave regions in reconstructed model, other researchers employ a collection of depth sensors [2] and/or stereo cameras [3]. Considering the good accuracies obtained in these experiments, this paper proposes an approach which reduces the cost of devices as well as avoids unnecessary resource redundancies. In detail, only one depth sensor is required while the others are replaced by mirrors. This work guarantees obtaining depth information from different view points and does not need a synchronization solution as when using multiple depth sensors (e.g. a time server using NTP protocol in [2]). In addition, using multiple depth cameras may cause severe IR interferences.

There are wide varieties of depth sensors together with different estimation techniques such as stereo matching and ToF. In this work, a Microsoft Kinect 2, which uses ToF, is employed because of its cheap cost, good manufactured calibration, and good depth estimation. An approach for 3D reconstruction using mirrors has been performed in [4] with the previous generation of Kinect. The depth map provided by a Kinect 1 is measured based on a structured light technique. Such depth map thus contains less details compared with the one obtained by ToF [5]. Therefore, the Kinect 2 with ToF depth estimation is considered in our work. However, with ToF camera, we need to solve depth measurement ambiguities which occur from unwanted multiple reflections [6]. Such solutions usually require prior knowledge of the ToF camera characteristics (e.g. modulation frequency [7]) or performing low-level modifications as well as using additional devices (e.g. a projector [8]). This paper presents a simple solution for reducing such ambiguities based on some basic assumptions. Although this method may not solve all depth distortions, it still provides an obvious improvement versus the raw initialized model. It is important to recall that our approach focuses on providing an acceptable 3D model for practical applications instead of reconstructing a detailed object or absolutely removing all depth distortions. Using mirror for 3D reconstruction has been introduced in related

works such as [9] and [10]. Unlike our work, these studies focused on alternative implementations of silhouette-based reconstruction using multiple cameras.

Let us introduce briefly the way a ToF sensor measures depth information to provide an overview of possible depth distortions. A ToF depth sensor contains two important parts that are infrared (IR) emitter and receiver. A signal is emitted by the former device and is then received by the latter one. There are two common types of such signal: *high-speed pulse* and *continuous wave*. Distance between the sensor and an object point is approximated as a half of traveled length based on time delay of the pulse or the phase shift between retrieved and emitted waves. Because of this measurement way, if such signal travels in a multipath trajectory, the obtained depth may be significantly changed. This scenario occurs in our configuration with mirrors under several conditions. The details of such depth distortion and our solution are presented in next two sections. Because the Kinect 2 employs a continuous wave modulation, the remaining content of this paper only mentions this technique.

II. DEPTH DISTORTION BEHIND A MIRROR

Let us consider a scenario using only one mirror without any environment reflection (e.g. a white wall), an overview of possible returned signals corresponding to a pixel in the depth image is illustrated in Fig. 1, in which C and C_m are the real and mirrored camera centers, P and P_m are the considered point and its reflection behind the mirror, P_K is the estimated result of the Kinect, and M is the point where the emitted signal touches the mirror. The term *mirrored point* indicates the image (behind a mirror) of a real point.

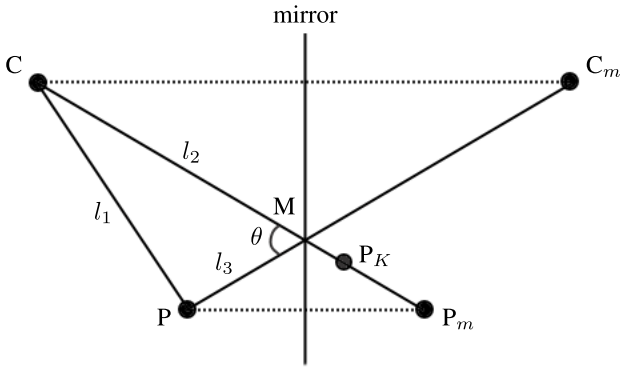


FIGURE 1. Depth estimation of a point in front of a mirror and distortion of corresponding mirrored point depth.

As mentioned in the previous section, the distance between the depth sensor and a point is approximated by half of the traveled distance of the signal, i.e.

$$distance(\mathrm{C}, \mathrm{P}) = \frac{1}{2}(\|\overrightarrow{\mathrm{CP}}\| + \|\overrightarrow{\mathrm{PC}}\|) = l_1 \qquad (1)$$

With the reflected point P_m, the trajectory of the corresponding signal is $\overrightarrow{\mathrm{CM}} + \overrightarrow{\mathrm{MP}} + \overrightarrow{\mathrm{PM}} + \overrightarrow{\mathrm{MC}}$, thus the expected distance is $l_2 + l_3$. The value measured by the Kinect, however, is significantly decreased, and a unreliable point P_K is obtained instead of the true point P_m. This distortion occurs because of another signal, which travels along the following way $\overrightarrow{\mathrm{CP}} + \overrightarrow{\mathrm{PM}} + \overrightarrow{\mathrm{MC}}$. We empirically found that if there is a significant difference of length between these two trajectories, the obtained depth value is approximated by the shorter one. This is indicated by the term *geometrical distortion* in this paper. In the other case, i.e. if the difference is small, the measured depth is affected by multipath ambiguity. We use the term *phase distortion* to denote this effect. In Fig. 1, the estimated distance between C and P_m becomes

$$distance(\mathrm{C}, \mathrm{P}_m) = \frac{1}{2}(l_1 + l_2 + l_3) \qquad (2)$$

Due to this distortion, a shape behind a mirror could be very different compared with the original one (e.g. a planar surface becomes curved, see Fig. 2 and Appendix). Thanks to the relation between the camera and the mirror, the estimated distance between C and P_K can be used to approximate a better position of P_m.

