IR Stereo Kinect: Improving Depth Images by Combining Structured Light with IR Stereo

Faraj Alhwarin Alexander Ferrein Ingrid Scholl
Mobile Autonomous Systems & Cognitive Robotics Institute
FH Aachen University of Applied Sciences
Aachen, Germany
Email: {alhwarin, ferrein, scholl}@fh-aachen.de

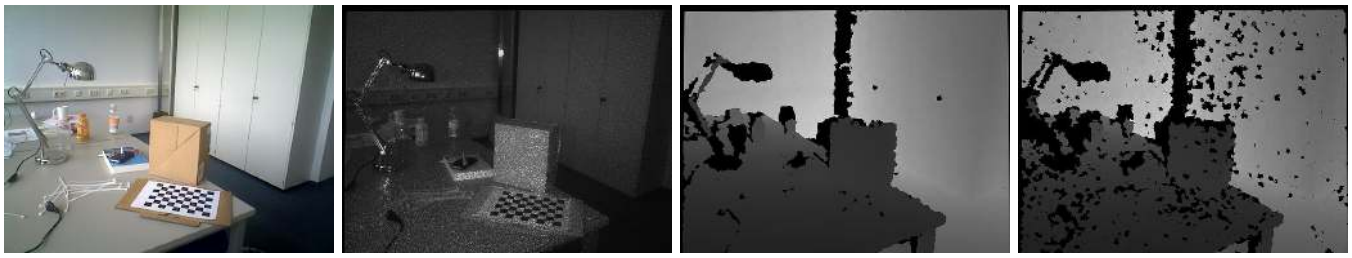
***Abstract*—RGB-D sensors such as the Microsoft Kinect or the Asus Xtion are inexpensive 3D sensors. A depth image is computed by calculating the distortion of a known infrared light (IR) pattern which is projected into the scene. While these sensors are great devices they have some limitations. The distance they can measure is limited and they suffer from reflection problems on transparent, shiny, or very matte and absorbing objects. If more than one RGB-D camera is used the IR patterns interfere with each other. This results in a massive loss of depth information. In this paper, we present a simple and powerful method to overcome these problems. We propose a stereo RGB-D camera system which uses the pros of RGB-D cameras and combine them with the pros of stereo camera systems. The idea is to utilize the IR images of each two sensors as a stereo pair to generate a depth map. The IR patterns emitted by IR projectors are exploited here to enhance the dense stereo matching even if the observed objects or surfaces are texture-less or transparent. The resulting disparity map is then fused with the depth map offered by the RGB-D sensor to fill the regions and the holes that appear because of interference, or due to transparent or reflective objects. Our results show that the density of depth information is increased especially for transparent, shiny or matte objects.**

I. Introduction

The introduction of RGB-D cameras such as the Microsoft Kinect or the Asus Xtion did not only influence consumer electronics, but also had an impact on several research disciplines such as robotics research, image processing, game design, and virtual reality applications [25, 14, 1, 15, 16]. RGB-D cameras consist of an infrared (IR) projector which emits a know pattern of structured IR light, an IR and an RGB camera. The estimation of depth is based on an internal triangulation process. The IR structured light source emits a constant pattern of speckles projected onto the scene. This pattern is acquired by the infrared camera and is correlated against a reference pattern. The reference pattern is generated by capturing a plane at a known distance from the Kinect sensor, and is stored in the camera's memory. When a speckle is projected on an object whose distance to the sensor is smaller or larger than that of the reference plane the position of the speckle in the infrared image will be shifted in the direction of the baseline between the IR projector and the projection centre of the IR camera. These shifts are measured for all speckles by a simple image correlation process to generate a disparity map. For each pixel the distance to the sensor can then be retrieved from the corresponding disparity pixel.

This revolutionized in some way the robotic and computer vision research scene, as now a sensor producing dense 3D point clouds in a reasonable quality is available at a price below USD 200. While the sensor is a great device, it has some restrictions: (1) it has a minimum range of about 80 cm and a maximum range of up to 5 m; (2) it is frail when used under real light conditions; and (3) it has problems to generate depth information on certain surfaces. In particular, when the observed surface is transparent, reflective or absorptive, the measurement is quite bad. This is because the appearance of projected speckles on such surfaces depend not only on their distances to the sensor, but also on multiple factors including viewpoints. Another problem occurs, when two or multiple RGB-D cameras work with overlapping views at the same time, because each sensor will see its own IR light pattern as well as the patterns of the other IR projectors and then will be unable to distinguish its own pattern. While some solutions to the latter problem have been proposed (e.g. in [16]) they do not solve the problem. And still the problem of the minimal distance of about 80 cm remains.

To overcome these problems, in this paper we present a novel approach to combine ordinary stereopsis with the RGB-D sensor. We use two low-cost RGB-D cameras (ASUS Xtions in our case) and arrange them as a stereo system. The new system yields much better depth image results than a single RGB-D camera alone. This solves, moreover, the problem of not detecting transparent objects as well as those which have bad emittance properties such as black objects or mirrors. The proposed approach is rather simple, yet very effective. All we need to do is to calibrate a pair of RGB-D cameras as a stereo camera system. One of the cameras serves as a reference camera. The 3D depth information from the scene is reconstructed using ordinary stereo registration. As one needs texture information for stereo registration, we did not choose to use RGB images from the cameras, but we make use of the IR image. This image yields very rich texture information, as the IR projector of the RGB-D camera projects an IR pattern into the scene. This helps to even detect depth information where in an ordinary RGB image not much texture information is available. Besides scenes with little to no texture, we can also compute depth information of transparent objects such as



(a) RGB image (b) Corresponding IR image (c) Depth map (one RGB-D camera) (d) Depth map (two RGB-D cameras)

Figure 1. Illustrating some of the RGB-D-camera weaknesses. This scene shows different kinds of surfaces (reflective, absorptive and transparent from glass and plastic) which cannot be captured by RGB-D camera. Note the interference (more areas with invalid depth information) when the scene is captured with two RGB-D cameras.

glasses or reflective objects such as mirrors.

