Int J Interact Des Manuf
DOI 10.1007/s12008-017-0377-0

ORIGINAL PAPER

Improved virtual reality perception with calibrated stereo and variable focus for industrial use

Álvaro Segura[1] · Javier Barandiaran[1] · Aitor Moreno[1] · Iñigo Barandiaran[1] · Julián Flórez[1]

Received: 11 January 2017 / Accepted: 19 January 2017

Abstract 3D image rendering for virtual reality HMDs is typically generated based on a set of parameters taken from the manufacturer-supplied specifications and an idealised perspective model. But actual displays may vary in build features that alter spatial perception. Moreover, the eye focusing effort is often ignored. The resulting visual discomfort and incorrect geometry perception has discouraged use of immersive virtual reality in industrial applications. This work addresses these issues and describes a system with per-device calibration and control of stereo perspective projections and focus. The ideas presented may improve the usefulness of VR in industrial training and visualization.

Keywords Virtual reality · Stereo vision · Industrial training · 3D rendering

1 Introduction

The Industry 4.0 paradigm initially introduced in Germany [12] is quickly spreading world-wide. It brings an opportunity to boost manufacturing productivity by leveraging modern ICT technologies that have been maturing in the last decade. Posada et al. [19] specifically propose visual computing as a key enabling technology for Industry 4.0.

Among the technologies advocated, the Industry 4.0 concept includes simulation together with virtual and augmented reality to enable digital twins of manufacturing assets. Several applications exist in different stages of the product life cycle: from the factory planning phases, through the plant or process visualisation, to the training of factory workers in machine handling. Choi et al. [2] recently surveyed the industrial applications of virtual reality. VR can also be used as a place for collaboration in a shared virtual space, as introduced by Galambos et al. [7]. We argue that immersive VR has a potential use in manufacturing engineering and training but needs to overcome some issues.

With the availability of a new generation of immersive head-mounted displays (HMD), virtual reality is gaining interest for industrial applications. However, some issues still prevent virtual reality from being perceived similar enough to actual reality. A well known issue is the vergence–accommodation mismatch [21], which refers to the fact that displays are flat surfaces so the eyes have to focus (or accommodate) at a fixed distance while stereoscopic disparity makes the eyes converge at varying distances. Additionally, stereoscopic projection may not be correctly computed to make the user perceive correct distances and shapes as in the real world. These problems lead to the necessity of having controlled variable focus in the display hardware. But this feature has rarely been implemented in VR systems in practice.

Rendering correct stereoscopic 3D images for HMDs requires the graphics engine to know its projection parameters. These parameters, such as the field of view (FOV) are usually taken from the manufacturer-supplied nominal values. For higher accuracy some methods for calibrating actual values from specific HMDs have been proposed, such as [15]

✉ Álvaro Segura
asegura@vicomtech.org

Javier Barandiaran
jbarandiaran@vicomtech.org

Aitor Moreno
amoreno@vicomtech.org

Iñigo Barandiaran
ibarandiaran@vicomtech.org

Julián Flórez
jflorez@vicomtech.org

[1] Vicomtech-IK4, Mikeletegi 57, 20009 San Sebastián, Spain

Published online: 08 February 2017

(which is based on subjective user impressions) and [9] (using objective measurements from a camera looking through the HMD).

These aspects have to work in harmony for images to be perceived correctly in terms of geometry, perspective, depth, vergence and accommodation. This paper proposes a virtual reality system with improved visual 3D perception that can mitigate the effects of the convergence–accommodation mismatch problem. Two aspects are considered: (1) correct stereoscopic projection is achieved by an objective calibration procedure, and (2) correct focus is achieved by a variable optical focus mechanism. This work contributes a stereo calibration method for HMDs based on a stereo camera and suggests a strategy for controlling focus.

2 Objective stereo display calibration

In 3D computer graphics, images are rendered based on a set of parameters that define the projection characteristics. At least the field of view (FOV) must be defined to control perspective. A small FOV produces a view similar to a *tele* setting in a camera lens, while a large FOV creates a more perspective-distorted view similar to a *wide angle* lens. In order to create a faithful view in virtual reality the FOV used in rendering must match the field of view that the display appears to have from the point of view of the user wearing the HMD.

When producing stereoscopic projections, additional parameters are used to control the final effect. They should correspond to the viewing conditions. If the stereoscopic projections do not match viewing conditions, a different virtual space will be perceived, virtual distances, sizes and shapes will be distorted. These distortions have been studied by Benzeroual et al. [1] (for 3D films) and by Kelly et al. [14] (for VR environments).

In real life, when staring at a point in space, the eyes rotate such that their visual axes intersect at that point. The distance to this intersection point, the convergence distance, is the distance at which we perceive the point to be. When looking at a very distant object, visual axes remain approximately parallel. A correct stereoscopic perspective projection should have the same properties.

Modelling visual axes and their intersection in a stereoscopic *screen* only requires knowing the position of the user in front of the screen, the user's interpupillary distance and the screen dimensions. For more precision, the user's head orientation can be used to find the location of each eye relative to the screen.