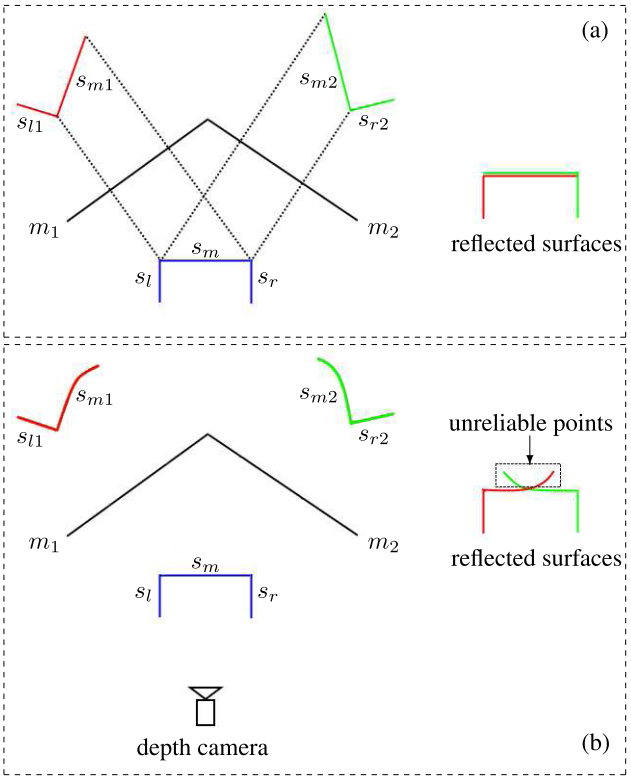


FIGURE 2. Practical situation of two reflections with mirrors m_1 and m_2: (a) physical reflection of an object in two mirrors, (b) depth information measured by a ToF sensor. The camera is placed in front of the object and the 3 illustrated object parts (i.e. the 3 surfaces s_l, s_m, and s_r) are not directly seen by the depth sensor (e.g. occluded by front parts (not shown) of the object). The mirrored surfaces of s_r in m_1 as well as s_l in m_2 do not appear in the figure because the depth camera cannot see them due to occlusions.

First, the equation of the mirror surface is determined using some markers placed on it, with their 3D coordinates measured by the Kinect. The position of M is then localized

by intersecting the direction $\overrightarrow{\mathrm{CP}_K}$ with the mirror plane to get the length l_2. Let us consider the triangle $\triangle$CMP. The angle θ is determined based on the two vectors $\overrightarrow{\mathrm{MC}}$ and $\overrightarrow{\mathrm{C}_m\mathrm{M}}$. With the estimated depth of P_K, we have:

$$l_1 + l_2 + l_3 = 2\|\overrightarrow{\mathrm{CP}_K}\| \quad (3)$$

$$\Leftrightarrow l_1 + l_3 = 2\|\overrightarrow{\mathrm{CP}_K}\| - l_2 = k \quad (4)$$

The law of cosines in the triangle $\triangle$CMP leads to the relation

$$l_1^2 = l_2^2 + l_3^2 - 2l_2 l_3 cos\theta \quad (5)$$

By combining eq. (4) and (5), the length l_3 is obtained as

$$l_3 = \frac{1}{2} \cdot \frac{l_2^2 - k^2}{l_2 \cos\theta - k} \quad (6)$$

Finally, the point P_m along the straight line CM can be localized together with its real point P. This solution will be tested in Section IV-A.

In practical situations, e.g. reconstructing an object with several mirrors, the depth measurement is slightly different. The described depth distortion, however, is useful for removing unreliable measured points. The details of our practical configuration together with the reconstruction of object's point cloud are presented in the next section.

III. UNRELIABLE POINT REMOVAL

Let us consider a practical scenario with a Kinect and two mirrors as in Fig. 2. According to geometrical optics, the object model can be formed by combining the front part, which is directly seen by the depth sensor, and reflected parts of the back through corresponding mirrors. The 3D cloud measured by a ToF depth sensor, however, contains a lot of unreliable points due to the *geometrical* and *phase* distortions defined in Section II.

A. GEOMETRICAL DISTORTION

With a given 3D point P on the back of the object and two mirrors m_1 and m_2 as in Fig. 3, the camera provides depth measurements of two mirrored points P_1 and P_2. Because the depth camera C does not directly see the point P, the measured distances of P_1 and P_2 are expected to be $l_1 + l_2$ and $l_3 + l_4$, respectively. The obtained values, however, are only exact for the point P_1, while the corresponding depth of P_2 decreases to P_{K2} with a significant deviation. This distortion occurs because there are two returned signals in the direction $\overrightarrow{\mathrm{P}_2\mathrm{C}}$ with traveled length $2l_3 + 2l_4$ and $l_1 + l_2 + l_3 + l_4$. The depth information is thus estimated based on the shorter way. In summary, a 3D point P, which is not seen by the depth camera, can create two mirrored points P_1 and P_2 containing at least one reliable point (e.g. P_1 in Fig. 3 because $l_1 + l_2 < l_3 + l_4$).

B. PHASE DISTORTION

We empirically found that most mirrored points were affected by geometrical distortion, thus our restoration approach for the other distortion is presented as an additional post-processing (see Section III-D and Appendix).

