Depth-Awareness in a System for Mixed-Reality Aided Surgical Procedures

Mauro Sylos Labini[1,2,3], Christina Gsaxner[1,2,4], Antonio Pepe[1,2], Juergen Wallner[2,4], Jan Egger[1,2,4(✉)], and Vitoantonio Bevilacqua[3]

[1] Institute for Computer Graphics and Vision, Faculty of Computer Science and Biomedical Engineering, Graz University of Technology, Graz, Austria
egger@tugraz.at
[2] Computer Algorithms for Medicine Laboratory, Graz, Austria
[3] Department of Electrical and Information Engineering, Polytechnic University of Bari, Bari, Italy
vitoantonio.bevilacqua@poliba.it
[4] Department of Oral and Maxillofacial Surgery, Medical University of Graz, Graz, Austria

Abstract. Computer-assisted surgery is a trending topic in research, with many different approaches which aim at supporting surgeons in the operating room. Existing surgical planning and navigation solutions are often considered to be distracting, unintuitive or hard to interpret. In this work, we address this issue with an approach based on mixed reality devices like Microsoft HoloLens. We assess the depth sensing capabilities of Microsoft HoloLens, and the potential benefit they could bring to computer-assisted surgery applications.

Keywords: Computer-assisted surgery · Mixed reality · HoloLens · Time-of-flight camera · RGB-Depth mapping · Pattern recognition

1 Introduction

Surgical resection of the tumoral mass is one of the primary treatments that patients affected by cancer in the head and neck area must undergo. Due to the high invasiveness of the procedure and the risk of relapses, it is crucial for the surgeon to quickly and precisely evaluate the location and extension of the tumor. As noted by the American National Institutes of Health (NIH) [1], while medical imaging and operating microscopes currently aid the surgeon during this operation, neither of these tools can provide direct visualization of the mass to be removed. This work wants to further investigate the usability of mixed-reality (MR) devices as a visualization aid for a

Supported by FWF KLI 678-B31 (enFaced), COMET K-Project 871132 (CAMed) and the TU Graz Lead Project (Mechanics, Modeling and Simulation of Aortic Dissection).

D.-S. Huang et al. (Eds.): ICIC 2019, LNAI 11645, pp. 716–726, 2019.
https://doi.org/10.1007/978-3-030-26766-7_65

surgeon in the aforementioned scenario [2]. Recent advancements in MR sparked new research effort towards the introduction of this technology in the operating room, with Microsoft HoloLens representing a common hardware choice. An example of this trend is given by Perksin et al. [3], where they evaluate the role of mixed reality during breast cancer surgery, or the work from Pratt et al. [4], on augmented reconstruction surgery. A relevant contribution regarding head and neck surgery is given by Wang et al. [5], who employed a video-see-through AR headset to visualize CT imaging data directly on the patient. Pepe et al. [6] suggested a MR-based application, which features automatic marker-less image registration and a hands-free interface for the facial surgeon. However, their approach is limited by the spatial reasoning capabilities of the device, which only exposes a coarse map of the environment. This is unsuitable for the accuracy required for medical applications. The capabilities of the device have recently been expanded, allowing researchers to access more of its onboard sensors data, therefore we aim at exploiting these new capabilities. In particular Time-of-Flight (ToF) depth sensor, which enabled us to turn the HoloLens into an all-in-one visualization and measuring tool. Furthermore, an approach for improved object-to-patient registration is proposed. Our method, in fact, combines the newly enabled depth-awareness of the headset with insight from pattern recognition algorithms to accurately detect a patient's facial traits and locate them in the user's frame of reference.

2 HoloLens Measuring Capabilities

Following the idea from Pepe et al. [6], we develop a MR image registration system based on the Dlib face recognition library [7–10], which detects a number of the patient's facial landmarks in the frames obtained from the device RGB camera. These landmarks are then identified on the 3D model built from the patient's CT scan so that each key point can be matched to its corresponding point in the camera frame. A major difficulty with this approach lies in the assessment of the patient's position in the camera's optical axis direction. To reconstruct the 3D coordinates of the detected landmark points, Pepe et al. use the combined information of the camera intrinsic parameters and the rough spatial mapping depth estimation. Spatial mapping however, was not designed for fine distance measurements and it is therefore not the ideal candidate for this task.