In the case of a *head-mounted display* the situation becomes more complicated due to the fact that the left and right images use separate microdisplays which are seen through separate optics. The lenses of the HMD distort and magnify the image of the microdisplays. The dimensions and precise position of the microdisplays are also distorted and magnified. So, small positioning errors of each display may lead to significant misalignments affecting the stereoscopically perceived geometry. Horizontal misalignment alters depth perception as the vergences will be altered leading to a different convergence distance. And vertical misalignment may lead to double vision and no depth perception at all: the eyes can only fuse if left and right corresponding stimuli are within a small angular elevation difference. This is known as vertical disparity amplitude tolerance (see [6,22]).

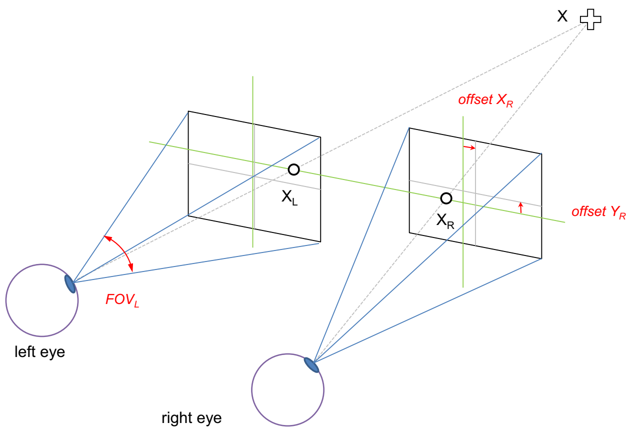

Fig. 1 View of the display from each eye. Each display is offset with respect to the perfectly centred position. Thus, in order to represent a point that is perceived at position **X**, its *left* and *right* positions $\mathbf{x_L}$, $\mathbf{x_R}$ have to be drawn at vertically and horizontally offset positions in each display

Most VR rendering systems assume that the microdisplays of an HMD are centred in front of each of the user's eyes. So, the projection matrices that generate the images on the HMD are known to be symmetric perspective projection matrices.

However, usually the microdisplays have small offsets (see Fig. 1). Even HMDs of the same model can have different offsets. In order to deliver a correct view in the HMD, the VR rendering system must know these offsets to compute the correct projection matrices. So we need a simple, precise and objective calibration method that finds these data for each HMD. Figure 1 shows the correct view frustums that must be found.

Figure 2 (top) shows what happens if the offsets are not considered. In the figure the offsets are exaggerated to illustrate the concept. However, very small offsets affect viewing perception and comfort. Figure 2 (bottom) shows that a correct compensation of the frustums create aligned images.

Several calibration procedures have been proposed for different stereoscopic displays. For CAVE-like setups Ponto et al. [18] introduced a perceptual calibration procedure. The calibration of see-through HMDs has been addressed by many authors, such as [3,4,8,10,13,16]. Indeed, correct pro-

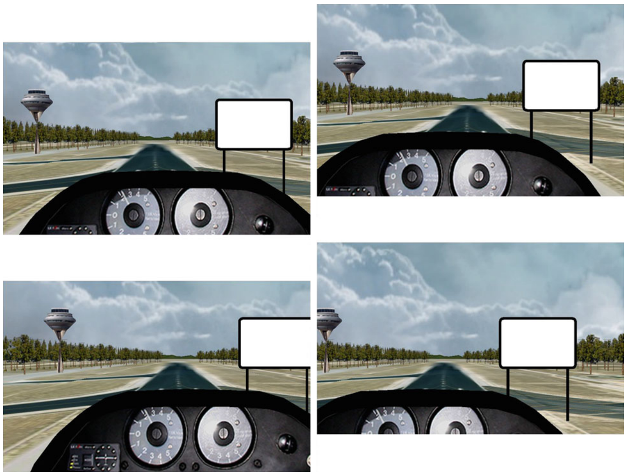

Fig. 2 *Top* the effect of misaligned HMD displays with conventional stereo rendering. The right display is above the left display so that the *left* and *right* views are vertically misaligned, show incorrect disparity and cannot be fused by the user. The offsets are exaggerated to illustrate the concept. *Bottom* the effect of misaligned HMD displays calibrated to provide offset-corrected stereo rendering. Now the renders apply offsets to generated images and they appear aligned and with the correct stereo disparity

jection in those devices used in augmented reality is crucial or otherwise virtual and real objects will appear misaligned. These methods take advantage of the see-through nature of those devices and require users to look at physical targets placed in front of the HMD. They are thus not usable in immersive HMDs (i.e. non-see-through) in which there is no view of the real world. In immersive HMDs the lack of real references makes projection errors less noticeable but still a correct geometric perception requires correct projection parameters. In this case applications usually either use the device's nominal parameters or employ subjective calibration methods, such as [15], because of the lack of externally observable features (only the wearer sees the HMD's displays).