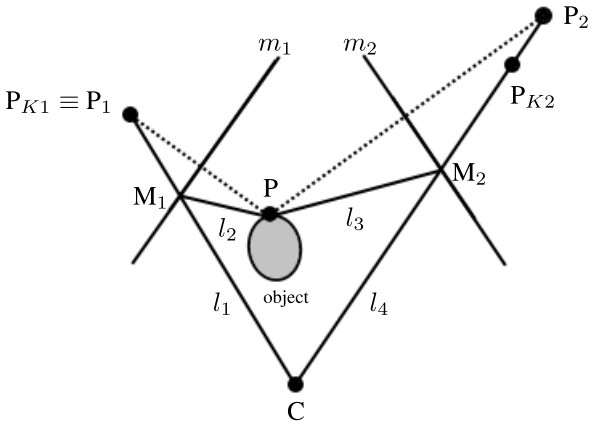


FIGURE 3. Depth measurement of a 3D point P in two mirrors m_1 and m_2. Let us note that P is not seen by the depth camera C. Two points P_{K1} and P_{K2} are Kinect measured points of P_1 and P_2, respectively.

C. RECONSTRUCTING RAW POINT CLOUD

In the scenario illustrated in Fig. 2, the raw estimated point cloud of the object is obtained by combining two components:

- Points (in front of the object) which are directly seen by the depth camera (not shown in the figure)
- Points (on the back of the object) which are reflected through corresponding mirrors m_1 and m_2

First, a 3D region of the reconstructed object is defined. Let us consider a point P in the cloud mentioned above. If P comes from the first component, i.e. P can be directly seen by the depth sensor, it is a *reliable point* lying on the object surface. If the camera sees a mirrored point P_m of P in an arbitrary mirror, the measured depth of P_m is significantly reduced, but P_m is always behind the mirror. The reflection of P_m is thus in front of this mirror. Our experiments (see Section IV-A) show that the distance between this reflected point and the corresponding mirror is very small, thus P_m can be easily removed by checking if its reflection is outside of the defined 3D object region. Therefore, there remains two cases which need to be focused on: a 3D point can be seen in only one mirror (e.g. point on surfaces s_l and s_r in Fig. 2) or in both mirrors (e.g. point on s_m).

In the first case, the signal corresponding to such point always travels along the shortest way, thus the reflected point is reliable. In the second one, it is important to recall that we have proved that a 3D point, which is not seen by the depth camera, can create two mirrored points containing at least one reliable point. Our goal thus becomes simpler since we just need to remove these false-estimated points.

Our idea for deciding a point in the raw reflected cloud to be removed or be kept is quite simple. Assume that a point P in cloud is recovered (i.e. reflected) from a mirrored point P_i through a mirror m_i with $i \in \{1, 2\}$, the corresponding mirrored point P_j of P in the other mirror is localized. According to the given coordinates of the camera center C, the point P is kept in the cloud if $distance(\mathrm{C}, \mathrm{P}_i) \leq distance(\mathrm{C}, \mathrm{P}_j)$, and otherwise is removed. This idea can be proved with the

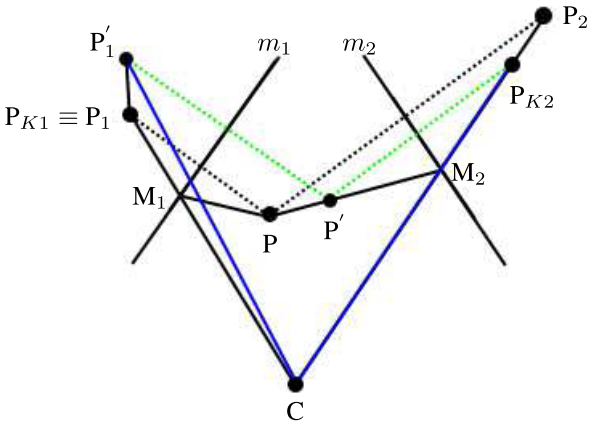


FIGURE 4. Reconstruction of a bad-measured Kinect point and its images corresponding to the two mirrors.

illustration of Fig. 4 (extended from Fig. 3). Let us assume that P$^{'}$ is the reflected point of P$_{K2}$ through m_2, and P$_1^{'}$ is the image of P$^{'}$ in m_1. As presented in Section II, the distance between camera center C and estimated point P$_{K2}$ satisfies the following condition:

$$\begin{aligned} 2\|\mathrm{CP}_{K2}\| &= \|\mathrm{CP}_1\| + \|\mathrm{CP}_2\| \\ \Rightarrow \|\mathrm{CP}_{K2}\| &= \|\mathrm{CP}_1\| + \|\mathrm{P}_2\mathrm{P}_{K2}\| \end{aligned} \tag{7}$$

The three segments P$_2$P$_{K2}$, PP$^{'}$, and P$_1$P$_1^{'}$ have the same length, thus eq. (7) is equivalent to

$$\|\mathrm{CP}_{K2}\| = \|\mathrm{CP}_1\| + \|\mathrm{P}_1\mathrm{P}_1^{'}\| \tag{8}$$

According to the triangle inequality [11] in $\triangle$CP$_1$P$_1^{'}$, we have

$$\|\mathrm{CP}_1\| + \|\mathrm{P}_1\mathrm{P}_1^{'}\| > \|\mathrm{CP}_1^{'}\| \tag{9}$$

By combining eq. (8) and (9), the length of CP$_{K2}$ is always greater than the distance between C and P$_1^{'}$. In other words, a point in the raw reflected cloud can be considered to be a reliable or unreliable one by checking distances between the camera center to mirrored points behind the two mirrors.