The rest of the paper is as follows. Next, we review the related work that have addressed both problems, the transparent object detection and the interference problems as well as some works on improving RGB-D depth information with stereo in Section II. In Section III, our novel method is presented. We first describe the procedure to calibrate the IR cameras, to rectify the IR images and then to estimate a disparity map based on IR correspondence matching. After that we outline how to fuse the obtained disparity map with the depth map offered by the RGB-D sensor. In Section IV, our method is verified through a number of experiments on images which include interferences and different types of objects that cannot be correctly sensed by an RGB-D sensor alone and the results are discussed. We conclude with a summary and an outlook to future work in Section V.

II. Related Work

The fusing of multiple RGB-D cameras in order to observe a wider field of view and to increase the reliability and robustness of 3D depth measurements is required for several applications such as human motion estimation [2, 4, 24], face recognition [13], gesture recognition [5], 3D simultaneous localization and mapping (SLAM) [11, 3], and many others. Deploying multiple RGB-D sensors with overlapping views will produce interference effects because the IR patterns of the different cameras overlap. The interference dramatically degrades the depth quality causing many invalid (black) depth pixels as shown in Figure 1(a)–1(d). The left two images of the figure show the RGB and the IR image of the scene, the two images to the right in Figure 1 show the reconstructed depth maps of the scene. Figure 1(d) was captured with two RGB-D cameras which leads to many invalid depth pixels due to the described interference effects.

Recently, some methods have been proposed to tackle this problem. In [19], Rafibakhsh et al. propose a geographical configuration of sensors to reduce interferences. They recommended an angle of 35° between two sensors mounted at the same height. Assuming that the interference holes are small and isolated, and the observed surface is smooth and continuous, Maimone and Fuchs [15] devised a filling and smoothing algorithm by modifying median filters to fill all holes while preserving edges. In [8], the problem of filling in depth information at spots where the RGB-D sensor does not provide any depth values is addressed. This happens, for example, at object borders. They propose to use a bilateral filter that combines spatial and temporal information gathered from the RGB-D video stream. A different approach to fill the holes in the depth map is taken in [17]. They align texture and depth boundary in order to estimate missing depth information. In [7, 16] the idea to minimize interference by exploiting motion blur introduced by additional hardware components is proposed. The motion blur is induced by vibrating the RGB-D camera unit using an offset-weight vibration motor. A small amount of motion is applied to a subset of the sensors so that each unit sees its own IR projected pattern sharply, while seeing a blurred version of the IR patterns of the other units. In [21] another hardware solution is proposed by Schroder et al. They use hardware shutters for mitigating interference between concurrently projecting sensors, where IR emitter on each RGB-D sensor is blocked in turn so that the IR patterns do not interfere. However, the frame rate of depth maps are reduced with the number of Kinects.

Another problem of RGB-D sensors is its high sensitivity to the visual characteristics of observed objects, such as transparency, absorption and reflection. These objects are quite common, for instance, in household environments in the form of drinking glasses, bottles or vases. Until now no robust solution for detecting transparent objects and their reconstruction have been proposed.

In [14], Lysenkov et al. propose a Kinect-based method for transparent object detection which exploits the fact that many transparent objects appear as holes in the depth map. These holes are used as candidates for transparent objects and serve as an initialization for a segmentation process to extract the object contours in the RGB image. Alt et al. [1] propose an algorithm to reconstruct transparent objects in unstructured environments based on identifying inconsistent depth measurements caused by refractive or reflective effects on the surfaces while moving a depth sensor around the scene. The detection in this case is limited to transparent objects with smooth and curved surfaces where refractive effects dominate. Recently, a couple of papers [9, 22] propose combining the Kinect depth sensor with an external stereo system, aiming to

Figure 2. The setup of the Asus Xtion IR image stereo system used in our experiments.

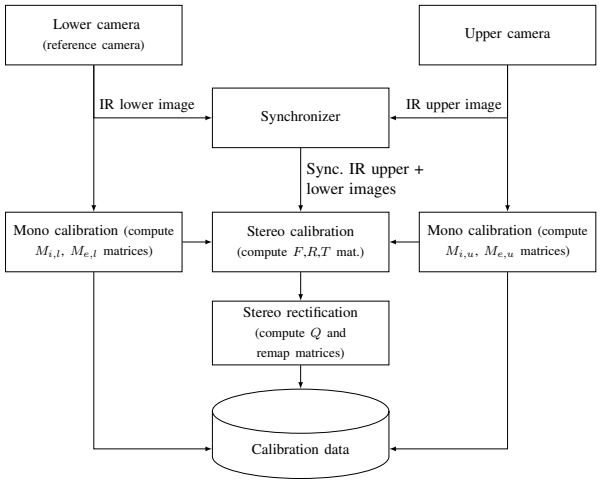


Figure 3. Calibration procedure

improve the accuracy of the depth map. In [23], Chiu et al. propose a cross-model stereo vision approach for the Kinect. They present a cross-modal adaptation scheme to improve the correspondence matching between RGB and IR cameras. The combination method produces depth maps that include sufficient evidence for reflective and transparent objects. The idea is similar to ours. However, while they improve the depth information from the RGB-D camera up to 30%, they still fail to return depth information for transparent, shiny, or matte objects.