In contrast, Gilson et al. [9] proposed an objective calibration method for immersive HMDs that uses one camera placed in the position of the user's eye and is an evolution of their earlier method for see-through HMDs [8]. The camera captures a pattern presented in the HMD display. From analysis of the captured pattern, they compute a mapping between the camera image and the HMD display coordinates. Then the HMD is removed while keeping the camera still and a marker is placed in the space in front. The marker is tracked with an external tracking system while it moves in the space in front of the camera. By relating the projection of the marker in the camera image, its tracked position in space and the position of the displayed pattern in the same image, the method computes the theoretical projection matrix of one display of the HMD. It then proceeds with the other display to create a stereo projection pair.

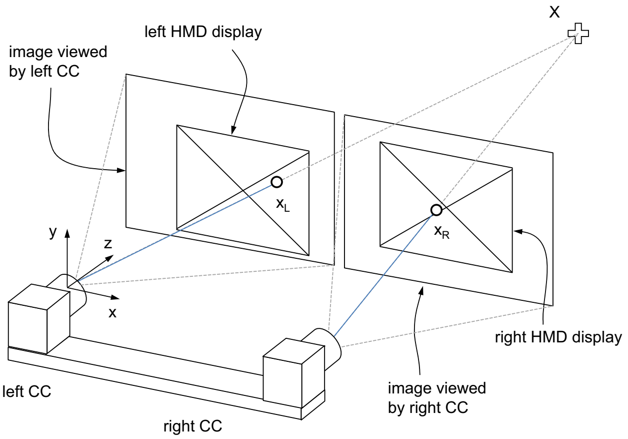

Fig. 3 HMD stereo projections calibration. A pair of rigidly attached calibration cameras (CC) substitute the eyes. The HMD displays' misalignment is exaggerated in the illustration

We see several limitations in this approach. First, it requires a complex setup including an external high-end position tracking system and a precise positioning of the camera with respect to the HMD. Then, the method calibrates the left and right displays in sequence. Stereo depth perception depends on the relation between the apparent position of points in each display. Then, such a sequential calibration can easily distort depth percepts. The authors acknowledge that a slight modification in camera position with respect to the HMD results in a different computed view frustum. Thus, the unintended different position or orientation of the calibration camera in front of each of the displays will lead to shifted visual axes and modified stereo distances.

The following subsection describes our proposed calibration method to obtain the parameters that allow the stereo system to compute view frustums for acceptable depth percepts.

2.1 New stereo view calibration procedure

Our approach shares some concepts with the one proposed by Gilson et al. but does not require an external tracking system and eliminates the uncertainty in position of the camera with respect to each of the displays, among other differences. We propose the use of a pair of rigidly attached cameras to simulate the user's eyes (*calibration cameras* or CC in Fig. 3). This pair gets a view of the virtual space projected by the rendering system as perceived by a user. The camera pair is previously calibrated, intrinsically (each camera) and extrinsically (the right camera with respect to the left). This calibration step is performed only once for our CCs and is used later on for the calibration of any HMD. Its results are the matrices of intrinsic parameters ($\mathbf{K_L}$, $\mathbf{K_R}$) and extrinsic parameters ($\mathbf{R_R}$, $\mathbf{t_R}$) that will be later referenced in Eq. (1).

We have defined a simplified projection model, suitable for low FOV and low distortion displays with small orientation and position offsets from an ideal position. The rendering module of the VR system will use view frustums defined by 6 parameters. The projection model is defined by two virtual cameras separated by the interpupillary distance. These virtual cameras have their principal axes parallel. Our *view frustum pair* (VFP) parameters are the following:

- *Interpupillary distance* or IPD. This is the separation between the two eye centres. Will be set to a fixed typical value.
- *Displays aspect ratio*. This is the ratio of width to height of the microdisplays.
- *Horizontal FOV angle of the left and right displays*. Note that these can be different, as we will see.
- *Horizontal and vertical offsets between the displays*. These parameters describe the relative displacement in both axes between the view of the microdisplays with respect to a perfectly centred position in front of each eye. They are measured with respect to display size so that, for example, a vertical value of 0.5 would mean the right display appears shifted vertically a length equal to half the screen height.

This model does not define separate offsets for each display but a relative displacement between them. This enables a much simpler operation and still ensures depth perception.

Interpupillary distance and display aspect ratio are known in advance. However FOV and offsets can vary for each specific HMD. These parameters must be known by the rendering module in the VR system in order to create correct virtual world projections (see Fig. 2). The calibration process we propose measures these parameters in a specific HMD. Even HMDs of the same model may have slightly different values. This fact shows the importance of using a simple and precise HMD calibration system.

