

# Improving the Quality of Image in Virtual Reality Applications for Industry

MIHALACHE GHINEA

Department of Machines and Manufacturing Systems  
University POLITEHNICA of Bucharest  
Splaiul Independenței, 313, sector 6, 060042, Bucharest  
ROMANIA  
[ghinea2003@yahoo.com](mailto:ghinea2003@yahoo.com), [www.pub.ro](http://www.pub.ro)

GICU CĂLIN DEAC

Department of Machines and Manufacturing Systems  
University POLITEHNICA of Bucharest  
Splaiul Independenței, 313, sector 6, 060042, Bucharest  
ROMANIA  
[gicu.deac@gmail.com](mailto:gicu.deac@gmail.com), [www.pub.ro](http://www.pub.ro)

CRINA NARCISA GEORGESCU

Department of Machines and Manufacturing Systems  
University POLITEHNICA of Bucharest  
Splaiul Independenței, 313, sector 6, 060042, Bucharest  
ROMANIA  
[crina.deac@gmail.com](mailto:crina.deac@gmail.com), [www.pub.ro](http://www.pub.ro)

*Abstract:* One of the pylons of the Industry 4.0 is augmented and virtual reality. It can improve the perception on the industrial processes in real time and more. The purpose of this research is to provide a practical guide and an effective starting point regarding the real possibilities to extend the way of using VR HMDs (e.g.: Oculus Rift, HTC Vive etc.), gathering together Virtual Reality, Augmented Reality and Gesture Interaction in a unique project. The project relies on the Leap Motion controller. This controller has an integrated infrared depth camera which can be used as an input video feed for stereoscopic view on Oculus Rift, in the same time, projecting the raw images of your hands into a 3D mesh that can interact with other objects in real-world space (in our case a 3D model of a complex product). The part of the mesh that you can see is covered by the passthrough from the twin Leap Motion cameras. Each camera provides a separate view of your hands, so that what you see has actual depth. We can interact with the 3d model and show some functional animations. We present a part of an industrial application of VR used into a common industry (door locks manufacturing industry), which can be useful for research and for training and advertising, too.

*Keywords:* virtual reality, augmented reality, Oculus Rift, Leap Motion, Industry 4.0

## 1 Introduction

Augmented reality (AR) is a live direct or indirect view of a physical, real-world environment whose elements are augmented (or supplemented) by computer-generated sensory input such as sound, video, graphics or GPS data [8][11][14]. It is related to a more general concept called mediated reality, in which a view of reality is modified (possibly even diminished rather than augmented) by a computer. As a result, the technology functions by enhancing one's current perception of reality [1][2].

By contrast, virtual reality replaces the real world

with a simulated one. Augmentation is conventionally in real-time and in semantic context with environmental elements, such as sports scores on TV during a match [3][4].

With the help of advanced AR technology (e.g. adding computer vision and object recognition) the information about the surrounding real world of the user becomes interactive and digitally manipulable. Information about the environment and its objects is overlaid on the real world [1][3][4][5][6][7]. This information can be virtual or real, e.g. seeing other real sensed or measured information such as

electromagnetic radio waves overlaid in exact alignment with where they actually are in space [9][10].

The emergence of new devices such as e.tablets, smartphones or soon to appear Augmented Reality eye-wear has encouraged the development of applications especially in the area of industrial maintenance and informative or educational geolocation. Uses of virtual reality in systems such as parts analysis and simulation, staff support, layout and construction planning or supervising can no longer be set in the future. Augmented Reality as a Virtual Support to Industries.

The development of Augmented Reality Technology to assist industries with precise site or field information in real-time is today a foreseeable reality to be used in almost any domain. For instance, a project management in building and construction will be more easily and safely carried out when site managers can virtually view and monitor work in progress in real time through Augmented Reality markers placed on parts or equipment being built. Pointing a camera to factory on-site piece of equipment can match it to the digital map of the plant and verify if it is in its designated location, not only freeing the staff from cumbersome paper layout plans but also providing the operators with virtual reality contextual information.

Industry decision makers can make timely decisions when management foresees how a piece of equipment or a machine once built will fit in its final environment, by merely looking at the superimposition of field data fed through Augmented Reality Systems.

