UNIVERSITY OF GENOA

'MASTER'S PROGRAM IN BIOENGINEERING

Thesis submitted in partial fulfilment of the requirements for the title of

Master of Bioengineering

Development of an adaptive virtual reality scenario for the treatment of phobias

Guerrini Nicolò

10/2023

Thesis advisor: Prof. Manuela Chessa

Thesis co-advisor: Prof. Fabio Solari

Contents

Abstract

Virtual reality (VR) is the use of computer modelling and simulation that enables a person to interact with an artificial three-dimensional (3-D) visual or other sensory environments. Since its inception, virtual reality has been used in many application areas, including education, gaming, and healthcare delivery. In recent years, virtual reality constitutes an alternative, effective, and increasingly utilized treatment option for people suffering from neurological illnesses and specific phobias. One of the latest innovations in this field is the delivery of therapies through patient exposure in virtual reality adaptive scenarios. The latter work as closed loop systems able to adjust the virtual world according to the 'patient's emotional state: patients confront and keep in touch with what they fear and avoid until anxiety gradually decreases through a process called habituation.

In the wake of the positive results reported by several papers in the literature, the aim of this work was to create an adaptive VR scenario that could work as exposure therapy for people affected by social anxiety disorder (SAD). The system uses heart rate to make inferences about the patient's anxiety state and modifies some fear related features of the scenario to ensure that the patient's anxiety remains controlled. The system adapts the features without disrupting the immersive experience. According to the simulations performed, it seems that the VR application developed could be suitable for being embedded in a system for SAD therapy, including more complex biofeedback signal. However, further investigations are needed, to understand which are the parameters that affect the patient anxiety worth to be modulated.

Chapter 1

Introduction

Definition of Virtual Reality

As the medium of Virtual Reality grows, different people and groups of people have different ideas and different points of view about the real meaning of the term. In fact, it is not so trivial to give a unique definition, however some key elements distinguish virtual reality regardless of its area of use: the participants, the creator, the virtual world, the interactivity and the immersion. Merging these elements, Sherman at al [1], gave a really good definition of virtual reality:" Virtual Reality is a medium composed of interactive computer simulations that sense the participant's position and actions and replace or augment the feedback to one or more senses, giving the feeling of being mentally immersed or present in the simulation (a virtual world) ".

Probably, the most important element of any VR experience are the persons involved in the experience. VR perception happens in the mind of the participants and every VR experience is different for each of them, because each brings their own capabilities, interpretations, and thus experiences the virtual world in their own unique way. With the same importance but from a different perspective, the person or team that develops virtual reality is also a key element as creator of the work experienced by the user.

While making sense of these elements may be quite intuitive, immersion is a confusing abstract concept. The term ""immersion"" refers to a feeling of being involved in an experience and it can be analysed from two different prospective: mental immersion and physical immersion [1].

Mental immersion is usually related to the so-called sense of presence. Sense of presence is the subjective sense of being in a virtual environment. Presence is important because the greater the degree of presence, the greater the chance that participants will behave in a VE in a manner similar to their behaviour in similar circumstances in everyday reality.

On the other side physical immersion refers to the ability of a VR system of replacing or augmenting the stimuli to the 'participant's senses. This means that the VR should be interactive and able to respond to user’s action.

1 Virtual reality technologies

From a technical perspective, Virtual Reality development, rise thanks to the combination of computer graphics and computer vision: computer graphics (CG) is used to create the virtual world and computer vision (CV) is used to track the user. Indeed, to have an immerse interactive experience we need creating virtual 3D environments coherently aligned with the real ones (e.g., the user).

1.1 Computer Vision

Computer vision is a field of study that centres around enabling computers to "see" and understand the world around them, which is at the heart of VR technologies. Therefore, computer vision in VR is critical for creating compelling and immersive experiences that bridge the physical and digital worlds. Computer vision is a very spread discipline that goes from raw data recording to digital image processing, pattern recognition and machine learning [2]. However, the two most influential computer vision topics for the creation of virtual realities are feature/object detection and visual tracking. Computer vision is utilized in VR to identify and detect real-world objects and integrate virtual content onto them, a process known as object detection. On the other side tracking refers to the ability of VR systems to track the user's head rotation and motion to adjust the virtual environment accordingly. Object detection process is usually the starting point of tracking algorithms however it deserves its own explanation because it is not so trivial.

1.1.1 Feature/Object Detection

A feature is a piece of information about the content of an image; typically, about whether a certain region of the image has certain properties. Features may be specific structures in the image such as points, edges or objects. Features may also be the result of a general neighbourhood operation. The desirable property for a feature

detector is repeatability [30]: whether or not the same feature will be detected in two or more different images of the same scene. Feature detection is a low-level image processing operation. It is usually performed as the first operation on an image and examines every pixel to see if there is a feature present at that pixel. If this is part of a larger algorithm, then the algorithm will typically only examine the image in the region of the features [31]. Indeed, there are many computer vision algorithms that use feature detection as the initial step: some areas of application are video surveillance, robotics, self-driving, biometrics data recognition, AR and obviously VR.

As a result, a very large number of feature detectors have been developed. The latter vary widely in the kinds of feature detected, the computational complexity and the repeatability.

The basic principles of feature detection lie in image filtering. Filtering an image means to perform a convolution between the image itself and a small matrix called "mask", which will be constructed differently depending on which feature of the image you want to detect (see fig. 1).

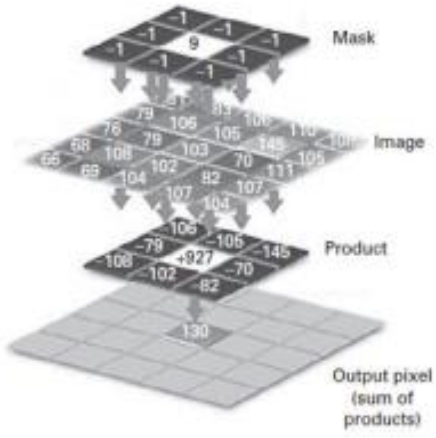


Figure 1 graphic representation of image convolution

$$g(x,y) = \sum_{m=-a}^{a} \sum_{n=-a}^{a} f(m,n)h(i-m,j-n)$$

Equation 1 Convolution formula, f is the function describing the image while h is the mask.

The idea is based on the concept of "template matching". Convolution with a filter can be viewed as comparing a little "picture" of what you want to find against all local regions in the image: when convolving an image with a mask, the local output will be greater the more the area of the image taken into consideration resembles the mask. Therefore, we can think of performing a filtering on the image using the feature we want to detect as a mask. After that, if thresholding is applied, only the area of the image that resembles the feature are kept.

For example, let's say we want to detect a blob. Blob detection should extract regions that differ in properties compared to surrounding regions. Therefore, we can think of using as a mask the Laplacian of Gaussian since the shape of this function resemble the intensity of a blob in an image (see fig. 2,3). Still, it is needed to choose a Gaussian with a standard deviation that matches the radius of the blob we want to detect, otherwise, if the Gaussian is too large or to narrow the blob will not be detected. Therefore, if the aim is to detect all blobs in the image, the only solution is to convolve with several Gaussians of different radius which might be a quiet computationally expensive operation.