Our calibration process is based on the projection of an arbitrary point in 3D space onto two different image planes, each with its own coordinate system: the HMD *displays coordinate systems* and the *calibration cameras coordinate systems*. In the equations in this section $\mathbf{x^{cam}}$ will denote points projected on a CC image, and $\mathbf{x^{disp}}$ will denote points projected on an HMD display. Both are in homogeneous coordinates, as they are used in projective geometry equations.

b) CC image coordinate system

Given a point $\mathbf{X}$ in 3D space and knowing the CC parameters, its projection on each CC image coordinate system is determined by Eq. 1.

$$\mathbf{x^{cam}} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X} \tag{1}$$

We have a pair of cameras, so there are actually two sets of parameters: $\mathbf{K_L}, \mathbf{K_R}, \mathbf{R_L}, \mathbf{t_L}, \mathbf{R_R}, \mathbf{t_R}$. In our case, since our extrinsic parameters take the left camera as reference, $\mathbf{R_L}$ is an identity matrix and $\mathbf{t_L}$ is a null vector. $\mathbf{K_L}$, $\mathbf{K_R}$, $\mathbf{R_R}$ and $\mathbf{t_R}$ are computed once for the CC pair and remain constant for the calibration of any HMD.

a) Display coordinate system

On the other hand, in the VR rendering system, the view frustum for each eye can be defined by a projection matrix $\mathbf{P}$. For this matrix we have chosen the form typically used in computer vision, as expressed in Eq. 2.

$$\mathbf{P} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

This matrix should be derived from the 6 model parameters we have defined. However, we do not know their values until the HMD calibration process is completed.

Each eye sees the virtual world from a different point of view. There is no rotation (i.e. the rotation matrix is an identity matrix) because we have defined our virtual cameras to be parallel and aligned with the CC Z axis. Combining projection matrix and point of view, the projection of a point $\mathbf{X}$ in 3D space by the rendering system into one of the displays is expressed by Eq. 3.

$$\mathbf{x^{disp}} = \mathbf{P}[\mathbf{I}_3|\mathbf{t}]\mathbf{X} \tag{3}$$

Again there are actually two matrices, $\mathbf{P_L}$ and $\mathbf{P_R}$ and two translation vectors $\mathbf{t_L}$ and $\mathbf{t_R}$ for the left and right displays, respectively. These subscripts will be omitted for clarity throughout this section when the specific side is irrelevant as both are processed the same way. As the equation expresses, there is no rotation as both virtual cameras have their principal axis parallel to the Z axis. And, as before, the reference system is the left eye, so $\mathbf{t_L}$ is a null vector and $\mathbf{t_R}$ is $(-\text{IPD}, 0, 0)$. $\mathbf{P_L}$ and $\mathbf{P_R}$ are the unknowns that will be computed by the calibration process.

c) Mapping between display and CCimage coordinates

For each of the left and right sides, any virtual 3D point $\mathbf{X}$ is projected onto a point $\mathbf{x^{disp}}$ in the *display coordinate system* as expressed in Eq. 3. The same 3D point is projected onto a point $\mathbf{x^{cam}}$ in the *camera image coordinate system*, as expressed in Eq. 1. These two projected points have to be equivalent but they are in different coordinate systems so their equations cannot be combined. A mapping between these two coordinate systems is needed to solve the problem. In a low distortion environment this mapping can be approximated by a 3 × 3 transform matrix **M** in homogenous coordinates (i.e. a homography) as shown in (Eq. 4).

$$\mathbf{x}^{\mathbf{disp}} = \mathbf{M}\mathbf{x}^{\mathbf{cam}} \tag{4}$$

Now we explain how this mapping **M** can be estimated. The camera images contain a view of the microdisplays (see Fig. 3). By finding a set of corresponding points in both display coordinates and CC image coordinates, the transformation **M** can be approximated. At least 4 point correspondences are needed. Actually, since the displays are usually nearly perpendicular to the camera view orientation, the mapping is approximately equivalent to an affine transformation. In that case the minimum requirement can be simplified to 3 point correspondences and the bottom row of **M** would be [0, 0, 1].

d) Computing the mapping

Our proposed calibration procedure employs an interactive process to determine this mapping. A fixed grid pattern is presented on both displays and a pair of images is grabbed with the CC cameras. Then these images are presented to a user who has to click on at least 4 specific pattern points with the mouse. We know the display coordinates of the points, as we have rendered them, and the user marks their corresponding camera image coordinates. From these point correspondences between display coordinates and camera image coordinates the algorithm computes the transform matrix **M** that relates the two images (display and camera).

e) Finding the projection parameters

Once the mapping **M** is known, we can find the elements of matrix **P** by minimization of the reprojection error as defined by the following expression.

$$\min \sum_i \left\| p(\mathbf{x}_i^{\mathbf{cam}}) - p(\mathbf{M}\mathbf{x}_i^{\mathbf{disp}}) \right\| \tag{5}$$

where $\mathbf{x}_i$ for $i = 1 \ldots N$ are the projections of a set of arbitrary virtual 3D points $\mathbf{X}_i$ and $p(\mathbf{x})$ is given by Eq. 6:

$$p(\mathbf{x}) = p([x, y, z]^T) = \begin{bmatrix} x/z \\ y/z \end{bmatrix} \tag{6}$$

Substituting the expressions from Eqs. 1 and 3 Eq. 5 expands to:

$$\min \sum_i \| p(\mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{X}_i) - p(\mathbf{M}\mathbf{P}[\mathbf{I}_3|\mathbf{t}]\mathbf{X}_i) \| \tag{7}$$

where the only unknowns are the 4 unknown elements of matrix **P**, the rendering projection matrix. The Levenberg–Marquardt algorithm [17] is applied to solve the minimization problem and estimate the 4 unknowns.