2.1 Research Mode

In April 2018, Microsoft released a Windows 10 update, which unlocks the so-called Research Mode on the HoloLens headset. Research Mode is a tool aimed at granting developers with an extended access to the data collected by the headset built-in sensors [11]. This provides APIs to three different data sources:

– Four environment tracking cameras used by the system for map building and head tracking.

- Two ToF depth cameras, one for high-frequency (30 FPS) near-depth sensing, commonly used in hand tracking, and the other for lower-frequency (1FPS) far-depth sensing, currently used by the SLAM-based Spatial Mapping.
- Two versions of an IR-reflectivity stream used by the HoloLens to compute depth.

As depth perception was found to represent a major obstacle in previous studies [6], in this work we only considered the short-range depth data, streaming at 30 frames per second (FPS). This choice was also driven by accuracy reasons - ToF sensors performances exponentially degrade with the distance and by the fact that surgeons generally operate within the arm distance from the patient.

2.2 Range Images Un-Projection

Cameras produce 2D images projecting a set of 3D points in the physical world onto a plane, thus reconstructing a 3D model of the camera view from its 2D representation requires some knowledge about the camera projection model. This model depends on physical and optical features of the camera, usually device-specific and hard to retrieve without the manufacturers aid. Microsoft provides a representation of the HoloLens depth camera model in the form of an un-projection mapping, namely a transformation that maps pixel coordinates to a unit-depth plane [12].

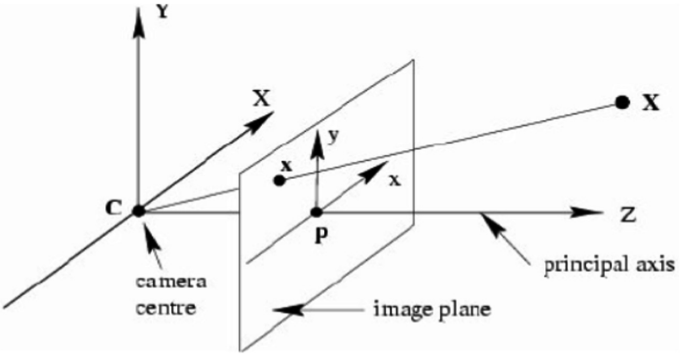

Fig. 1. Illustration of the geometry of the projection process.

With reference to Fig. 1, let us call *[X, Y, Z]* the 3D coordinates of a point in the real world and *[x, y, 1]* the coordinates of the point projected on the unit-depth plane. The un-projection mapping specifies, for each point on the projection plane, a pair of [u, v] values such that *[X, Y, Z] = Z * [u, v,1]*. The mapping values come arranged in two 448 $\times$ 450 matrices – one for u values and one for v values – so that to each pixel in the depth frame correspond a unique pair *[u, v]*. This means that all we need now in order to reconstruct the cameras 3D view is the Z coordinate of the un-projected points. As before mentioned, research mode provides a stream of range images, which define

for each pixel a value of distance, measured from the target point to the camera center. However, Fig. 1 clearly shows that this distance does not correspond to the Z coordinate of the target, which we can retrieve through the (u, v) values.

Now, to retrieve the desired 3D coordinates, we apply the mapping over the whole depth frame as follows: if Z_{ij} is the Z value of the pixel on the *i-th* row and the *j-th* column and u_{ij} and v_{ij} are the un-projection mapping parameters, we transform the pixel in Z_{11} with the corresponding *(u, v)* values so that $[X_1, Y_1, Z_1] = Z_{11} * [u_{11}, v_{11}, 1]$. This process theoretically leaves us with $448 * 450 = 201600$ – 3D points, of which, in practice, more than a half are discarded through background removal.