In this paper, we propose a simple yet novel method to overcome almost all RGB-D sensor weaknesses with the single hardware requirement to use two RGB-D cameras instead of one, but without any conditions or limitations on the observed environment. The idea is to use each two RGB-D cameras as a stereo system and carrying out correspondence matching between IR images to build an additional depth map. Experimentally, we found that this depth map is insensitive to transparent, absorbing and reflective surfaces. In addition when this map is combined with the depth map offered by the RGB-D sensor, the problem of interference is completely resolved.

III. Combining Structured Light with Stereo

As mentioned above the RGB-D camera fails to capture objects and surfaces made from transparent, reflective and absorptive materials. In addition when at least two cameras are used to view the same scene, the interference problem decreases the quality of the estimated depth map. The RGB-D camera measures the depth by projecting a constant speckle pattern on a standard plane surface and saves the IR image as a reference image in the internal memory. The reference and the live captured image are used for triangulation to estimate depth under the assumption that the appearance and the relative shifts of speckles is only related to the depth of surface where they are projected.

This assumption is violated in two cases: (1) if the projected plane is reflective, refractive, absorptive or transparent; (2) several RGB-D cameras interfere with each other. In the first case the appearance of speckles is dramatically modulated by the visual properties of the observed objects. In the latter case when two or more cameras share a common field of view, each camera cannot distinguish between its own pattern and that of another camera.

We propose a new method that tackles both issues. Our idea is to consider the IR images that capture the projected IR speckles at the same time. The appearance and interference of speckles will seem almost identical in both IR synchronized images. If these images are calibrated and rectified, we can utilize them as a stereo pair to generate an additional depth map. Figure 2 shows our camera setup. The speckles and their interference will improve the correspondence matching results even if the observed surfaces are texture-less. The projected IR pattern of the two cameras yield a lot of texture information that helps to solve the correspondence problem for the stereo camera. Figure 1(b) gives an impression of how much additional texture information is provided by the IR pattern. In the next section we outline the camera calibration for the proposed camera setup before we describe the imaging pipeline in Section III-B. Some implementation details are given in Section III-C.

A. Stereo Camera Calibration, Rectification and Matching

To compute a depth map from a stereo image pair, it is necessary to calibrate the cameras and then to rectify the images. The goal of the stereo calibration process is to estimate the projection matrix of each individual camera that describes the projective transformation between the 3D scene and its image and to estimate the fundamental matrix that describes the epipolar geometry between two corresponding images. The estimation of both matrices can be done by generating 3D artificial points that can be easily, reliability and rapidly detected in the captured images. Therefore, available implementations, for instance in OpenCV or Matlab, use

checkerboards for calibrating these parameters. The offline stereo camera calibration routine is shown in Figure 3. We follow the notation of [10]. In our calibration method, the IR emitters are covered during the stereo calibration and rectification processes. This improves the reliability of the checkerboard corner detection.

1) Mono Calibration: For each camera, we assume a pinhole camera model, describing the perspective projection from a 3D point P in the world coordinate frame to its 2D points onto the image planes p_u from the upper camera and p_l from the lower camera, respectively. The pinhole camera model considers the intrinsic camera parameters in a matrix $M_{i,u}, M_{i,l}$, such as focal length and the lens distortions, and extrinsic parameters in a matrix $M_{e,u}, M_{e,l}$ corresponding to the translation and rotation of each camera between the camera and the world coordinate system. The matrix $M_{e,u}$ includes the 3×3 rotation matrix R_u and the 3D translation vector T_u from the upper camera origin to the world coordinate centre with $M_{e,u} = [R_u \mid T_u]$ and $M_{e,l} = [R_l \mid T_l]$ for the lower camera, respectively. The perspective projections of the mono calibration step can be described by the following equations, calculating the projection of a 3D point P to the 2D points p_u and p_l on the image planes from the upper and the lower camera with $p_u = M_{i,u} \cdot M_{e,u} \cdot P = M_u \cdot P$ and $p_l = M_{i,l} \cdot M_{e,l} \cdot P = M_l \cdot P$, respectively.

2) Stereo calibration: The stereo calibration step computes the rotation matrix R and the translation vector $T = [T_0\ T_1\ T_2]^T$ between the two camera coordinate systems. In both coordinate systems the vectors $P_u = [X_u\ Y_u\ Z_u]^T$ and $P_l = [X_l\ Y_l\ Z_l]^T$ represent the same 3D point P resulting in $P_u = R \cdot (P_l - T)$. The epipolar plane is spanned by the two vectors T and P_l. The vector $(T - P_l)$ is inside this plane, so the dot product must be zero: $(P_l - T) \cdot (T \times P_l) = 0$. Hence,

$$(T \times P_l) = \begin{bmatrix} 0 & -T_2 & T_1 \\ T_2 & 0 & -T_0 \\ -T_1 & T_0 & 0 \end{bmatrix} \cdot P_l = A \cdot P_l$$

and $P_u = R \cdot (P_l - T)$ which is $R^{-1} P_u = P_l - T$. Substituting $(P_l - T)$ with $R^{-1} P_u$ and the cross product $(T \times P_l)$ with $A \cdot P_l$ and rearranging the equation we obtain $(P_l - T) \cdot (T \times P_l) = 0 \Leftrightarrow R^T P_u \cdot A \cdot P_l = P_u^T \cdot R \cdot A \cdot P_l = 0$ where the matrix $E = R \cdot A$ is called the essential matrix. The corresponding homogeneous points on the image planes p_u and p_l can be described through the fundamental matrix F with $p_u \cdot F \cdot p_l = 0$. The fundamental matrix considers the intrinsic parameters from the upper and lower camera $M_{i,u}$ and $M_{i,l}$ and can be calculated with the essential matrix E by $F = M_{i,u}^{-T} \cdot E \cdot M_{i,l}^{-1}$.