### 1.1 Enhancing productivity and work behavior

It is now a technological feasibility to mix Augmented Reality with geolocation, recognition and tracking technologies. The novelty of instant information coupled with enhanced perception will ensure that Augmented Reality systems play a big role in how people and companies work in the future.

Milgram defined a continuum of Real to Virtual environments, where Augmented Reality is one part of the general area of *Mixed Reality* (Figure 1). In both Augmented Virtuality and Virtual Environments (a.k.a Virtual Reality), the surrounding environment is virtual, while in AR the surrounding environment is real. This survey focuses on Augmented Reality and does not cover Augmented Virtuality or Virtual Environments [12][13][14].

Until recently, most AR interfaces were based on the desktop metaphor or used designs from

Virtual Environments research. One main trend in interaction research specifically for AR systems is the use of heterogeneous designs and tangible interfaces. Heterogeneous approaches blur the boundaries between real and virtual, taking parts from both worlds. Tangible interfaces emphasize the use of real, physical objects and tools. Since in AR systems the user sees the real world and often wishes to interact with real objects, it is appropriate for the AR interface to have a real component instead of remaining entirely virtual.

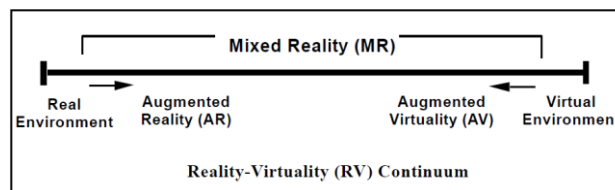


Fig.1 Milgram's Continuum of real to virtual environments

Virtual Reality (VR) is all about the creation of a virtual world that users can interact with. Users are isolated from the real world while immersed in a full synthetic environment, as far as the immersive experience is achieved by the wearing of a VR helmet (e.g.: Oculus Rift).

On the other hand, Augmented Reality (AR) blends virtual reality contents with the real world. As a result, users can interact with virtual contents while continuing to be in touch with the real life around them. This experience is made possible by the wearing of AR glasses (e.g.: Google Glasses).

Technically speaking, the main difference between the two systems seems to be a transparency mask issue. VR scenes are completely opaque over the entire display (i.e. you cannot see the real world around you). AR glasses are transparent (you see the real environment) and pixels where virtual contents are drawn have an opacity value in order to overwrite the real world light information. From this point of view, VR is simply a transparency mask limit case of AR, where transparency has a null value over the entire display.

Often it seems that VR and AR are two different worlds that do not overlap. But what if we imagine VR and AR in the same next generation device? Why should not have a unique system that could provide both AR and VR experiences? Simply because it is not so simple as it looks like.

## 2 Research activity

The purpose of these research is to provide a practical guide and an effective starting point regarding the ambitious and real possibilities to

extend the way of using **VR HMDs** (e.g.: Oculus Rift, HTC Vive etc.), gathering together Virtual Reality, Augmented Reality and Gesture Interaction in a unique project (Figure 2).

The project relies on the **Leap Motion** controller, a small puck-like device that tracks the motion of your arms and hands.



Fig.2 Leap motion controller mounted on Oculus Rift

This controller have an integrated infrared depth camera wich can be used as an input video feed for stereoscopic view on **Oculus Rift**, in the same time, projecting the raw images of your hands into a 3D mesh that can interact with other objects in real-world space. The part of the mesh that you can see is covered by the passthrough from the twin Leap Motion cameras. Each camera provides a separate view of your hands, so that what you see has actual depth.

This hybrid effect has a powerful impact in VR because your real hands can now actually pass through (or disappear behind) virtual objects. The hands interact properly with other 3D objects because they are 3D – complete with the interactive and visual capabilities that you expect. This approach also reduces jitter effects, since there is no longer an artificially generated rigged hand to worry about. While the hand image in VR might shift or become transparent, it will not suddenly shift in the wrong direction.