In each iteration the algorithm uses a set of N random points in the space in front of the viewer and visible by both eyes and projects them with the current value of the projection parameters. Their projection in display coordinates is transformed to camera coordinates using the computed mapping **M** and the distance to their projection by the cameras in pixels is used as reprojection error. The projection is initialized with a fixed set of parameters corresponding to the HMD's nominal perspective projection with no offsets or a standard guess if there is no such information. The Levenberg–Marquardt algorithm finds the projection matrix elements f_x, f_y, c_x, c_y that minimize reprojection error for each display. The process is executed for the left and right displays separately.

2.2 Extraction of projection parameters

The algorithm in the previous section computes a projection matrix **P** for each display. That should be sufficient as such a matrix can be converted into, for example, an OpenGL projection matrix ready for rendering graphics. But our rendering system does not use these projection matrices directly for several reasons. As the following sections will show, the projection characteristics are not fixed, but vary when varying optical power (i.e. setting different accommodation distances). For this reason the rendering engine has to continually adjust projections by interpolating values obtained in different conditions of optical power.

The computed projection matrices are affected by the orientation of the cameras with respect to the HMD. This means that the projections obtained for different optical powers will have unintended variations if the cameras are not always positioned with the exact same pose. This is why we prefer to compute a relative, inter-display offset that is unaffected by small camera deviations.

Instead of the raw projection matrices our VR system uses the 6 parameters described at the beginning of Sect. 2.1 to recreate new projection matrices for rendering. The first two parameters *Interpupillary distance* and *display aspect ratio* are known in advance. The remaining 4 (left and right *FOVs* and *horizontal and vertical interdisplay offsets*) are extracted from the elements of matrix **P** as follows.

The horizontal and vertical FOVs are related to the f_x and f_y elements. They can be extracted from the computed projection matrix as shown in Eqs. 8 and 9, where w and h are the width and height in pixels of the displays.

$$FOV_H = 2 \arctan \left(\frac{w}{2 f_x} \right) \tag{8}$$

$$FOV_V = 2 \arctan \left(\frac{h}{2 f_y} \right) \tag{9}$$

And the relative offsets are related to the principal point parameters c_x and c_y. They are the difference of these parameters between the left and right projections, divided by display resolution in order to make them resolution independent (Eqs. 10 and 11). The subscripts L and R denote the parameters for the left and right sides, respectively, which are separately obtained.

$$\text{offset}_x = \frac{c_{xL} - c_{xR}}{w} \tag{10}$$

$$\text{offset}_y = -\frac{c_{yL} - c_{yR}}{h} \tag{11}$$

The rendering engine uses the field of view value to compute a standard *OpenGL* projection matrix, and then modifies this matrix with added horizontal and vertical offsets. As we have relative offsets between displays, we apply half of each offset to the left image and half to the right image with opposite signs.

Actually, the cameras can have some optical distortion and a set of distortion coefficients are obtained as part of the camera calibration. The above camera projection equations, such as Eq. 1 do not show this effect for simplicity but it is taken into account. We apply these distortions when computing the cost function in the minimization process.

3 Focus control

Some known commercial HMDs have a manually adjustable focus setting (e.g. “dioptre adjustment” in the *Sensics zSight* [20]). Rather than simulating different virtual object focus distances this setting is added to let users with refractive problems (e.g. short-sighted or farsighted) see a decent image without wearing their glasses. The setting is often purely subjective: users turn a knob until they can see the image comfortably. What these settings alter is the way in which light beams from the display’s pixels arrive at the user’s eye. Light from a very distant point form a beam of parallel rays, and the eye’s cornea and lens focus them back into a dot on the retina to produce a sharp point. Light from a near point arrive as a beam of diverging rays and the eye’s lens has to perform an added effort to bend them into a point. A short-sighted person bends incoming light rays excessively to properly project on their retina. In order to focus at far distances they wear glasses with concave diverging lenses that compensate it. Lenses are characterized by their optical power, describing how much they bend light ray, and expressed in dioptres. Convex lenses have a positive power and concave lenses have a negative power. The amount of accommodation effort the crystalline lens has to perform depends on the distance to the object looked at, and on the lenses (such as glasses) that may be in place. Accommodation effort is usually described as an optical power, expressed in dioptres.

Fisher et al. [5] listed eight theoretically feasible methods to achieve variable focus in HMDs. We have studied two of them: axial translation of the display (microdisplays move closer or further away), and liquid filled lenses (lenses that deform by the effect of an electrical current).