In summary, given a 2D array *pts* (depth image) of 3D points measured by the Kinect, two mirror plane equations mir_1 and mir_2, position of camera center C, and a predefined 3D object region of interest *reg*, our algorithm for reconstructing a point cloud representing an object is as the Algorithm 1.

D. INCREASING POINT DENSITY BY SPACE CARVING

An obvious limitation of the reconstructed object point cloud in Section III-C is that the farther the object is from a mirror, the larger is the distance between two neighbor 3D points corresponding to this mirror in the obtained cloud. To increase the density of such points, the space carving approach can be applied together with the algorithm described in the previous section. Given a voxel volume V and input components of the algorithm of unreliable point removal, the overall processing is performed as the Algorithm 2.

In practical applications as well as when working on specific objects, some additional operations can be integrated into the two presented algorithms to improve reconstruction accuracy such as color filtering and defining object boundary.

As mentioned in the end of Section III-B, a post-processing could be applied to improve the model quality. This processing requires a correspondence of two mirrored points which are created based on one real 3D point, thus it is appropriate to apply the post-processing in the presented space carving approach. This stage can be easily performed based on the eq. (13) (see Appendix). However, let us recall that most estimated points are not affected by this distortion, thus this post-processing is not necessary if our goal is to provide an acceptable intermediate model for practical applications. Moreover, the method in Section III-C could be integrated into real-time systems while it takes much time to perform the space carving technique.

Algorithm 1 Unreliable Point Removal

```
procedure GetCloud(pts, mir1, mir2, C, reg)
    cloud ← null
    for each point P in pts do
        if P inside reg then
            cloud ← push(P)
        else if P behind mir1 then
            Pr ← reflect(P, mir1)
            if Pr not inside reg then
                continue                    ▷ check another point
            end if
            P2 ← reflect(Pr, mir2)
            if CP < CP2 then
                cloud ← push(Pr)            ▷ reliable point
            end if
        else if P behind mir2 then
            Pr ← reflect(P, mir2)
            if Pr not inside reg then
                continue                    ▷ check another point
            end if
            P1 ← reflect(Pr, mir1)
            if CP < CP1 then
                cloud ← push(Pr)            ▷ reliable point
            end if
        end if
    end for
    return cloud                            ▷ Return object point cloud
end procedure
```

IV. EXPERIMENTAL RESULTS

This section demonstrates the results of solving depth distortion in the cases of using one and two mirrors. The former experiment was performed by comparing distances between a real 3D point and its raw reflected point as well as the one relocated by our proposed approach [Section II, eq. (6)]. In order to obtain a high generalization, a set of points, which consists of markers located on a small flat board, was employed to calculate the distance deviation instead of using

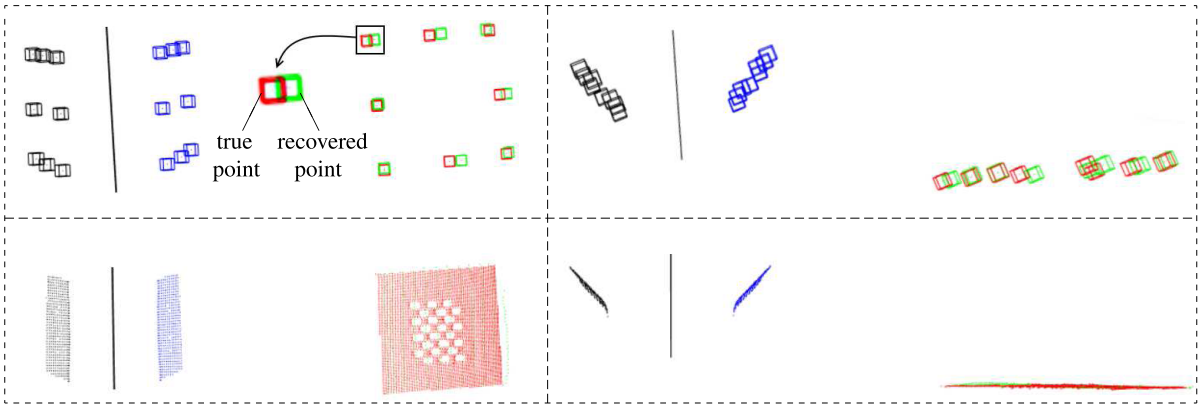


FIGURE 5. From left to right: tested scenes with top view and side view. The top row shows result of solving one-mirror distortion on a set of 8 markers lying on a flat board of size 30 *cm* × 30 *cm* while the second row is the result of recovering all points in the board. In each sub-figure, the mirror is represented by a straight segment, and each marker position is shown as a point bounded by a cube-shape-wireframe for better visualization. The red points are real 3D markers and the black ones are their mirrored points determined in the captured depth map. The blue and green markers respectively indicate the reflection of mirrored points and our recovered ones. Notice that in the bottom-left sub-figure, the board contains some holes because there was a chessboard on the surface, and processing of black pixels was avoided due to low-reflection of black regions.