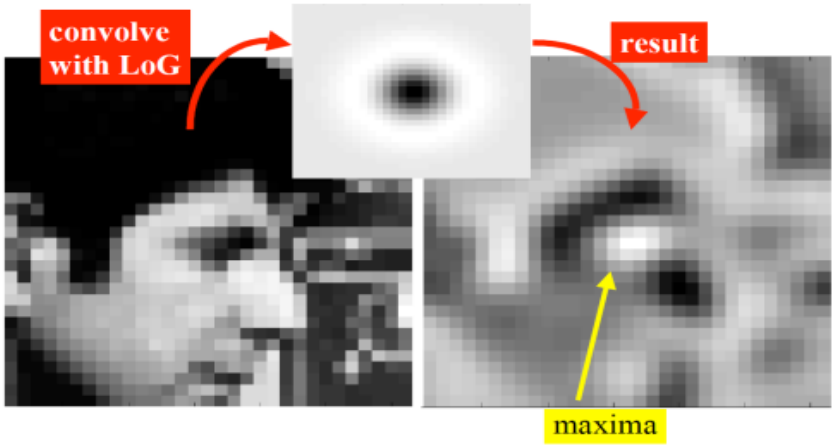
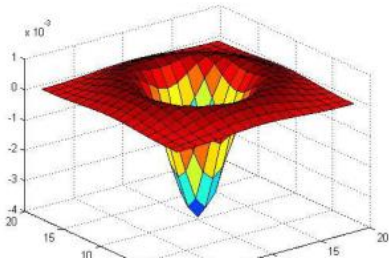
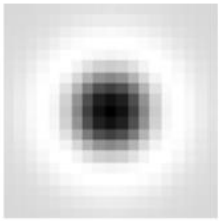


Figure 2 Graphic visualization of blob detection's result





$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

Figure 3 LoG, Laplacian of Gaussian

This leads to one of the crucial issues of feature detection, i.e., what are the characteristics that a feature must have to be a "good"" feature.

In many computer vision applications, features are found in a reference image and then found modified (translated or rotated or scaled) in other images. This, for example, happens in tracking, when you want to reconstruct the movement of an object from one frame to another and also in panorama stitching when you want to join two images taken from different perspectives of the same panorama.

For this reason, as mentioned before, region extraction needs to be repeatable and accurate: features should be invariant to scaling, translation, rotation and robust to lighting variations, noise, blur, quantization. It is not known on which other image locations the feature will end up being matched against. However, we can compute how stable a location is in appearance with respect to small variations in position.

1.1.1.1 Harris Corner detector

Harris corner detector is a standard technique for locating interest points on an image. Despite the appearance of many feature detectors in the last decade, it continues to be a reference technique, which is typically used for camera calibration, image matching, tracking or video stabilization [32]. Harris algorithm is based on the idea that regions around the feature should contain "interesting" and distinctive structure. Points are detected based on the intensity variation in a local neighbourhood: a small region around the feature should show a large intensity change when compared with windows shifted in any direction. This idea can be expressed by the autocorrelation function.

$$f(\Delta x, \Delta y) = \sum_{(x,y)\in W} (I(x_j, y_k) - I(x_k + \Delta x, y_k + \Delta y))^2$$

$$f(\Delta x, \Delta y) \approx (\Delta x\ \Delta y) M (\Delta x\ \Delta y)$$

Equation 2 Minimization function, Δx and Δy are the variations with respect to the current position while I(x,y) is the light intensity at pixel in position x,y.

So, this expression depends on the gradient of the image through the autocorrelation matrix M:

$$M = \sum_{(x,y)\epsilon W} \begin{matrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{matrix}$$

Equation 3 Autocorrelation or tensor matrix, the entries of the matrix are the four second derivatives of light intensity.

Eigenvalues of this matrix provides some key information: the largest eigenvalue corresponds to the direction of largest intensity variation, while the second one corresponds to the intensity variation in its orthogonal direction. If both eigenvalues are greater than zero, the local region probably contains a corner. The algorithm for corner detection with autocorrelation matrix follows these steps:

- M is computed within all image windows to get their corners scores.
- Find points with large corner response (R > threshold)
- Take the points of local maxima of R.

Speaking of the properties that define a "good" feature we can say that corners are covariant to translation and rotation while invariant to light intensity. However, corners are variant with scaling.

1.1.1.2 Harry-Laplace detector

As an improvement of Harris corner detector, the Harris Laplace detector was designed. The Harris-Laplace detectors integrates the automatic scale selection with LoG into the Harris detector. Therefore, it could detect corners regardless of the image scale, meaning that a corner in an image will be constantly detected after resizing the image. Harris-Laplace detector is a multi-scaled version of Harris

detector where the Harris response is computed repeatedly on the same images but convolved with LoG filters from varying scales of standard deviation (see fig. 4). The points that will be considered as corners are only the points that not only have a large corneress in one scale but also in the adjacent scales.

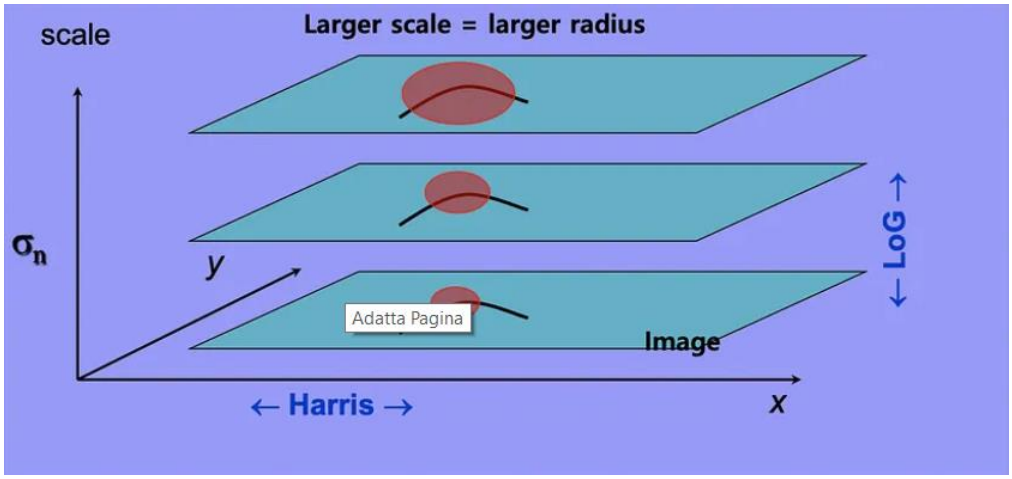


Figure 4 Harris Laplace detector scheme

1.1.1.3 Deep learning Methods

A common traditional object detection pipeline follows these steps: informative region selection, feature extraction and classification. However, each one of them carries some limitations. The processes of scanning the whole image with multi-scale windows, the extraction of robust features and thresholding are difficult-designed and computationally very expensive procedures. Later in the years, with the rapid development of deep learning, more powerful techniques have been developed in the field of object detection. Indeed, convolutional neural networks technique have several advantages against traditional methods [33]:

- Hierarchical feature representation, which is the multilevel representations from pixel to high-level semantic features learned by a hierarchical multi-stage structure.
- The architecture of CNN provides an opportunity to jointly optimize several related tasks together.
- Compared with traditional shallow models, a deeper architecture provides an exponentially increased expressive capability.

In literature there are several deep learning algorithms for feature detection which will not be analysed in detail because it would require an entire separate study.

1.1.2 Visual Tracking

Visual tracking is a branch of computer vision made up of different algorithms capable of keeping track of the 3D motion of objects. Visual tracking is a very challenging problem, but it can provide "intelligent" systems useful in some of the most advanced field of research, ranging from Virtual Reality to robotic perception [22]. These types of algorithms can be based on different methodologies but generally they follow a common pipeline. The starting point of visual tracking is the detection, in a 2D image, of the object of the scene that you want to trace. This step usually lies on image processing and feature detection (see section 1.1.1). After detection, the object will be searched in the next frame. Knowing the position of the same object in two images taken at different moments but from the same camera allows us to reconstruct the 2D motion of the object. Usually, this search step is carried out using inference techniques, such as the Kalman filter, to narrow the search area. Once the position of the object is known in both frames, its 3D position in space can be reconstructed by triangulation. These steps are then iterated throughout the all-frame's sequence.

1.1.2.1 2D Motion reconstruction: optical flow estimation

The idea of 2d motion reconstruction is that, given two images, we would like to find the location of a world point in a second close-by image with no camera info. This problem is usually referred to as optical flow estimation. Optical flow estimation can be a very good estimation of the motion field, which is the projection of 3d vectors velocity onto the 2d image plane so what we really want to know.