2.3 Measurements on a Reconstructed 3D Scene

As we propose to use Microsoft HoloLens as a high-precision depth sensing tool, it is crucial that we assess the accuracy performance of the device in different measurement scenarios. Here we consider three scenarios:

- planar surfaces,
- simple 3D objects,
- a complex 3D object.

To perform the measurements, the recordings of the depth sensor were downloaded from the HoloLens, then the relative point clouds were extracted and analyzed in MATLAB, using the built-in point clouds visualizer. As light interferences can negatively affect IR-based depth sensing, the recordings were carried out in an environment as isolated from sunlight as possible, using the on-board IR projector for illumination. For planar surface measurements, we observed a wall in the laboratory, assumed to be perfectly flat. We then employed a RANSAC-based algorithm [13] to fit a plane to the extracted point cloud and we measured the mean squared error over all the inlier points. The depth sensor is located on top of the user's head, and therefore moves together with it. Due to this fact, it is difficult to accurately establish a ground truth for absolute distance measurements. Thus, for the remaining measurements, we decided to maintain a simple setup and to only consider relative distances. The last measurement scenario is worth of particular attention, as this is also a testbed for the actual medical application. The used object was a 3D-printed model built from a high-precision scan of a head-cancer patient; the same employed by Pepe et al. [6] to test the final application. To recover the ground truth for the measurement, we loaded the mesh from which the model was printed into a 3D visualization software, as shown in Fig. 2. Here, we determined the distance of the nose tip from the flat back of the head, to simulate a patient lying face-up on an operating bed. Then, for the actual measurements, we analyzed several reconstructed views of the 3D head, recorded at arm distance. Eventually, we exploited MATLAB plane fitting to determine the parameters of the head's bearing plane, selected the farthest point from such plane and took the distance as our "nose tip - to - plane" measure.

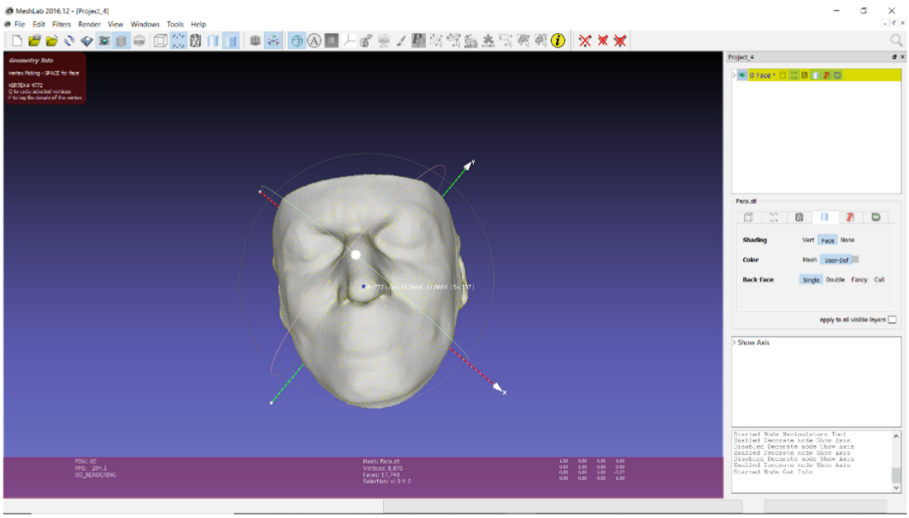

Fig. 2. The 3D model of a patient head visualized in MeshLab to define a ground truth for the measurements.

2.4 RGB-Depth Mapping Pipeline

In order to enhance the RGB camera-based face detection algorithm developed by Pepe et al. [6], it was necessary to map depth values produced by the ToF sensor to the frames shot by the HoloLens front-facing camera. The task was not trivial due to the misalignment between the two camera views and to the differences in field of view (FOV) and resolution, highlighted in Fig. 3.