Algorithm 2 Space-Carving-Based Reconstruction

procedure SpaceCarving(V, pts, mir_1, mir_2, C)
  $th \leftarrow th_0$ ▷ define a threshold of distance deviation
  for each voxel P in V **do**
    $pixel \leftarrow$ project(P) ▷ 3D to 2D projection
    $P_K \leftarrow$ get3Dpoint(pts, $pixel$) ▷ 3D Kinect point
    if $\|CP_K - CP\| < th$ **then**
      $V \leftarrow$ keep(P)
      continue ▷ check next voxel
    else
      $P_1 \leftarrow$ reflect(P, mir_1)
      $P_2 \leftarrow$ reflect(P, mir_2)
      $pixel_1 \leftarrow$ project(P_1)
      $pixel_2 \leftarrow$ project(P_2)
      $P_{K1} \leftarrow$ get3Dpoint(pts, $pixel_1$)
      $P_{K2} \leftarrow$ get3Dpoint(pts, $pixel_2$)
      if $CP_1 < CP_2$ and $\|CP_{K1} - CP_1\| < th$ **then**
        $V \leftarrow$ keep(P)
        continue ▷ check next voxel
      end if
      if $CP_2 < CP_1$ and $\|CP_{K2} - CP_2\| < th$ **then**
        $V \leftarrow$ keep(P)
        continue ▷ check next voxel
      end if
    end if
    $V \leftarrow$ remove(P)
  end for
  return V ▷ Return voxel volume
end procedure

only one point at a time, and the board was also placed in front of the mirror at different tilt angles. The latter experiment was evaluated by fitting a surface based on raw reconstructed point cloud as well as voxel volume and then estimating

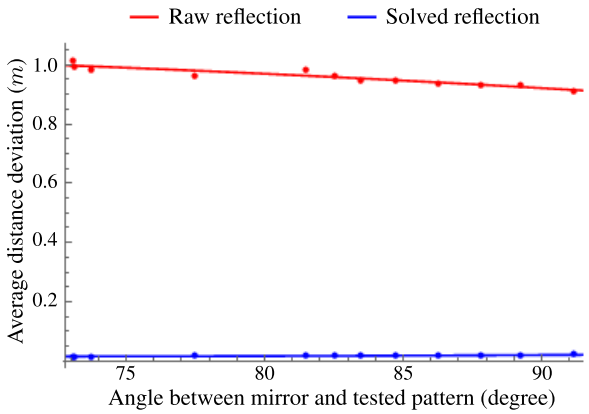


FIGURE 6. Measured reflection errors before and after applying our solution, in which deviation values were decreased about 53 times (0.959 and 0.018 on average, respectively).

the corresponding error according to prior knowledge of the object shape. In order to simplify the calculation, this work employed two simple objects including a flat board and a cylinder. The testing process was also performed with different distances between the object and the two mirrors.

A. SOLVING DEPTH DISTORTION WITH ONE MIRROR

For each real marker P on a flat pattern placed in front of the mirror, our processing flow in this experiment consists of the following steps (see Fig. 1): (a) reflecting P to get the true position of its image P_m behind the mirror, (b) determining the corresponding measured Kinect point P_K, (c) re-estimating a corrected point P_C of P_K ($P_C \equiv P_m$ in the ideal case), and (d) calculating *distance*(P, P_{Cm}) and *distance*(P, P_{Km}) where P_{Cm} and P_{Km} are reflections of P_C and P_K through the mirror, respectively. In summary, a set of n corners provides n pairs of such values. Finally, average distances are compared together to evaluate the proposed solution.

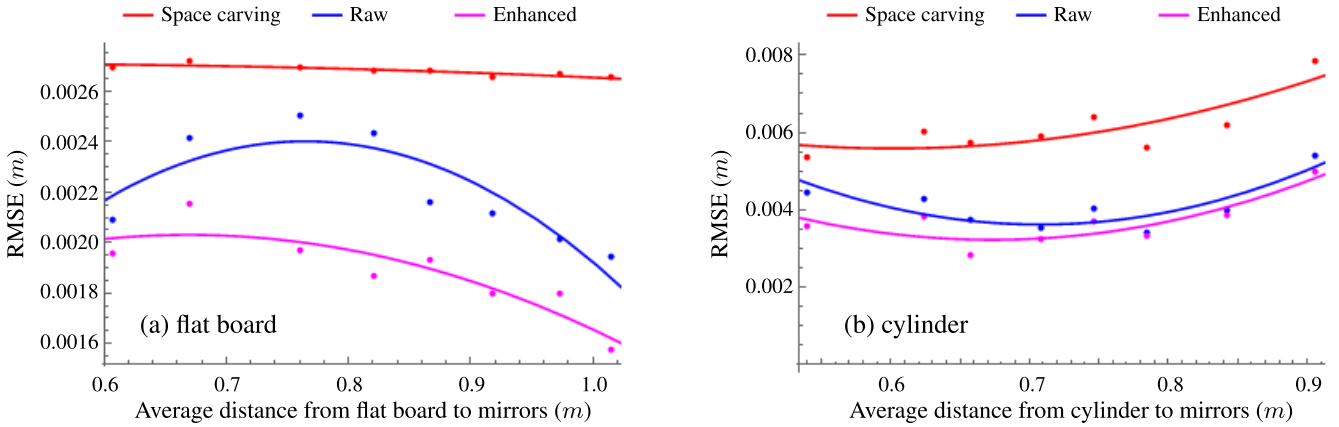


FIGURE 7. Reconstruction errors of three types of clouds: raw, distortion removal, and space carving. The cylinder radius, which was manually measured, was 150 *mm* and the average radius of the reconstructed clouds was 147.4 *mm*.