Common key assumptions for optical flow estimation are that pixel intensities are translated from one frame to the next and that points do not move very far from one image to the other, so we can narrow the search in nearby pixels [34]. These assumptions allow us to write the following equation for each pixel.

$$I_x u + I_y v + I_t = 0$$

Equation 4 Luminance Constancy equation also called optical flow equation. u,v are the velocities and Ix,Iy,It are the derivatives of light intensity rispectively to x-axis,y-axis and time.

For each pixel we have one equation and to unknown (u,v), in fact we can only evaluate the component of the velocity perpendicular to the edge(expressing the formula with vector notations). This impossibility in uniquely defining the velocity vector in the 2d image plane is also known as the "aperture problem".

Over the course of the year's algorithms solved the under determined aperture problem following different strategies. Indeed, traditional optical flow estimation algorithms can be divided in different categories depending on their solution of the aperture problem:

- **Global algorithms** build a global (entire image) energy functional whose minimizing scheme yields to the optical flow field. In 1981 Horn and Schunck [36] solved the aperture problem by providing additional smoothness constraint. The idea was to minimize the unknown velocities by minimizing the following global functional.

$$Ehs(u,v) = \int_{\omega} \left(\left(f_x u + f_y v + f_t \right)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2) \right) dxdy$$

Equation 5 Sanchez et al. 2018 [32] Minimization function solved by HS algorithm for optical flow estimation.

Where the smoothness weight α serves as regularisation parameter: Larger values for α result in a stronger penalisation of large flow gradients and lead to smoother flow fields.

Horn and 'Schunck's approach was followed by many researchers; however, the algorithm faces difficulties in many practical situations such as large displacements.

-Local algorithms are based on the assumption of constant essential flow within a small neighbourhood region of pixels.

The solution of Lucas and Kanade algorithm to the aperture problem, was to suppose that the N pixels belonging to the same small area have the same velocity. In this case we end up with N equations and two unknowns and the solutions, i.e., the velocities, can be calculated using least squares.

Due to the aperture problem, we would like to include points in the image that have different gradient directions: corners can be a good candidate as points to track from

frame to frame. Indeed, an evolution of Lucas-Kanade is the Kanade-Lucas-Tomasi algorithm. The latter first finds corner, then extracts the intensity of the area surrounding each corner and finally performs Lucas-Kanade algorithm using the corners found.

It must be noted that the assumptions of "luminance constancy" and "small motion", used by both local and global algorithms, are both very stringent and carry some limitations.

As for feature detection, also in optical flow estimation, the spread of deep learning has given rise to much more efficient algorithms [35]. Based on machine learning principles, these algorithms learn to compute optical flow from a pair of input images. Optical flow estimation requires the convolutional neural networks (CNN) to learn feature representations and match them at different locations in two images. One of the main advantages of CNN is the division in layers. This allows to easily perform coarse to fine analysis and consequentially to easily detect motion at different scales. However deep learning methods requires a huge amount of training data and parameters, for this reason they are very expensive in terms of computational memory needed.

From what has just been reported the estimation of the speed and position of objects are not deterministic measures in computer vision tracking. For example, just think that Lucas Kanade's solution for velocity is obtained by minimizing the least squares or that the detection algorithms take into consideration a descriptive window that is not uniquely defined. For this reason, optical flow estimation is carried out with the help of optimization algorithms, such as Kalman filter.

1.1.2.2 Kalman Filter

The Kalman filter is a powerful mathematical technique employed in various fields, primarily in the realm of estimation and control systems. This sophisticated algorithm is particularly valuable when dealing with linear system models and any data derived from these systems, including measurements. The Kalman filter hinges on the incorporation of statistical models that capture the inherent errors associated with both the system's dynamics and the measurements taken.

At its core, the Kalman filter serves as a recursive process for assimilating measurement data into the evolving estimate of the system's state. This recursive nature is a key aspect of its practicality, enabling it to adapt and refine its predictions as new measurements become available. As a result of this recursive operation, the Kalman filter continually refines its estimates, converging over time towards a more accurate representation of the true values of the variables being measured.

In computer vision, Kalman filter is widely used for tracking objects in video sequences. The Kalman filter is applied to predict the future position and velocity of a tracked object based on their previous-frame values. As new measurements from the video frames are obtained, the Kalman filter continually updates its estimate, considering the measurement uncertainties and the dynamic characteristics of the object's motion (see fig. 5). This results in a more robust and accurate tracking of objects, even in the presence of noise or occlusions.

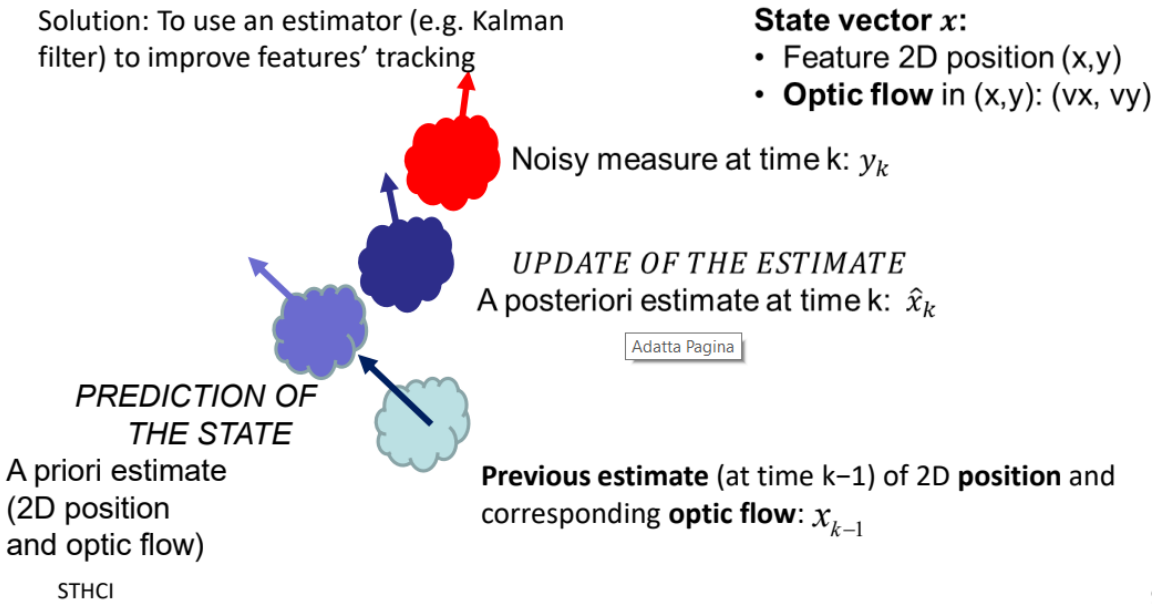


Figure 5 Kalman filter graphical scheme for its implementation in computer vision.

1.1.2.3 3D Pose estimation

Once optical flow has been estimated the next step is the 3d reconstruction of motion. 3D reconstruction is strictly related to stereoscopic vision (two cameras), which allows human and computer to perceive depth. Stereoscopic depth estimation is a technique that uses two cameras placed side by side to capture images of the same

scene. These cameras simulate the way human eyes perceive depth through binocular vision. The relationship between depth and stereoscopic vision lie on the geometry constrains between the two image planes, a topic called “Epipolar geometry”.

Epipolar geometry is a set of principles that relate the images captured by two cameras in a stereo setup. It defines the geometric relationship between these two cameras and their images.

In the context of depth estimation, epipolar geometry is particularly useful. It introduces the notion of epipolar lines, which are straight lines in one camera’s image along which corresponding points in the other camera’s image must lie (see fig. 6).

Epipolar lines constrain the possible locations of corresponding points, making it easier to find matching points in the two images.