3) Stereo Rectification and Matching: The next step is the stereo rectification. We need to compute the rotation matrices for each camera so that the corresponding epipolar lines in all viewing planes become collinear with each other. It takes all epipoles of an original stereo setup to infinity. After the rectification, corresponding points in the upper and lower images are on the same vertical line. The output is a 4×4 reprojection matrix Q, which transforms the disparity value into a depth map. The reprojection from the disparity map D to a 3D point cloud is calculated as $Q \cdot [x\ y\ d\ 1]^T = [X\ Y\ Z\ W]^T$. The 2D point is reprojected to 3D space as $[X/W\ Y/W\ Z/W]^T$ by dividing through the homogeneous component.

Once the IR images are rectified, a disparity map can easily be computed by searching for correspondence pixels on the corresponding epipolar parallel lines. The decision that two pixels are corresponding pixels is related to the similarity of the local appearance around them. There are many ways to measure the similarity cost including the Sum of Absolute Differences (SAD), which is the most common used similarity cost due to its computational simplicity.

$$SAD(p(x,y), D_p) = \sum_{i=-w}^{w} \sum_{j=-w}^{w} (|Img_l(x+i, y+j) - Img_u(x+i, y+j+D_p)|)$$

where $(2w+1) \times (2w+1)$ is the local block size and Img_l, Img_u are the lower and upper rectified stereo infrared image pair. Current stereo methods can be divided into two major classes: local and global methods. Local methods try to estimate optimal disparities for each point only based on the local appearance around the point which can lead to discontinuities in the estimated disparity map. Global approaches optimize all disparities at once by making explicit smoothness assumptions of the disparity map and then calculating it by minimizing a global energy function. However, the required computation time is considerably higher.

The Semi-global Block Matching [12] incorporates the advantages of both method classes, achieving good trade-off between the low complexity and the high quality. The semi-global matching method aims to minimize a global $2D$ energy function by solving a large number of $1D$ minimization problems.

$$E(D) = \sum_p (SAD(p(x,y), D_p) + \sum_{N_p} P_1 T[|D_p - D_q| = 1]) + \sum_{N_p} P_2 T[|D_p - D_q| > 1])$$

where $T[.]$ returns 1 if its argument is true and 0 otherwise. The first term is the similarity cost for all pixels p at their disparities D_p. The second term penalizes small disparity differences of neighbouring pixels N_p of p with the constant weight P_1. Similarly the third term penalizes large disparity steps with a higher constant penalty P_2. Using a lower penalty P_1 for small disparity changes permits an adaptation to curved surfaces. The penalty P_2 for larger disparity changes preserves discontinuities. The $2D$ energy function is computed along $1D$ paths from 8 directions towards each pixel of interest using dynamic programming. The costs of all paths are summed for each pixel and disparity. The disparity is then determined by winner takes all, then a sub-pixel interpolation is performed by fitting a parabola to the winning cost value and its neighbours. After that a left-right consistency check is performed for mismatches and occlusion invalidation.