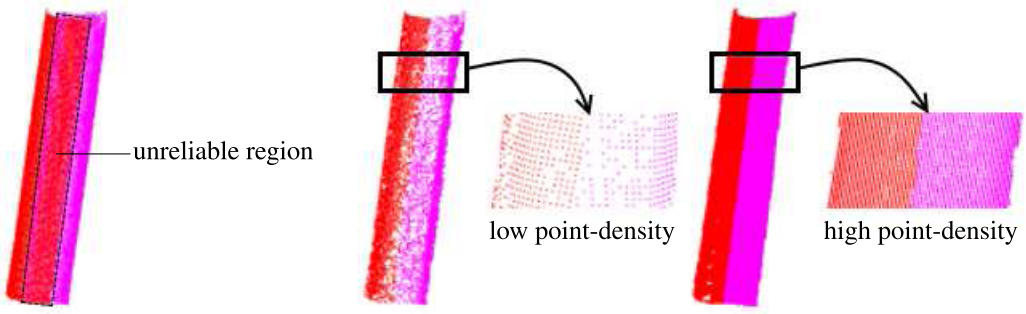


FIGURE 8. From left to right: raw cloud, cloud after removing unreliable points, and space carving. Points directly seen by the Kinect are not shown in this figure since they are not affected by any of mentioned distortions.

Processing a set of markers as well as all points on the flat board are illustrated in Fig. 5. In the top row, the recovered points were almost at their corresponding true points though there were significant distance deviations in the Kinect measurement. In the bottom row, the points provided by our solution and the true points also fit a plane. The small position deviations of our recovered points in Fig. 5 come from the following reasons. First, the mirror was not an absolute planar surface, a point displacement might thus occur. Besides, this experiment was performed on raw captured data without any improvement (e.g. depth smoothing or enhancement). In addition, different 3D positions could be mixed into one point by Kinect due to the low resolution of the IR camera (512×424 pixels). To overcome these limitations, a depth improvement procedure could be applied (e.g. [12]), and a high-resolution camera could also be employed as an additional view (e.g. mapping between color and depth cameras of Kinect to investigate a higher density of recovered points).

Figure 6 shows experimental results corresponding to 12 different pattern poses in front of the mirror. It is obvious to see that distance deviations between true points and reflected ones were significantly reduced by our proposed solution.

B. REDUCING DISTORTION IN THE CASE OF TWO MIRRORS

In this experiment, the angle between two mirrors was about 120 degrees. The distance from a tested object to mirrors was defined as the mean of all distances between the final reconstructed object points and the two mirrors. Given knowledge about the object shape (either plane or cylinder), the evaluation was performed by fitting a surface based on RANSAC [13] and estimating root-mean-square errors (RMSE). Our experimental results when testing these two objects are shown in Fig. 7. Fitting errors were reduced after applying our approach on raw reconstructed point cloud. Notice that the error corresponding to the space carving method was always larger than the two others because of object's thicker borders. Measured errors were less than 1 *cm*. The cylinder radius was 150 *mm*.

A visual comparison of reconstructed point clouds of a cylinder before and after performing our method is also presented in Fig. 8. The proposed approach removed a large number of noisy points from the raw reconstructed models.

A visualization of point clouds representing a human body with different postures is also presented in Fig. 9. These clouds are reconstructed by the algorithm of unreliable point removal presented in Section III-C.

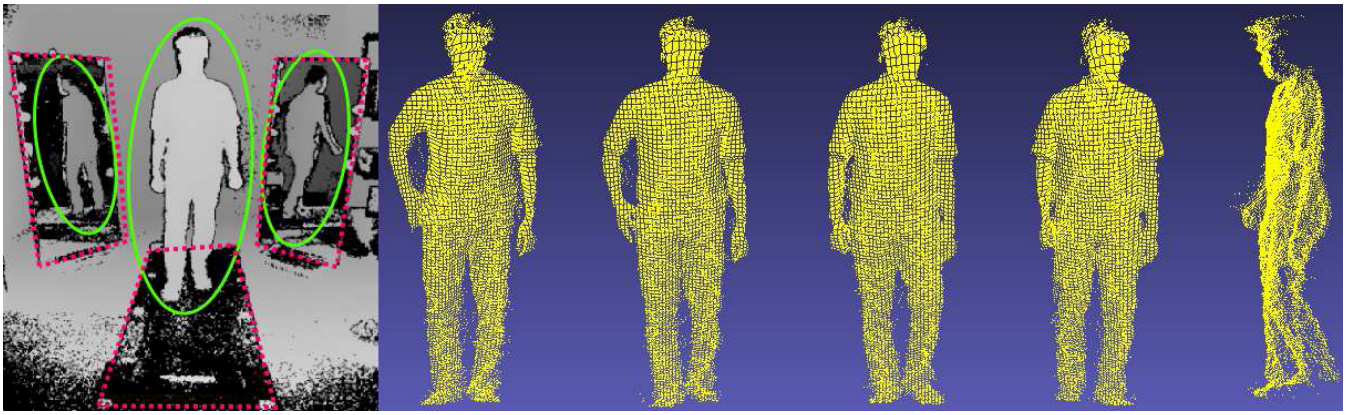


FIGURE 9. Left: our realistic setup of a 3D reconstruction system for the task of gait analysis including a treadmill and two mirrors (highlighted by dotted red rectangles). Right: reconstructed point clouds corresponding to 4 nearby poses of a walking gait, and the last cloud is the 4^{th} one seen from side view. These point clouds were acquired at 13 fps using the computer mentioned in Section IV-C. These clouds are extracted from our huge dataset (nearly 100,000 postures) of human walking gaits that is available at http://www.iro.umontreal.ca/~labimage/GaitDataset. Details of data acquisition is clearly described in [14].