We define the system optical power, p as the power perceived by the user and that his/her eyes have to accommodate. In the absence of glasses, when looking at a distant object, the system optical power must be zero, so that there is no accommodation demand. A near object at distance d should produce a negative power $-1/d$ forcing the eye to apply a positive accommodation power $1/d$ to compensate. Thus, the complete definition of the system optical power is:

$$p = -\frac{1}{d} + p_{sc} \tag{12}$$

where d is the distance of the presented object, p_{sc} is the spherical correction power of the user’s glasses (negative for myopia, positive for hyperopia).

These techniques are known to modify the optical power of the display system and thus vary the accommodation effort induced on the user. In order to set optical power to a desired value there needs to be a known relation between action and effect: how optical power varies as a function of display position, in the first case, or how it varies as a function of the electrical lens intensity, in the second. Let us define the focus command as either the display position for the first setup, or the lens current for the second.

One way to establish those mapping functions would be to accurately model the theoretical optical system and determine those mathematically. Alternatively, our proposed method approximates the relations experimentally. The basic idea is to measure several points along those functions and fit parametric curves to them that can later be used for control.

To do this we take a camera with manual focus set to infinity and place it in front of the eyepiece. If the camera sees the display image sharp, then the current system optical power is zero (i.e. light exits as collimated beams). If a lens of known power p is placed between the camera and the eyepiece and the camera gets a sharp image, then the system optical power is $-p$.

The calibration procedure should proceed as follows. The camera captures the display and the focus power is adjusted manually (in terms of electrical current or display position) until the image is sharp. This focus value corresponds to zero power. Then, we place lenses of different positive and negative powers in front of the eyepiece and for each of them, focus control is modified to produce a sharp camera images. For each of the lenses we obtain an association of a focus command and a focus power. And using all the data we can fit a parametric curve that approximates the mapping.

Our graphics system should have control, not only on image generation but also on real focus control. The application should be able to select a focus distance at any time and apply it on the hardware, or continuously vary it frame after frame if the main object moves. When doing so, the display field of view may change. So, the rendering engine needs to adapt to the new FOV.

The control flow should be as follows. The application requests a focus distance. The optical power controller computes a system optical power, obtains the corresponding focus command (i.e. a display position or a lens current, depending on the hardware) using the calibration curves and applies it. Then the controller computes the new FOV corresponding to the applied optical power, from the calibrated FOV curve and passes it to the graphics rendering engine.

4 Results

Field of view and stereoscopic parameters of five experimental hardware setups were calibrated using the algorithms provided in Sect. 2. This section describes how the method was applied to our hardware and highlights implementation details of the intermediate steps.

A compact stereo camera is placed in the HMD where a user's eyes are normally located (a Fujifilm Finepix Real 3D was used in the experiments). This is the calibration camera pair (CC). The camera captures the two HMD displays. The HMD displays are in front of this calibration camera so that each of the two CC lenses captures one of the displays.

As a preliminary step, the intrinsic and extrinsic parameters of the calibration camera pair (CC, see Fig. 3) were obtained using a typical checkerboard pattern and functions from the *OpenCV* [11] computer vision library.

In the calibration process of an HMD, the first step is to capture images of the HMD displays with the CC pair. It was placed in the HMD and a pair of images were captured while the displays presented a calibration grid. Figure 4 shows the images of the displays captured by the left and right cameras.

The second step was to compute a mapping between CC image coordinates and display coordinates. Each image is presented to a user who has to click on four specific points of the grid. In this way we get the CC image coordinates of the four points. On the other hand, as we have generated the grid, we know the display coordinates of these points. So we can compute the mapping homography **M** as explained before.

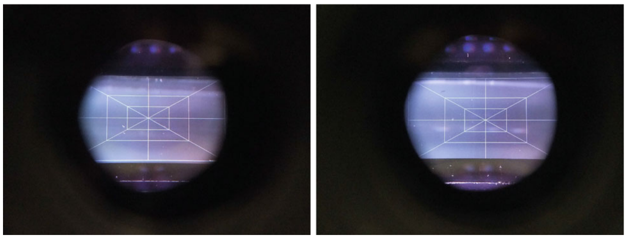

Fig. 4 *Left* and *right* displays showing the calibration grid as seen in the *left* and *right* CC images