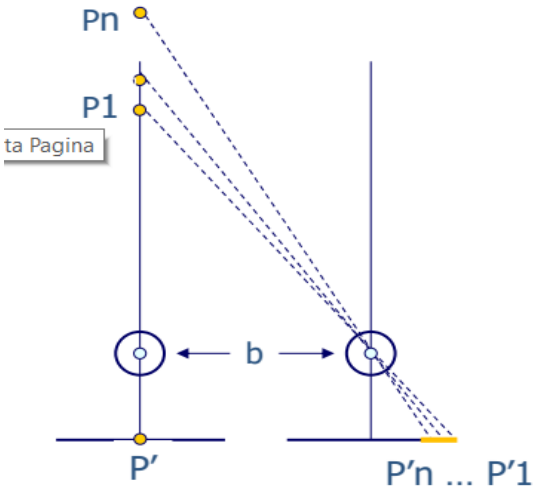


Figure 6 Epipolar lines, it is visible how a point in an image plane can correspond to a line in another image plane.

Using epipolar geometry, it's possible to derive a formula that relates the projections of an object in two parallel cameras with its depth. This formula is based on the disparity, which is the horizontal difference between the positions of corresponding points in the two images.

The relationship can be expressed as:

$$\delta = \frac{fB}{Z}$$

In this formula, the baseline (B) represents the separation between the two cameras, the focal length (f) is a property of the cameras, and the disparity (δ) is determined by measuring the horizontal shift of corresponding points in the stereo images.
By calculating this depth estimation formula for different points in the scene, it's possible to create a depth map, which represents the 3D structure of the objects in the environment.

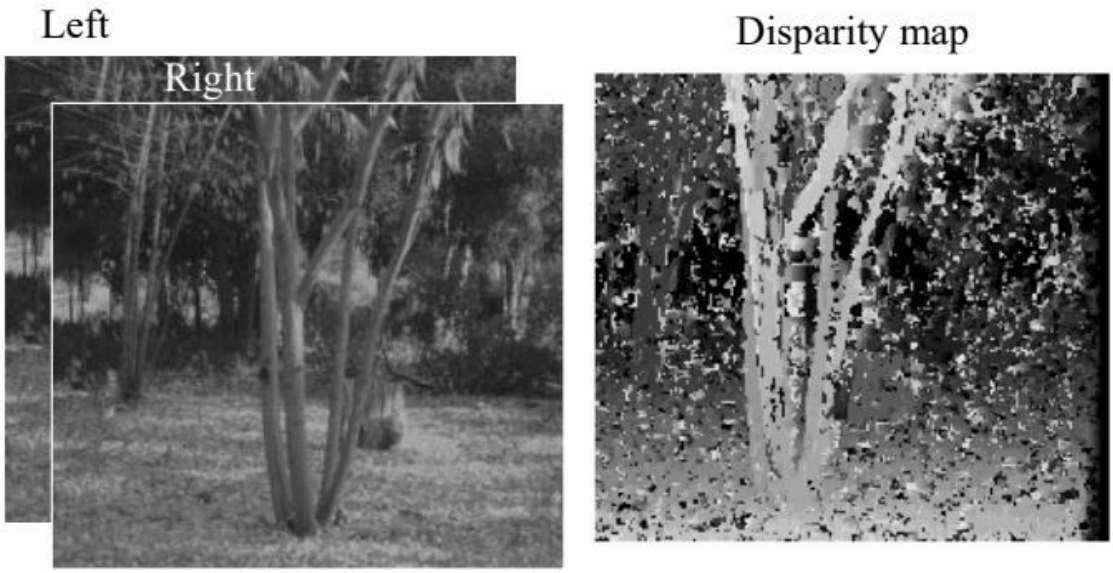


Figure 7 Depth map of the same image taken from two shifted cameras.

Moreover, it is now well established that the coordinates of the projection of a point and the two camera optical centres form a triangle, a fact that can be written as an algebraic constraint involving the camera poses and image coordinates. This mathematical constraint ensures that, given the projection candidates of the same object in two image planes, it is possible to estimate relative position and orientation between the cameras. This relationship is expressed by the essential matrix, a matrix that defines the homogeneous transformation from one camera to the other (see fig. 8).

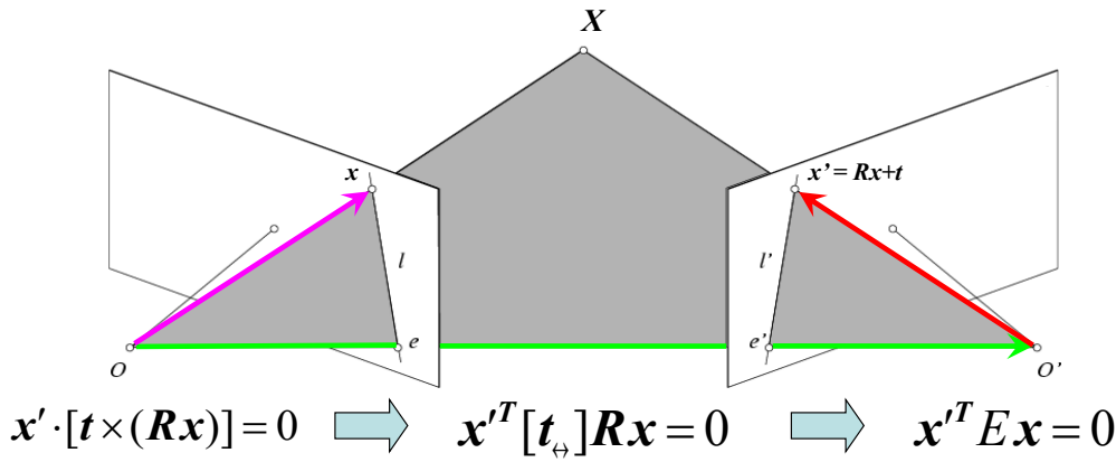


Figure 8 Scheme of triangulation constrains; E is the so-called essential matrix.

1.1.2.4 Tracking algorithm classification

After this brief excursus on the different topic on which lies visual tracking, it is possible to proceed with a classification of the algorithms.

Tracking algorithms can be divided into two macro categories depending on how sensors (cameras) and sources are arranged and managed. Outside-in tracking uses sensors mounted stationary in the environment, while inside inside-out tracking uses sensors mounted to a mobile or body-worn device.

Between the most used tracking algorithms there are:

-**Marker-based Tracking**: Marker-based tracking is a method where 2D markers, such as QR codes or fiducial markers, are used as reference points in the scene. These markers have known geometric properties. By analysing the transformation (homography) between the 2D marker's appearance in an image and its 3D model, the camera's position and orientation (pose) can be accurately estimated.

-**Stereo Camera Tracking**: Stereo camera tracking uses two cameras to estimate the 3D position of objects in the scene. This technique is particularly valuable for tracking human bodies or objects marked with infrared lights. It relies on the concept of disparity. By analysing disparity, it's possible to determine the depth of objects in the scene and reconstruct 3D position.

-Incremental Tracking: Incremental tracking is essential for real-time applications like Virtual Reality. It leverages the continuity of information between consecutive video frames.

In incremental tracking, it's assumed that the camera's position and the appearance of feature points don't change significantly between frames, simplifying the tracking process.

The system uses data from the previous frame as a starting point for tracking in the current frame, making use of two main components: an incremental search component and an interest point matching component.

Incremental tracking is advantageous in VR because it optimizes computational efficiency and tracking accuracy by building on prior knowledge. This is particularly important for real-time VR experiences where virtual elements must align accurately with the real world.

Simultaneous localization and mapping (SLAM) it's a widely used incremental tracking technique. Localization means constantly knowing the orientation and position of the camera with respect to a fixed reference system. Mapping, on the other hand, refers to the creation of a scene map through detection of all the key features present.

SLAM algorithm follows this pipeline:

- **Detection of points of interest** (feature detection algorithm)
- **2D tracking of the detected points** through optical flow estimation (Lucas Kanade Tomasi)
- **Estimation of the essential matrix** that defines the homogeneous transform of the camera pose from the previous frame to the current one.