The figure shows that it is reasonable to expect that our approach could be used to provide intermediate (real-time) models in systems which process 3D information. A huge dataset (nearly 100,000 samples) of such point clouds representing human walking gaits performed on a treadmill is also available online.[1]

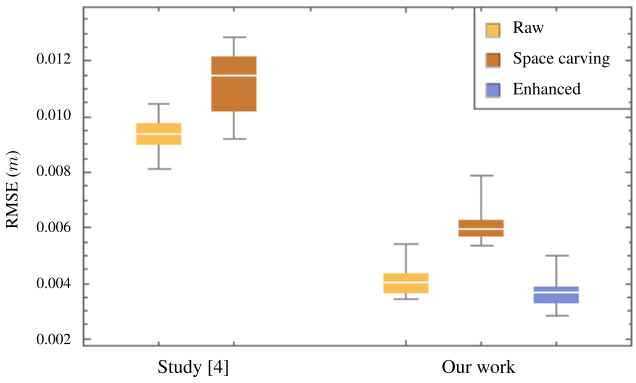


FIGURE 10. Reconstruction errors corresponding to our work and the study [4]. The comparison is performed on three types of clouds: raw reflection, distortion removal (only our work), and space carving.

Figure 10 shows a comparison of reconstruction error (RMSE) between our system and the similar setup in [4], where a Kinect 1 with structured-light depth estimation was employed instead of a Kinect 2. Both reconstructions were performed on the same cylinder, and the statistical information presented in Fig. 10 was calculated on different distances between the cylinder and mirrors. The study [4] also provided point clouds corresponding to raw reflection and space carving. Notice that the depth distortion, which has been dealt with in our study, does not occur in the setup [4]. This comparison shows that our system with a Kinect 2 provided better point clouds. This is because the depth map of Kinect 1 is noisier and has less details compared with the next generation [5].

Finally, let us note that our algorithm makes a trade off between the simplicity of processing flow and a constraint in scene configuration. For example, in the case where the object in Fig. 2(b) is placed nearer the mirror m_1 (large deviation of distances between the object and each mirror), the proposed algorithm might fail to reconstruct the surface s_r from s_{r2}. In detail, the idea of checking point reliability in Section III-C is sometimes not appropriate for object points which are seen in only one mirror. This drawback, however, could be easily avoided by placing the object near the center of a *balanced* (approximately) configuration. All our experiments satisfy this constraint without any complicated additional processing. In addition, we should notice that if the setup contains more than 2 mirrors, the depth distortion would be more complicated due to the increasing number of unwanted reflections. Such setup may even reduce the quality of reconstructed 3D point clouds.

C. IMPLEMENTATION

Our system was built on a medium-strength laptop using C++ (non-optimized code) and the two open source libraries OpenCV [15] and Point Cloud Library [16]. All Kinect depth images in our experiments were captured with a resolution of 512×424 pixels. The process of reconstructing raw point cloud (as in Section III-C) was performed with an average speed of 0.07 seconds per frame. This processing time could be significantly reduced with the support of parallel (and multi-threading) programming. The proposed approach thus could be expected to be appropriate for creating a real-time reconstruction system.

V. CONCLUSION

Throughout this paper, a new approach for reconstructing a 3D object using only one ToF depth sensor together with

[1] http://www.iro.umontreal.ca/~labimage/GaitDataset

mirrors has been presented. An overview of depth distortion occurring with one and two mirrors and corresponding solutions are also mentioned. Beside avoiding the problem of synchronization (i.e. all depth data from different viewing directions are provided by only one Kinect) and possible severe IR interferences caused by multiple depth cameras, our method can be applied on dynamic objects (e.g. a walking person). The experiments and evaluations show that the proposed approach improves significantly the quality of Kinect depth estimation. In summary, our method can serve as a basic system for cheap 3D reconstruction as well as for providing intermediate object models in practical applications. In future work, we intend to use the reconstructed data for various applications, such as human gait analysis and assessment.

APPENDIX
ANALYSIS OF PHASE DISTORTION

According to [8], the depth of a point is measured based on the phase delay of optical trajectories as

$$d = \frac{c\varphi}{4\pi f} \tag{10}$$

where the constant c is the speed of light, f is the modulation frequency of the IR emitter, and φ is the phase shift. The measured phase shift in the case of multipath interference is

$$\widetilde{\varphi} = tan^{-1}\Big(\frac{\alpha_0 sin\varphi_0 + \sum_{i=1}^{K} \alpha_i sin\varphi_i}{\alpha_0 cos\varphi_0 + \sum_{i=1}^{K} \alpha_i cos\varphi_i}\Big) \tag{11}$$

where K is the number of signals returning to the corresponding pixel of the considering point, and α denotes the amplitude.

In our setup, there are only two signal paths: the direct way which provides a true depth and the indirect one which affects this value (e.g. the two mentioned trajectories in Section III-A). Besides, we also assume that these two signal amplitudes are similar because they touch the object only once. Eq. (11) thus could be simply approximated as

$$\widetilde{\varphi} = tan^{-1}\Big(\frac{sin\varphi_D + sin\varphi_I}{cos\varphi_D + cos\varphi_I}\Big) \tag{12}$$

where the subscripts D and I denote parameters of the direct and indirect signals, respectively.