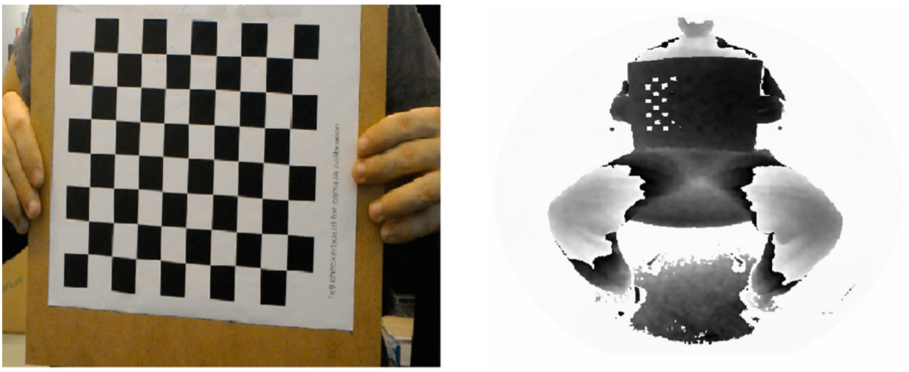

Fig. 3. A quasi-synchronized shot from the RGB camera (left) and the short-range depth camera (right).

For devices developed with depth measurements in mind, manufacturers usually provide in-house produced calibration results, as the procedure requires rather complex setups and high-precision instrumentation. This was not the case for HoloLens, for which Microsoft only released scarcely documented, partial pieces of calibration data. For this reason, a significant part of the present work was devoted to delineating the RGB-to-Depth mapping pipeline.

2.5 The Mapping Pipeline

In order to map depth information on RGB frames, we need to find a transformation between the two camera views. This transformation can be expressed through a 4×4 roto-translation matrix, composed by a 3×3 matrix and a 1×3 vector, which hold information about the relative rotation and translation, respectively, between two cameras' frames of reference. In practice, as illustrated in Fig. 4, the whole process can be reduced to a series of transformations: from the depth camera 2D projection space to its relative 3D Coordinate System, then to the RGB Coordinate System and finally back to the RGB projection space. One major issue with this approach lies in the temporal misalignment between the recordings of the two sensors. In fact, the HoloLens API does not allow access to the RGB and Depth data streams at the same time [14], leading to a fluctuating mismatch between the frames acquisition time. Because of this, we are forced to consider the sensors as if they were constantly moving with respect to each other. Therefore, the sensors relative position has to be calculated for each pair of frames we want to map between.

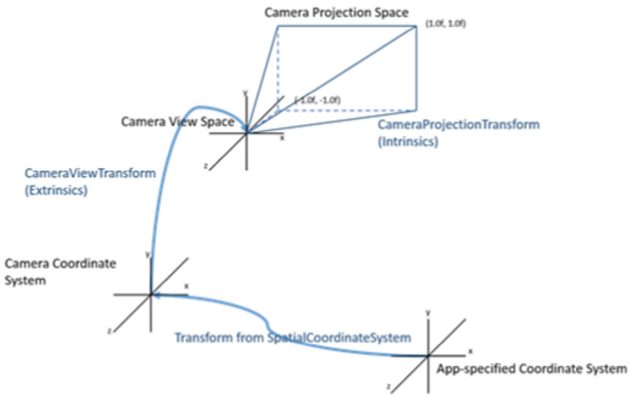

Fig. 4. A scheme of the process of locating frames acquired with HoloLens in the real world (https://docs.microsoft.com/en-us/windows/mixed-reality/locatable-camera).

Initially, we un-project the depth frame pixels to 3D coordinates as discussed in the previous section. Then, we calculate the absolute poses of the sensors. We achieve this by combining the Frame to Origin and the Camera View Transformation 4×4 matrices, accessible through the HoloLens API [15]. Finally, we calculate the

transformation between Depth and RGB cameras coordinates, from which the relative pose can be derived as follows:

Let the cameras C1 and C2 have respective camera poses $P_{w1} = \begin{pmatrix} R_1 & t_1 \\ 0 & 1 \end{pmatrix}$ and $P_{w2} = \begin{pmatrix} R_2 & t_2 \\ 0 & 1 \end{pmatrix}$, where W denotes the world's frame of reference, R is a 3×3 rotation matrix and t a 3×1 translation vector. We want to find the transformation matrix P_{21} that defines the transformation from C1 to C2. We can just use P_{W1} and P_{W2} to find this, since they share a similar view. The first step is to convert a point q_1 in the C_1 space to a common world space through P_{W1}:

$$q_w = t_1 + R_1 * q_1 \tag{1}$$

Now, given a point q_w in the world's frame of reference, we can invert the camera pose transformation to write a point q_2 in C_2 as:

$$q_2 = R_1^{-1} * (q_w - t_2) \tag{2}$$

Substituting (1) in (2) we obtain P_{21}, the searched transformation from a source camera space - C_1 - to a target camera space - C_2 - so that:

$$q_2 = P_{21} * q_1 \tag{3}$$

where:

$$P_{21} = \begin{bmatrix} R_2^{-1} & R_2^{-1}(t_2 - t_1) \\ 0 & 1 \end{bmatrix} \tag{4}$$

Applying (4) to our point cloud through simple matrix multiplication brings the 3D points to the RGB camera's coordinate system. Finally, the transformed points are projected back to the RGB frame through the RGB camera's intrinsic parameters [16] provided by the camera API.

Once we have projected the points back to the RGB frame, we have to make sure that our depth values refer to points that are actually in the RGB camera view. In fact, as the depth sensor produces ultra-wide FOV images, some of the obstacles detected will not appear at all in the correspondent RGB camera frame. Moreover, as the RGB camera captures frames at a much higher resolution than the depth sensor, many pixels in the RGB frame just will not have a depth value assigned. Figure 5 shows an example of the proposed RGB-Depth mapping performed on a separate machine with the data recorded on the HoloLens.

2.6 Depth-Enhanced Landmark Detection

This section will discuss the steps taken in order to enhance the hologram to patient registration algorithm proposed in [6] with the acquired depth information. The RGB-Depth mapping process illustrated above, enabled us to assign a "distance" value to

pixels in the RGB frames, which we can use to better estimate an object's location in the world. Moreover, through this approach, we could directly build upon the foundation laid by Pepe's team's work, developed around the Dlib RGB-based face detection algorithm.

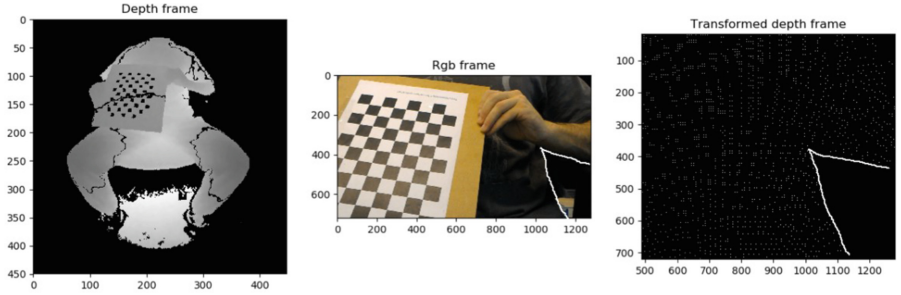

Fig. 5. An example of our RGB-Depth mapping. The white markings highlight the correspondence between the frames.

In particular, we decided to use the depth data to correct the estimated position of the patient's nose tip, one of the landmark points searched by Dlib in order to recognize a human face, like in Fig. 6.

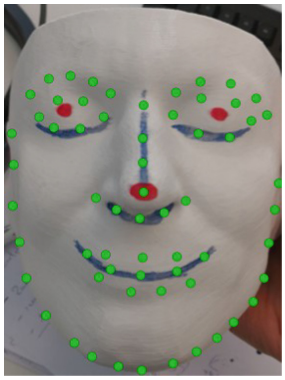

Fig. 6. The Dlib face detection software locating the face landmarks on a 3D-printed head [6].