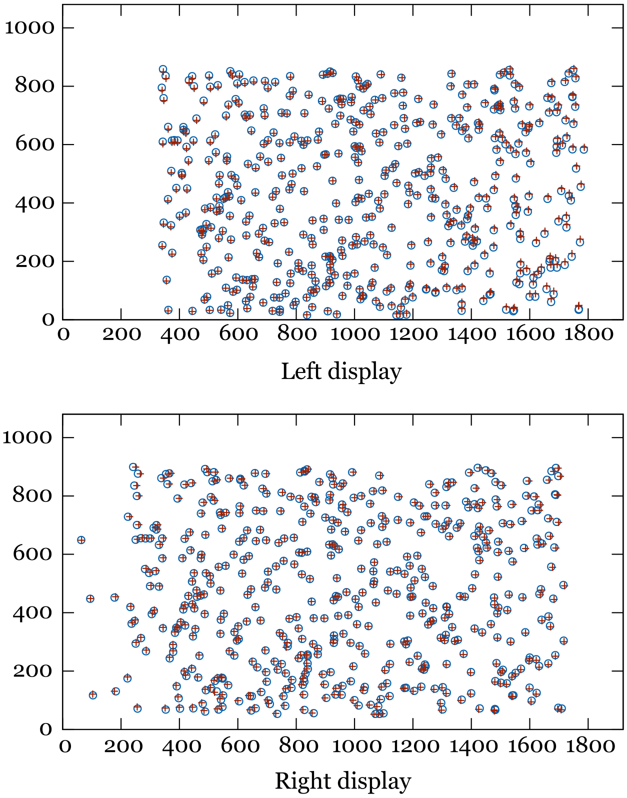

Fig. 5 A set of 500 random 3D points projected into the displays by the calibrated view frustums (*circles*) and as seen by the cameras (*crosses*) for the *left* and *right* sides (*top* and *bottom*, respectively) in display pixel coordinates

Then, the left and right view projection matrices (Eq. 2) are obtained by minimizing the reprojection error of a set of 3D points. Our software uses a set of 500 random points in 3D space. They are inside a volume that is visible by both eyes. They suitably fill the virtual space seen in the displays.

In order to verify the quality of the calibration process we perform the following test. Figure 5 shows the above points projected into the left and right displays by the calibrated view projection matrices (as circles). Then the mapping homography **M** is applied to these points as seen by the cal-

Table 1 Left and right display FOVs and interdisplay relative offsets in each hardware setup

Setup no	Type	FOV-L	FOV-R	offsetX	offsetY
1	L	27.1	26.5	0.039	0.032
2	L	27.5	27.0	0.007	0.044
3	L	26.6	26.5	0.031	−0.006
4	D	26.8	27.1	0.053	−0.017
5	D	25.6	25.5	0.130	−0.054

The *type* column indicates if that hardware uses a tunable lens (L) or display axial motion (D) for accommodation control

ibration cameras (in CC image coordinates) and the result, now expressed in the display coordinate system, is also drawn in the figure (as crosses). The figure shows a good fit between the projected points on both sides, with RMS errors of 4.2 and 3.2 pixels, respectively (4 pixels in these views are equivalent to about 0.06 degrees deviation).

In Fig. 5 we can see that there is a small vertical disparity of any given projected point in the left and right displays. This means that our computation of the projection matrices creates a vertical displacement of the rendered images that compensates the vertical offset of the HMD displays. Horizontally there is a significant bias as well, which is due in part to the mere stereo projection disparity and in part to the horizontal interdisplay offset. Two specific parameters in our view frustum pair parameters (VFP) control these offsets.

These results show that our method corrects the horizontal and vertical small misalignment of the physical displays (see Fig. 1 in Sect. 2). The rendering software places the images in the displays correctly aligned to the user. Using the offset parameters, the VR rendering software displaces the images compensating the physical offset.

Table 1 shows the calibration results for the 5 hardware setups: fields of view (FOVs) for the left and right displays and interdisplay horizontal and vertical offsets. The remaining two parameters, *interocular distance* and *display aspect ratio* are known in advance to be 65 mm and 16:9, respectively. Note that the offset values are relative to the screen size so, for instance, setup 5 has a horizontal interdisplay offset of 13% of screen width and a vertical offset of −5.4% of screen height.

5 Conclusions

Our stereo perspective calibration method provides objective measures unlike typical methods which are subjective, and allows using affordable, imperfect HMDs. Our method is simpler than existing similar ones. Even more, as explained in Sect. 2, it is more precise and avoids calibration camera position errors because we calibrate both displays together. We should strain the fact that a few pixels of misalignment between the stereo displays can disrupt the stereo depth perception.

The visual comfort and spatial perception issues in HMDs, including the vergence–accommodation mismatch problem, may not be fully solved with current hardware designs. Nevertheless we have suggested means to partially overcome them.

Several lines for future research have been identified. On the one hand, the calibration procedure needs a manual step of picking three reference display points in the camera images. First, using more than three points would enable a better mapping between the images, maybe needing a more complex model involving distortions. This step could be automated by presenting recognizable patterns (e.g. checkerboard patterns) and having an algorithm automatically locate its corner points.

On the other hand we will make a proof of concept for our focus control proposal. We will use variable optical power lenses and moving displays for this purpose. Anyway, the strategy for controlling focus based on the distance to the main object is limiting. Users looking at objects at different depths will be unable to correctly focus as they would in the real world and in many applications there is no main object. A more realistic effect would involve the use of an eye tracking system to sense the user's 3D gaze point and control focus based on the distance to the point the user is looking at.

The use of virtual reality in advanced manufacturing scenarios, including collaborative engineering and training can be enhanced by the techniques presented in this work. They should provide a more correct perception of virtual space and shapes, as well as higher visual comfort.