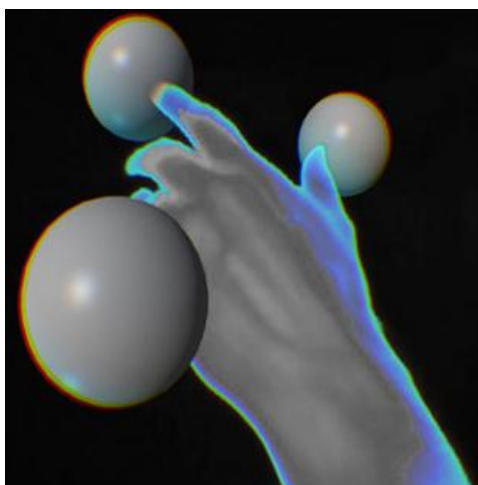


Fig. 3 Image hands preview

The **Image Hands** are also designed to provide

users with dynamic feedback on the Leap Motion Controller's tracking confidence [1]. When your hand assumes a high-confidence pose, it glows in blue. The glow disappears as confidence values drop (Figure 3).

The Leap Motion controller uses infrared stereo cameras as tracking sensors. You can access the images from these cameras using the **Controller.Images** or **Frame.Images** functions [1][16]. These functions provide an **ImageList** object, containing the **Image** objects. **Controller.Images** provides the most recent set of images. **Frame.Images** provides the set of images analysed to create that frame and can be slightly obsolete compared to the images returned directly by the Controller.

When it obtains an image from one of the cameras, a grid highlighting the significant and complex distortion is superimposed on the image (Figure 4). The images can be used for: **Head-Mounted Display** video pass-through, **Augmented Reality**, **Computer Vision**. The Image API (Application Programming Interface) provides a buffer containing the sensor brightness values and a buffer containing the camera calibration map, which can be used to correct lens distortion and other optical imperfections in the image data[1][16].

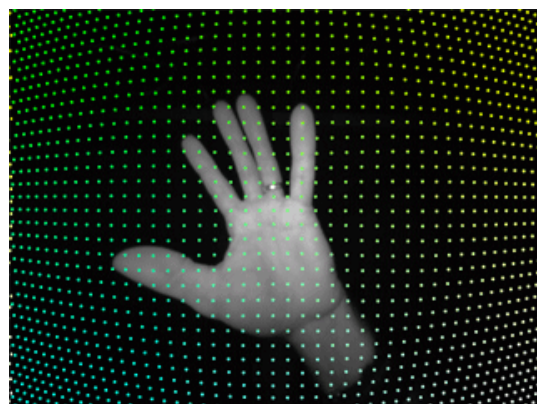


Fig.4 Distorted image from camera

## 2.1 Image Distortion

When a ray of light enters one of the Leap Motion cameras, the lens bends the light ray so that it hits the sensor, which records it as a greyscale brightness value at a specific pixel location. Of course, no lens is perfect, so a ray of light does not land on the sensor in the optically perfect spot. The calibration map provides data to correct this imperfection, allowing you to calculate the true angle of the original ray of light. You can use the corrected angle to generate a distortion-free image, and, using the angles from both images in the stereo pair, you can triangulate the 3D location of a feature identified in both images. Note that the calibration

map corrects lens distortion; it does not correct perspective distortion [1][15].

### 2.2 Image Ray Correction

After image orientation, getting raw images and calibration map, it must make the correction of the image ray.

You can correct the raw image distortion in two ways [1][15][16]:

- Use the **Image.Warp()** and **Image.Rectify()** functions.
- Use the data in the **Image.Distortion** buffer directly.

The **Warp()** and **Rectify()** functions are the simpler method, but processing each pixel individually on the CPU is relatively slow. Use these functions if you are only correcting a few points, you don't need to process data in real time, or when you cannot use GPU shaders. The Distortion buffer is designed to be used with a GPU shader program and can correct the entire raw image while maintaining a good application frame rate.

#### 2.2.1 Correction using bilinear interpolation

In situations where shaders are not feasible, you may be able to correct image distortion faster using well-optimized bilinear interpolation than when using the **Warp()** function. (As with any such optimization, you should verify your results with performance testing.)