-Bundle Adjustment. Bundle adjustment is an expensive computational optimization technique that produces jointly optimal 3D structure and viewing parameter (camera pose and/or calibration) estimates [23].

1.2 Computer Graphics

Computer graphics plays a pivotal role in the development of Virtual Reality by enabling the transition from 3D graphics data to 2D images. This process is at the heart of rendering immersive virtual environments.

The computer graphics pipeline is a well-defined series of operations that bridges the gap between 3D graphics data and the 2D images displayed on a computer screen. It is a fundamental component of rendering in computer graphics.

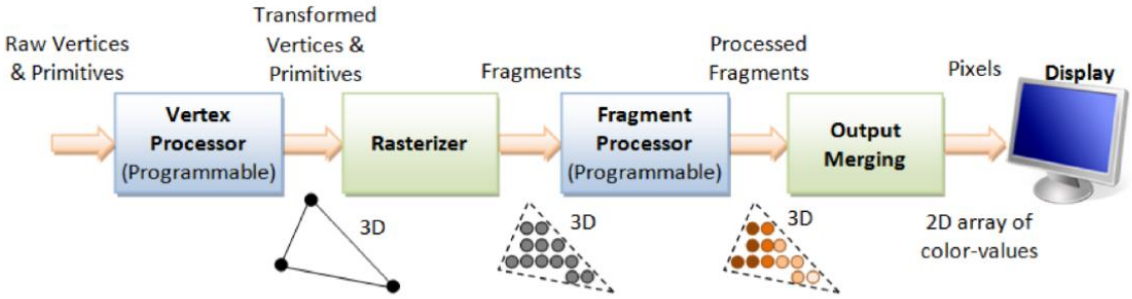


Figure 9 Computer Graphics pipeline

In the context of VR development, the process begins with 3D graphics data, which represents the geometry and properties of objects in a virtual environment.

The first step involves taking 3D points corresponding to the vertices of these objects' surfaces.

Coordinate transformations are then applied to these points to move them from the individual object's reference system to a common reference system, often referred to as the World reference system. This ensures that all objects are correctly positioned in the same global coordinate space.

Before passing from 3D to 2D the surfaces undergo a process called rasterization. The core of the rasterization process involves determining which pixels on the 2D plane (the screen) are "covered" by the projected polygons. This is done by breaking down polygons into smaller primitives, typically triangles.

For each primitive, the algorithm determines which pixels within the primitive's bounding box (a rectangle in 2D) are part of the polygon. These pixels are then shaded to determine their colours.

Rasterization includes shading and texturing steps, where the colour of each pixel is calculated based on lighting models and textures applied to the 3D scene. Shading models like Phong or Lambertian shading are often used to simulate how light interacts with the surfaces.

Following rasterization, the next step is converting the 3D data into 2D images that can be displayed on a screen or within a VR headset.

This conversion involves projecting the 3D points onto a 2D canvas, mapping the 3D scene onto a 2D plane, which is typically called the "viewport" or "screen space.". There are two primary projection techniques: perspective projection and orthographic projection (see Fig. 10).

-**Perspective projection** is the most commonly used technique in VR. It simulates how objects in the real world appear smaller as they move farther away, creating a sense of depth and perspective.

It is well-suited for VR because it mimics how we naturally perceive depth and allows for realistic 3D representations of virtual environments.

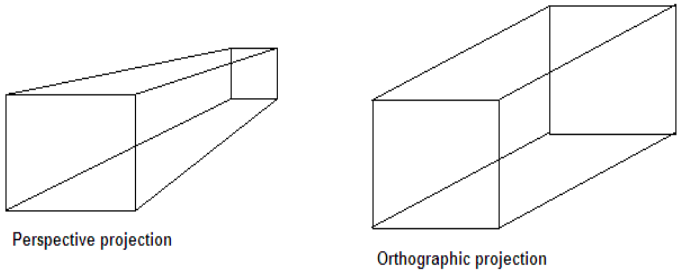


Figure 10 Visualization of the two different types of projection

-**Orthographic projection**, on the other hand, is a more general technique that doesn't account for perspective and instead projects objects uniformly from a "box camera's" view frustum.

While not typically used in VR for creating immersive experiences, orthographic projection is valuable in certain technical or architectural visualization applications.

1.3 Head-Mounted Displays (HMDs)

Nowadays the most used virtual reality devices are the so-called head mounted displays (HMDs). Such devices are helmets which may or may not allow a view of the outside world. Images are displayed on a screen or a pair of screens in the helmets or glasses (see fig. 11). The output of tracking algorithms tells the computer system where the participant is located in the real space, or at least, where he is looking at. In fact, two different tracking modes can be recognized for systems with HMD: three degrees of freedom (3DoF) tracking and six degrees of freedom (6DoF) tracking.

- **3DoF headsets** provide the simplest form of user tracking in VR. They rely mostly on sensors (accelerometers, gyroscopes and magnetometers) embedded in the devices (such as a smartphone). 3-DoF headsets allow us to track rotational motion but not translational. With a user wearing a VR headset, it is tracked whether a user looks left or right, rotates their head up or down, pivots left or right.
Some of the most common HMDs that use 3DoF tracking are Samsung Gear VR, Oculus GO and Google Cardboard.

-6-DoF headsets allow us to track translational motion as well as rotational motion. We can determine whether a user has rotated their head and moved around in the scene.
The most commercialized 6-DoF VR headsets are: Oculus Rift, Oculus Quest, HTC Vive and Windows Mixed Reality.

The computer quickly displays a visual image from the vantage point appropriate to the participant's position. Thus, the participant is able to look a computer-generated world in a manner similar to the real world (within the limits of current technology), making this a natural, intuitive interface. Among the most popular 6DoF HMDs we can distinguish very different tracking algorithms.

Figure 11 Oculus rift HMD and its hand tracking controllers

-**Oculus Rift** [41] for example uses an outside-in system called "constellation tacking". Each tracked device has a pre-defined "constellation" of infrared LEDs hidden under the external plastic. Sensors, which are basically cameras with filters to see only IR light, send frames to the 'user's PC over a USB cable at 60 Hz. The PC processes each frame, identifying the position of each IR LED, and thus the relative position of each object. The software can easily recognize which LEDs 'it's seeing because it knows the shape of the "constellation", it remembers where the object was in the previous frame, and it knows its direction of acceleration (from the accelerometer), and its rotation. Oculus Rift is no more available on the market, though many research applications are still using it.

- **Oculus Quest** [42] uses an Inside-Out tracking system. In each corner of the device is present a camera. The 4 images are used to compute the position of the headset (and thus of the user) with respect to the real environment through the SLAM algorithm.

- **HTC Vive** [40] uses an inside-out tracking system called lighthouse tracking. Two stations, still in the environment, emit precisely timed IR pulses (blinks) and X/Y axis IR laser sweeps. HMD and hand controllers are equipped with Infrared sensitive photodiodes. As the IR light generated by the stations scans the scene, the photodiodes that receive this light provide outputs that are amplified and sent to an internal ASIC. Each time a sensor is hit by a laser beam it resets itself and start counting. The ASIC integrates the time of flight needed for the light to reach each photodiode. After processing this information, the ASIC is able to estimate its own positions and orientation in the room. This type of tracking is sensitive to occlusion. If, materials that are able to absorb IR light are present, the tracking mechanism might not work as desired.

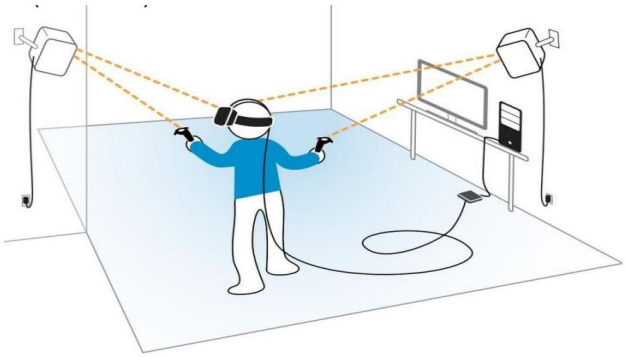


Figure 12 HTC Vive tracking setup.