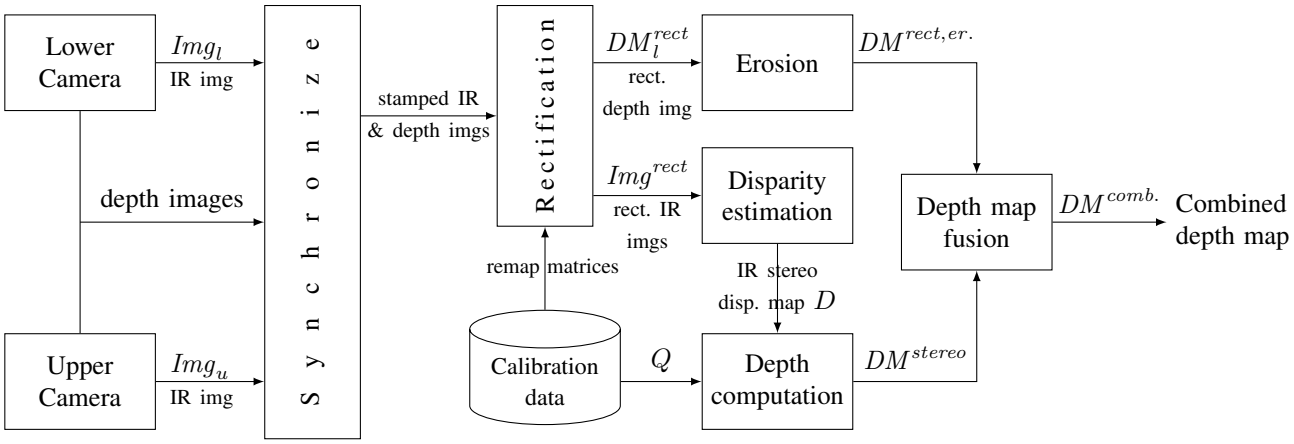


Figure 4. Imaging pipeline

B. Fusion Algorithm

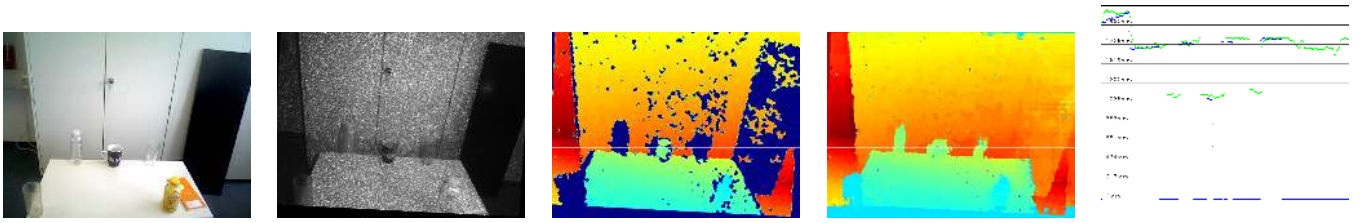
The idea is to fuse the depth information from the RBG-D sensor and from the proposed IR stereo matching to tackle the problem of interference and material-sensitivity. The complete pipeline and the basic algorithm is shown in Figure 4. The depth images as well as the IR images from both cameras are registered and synchronized. After that they provide corresponding timestamps. The stamped images are being rectified making use of the extrinsic and intrinsic camera parameters estimated in the calibration step, $Img_l^{rect} = remap_{lower}(Img_l) Img_u^{rect} = remap_{upper}(Img_u)$, where Img_l and Img_u, resp., are the lower and upper raw IR images and Img_l^{rect} and Img_u^{rect}, resp., are their rectified versions. $remap_{lower}$ and $remap_{upper}$ are transformation functions for the lower and the upper camera. After the rectification of the IR images, we use them to estimate a disparity map, $D = estimate\_disp(Img_l^{rect}, Img_u^{rect})$. which is then used to compute the stereo depth map $DM^{stereo} = D \cdot Q$ with Q the reprojection matrix computed in the stereo rectification process for calculating 3D information from disparity maps. Because the DM^{stereo} is computed in the coordinate system of the lower camera, the $remap_{lower}$ is also applied on the lower depth map offered by the Xtion to keep an identical arrangement of the depth pixels in both depth maps: $DM_l^{rect} = remap_{lower}(DM_l)$. For the remapping process, an interpolation function is used to compute pixels coordinates for non-integer points. This leads to introduce non-zero invalid pixel into the depth map around the black invalid pixel regions. These pixels are eliminated before conducting the fusion process by using a morphological erosion function $DM^{rect,er.} = erode(DM^{rect})$. The fusion process is then done by replacing the values of invalid depth pixels by corresponding one from the IR stereo matching:

$$DM^{comb.} = \begin{cases} DM^{rect,er.}, & \text{if } DM^{rect,er.} \neq 0 \\ DM^{stereo}, & \text{otherwise} \end{cases}$$

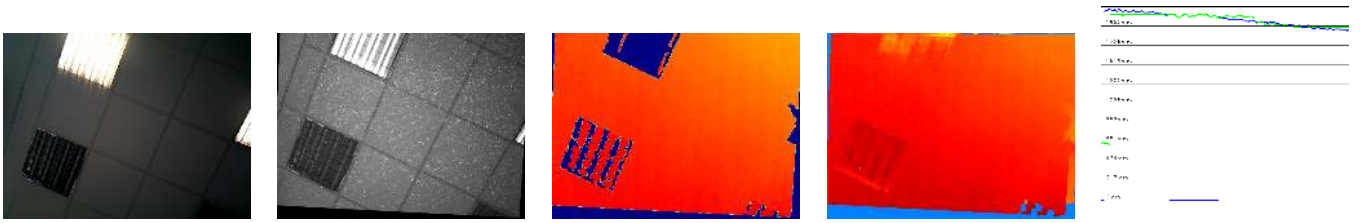
C. Implementation Details

In the implementation of the proposed method we used standard functions from the OpenCV library [6], the Robot Operation System ROS [18] and the Point Cloud Library (PCL) [20]. From the ROS framework two packages are used, the `cv_bridge` and the `message filters`. `cv_bridge` is as its name implies only used for bridging between OpenCV image data types and ROS image messages. The `message filters` package consists of few filter implementations including `Synchornizer` which synchronizes multiple messages by their timestamps and only passing them through when all have arrived. In our implementation, the `Synchornizer` filter is used to synchronize lower and upper IR images for stereo calibration and the disparity estimation procedures and to synchronize lower IR and depth images for the depth map fusion process.