Recall that the distortion map contains a 64x64 element grid. Imagine these grid elements evenly spread out over your target image (with element [0, 0] in the lower-left hand corner and [64,64] in the upper-right). Each element contains a horizontal coordinate and a vertical coordinate identifying where in the sensor image data to find the recorded brightness for that pixel in the target image. To find the brightness values for pixels in between the distortion grid elements, you have to interpolate between the four nearest grid points (Figure 5).

The base algorithm for finding the distortion-corrected brightness for a given pixel in the target image is:

- 1) Find the four points in the calibration grid surrounding the target pixel.
- 2) Calculate the interpolation weights based on the distance of the target to each surrounding grid point.
- 3) Lookup the horizontal and vertical values at each of the four grid elements.
- 4) Bilinearly interpolate the horizontal value using the distance-based weighting factors.
- 5) Repeat this interpolation for the vertical value.
- 6) Reject any points where either the horizontal or vertical value is outside of the range [0..1]. There is no recorded data for such points.

- 7) Denormalize the values so that they represent pixel coordinates into the raw sensor data.

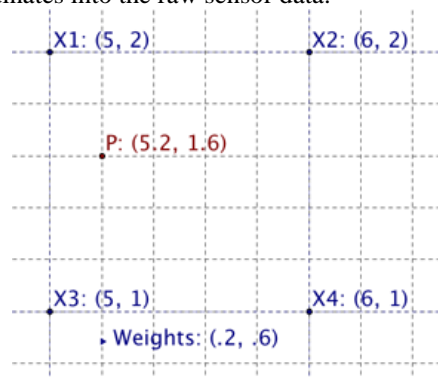


Fig. 5 The grid with four closest points

- 8) Look up the sensor value at the computed pixel coordinates.

- 9) Set this brightness value at the original coordinates in the target image.

#### 2.2.2 Draw Tracking Data over Image

It is reasonably straightforward to draw representations of the Leap Motion tracking data over the camera image [15]. If you have drawn the raw image data to a bitmap, you can find the pixel corresponding to a Leap Motion position using the **Warp()** function [15][16].

Converting a position in Leap Motion coordinates to horizontal and vertical slopes (from the camera perspective) requires knowing how far the cameras are from the origin of the Leap Motion coordinate system. For the current peripheral version, the offset on the x axis is 20mm to either side. The cameras are on the x-axis, so there is no z offset. The slope is simply the distance from the camera in the image plane – the x-coordinate for the horizontal slope; the z-coordinate for the vertical slope – divided by the distance to the image plane, the z-coordinate. The diagram from figure 6 illustrates the geometry for the horizontal slope.

The calculation is shown for the left camera; add the offset distance instead of subtracting for the right camera (Figura 6).

Once we know the ray slope values, we can get the pixel coordinates using **Warp()**.

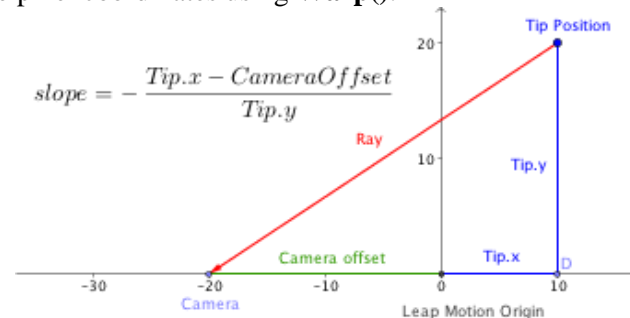


Fig.6 The geometry for the horizontal slope

If you have rendered the corrected image data, then correlating tracking data to the image data depends on how you rendered the image. For 3D scenes, this is a matter of using a consistent scale and correct placement of the textured quad showing the camera image. For other types of rendering, you must convert the ray slopes representing a Leap Motion position to a target image pixel according to the way that you corrected the image data.

### 2.2.3 Calculate the direction to an *Image Feature*

We can get the direction to an image feature with the **Image.Rectify()** function. **Image.Rectify()** returns a vector containing the horizontal and vertical slopes (as defined from the camera point of view) given the pixel coordinates in the raw image data.