Chapter 2

State of Art

2 Virtual Reality in medicine

Since its inception in 1960s, when it was initiated for computer graphics, Virtual Reality has extended steadily to different disciplines: from gaming tools at homes to work-related tools for professionals and researchers. One of the areas where VR has made a significant beneficial impact is medicine [24]: VR has established over different medical applications, including medical training, clinical evaluation, rehabilitation and healthcare care delivery.

2.1 Medical education

Different VR applications were developed for medical education, including virtual patients and serious games.
Virtual patients are computer-based simulations of real clinical scenarios. They are designed to mimic actual patient encounters, enabling medical students to practice their clinical skills in a safe and controlled environment. These simulations can cover a wide range of medical cases, from routine check-ups to complex surgical procedures. Here are some key points to consider:

-Safe Learning Environment: Virtual patients provide a risk-free setting for students to make diagnostic and treatment decisions without the potential harm to real patients.

-Repetitive Practice: Students can repeat scenarios as many times as necessary to gain proficiency and confidence in their skills.

-Diagnostic Training: These simulations are invaluable for teaching students how to take patient histories, conduct physical examinations, and order appropriate tests.

-Interactivity: VR allows for a high degree of interactivity, including responses to student actions, which enhances the realism of the training.

Serious games are educational games designed with the primary purpose of teaching, rather than just for entertainment. When integrated into medical education, they introduce gaming principles such as difficulty levels, incentives, and feedback to make learning more engaging and effective.
Serious games leverage the principles of gamification to create an engaging learning experience. This can include rewarding students for completing tasks, offering points or badges, and providing feedback on their performance. Providing rewards or incentives in the form of in-game achievements or recognition can motivate students to stay engaged with the learning process.
Games can be designed to offer varying degrees of difficulty, allowing students to progress at their own pace. This ensures that both beginners and advanced learners find value in the educational experience.

The use of VR applications, including virtual patients and serious games, in medical education has revolutionized the way medical students learn and prepare for clinical practice. It offers a safe, interactive, and engaging environment that fosters knowledge and skill development, ultimately benefiting the training and readiness of young medical professionals.

2.2 Rehabilitation

The utilization of Virtual Reality (VR) technology and computer games in motor rehabilitation is indeed a promising and effective approach to enhancing the rehabilitation process. This concept revolves around creating enjoyable and motivating tasks that can be adjusted in complexity to facilitate the rehabilitation of individuals with motor impairments. Many studies have identified the benefits of using VR in rehabilitation to improve balance, endurance, dexterity, speed and range of motion [4][5].
One of the primary advantages of using VR and computer games in rehabilitation is their ability to engage patients in therapy. The immersive and interactive nature of VR creates

a more enjoyable and motivating environment for patients, which can be particularly beneficial for individuals who may find traditional rehabilitation exercises monotonous.

Figure 13 Patient undergoing VR rehabilitation.

Games and VR applications can tap into a patient's intrinsic motivation to perform tasks. Challenges, rewards, and a sense of achievement encourage individuals to actively participate in their rehabilitation.

Moreover, VR technology allows therapists to customize rehabilitation exercises to match an individual's specific needs and capabilities. Tasks can be graded or adjusted in difficulty, ensuring that patients are challenged without being overwhelmed.

In fact, VR applications often include tools for monitoring a patient's progress. Therapists can use this data to fine-tune the rehabilitation program and ensure that patients are making consistent gains.

2.3 Health care delivery

The application of Virtual Reality (VR) technologies in healthcare delivery focuses on providing clinical solutions to enhance or transform medical practices and procedures, with a wide range of applications that benefit both patients and medical professionals [24].

-Surgery: VR can be used for surgical training, allowing medical professionals to practice and refine their surgical skills in a virtual environment. It can also provide real-time guidance during complex surgical procedures.

-Body Examinations: VR technology can enhance the visualization of medical imaging data, making it easier for healthcare providers to understand and interpret complex medical images such as CT scans or MRIs.
Moreover, some of the most interesting areas of use are body swapping or assessments and treatments of eating disorders and obesities, anxiety disorders, stress related disorders, or pain management.

-Assessments: VR assessments are used to evaluate various aspects of a patient's physical or mental health. For example, VR can be employed in cognitive assessments for conditions like dementia.

-Body Image Modification: VR tools are used to analyse the effects of altering the perception of one's body. This can be particularly useful in the treatment of eating disorders, where individuals may have distorted body image perceptions.

-Food Craving Reduction: VR can be employed to modify the experience of the body in real-time to help reduce food cravings. This can include creating scenarios where patients learn to cope with food-related triggers and cravings in a controlled environment.

-Anxiety Disorders: VR-based exposure therapy is effective for individuals with anxiety disorders. It allows patients to confront their fears in a safe and controlled environment, gradually reducing their anxiety.

-Stress-Related Disorders: VR offers relaxation and mindfulness exercises, helping individuals manage stress and reduce its impact on their overall health.

-Pain Management: VR is used to distract patients from pain and discomfort during medical procedures. It can also provide a means of controlling pain perception through immersive experiences.

2.4 Phobias Therapy

A specific phobia consists of fear and anxiety about a particular situation or object. The situation or object is generally avoided when possible; but if exposure does occur, anxiety develops quickly. Anxiety can intensify to the level of a panic attack. People with a specific phobia usually recognize that their fear is irrational and excessive. There are several possible treatments that must be adequate according to the clinical condition. One such treatment is exposure therapy. Patients confront and keep in touch with what they fear and avoid until anxiety gradually decreases through a process called habituation. Typically, therapists start with moderate exposure. When patients are comfortable with an exposure level, the latter is increased. Therapists continue to increase the level of the exposure until patients are able to tolerate normal interaction with the situation or object. Through exposure therapy individuals are allowed to interact with the fear memory and, thanks to extinction and habituation, the fear structure is slowly modified to a less aversive memory. Using solutions implemented in Virtual Reality has significative advantages, as it is possible to expose patients to situations or objects related to phobias in a controlled manner. In the real world, the use of inducing-fear object (for example real animals) is very difficult and controlling their behaviours is a complex challenge [7].

For VR solutions, the reality of the experience is assessed through patient reports of anxiety symptoms such as "sweating, the butterflies, and weakness" during the virtual exposure, these reports serve as evidence of "realness" and immersion of the experience [29].

As mentioned before, a key word for the treatment of phobias is adaptation. In traditional therapies this phenomenon is generated over the course of several sessions by changing the scenarios proposed from one time to the other after evaluating the patient's feedback. In recent years, researchers have begun to try to reproduce the patient's adaptation to the phobia during the course of a single exposure to virtual reality. The scenarios developed for this purpose must be able to evolve in real time depending on the patient's level of fear. In general, this "intelligent" technology is called adaptive virtual reality. It is a new rising technology with great potential to improve the application of virtual reality in medicine.

3 Adaptive physiologically driven Virtual Reality scenarios

Nowadays it is well established that virtual reality is able to alter the emotional state of the person immersed in the environment. This aspect of VR can be exploited to develop an adaptive interactive mechanism, which alters aspects of the virtual environment in real-time to create personalized experiences calibrated to the individual user. Adaptive VR functions on the basis of a closed-loop design, where behaviour, psychophysiology, or neurophysiology is monitored to create a real-time model of the user. This quantification is used to infer the emotional state of the individual user and trigger adaptive changes within the VE during run time [8].