The OpenCV framework provides some functions for the purpose of camera calibration, stereo rectification and disparity estimation. The functions `findChessboardCorners` and `cornerSubPix` are used to detect checkerboard corners in sub-pixel resolution. The `cameraCalibrate` function is used to estimate the camera matrix and the distortion coefficients for each individual camera iteratively by minimizing reprojection errors over all detected corners in several images. The `stereoCalibrate` function conducts also a similar optimization procedure to compute the essential matrix E and the fundamental matrix F of the two views with $F = M_u^{-T} \cdot E \cdot M_l^{-1}$ and a rotation and translation matrices that project the coordinate system of the first camera onto the coordinate system of the other one that is selected as a reference camera. The stereo rectification process aims to reproject the image planes of the two cameras so that the epipolar lines in one image are parallel to epipolar lines in the other image. To this end, the epipolar points in both images must be shifted to the infinity. In OpenCV framework, this is done by the function `stereoRectify`. The function takes as input the camera matrices, distortion coefficients and projective transformations computed by `stereoCalibrate`

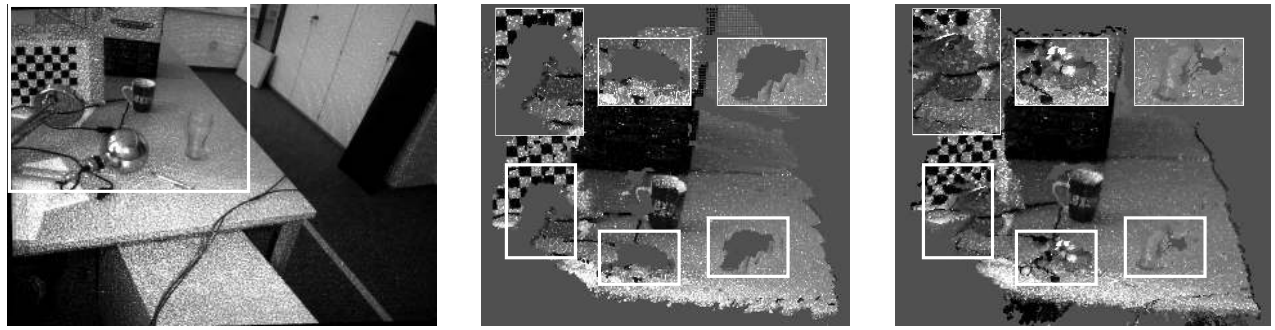


(a) RGB image (b) IR image (c) Depth image from Xtion (with interference from second projector) (d) Depth image combined (e) Comparison between (green) IR stereo and (blue) RGB-D Sensor depth at the scan line

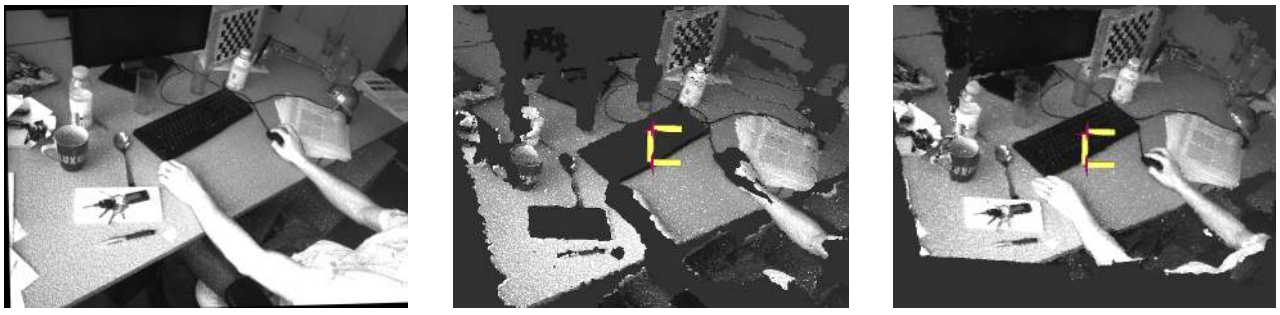


(f) RGB image (g) IR image (h) Depth image from Xtion (with interference from second projector) (i) Depth image combined (j) Comparison between (green) IR stereo and (blue) RGB-D Sensor depth at the scan line

Figure 5. Comparison of depth maps: Scene1 shows an office table top. Both IR projectors are turned on and lead to interference in the Xtion depth image; Scene2 shows the office ceiling with reflecting lamps. Only the IR projector from the reference camera was used.



(a) IR image of the scene 1 (b) Point cloud from depth sensor (c) Point cloud from IR stereo



(d) IR image of scene 2 (e) Point cloud from depth sensor (f) Point cloud from IR stereo

Figure 6. Comparison of point clouds; The first row shows how missing depth information are filled in for a table top scene. (b) shows the missing depth information in the point cloud acquired by the Xtion. In (c) most of the missing points were reconstructed by IR stereo; in the second scene wide areas such as the monitor and keyboard, the booklet and the keyring, the spoon and the arm holding the mouse cannot be detected by the RGB-D sensor, but with the combined depth map from RGB-D depth and stereo depth. The red marks in the centre of the image show the origin of the two camera coordinate systems.

and `cameraCalibrate` functions and returns the projections required to map the epipoles to infinity. To estimate the disparity map from rectified IR images, the OpenCV class `StereoSGBM` is used. Figure 4 shows the complete procedure.

IV. Empirical Evaluation

To evaluate the effectiveness of our approach, multiple experiments have been run on different scenes that include different types of objects that cannot be detected by an RGB-D sensor. Here two examples of these scenes are presented. The first scene (presented in Figures 5(a)–5(d)) involves two glasses and plastic bottle as example for transparent objects and table lamp made from stainless steel as an example for a reflective object; in the background there is a black plate as an example for an absorptive object. The second scene is presented in Figures 5(f)–5(i) and shows lamps (some are turned on and some off) in the ceiling of an office as another example for surfaces that cannot be sensed by the RGB-D depth sensor.