With the mapped depth values in hand, we moved on in a similar fashion to Pepe's study: We determined the direction of the nose tip landmark point through pixel unprojection. Then, we scaled its position according to the relative depth value, which for a front facing subject is easily found as the point of the patient's face closest to the HoloLens user. This process provides us with an estimate of the patient's head's position but doesn't tell us about its orientation.

The task of recovering an object's orientation with respect to the camera pose from a set of 3D points and their projection on the frame is called a Perspectiven-Point (PnP) problem and can be solved by common computer vision libraries like OpenCV.

3 Results

Due to the nature of the experience offered by HoloLens to the user, it can be hard to assess its visualization performance objectively. Moreover, due to the very recent release of Research Mode, there is, at the time of writing, close to no documentation about the HoloLens depth sensor's accuracy. For this reason, we decided to evaluate the accuracy also in scenarios not exactly related to our particular use-case, to the advantage of future studies relying on Research Mode data.

First, we considered a flat surface in our laboratory, to assess the smoothness of the relative point cloud reconstruction. The measurement was repeated 10 times, by analyzing the point clouds extracted from 10 different depth frames, taken at a distance of about 60 cm from a wall (Table 1).

Table 1. HoloLens' depth sensing accuracy - planar surface smoothness.

RMS error (mean ± standard deviation)	2.4 ± 0.4 mm

Next, we performed relative distance measurements for 3D objects, namely a sharp-edged wooden box and the 3D-printed head model already used for testing purposes. For the box, the measure was performed on all the 3 dimensions, while for the 3D-printed head only the nose tip-to-bearing plane distance was evaluated. Each measurement was repeated 5 times and performed at a distance of about 60 cm from the target (Table 2).

Table 2. HoloLens' depth sensing accuracy - wooden box dimensions.

Relative error in the back-front dimension (mean ± standard deviation)	0.033 ± 0.018
Relative error in the up-down dimension (mean ± standard deviation)	0.079 ± 0.028
Relative error in the right-left dimension (mean ± standard deviation)	0.044 ± 0.021

The primary goal of this work is to assess the effects of introducing additional depth information in the hologram-to-patient registration process, reportedly one of the major weak spots in Pepe's study [6] (Table 3).

Table 3. HoloLens' depth sensing accuracy - 3D-printed head facial features.

Relative error in the nose tip - to bearing plane distance (mean ± standard deviation)	0.018 ± 0.011

The technical results, shown in Table 4, are generally in line with what is found in the previous study, but still interesting considering that this preliminary work only exploits a very small portion of the available sensors' data. Again, the measures were repeated 5 times, at a distance of approximately 60 cm.

Table 4. Hologram-to-patient registration error: comparison.

Measured value	Proposed method	Previous method
Error in the back-front dimension (mean ± standard deviation)	3.8 ± 1.7 mm	−4.5 ± 2.9 mm
Error in the up-down dimension (mean ± standard deviation)	−8.6 ± 3.7 mm	3.3 ± 2.3 mm
Error in the right-left dimension (mean ± standard deviation)	−2.2 ± 1.5 mm	−9.3 ± 6.1 mm

4 Conclusion

We evaluated the potential of the Microsoft HoloLens depth sensing capabilities in enabling accurate and seamless imaging data visualization in maxillofacial surgical procedures. In this work, we addressed the most prominent issue of the previous implementations: the bottleneck in hologram-to-patient registration accuracy, which is limited depth perception. Invaluable for this purpose was HoloLens' Research mode, which, only recently released by Microsoft, provided us with a stream of depth data previously unavailable to researchers. First, the data, in the form of a stream of range images, was processed in order to obtain a point cloud representation of the user's view. Then, a conversion pipeline, to map the depth information onto the RGB frames' pixel was established. Given the only recent availability of the data, no information on the sensor's accuracy was found in literature, for this reason, an accuracy evaluation in different measurement scenarios was performed. Ultimately, we proceeded to integrate the mapped depth information into the RGB-based application developed in [6], as pattern recognition techniques like the ones here employed for face detection can be heavily affected by inaccurate spatial perception. Overall, our study found that the rich set of on-board sensors can be exploited beyond its intended use user interface, environment navigation - turning the headset in something more than a mere visualization device. The accuracy evaluation, in fact, demonstrated the device potential for millimeter-accuracy measurement, comparable to other commercially available sensors. With regard to hologram registration, although only slight improvements from the previous study were found, we see huge potential for more spatial aware HoloLens applications. Future developments, in fact, could easily make more extensive use of the headset sensors data, for example considering a 3D-to-3D registration approach, in order to overcome the inherent flaws of two-dimensional pose estimation.