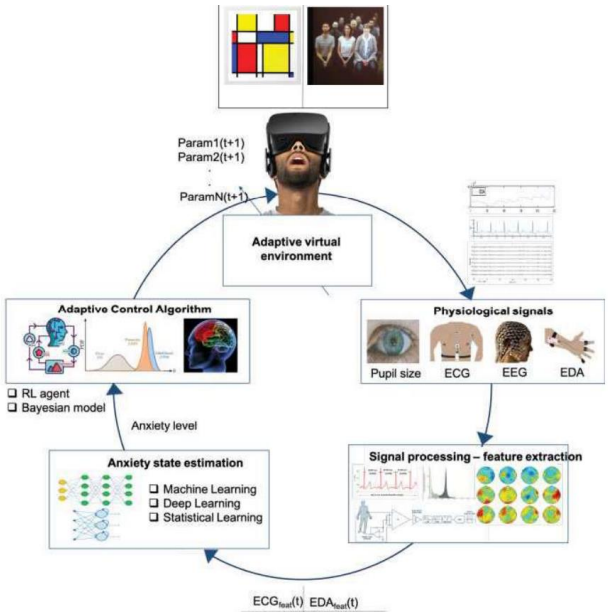


Figure 14 Scheme of closed loop physiologically driven VR system

It is easy to understand how these adaptive scenarios can be revolutionary for the use of virtual reality. Traditional techniques, that rely on adaptation phenomenon, presuppose that the users must undergo VR exposure several times and in different sessions with the effort of having to change the scenario from time to time.

When we talk about adaptive virtual reality scenarios we are talking about an extremely versatile technology. Since they are based on the alteration of the human emotional state, they can come into play in practically any application involving human interaction.

Up to date, the main areas of application of these scenarios concern learning and medical therapies for neurological illnesses.
Before looking to specific use cases in detail, it is important to go through the technical methodologies adopted for developing VR adaptive scenarios.

3.1 Technical aspects of physiologically driven VR

The technical choices for the creation of adaptive virtual reality projects can be very different from case to case. Control strategy, appropriate stimuli generation, and emotional state estimation must be appropriate for the purpose of the application [10]. Despite this, by analysing the state of the art, it is possible to recognize logical blocks that are common to almost all scenarios, regardless their area of use. Below are presented two works that studied the logical architecture of VR adaptive systems.

3.1.1 Architecture

Cosic and colleagues in 2010 [10] created a system with some aspects that are also recognizable in other subsequent works. The system was divided into four logical blocks working in a synchronized manner: an adaptive controller, an emotional state estimator, a stimuli generator and a reference knowledge database. Below a brief description these blocks.

-Stimuli Generator: Within the physiology-driven adaptive VR stimulation, the stimuli generator is responsible for finding the best-matching stimuli from its databases with respect to the semantics, emotional properties, and media form specified by control signals from the adaptive controller. In this process, it is important that the signals result in emotionally and semantically aligned stimuli, which are individually conformed to a specific participant's mental state.

-Emotional state estimator: Emotional state estimator provides crucial information for adaptation of VR. It estimates the 'participant's emotional state as physiological signals are acquired. For an interval extending a number of seconds backwards from present moment, physiological samples are collected and used in computing the various features

required for arousal estimation, such as mean, standard deviation, minimum, maximum, and slope. From these results, the emotional state estimator spits an output that can be seen as an indicator of arousal.

-Adaptive controller: The adaptive controller is the central subsystem for optimally individualized VR exposure treatment. It selects and adjusts all relevant parameters of the system according to the participant's physiology [7]. This block relies on the stimuli generator for actual stimuli delivery to the participant and the emotional state estimator for information regarding his or her emotional state.

-Archiving Database: The Reference Knowledge Database (RKD) is based on relevant data from literature and or on integrated data recorded of previous participants.

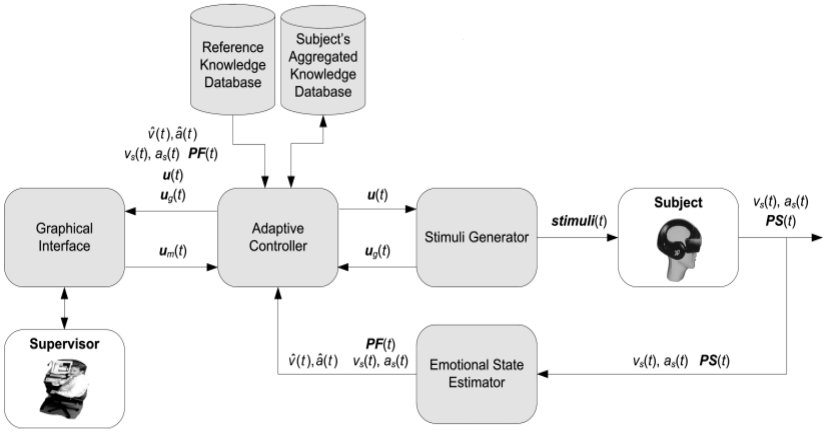


Figure 15 Cosic et al 2010 [10] Scheme of adaptive VR architecture

Ten years later Vega et al [11] proposed an architecture structured of three main components: user model (UM), virtual reality scenario and adaptive engine (AE).

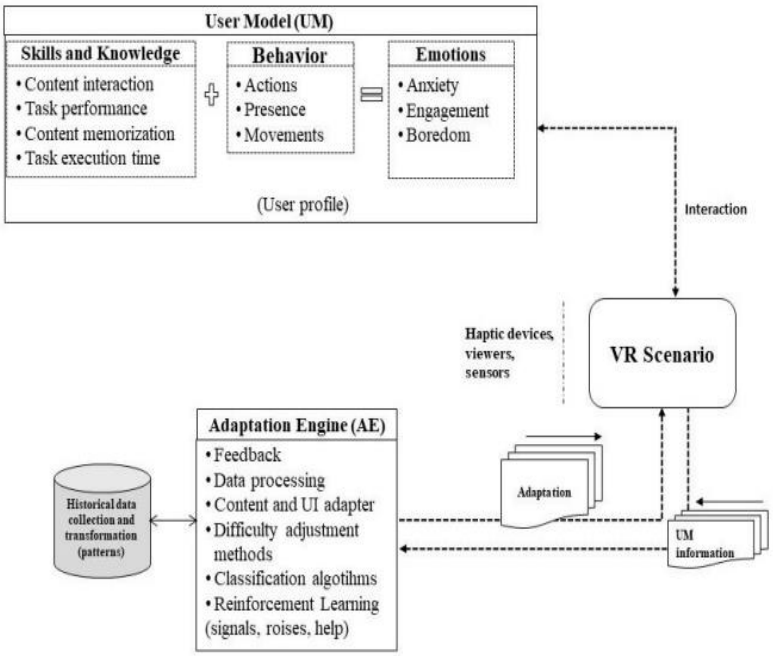


Figure 16 Vega et al 2020 [11]. Scheme of adaptive VR architecture

The UM provides user characteristics splits into three main collections: skills and knowledge, Emotions, and Behaviour. UM is in constant interaction with the other components and help to understand that the user has difficulties at some stage of training. The VRS acts as a bridge with the AE, while the AE analyse the information provided by the VRS. The AE processes the information obtained and is supported by a collection of data containing behaviour patterns for optimization and updating content and user interface. The interaction of the components works as a cycle that improves the historical data collection.