We use a 3d object modelled in Catia and imported in Cinema 4D where we created all the animations. We have two scripts, one script for reading point clouds of the 3d model and another one for extracting these points, addressing the interaction with the model with image hands [1].

## 3 Finding improvements

With a satisfactory precision, CAD models can be imported into certain programs for 3D animation. This fact is more and more important for applications in industry, medicine, biology (to name only few of them), as they need quite realistic image. Thus, Image API applications become increasingly useful when pretty realistic results are required. In figure 7, the real image of a door lock system, together with all its parts, is presented.

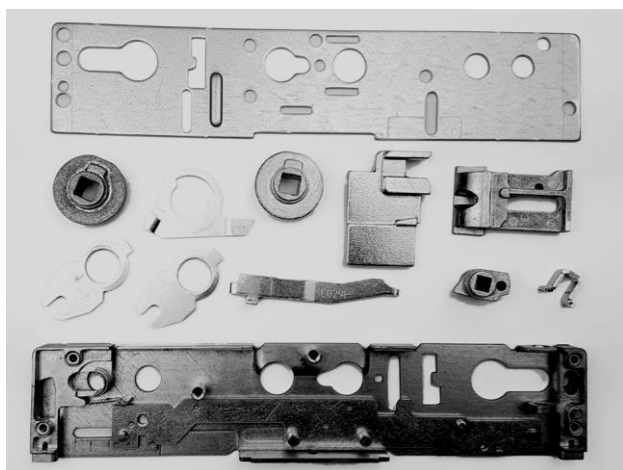


Fig.7 The parts of the door lock system

Conceiving a CAD model (Figure 8) by means

of an application such as CATIA could lead to a virtual representation of the real model with a 100% accuracy. Although 3D scanning could be a solution, as well, even faster and cheaper, the inconsistencies that introduce the real model must also be taken into consideration.

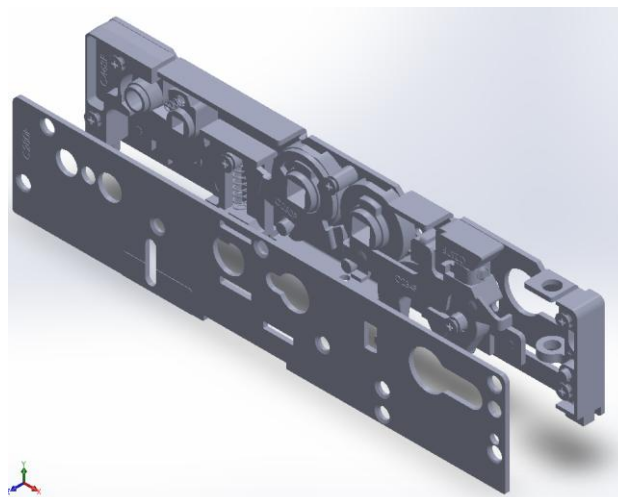


Fig.8 The CAD model of the door lock system

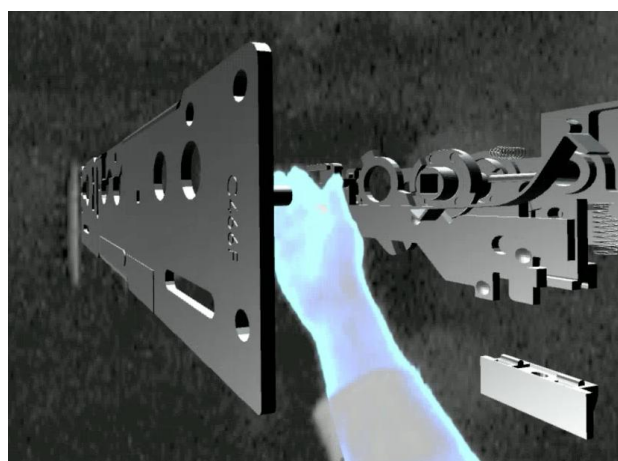
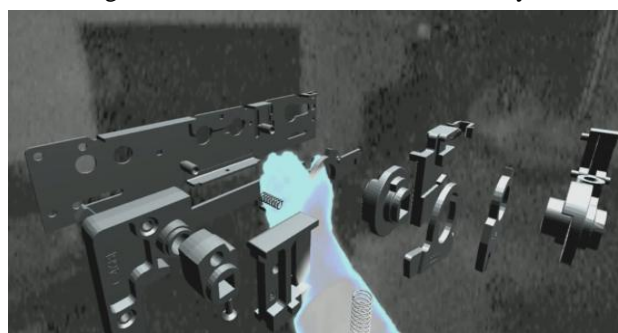