References

1. National Institutes of Health: Technologies Enhance Tumor Surgery: Helping Surgeons Spot and Remove Cancer. News in Health (2016)
2. Chen, X., et al.: Development of a surgical navigation system based on augmented reality using an optical see-through head-mounted display. J. Biomed. Inform. **55**, 124–131 (2015)
3. Perkins, S.L., Lin, M.A., Srinivasan, S., Wheeler, A.J., Hargreaves, B.A., Daniel, B.L.: A mixed reality system for breast surgical planning. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), Nantes, pp. 269–274 (2017)
4. Pratt, P., et al.: Through the HoloLens looking glass: augmented reality for extremity reconstruction surgery using 3D vascular models with perforating vessels. Eur. Radiol. Exp. **2**(1), 2 (2018)
5. Wang, J., Suenaga, H., Yang, L., Kobayashi, E., Sakuma, I.: Video see-through augmented reality for oral and maxillofacial surgery. Int. J. Med. Rob. Comput. Assist. Surg. **13**, e1754 (2017)
6. Pepe, A., et al.: Pattern recognition and mixed reality for computer-aided maxillofacial surgery and oncological assessment. In: 2018 11th IEEE Biomedical Engineering International Conference (BMEiCON), pp. 1–5 (2018). https://doi.org/10.1109/BMEiCON.2018.8609921
7. Dlib C++ library. http://www.dlib.net
8. Bevilacqua, V., D'Ambruoso, D., Mandolino, G., Suma, M.: A new tool to support diagnosis of neurological disorders by means of facial expressions. In: IEEE International Symposium on Medical Measurements and Applications, Medical Measurements and Applications Proceedings (MeMeA2011), pp. 544–549 (2011). https://ieeexplore.ieee.org/document/5966766/
9. Bevilacqua, V., Biasi, L., Pepe, A., Mastronardi, G., Caporusso, N.: A computer vision method for the italian finger spelling recognition. In: International Conference on Intelligent Computing (ICIC2015), vol. 9227, pp. 264–274 (2015). https://doi.org/10.1007/978-3-319-22053-6_28
10. Bevilacqua, V., et al.: A comprehensive method for assessing the blepharospasm cases severity. In: Santosh, K.C., Hangarge, M., Bevilacqua, V., Negi, A. (eds.) RTIP2R 2016. CCIS, vol. 709, pp. 369–381. Springer, Singapore (2017). https://doi.org/10.1007/978-981-10-4859-3_33
11. Microsoft: HoloLens Research Mode Tutorial at CVPR (2018). https://docs.microsoft.com/it-it/windows/mixed-reality/cvpr-2018
12. Microsoft: HoloLensForCV C#/C++ library. https://github.com/Microsoft/HoloLensForCV
13. MathWorks: pcfitplane - fit plane to 3D point cloud. https://www.mathworks.com/help/vision/ref/pcfitplane.html
14. Microsoft: Process Media frames with MediaFrameReader. https://docs.microsoft.com/en-us/windows/uwp/audio-video-camera/processmedia-frames-with-mediaframereader#setting-up-your-project
15. Guyman, W., Zeller, M., Cowley, E., Bray, B.: Locatable camera, Microsoft - Windows Dev Center, 21 March 2018. https://docs.microsoft.com/en-us/windows/mixedreality/locatable-camera
16. Microsoft: CameraIntrinsics Class. https://docs.microsoft.com/en-us/uwp/api/windows.media.devices.core.cameraintrinsics