By combining eq. (10) and (12), the relation between the measured depth d_K and the two elementary traveled ways d_D and d_I is

$$d_K = \frac{1}{2}\big(d_D + d_I\big) + k\frac{c}{4f} \tag{13}$$

where k is an integer. By performing some experiments, we found that 0 is the most appropriate value of k. It means that in the case of phase distortion, the measured depth could be approximated as a quarter of the total traveled lengths of the two elementary signals.

ACKNOWLEDGMENT

The authors would like to thank professor Sébastien Roy, 3D Vision Lab, DIRO, for his useful comments on this work.

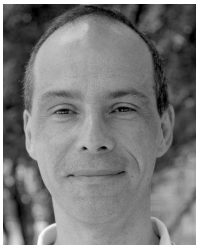
REFERENCES

[1] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3D full human bodies using Kinects," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 4, pp. 643–650, Apr. 2012.
[2] E. Auvinet, J. Meunier, and F. Multon, "Multiple depth cameras calibration and body volume reconstruction for gait analysis," in *Proc. 11th Int. Conf. Inf. Sci., Signal Process. Appl. (ISSPA)*, Jul. 2012, pp. 478–483.
[3] G. K. M. Cheung, S. Baker, and T. Kanade, "Visual hull alignment and refinement across time: A 3D reconstruction algorithm combining shape-from-silhouette with stereo," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2003, pp. II-375–II-382.
[4] T.-N. Nguyen, H.-H. Huynh, and J. Meunier, "Matching-based depth camera and mirrors for 3D reconstruction," *Proc. SPIE*, vol. 10666, pp. 10666-1–10666-10, Apr. 2018, doi: 10.1117/12.2304427.
[5] O. Wasenmüller and D. Stricker, *Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision*. Cham, Switzerland: Springer, 2017, pp. 34–45.
[6] D. Freedman, Y. Smolin, E. Krupka, I. Leichter, and M. Schmidt, *SRA: Fast Removal of General Multipath for ToF Sensors*. Cham, Switzerland: Springer, 2014, pp. 234–249.
[7] A. A. Dorrington, J. P. Godbaz, M. J. Cree, A. D. Payne, and L. V. Streeter, "Separating true range measurements from multi-path and scattering interference in commercial range cameras," *Proc. SPIE*, vol. 7864, p. 786404, Jan. 2011.
[8] N. Naik, A. Kadambi, C. Rhemann, S. Izadi, R. Raskar, and S. B. Kang, "A light transport model for mitigating multipath interference in time-of-flight sensors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 73–81.
[9] J. Gluckman and S. K. Nayar, "Catadioptric stereo using planar mirrors," *Int. J. Comput. Vis.*, vol. 44, no. 1, pp. 65–79, Aug. 2001, doi: 10.1023/A:1011172403203.
[10] X. Ying, K. Peng, Y. Hou, S. Guan, J. Kong, and H. Zha, "Self-calibration of catadioptric camera with two planar mirrors from silhouettes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1206–1220, May 2013.
[11] M. A. Khamsi and W. A. Kirk, *An Introduction to Metric Spaces and Fixed Point Theory*, vol. 53. Hoboken, NJ, USA: Wiley, 2011.
[12] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov, "Temporal filtering for depth maps generated by Kinect depth camera," in *Proc. 3DTV Conf., True Vis.-Capture, Transmiss. Display 3D Video (3DTV-CON)*, May 2011, pp. 1–4.
[13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[14] T.-N. Nguyen and J. Meunier, "Walking gait dataset: Point clouds, skeletons and silhouettes," Dept. d'informat. Recherche Opérationnelle, Univ. Montreal, Montreal, QC, Canada, Tech. Rep. 1379, Apr. 2018. [Online]. Available: http://www.iro.umontreal.ca/~labimage/GaitDataset/dataset.pdf
[15] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Cambridge, MA, USA: O'Reilly, 2008.
[16] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Shanghai, China, May 2011, pp. 1–4.

TRONG-NGUYEN NGUYEN received the B.Sc. degree in information technology from Danang University of Science and Technology, Danang, Vietnam, in 2012, and the M.Sc. degree in computer science from The University of Danang, Danang, in 2015. He is currently pursuing the Ph.D. degree with the University of Montreal, Montreal, QC, Canada.

From 2014 to 2015, he was a Research Assistant with the Vision Laboratory, Danang University of Science and Technology. In summer 2014, he was an Intern with the Image Processing Laboratory, DIRO, University of Montreal. His research focused on hand gesture recognition and human gait analysis. His current research interests include human gait analysis and deep learning.

HUU-HUNG HUYNH received the B.S. degree in computer science from the Hanoi University of Science and Technology, Vietnam, in 1998, the M.Sc.A. degree in computer science in 2003, and the Ph.D. degree in computer science from the Aix-Marseille University in 2010. He is currently a Lecturer with the University of Science and Technology, The University of Danang, Danang, Vietnam. His research interests include computer vision and health care systems.



JEAN MEUNIER received the B.S. degree in physics from the University of Montreal (UdeM), Canada, in 1981, the M.Sc.A. degree in applied mathematics in 1983, and the Ph.D. degree in biomedical engineering from Polytechnique Montreal in 1989.

In 1989, after his Ph.D. with the Montreal Heart Institute, he joined DIRO, UdeM, where he is currently a Full Professor and also a Regular Member of the Biomedical Engineering Institute. His current research interests include computer vision and its applications to medical imaging and health care.