Fig. 9 3d model moved in VR by image hands

The present article exhibits the collective endeavors of the research team aiming to achieve a post-processing software that enables the virtual immersion as being the real one. Unfortunately, only by using virtual immersion equipment one can understand how impressive the advancement is. The images in Figures 9, are the ones precessed by

means of the above mentioned method.

#### 4 Conclusions

All over the world, VR technology is used today for training applications in a variety of process industries, and enables personnel subjection to simulated hazardous situations in a safe, highly visual and interactive way.

Customized simulations of plants layouts, dynamic processes and comprehensive virtual environments can be set up and allow users to move within the virtual plants or systems, making operational decisions and investigating processes at a glance.

Our aim is to couple the activity of AVRENG lab (Augmented & Virtual Reality for Engineering), from University POLITEHNICA of Bucharest, with virtual reality and virtual environments applications helpful for future industrial workspaces. We want to gather expertise from partner members and determine the future research agenda for the development and use of virtual reality (VR) technologies. The working team on Education and Training is specifically focused on understanding how VR is used to support learning in educational and industrial contexts.

This paper represents the very first steps of VR technology currently in use or development for training in industry. It remains to identify potential future development of VR training applications and also to overcome the existent barriers.

For the next version of our application, we think about implementing a color stereoscopic camera to provide our industrial partners with an improved experience and a higher accuracy.

#### References:

- [1] Ghinea, M., Deac, G.C, Georgescu, C.N. *Training applications for industries by means of virtual reality technology*, In Proceedings of Augmented Reality for Technical Entrepreneurs-1<sup>st</sup> ed. ARTE, Bucharest, 2016, pp.26-32.
- [2] Graham, M., Zook, M., and Boulton, A. , *Augmented reality in urban places: contested content and the duplicity of code*, Transactions of the Institute of British Geographers, DOI: 10.1111/j.1475-5661.2012.00539.x 2012.
- [3] Steuer, J. *Defining Virtual Reality: Dimensions Determining Telepresence*, Department of Communication, Stanford University, 15 Oct., 1993.
- [4] \*\*\* Introducing Virtual Environments National Center for Supercomputing Applications, University of Illinois.
- [5] Chen, B.X. If You're Not Seeing Data, You're Not Seeing, Wired, 25 August 2009.
- [6] Maxwell, K. *Augmented Reality*, Macmillan Dictionary Buzzword.
- [7] Azuma, R. A Survey of Augmented Reality Presence: Teleoperators and Virtual Environments, pp. 355–385, August 1997.
- [8] Zhanpeng, H., Pan H., et al. Mobile augmented reality survey: a bottom-up approach.
- [9] *Phenomenal Augmented Reality*, IEEE Consumer Electronics, Vol. 4, No. 4, October 2015.
- [10] Archibald, C., Petriu, E. *Time-frequency perspectives, with applications*, in Advances in Machine Vision, Strategies and Applications, World Scientific Series in Computer Science: Volume 32.
- [11] Metz, R. *Augmented Reality Is Finally Getting Real* Technology Review, 2 August 2012.
- [12] Fleet Week: *Office of Naval Research Technology- Virtual Reality Welder Training*, eWeek, 28 May 2012.
- [13] Rolland, J. Baillott, Y. Goon, A..A., *Survey of Tracking Technology for Virtual Environments*, Center for Research and Education in Optics and Lasers, University of Central Florida.
- [14] Azuma, R., Balliot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B. *Recent Advances in Augmented Reality*, Computers & Graphics, November 2001.
- [15] <https://developer.leapmotion.com/documentation/csharp/index.htm>
- [16] <http://iiif.io/api/image/2.0>