References

1. Benzeroual, K., Allison, R.S., Wilcox, L.M.: Distortions of Space in Stereoscopic 3D Content Karim. In: SMPTE International Conference on Stereoscopic 3D for Media and Entertainment, vol. 1100 (2010)
2. Choi, S., Jung, K., Noh, S.D.: Virtual reality applications in manufacturing industries: past research, present findings, and future directions. Concurr. Eng. **23**(1), 40–63 (2015)
3. Figl, M., Birkfellner, W., Hummel, J., Ede, C., Hanel, R.A., Bergmann, H.: Calibration of an optical see through head mounted display with variable zoom and focus for applications in computer assisted interventions. In: Galloway, R.L., Jr. (ed.) Proceedings of SPIE—The International Society for Optical Engineering, vol. 5029, pp. 618–623 (2003)
4. Figl, M., Ede, C., Birkfellner, W., Hummel, J., Hanel, R., Bergmann, H.: Design and automatic calibration of a head mounted operating binocular for augmented reality applications in computer aided surgery. In: Galloway, R.L., Jr., Cleary, K.R. (eds.) Progress in Biomedical Optics and Imaging—Proceedings of SPIE, vol. 5744, pp. 726–730 (2005)

5. Fischer, R.E., Reiley, D.J., Pope, C., Peli, E.: Methods for improving depth perception in HMDs. In: Proceedings of the Workshop on The Capability of VR to meet Military Requirements, December, pp. 5–9 (1997)
6. Fukuda, K., Wilcox, L.M., Allison, R.S., Howard, I.P.: A reevaluation of the tolerance to vertical misalignment in stereopsis. J. Vis. **9**, 1.1–1.8 (2009)
7. Galambos, P., Weidig, C., Baranyi, P., Aurich, J.C., Hamann, B., Kreylos, O.: VirCA NET: A case study for collaboration in shared virtual space. 2012 IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom), pp. 273–277 (2012)
8. Gilson, S.J., Fitzgibbon, A.W., Glennerster, A.: Spatial calibration of an optical see-through head-mounted display. J. Neurosci. Methods **173**(1), 140–146 (2008)
9. Gilson, S.J., Fitzgibbon, A.W., Glennerster, A.: An automated calibration method for non-see-through head mounted displays. J. Neurosci. Methods **199**(2), 328–335 (2011)
10. Itoh, Y., Klinker, G.: Interaction-free calibration for optical see-through head-mounted displays based on 3D Eye localization. In: 2014 IEEE Symposium on 3D User Interfaces (3DUI), pp. 75–82. IEEE (2014)
11. Itseez: The OpenCV Computer Vision Library (2015). URL http://opencv.org/
12. Kagermann, H., Wahlster, W., Helbig, J.: Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Tech. Rep. April, Acatech (2013)
13. Kellner, F., Bolte, B., Bruder, G., Rautenberg, U., Steinicke, F., Lappe, M., Koch, R.: Geometric calibration of head-mounted displays and its effects on distance estimation. IEEE Trans. Vis. Comput. Graph. **18**(4), 589–596 (2012)
14. Kelly, J., Burton, M., Pollock, B.: Space perception in virtual environments: displacement from the center of projection causes less distortion than predicted by cue-based models. ACM Trans. Appl. Percept. **10**(4), 1–23 (2013)
15. Kuhl, S.A., Thompson, W.B., Creem-regehr, S.H.: HMD calibration and its effects on distance judgments. In: APGV '08 Proceedings of the 5th symposium on Applied perception in graphics and visualization, 212, pp. 15–22. ACM (2008)
16. Makibuchi, N., Kato, H., Yoneyama, A.: Vision-based robust calibration for optical see-through head-mounted displays. In: Proceedings of ICIP 2013, pp. 2177–2181. IEEE Comput. Soc (2013)
17. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. **11**(2), 431–441 (1962)
18. Ponto, K., Gleicher, M., Radwin, R.G., Shin, H.J.: Perceptual calibration for immersive display environments. IEEE Trans. Vis. Comput. Graph. **19**(4), 691–700 (2013)
19. Posada, J., Toro, C., Barandiaran, I., Oyarzun, D., Stricker, D., de Amicis, R., Pinto, E.B., Eisert, P., Dollner, J., Vallarino, I.: Visual computing as a key enabling technology for industrie 4.0 and industrial internet. IEEE Comput. Graph. Appl. **35**(2), 26–40 (2015)
20. Sensics: Sensics zSight Data Sheet. URL http://sensics.com/portfolio-posts/zsight/
21. Yang, S., Sheedy, J.E.: Effects of Vergence and Accommodative Responses on Viewer's Comfort in Viewing 3D Stimuli. In: Woods, A.J., Holliman, N.S., Dodgson, N.A. (eds.) Proc. SPIE 7863, Stereoscopic Displays and Applications, pp. 78,630Q–78,630Q–13 (2011)
22. Zhang, D., Sang, X., Wang, P., Chen, D., Jean Louis B.T.: Comparative visual tolerance to vertical disparity on 3D projector versus lenticular autostereoscopic TV. J. Disp. Technol. **12**(2), 178–184 (2016)