We consider two different cases: (1) the IR emitters of both cameras are active; and (2) the IR emitter of the non-reference camera is blinded to illustrate the performance of the proposed method for resolving the interference problem. Figures 5(a)–5(e) show the result of the first scene with taking into account the effect of interference. In Figures 5(f)–5(j) we present the result of the second scene without interference effects. Comparing the depth maps in both scenes the advantage of our approach to overcome the weaknesses of sensor depth becomes apparent. As shown in Figures 5(c) and 5(h), in the regions where transparent, reflective or absorptive objects appear in the scene, the sensor depth map shows black pixels (here represented in dark blue) which means that no depth information at all is available for these regions. In contrast, Figures 5(d) and 5(i) show that the depth of such objects can be measured by IR stereo with an accuracy similar to that achieved by the depth sensor. In case of Figure 5(d), one can see that the effect of interference is resolved completely. For both scenes, we also compare the accuracy of the depth information of the RGB-D sensor (blue dots with the stereo information (green dots) in Figs. 5(e) and 5(j) for a certain shown in Figs. 5(d) and Figs. 5(i), resp. As can be observed yield RGB-D and stereo very similar depth information, the RGB-D camera, however, has many regions where no information are available at all.

In Table I we show quantitative results of the IR stereo system. The table compares the invalid (black) pixels, where no depth information is available. For the two scenarios, we also compare the number of invalid pixels when both IR projectors are used (denoted by *w/ interference*) and only one is used (denoted by *w/o interference*). We compare *Sensor depth*, which denotes the depth map from the sensor acquired through structured light, *IR stereo* which shows the invalid pixels for the stereo depth computation based on the IR images, and finally *Combined depth* where invalid pixels coming from the sensor are filled by stereo IR information. As can be seen in the table, there is a dramatic decrease in the number of invalid pixels from RGB-D to IR stereo. Obviously, the combined depth map does not yield further improvements in the number of invalid pixels. The interference has only little effect on the disparity computation.

The second set of experiments even more show the power of our approach to sense reflective, transparent, and absorptive objects. The scenes in Figure 6 involve many objects including a table lamp made from stainless-steel, a transparent glass, black surfaces from a monitor and keyboard, shiny surfaces from a booklet, etc. We acquire point clouds by the Xtion depth sensor and the IR stereo system. The point clouds are shown in Figure 6. As one can see in the figures, IR stereo is nearly completely able to acquire depth information where the Xtion sensor only shows invalid pixels.

Further, we wanted to find out about the accuracy of the computed depth maps. We captured a texture-less plane surface from different distances. The plane was parallel to the sensor image plane so that all pixels had the same depth. After that we computed the distance from the camera to the plane by the depth sensor and by our IR stereo system. During this experiment, the IR emitter of the non-reference camera was blinded to neutralize the interference effects. The obtained results are listed in Table II. One result is that the IR stereo can overcome the RGB-D distance limitation. With a baseline of $0.045\,\mathrm{m}$ (see Figure 2 for our camera setup) we are able to measure minimal distances of $0.5\,\mathrm{m}$. The accuracy achieved by IR stereo is similar to that provided by the depth sensor.

The proposed method runs on a Core-i7 processor @ $3.40\,\mathrm{GHz}$ with a frequency of 5–6 Hz when the disparity is computed. In our prototype implementation, we compute the disparity for all pixels. A dramatic speed-up is expected by limiting the computation of disparity only to pixels of black holes in sensor depth map.

V. Conclusion

In this paper we proposed a novel, yet simple method to improve the depth information provided by an RGB-D sensor. Such sensors use structured infrared light to acquire depth information of a scene. However, RGB-D cameras suffer from several limitations: (1) they cannot detect transparent, shiny or absorptive surfaces, as the IR pattern gets distorted; (2) they have a minimal range of about $0.8\,\mathrm{m}$. Our approach is to use two RGB-D sensors in a stereo setting. Instead of computing a depth map based on RGB information, we make use of the IR images of the RGB-D sensor. The advantage is that even when there is little to no texture information in the RGB images for finding corresponding pixels to compute the disparity, the projected pattern in the IR image provides rich texture information. With our method we are able to robustly detect transparent objects such as glasses, shiny objects such as mirrors, or absorptive objects such as matte surfaces. Further, with a baseline of $0.045\,\mathrm{m}$ we are able to acquire minimal distances of $0.5\,\mathrm{m}$. This is an improvement over the Kinect's or Xtions's minimal distance of $0.8\,\mathrm{m}$. Our results show that the number of invalid pixels in the depth map can be decreased dramatically with combining the raw depth information provided by the

Table I
COMPARING THE NUMBER OF INVALID PIXELS WITH AND WITHOUT INTERFERENCE FOR THE TWO SCENES FROM FIG. 5

No. of invalid pixels	Scene 1		Scene 2	
	w/ interf.	w/o interf.	w/ interf.	w/o interf.
Sensor depth	69643	50610	44448	30240
IR stereo depth	298	219	673	708
comb. depth	298	219	673	708

Table II
ACCURACY COMPARISON BETWEEN PLAIN XTION DEPTH MAPS AND IR STEREO DEPTH MAPS AT DIFFERENT DISTANCES.

Camera type	Distance [mm]							
	500	1000	1500	2000	2500	3000	3500	4000
Plain Xtion	N/A	1003	1503	2002	2504	3007	3506	4009
IR stereo Xtion	499	997	1494	1989	2499	2995	3493	3991

sensor with the IR stereo computations. As another result, we show that interference problems when using more than one RGB-D sensor for acquiring point clouds can be overcome by filling in the missing depth information from our combined depth map. The stereo computation works with 5–6 Hz in our prototype implementation. This is a rather low frame rate compared to the 30 Hz of the RGB-D sensor. For future work, we plan to increase the frame rate of our system by a fair amount. For our prototype implementation, we computed the IR stereo depth map for all pixels. A dramatic increase in the speed could simply be achieved by only computing stereo depth information for pixels, where the ordinary RGB-D sensor is unable to provide depth information. For another future work, we plan to apply the proposed IR stereo method on a set of Kinect sensors distributed in an industrial robot arm workspace to enhance object manipulation and obstacle avoidance capabilities of the robot and in mapping and localization tasks for mobile systems.

ACKNOWLEDGMENTS

This work was funded by the Ministry of Innovation, Science and Research of North-Rhine Westfalia, Germany, under grant 321-8.03.04.02-2012/01/1.

REFERENCES

[1] Nicolas Alt, Patrick Rives, and Eckehard G. Steinbach. Reconstruction of transparent objects in unstructured scenes with a depth camera. In *Proceedings of the IEEE International Conference on Image Processing (ICIP-2013)*, pages 4131–4135. IEEE, 2013.

[2] Stylianos Asteriadis, Anargyros Chatzitofis, Dimitrios Zarpalas, Dimitrios S. Alexiadis, and Petros Daras. Estimating human motion from multiple kinect sensors. In *Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications*, MIRAGE '13, pages 3:1–3:6. ACM, 2013.

[3] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[4] Kai Berger, Kai Ruhl, Yannic Schroeder, Christian Bruemmer, Alexander Scholz, and Marcus A. Magnor. Markerless motion capture using multiple color-depth sensors. In Peter Eisert, Joachim Hornegger, and Konrad Polthier, editors, *Proceedings of the Vision, Modeling, and Visualization Workshop (VMW-2011)*, pages 317–324. Eurographics Association, 2011.

[5] K. K. Biswas and S.K. Basu. Gesture recognition using microsoft kinect. In *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, pages 100–103, Dec 2011. doi: 10.1109/ICARA.2011.6144864.

[6] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[7] D. Alex Butler, Shahram Izadi, Otmar Hilliges, David Molyneaux, Steve Hodges, and David Kim. Shake'n'sense: Reducing interference for overlapping structured light depth cameras. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1933–1936. ACM, 2012.

[8] Massimo Camplani and Luis Salgado. Efficient spatio-temporal hole filling strategy for kinect depth maps. In *Proc. of the SPIE*, volume 8290, pages 82900E–82900E–10, 2012.

[9] D. Y. Chan and C. H Hsu. Regular stereo matching improvement system based on kinect-supporting mechanism. *Open Journal Applied Sciences*, page 22–26, 2013.

[10] Boguslaw Cyganek and J. Paul Siebert. *An introduction to 3D Computer Vision - Techniques and Algorithms*. Wiley, 2009.

[11] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.

[12] Heiko Hirschmueller. Stereo processing by semi-global

matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[13] M. Hossny, D. Filippidis, W. Abdelrahman, H. Zhou, M. Fielding, J. Mullins, L. Wei, D. Creighton, V. Puri, and S Nahavandi. Low cost multimodal facial recognition via kinect sensors. In *Proceedings of the 2012 Land Warfare Conference: Potent land force for a joint maritime strategy (LWC-2012)*, 2012.

[14] Ilya Lysenkov, Victor Eruhimov, and Gary Bradski. Recognition and pose estimation of rigid transparent objects with a kinect sensor. In *Proceedings of Robotics: Science and Systems*, 2012.

[15] A. Maimone and H. Fuchs. Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR),*, pages 137–146, 2011.

[16] A. Maimone and H. Fuchs. Reducing interference between multiple structured light depth sensors using motion. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, pages 51–54, 2012.

[17] Dan Miao, Jingjing Fu, Yan Lu, Shipeng Li, and Chang Wen Chen. Texture-assisted kinect depth inpainting. In *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, pages 604–607, 2012.

[18] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.

[19] N. Rafibakhsh, J. Gong, M. Siddiqui, C. Gordon, and H. Lee. *Analysis of XBOX Kinect Sensor Data for Use on Construction Sites : Depth Accuracy and Sensor Interference Assessment*, chapter 86, pages 848–857. 2012.

[20] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA-2011)*. IEEE, 2011.

[21] Yannic Schröder, Alexander Scholz, Kai Berger, Kai Ruhl, Stefan Guthe, and Marcus Magnor. Multiple kinect studies. Technical Report 09-15, ICG, 2011.

[22] G. Somanath, S. Cohen, B. Price, and C. Kambhamettu. Stereo+kinect for high resolution stereo correspondences. In *International Conference on 3D Vision (3DV-2013)*, 2013.

[23] Ulf Blanke Wei-Chen Chiu and Mario Fritz. Improving the kinect by cross-modal stereo. In *Proceedings of the British Machine Vision Conference (BMVC-2011)*, pages 116.1–116.10. BMVA Press, 2011.

[24] Licong Zhang, Jürgen Sturm, Daniel Cremers, and Dongheui Lee. Real-time human motion tracking using multiple depth cameras. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-2012)*, pages 2389–2395. IEEE, 2012.

[25] S. Zug, F. Penzlin, A. Dietrich, Tran Tuan Nguyen, and S. Albert. Are laser scanners replaceable by kinect sensors in robotic applications? In *Robotic and Sensors Environments (ROSE), 2012 IEEE International Symposium on*, pages 144–149, 2012.