3.1.2Adaptive logic and Adaptive Variables

Although the authors of the two works presented above may have used names or components that are not exactly identical, it is clear that the two architectures presented are extremely similar. Both used a closed-loop system that takes the patient's physiological data as input and uses it, following a precise logic, to update the virtual reality environment. It must be noted that the beating heart of these models is the block capable of allowing adaptation to the scenario, it is called adaptive controller in the first work and adaptive engine in second work. To give a unique naming to this component from now on we will call it the "adaptive block". The adaptive block is usually

characterized by two parts: the adaptive logic and the adaptive variables. In literature most of the works implemented adaptive logic using machine learning or optimization algorithms including artificial neural network [9]. The most common algorithm basically follows this flow: key user's physiological features, decided a priori of the experiment, are extracted and sent to a machine learning classification algorithm in order to infer the general state of the user, then, based on the inferred state the VR scene is updated.
In literature it is also possible to find another type of approach that does not involve machine learning. This type of adaptive block is named "rule based". Rule based systems adaptation's mechanism is provided by means of conditional statements. These conditional statements are used to compare the performance/the status of the trainee/patient with thresholds. The latter can be derived either from a baseline of user's bio signals recorded previous the exposure or from literature.
Adaptive variables are the features of the scenario chosen to evolve consistently with the patient's biofeedback. It would not make much sense to speak generically about adaptive variables because they are strictly related to the specific use case. Indeed, it is possible that two teams that work on the same virtual reality scenario and with the same purpose could choose different adaptive variables. Obviously, depending on the use case, there are variables that are more likely to be chosen than others, however, much still remains in the hands of the developers. This is because there are situations in which it is not so trivial to choose scene variables that are likely to generate the desired effect on the patient's state of mind with a high probability. Indeed, speaking of the use of virtual reality in medical therapies, developments continue in parallel with psychophysics studies. The latter is the science that study quantitative relations between psychological events and physical events or, more specifically, between sensations and the stimuli that produced them. With psychophysics studies' results it would be easier to choose adaptive variable that are assessed to be more effective on the patient emotional state.

3.2 Adaptive VR for phobias treatment

After this brief excursus about the technical methodologies that make the development of adaptive virtual reality scenarios possible, we can go into detail on the use of the latter for the creation of a system useful for the rehabilitation and treatment of phobias.

Virtual reality creates a complete sense of presence and increases immersivity by using stereoscopic 3D view, motion tracking, and head tracking technologies, for this reason virtual reality is able to trigger emotional states, quantified by various physiological responses. Indeed, in psychophysical experiments, virtual reality has been widely used as a stimulus for eliciting emotions. The modalities of quantifying emotions are mainly two: self-reports (verbal descriptions of the emotions experienced or self-ratings questionnaires) and measures of the physiological signals originating in the Central Nervous System (brain activity collected by EEG) and Autonomic Nervous System (respiration, heart rate, body temperature).

Feeling of presence in the virtual environment, the degree of realism, and immersivity are highly correlated with the percentage of change in heart rate and skin resistance. Wiederhold [43] found a significant correlation between presence and skin conductance level in VR. Riva et al. [44] found that the sense of presence influences the emotions experienced in the virtual environment. Jang et al. [45] found that different virtual environments generate different affective states. Moreover, electrodermal activity (EDA) and heart rate (HR) variability are reliable measures of arousal. In Mehan et al. [46], the intensity perceived in the virtual environment was directly proportional to the heart rate. Peterson et al. [47] showed that high virtual height conditions increased heart rate variability, electrodermal activity, and heart rate frequency power compared to low virtual heights, inducing a certain level of psychological stress. Virtual characters, such as avatars, are capable of eliciting emotional responses in people who observe their facial expressions and postural gestures.

Looking at the results of the aforementioned works, it can be firmly said that not only are bio signals (HR and EDA) evidence of the state of anxiety, but also that virtual reality is able to bring changes in the state of anxiety/arousal, which is closely linked to level of fear experienced by a patient. On the basis of this evidence different groups of researchers have started to develop physiological driven VR scenarios capable of evolving differently depending on the patient’s level of fear [14].

3.2.1 3D TVAdaptive VR

In 2020 Rosa et al[17] the authors combined a 3D-TV exposure with physiology to lead to a deeply personal experience by linking the VR environment to the users physiological state. The aim of this study was to develop and refine a physiology-driven application embedded in a non-immersive VR environment by assessing its ability to induce fear of cockroaches in individuals with different fear levels. User's cardiac activity was used as input by the VR system to determine the number of cockroaches within the scenario: if the heart rate was high the number of bugs increased while on the opposite if the heart rate was low the number of bugs decreased. According to the results obtained, the adaptive VR environment was effective in eliciting fear of cockroaches. Furthermore, this study served as a proof of principle that measurement of HR is sensitive enough to emotional response (fear), from which it is possible to adjust a VR environment in real-time.

3.2.2 Adaptive VR driven by EDA

Kritikos and colleagues [16] developed a rule based closed loop system [9] for the treatment of patients suffering arachnophobia. Initially, the system was launched to record the user's electrodermal response in resting conditions, without any stimulus introduced. After the system initialization, phobic stimuli (e.g., spiders) of graded intensity were installed, triggering different stress reactions according to each person's perception of fear. As soon as the electrodermal activity (EDA) response was recorded and processed, the system adjusted and updated the virtual scenario parameters to meet the anxiety reaction levels of the user each time. Essentially, the user provided feedback to the system with their physiological response, and the system recalibrated itself to adapt the virtual scenario to the user's response. The "virtual spider's" adaptive variables chosen by the authors were: generation frequency, jumping Force, probability of moving towards user, size and velocity. The aim of the study was to generate appropriate anxiety-inducing stimuli to bring each user's sweat secretion, and, therefore, the intensity of their anxiety, close to certain state.

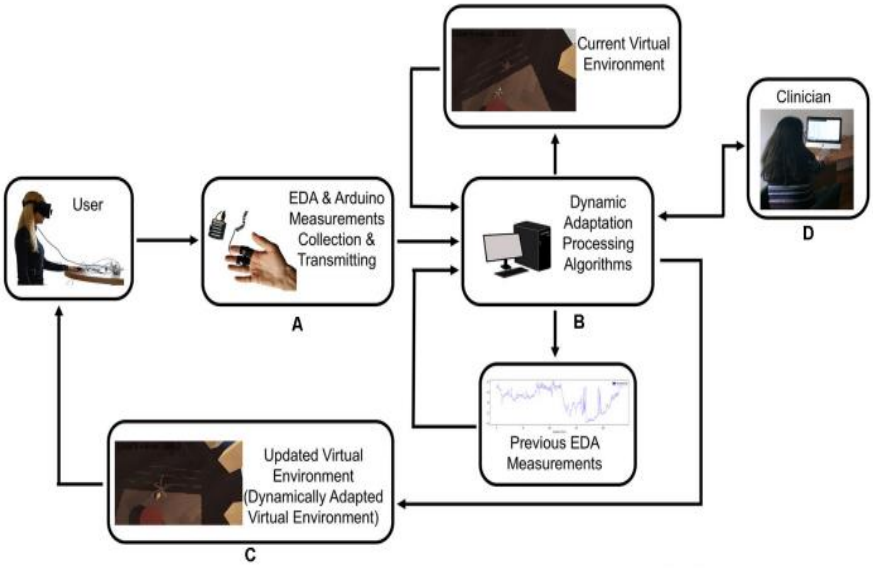


Figure 17 Kritikos at al. 2019 [16]. Eda-driven VR for arachnophobia treatment

The authors stated that the results of their work suggest it is possible to use a virtual reality scenario to adjust the psychiatric and neurological treatments according to each patient's unique characteristics and personality, in real-time, without the need for a system installation.

3.3 EEG driven adaptive VR scenarios

As previously mentioned, heart rate and EDA are not the only physiological signals capable of reporting the emotional state of a patient. In particular, EEG brain activity information, has been used to detect emotional states. Patterns of EEG activity have been found to distinguish emotions induced by stimuli with different valence and arousal levels [25]. Unlike adaptive reality scenarios guided by heart rate and EDA, the creation of an EEG-driven system requires some extra care and attention. The use of EEG in VR has been studied for a number of years, most of these studies are related to brain computer interfaces (BCI) and interaction, or monitoring user behaviour, not learning environments that adapt to user experience. However, the state of art regarding this topic is not completely empty: a little body of studies have used EEG to quantify user workload or user's emotional state [18]. EEG electrical signals are typically divided into bands of activity based on frequency, including Alpha, Beta, Theta, and Delta bandwidths. Alpha

activity is oscillatory brain activity in the range of approximately 9 - 13 Hz and is a valid and reliable measure of several key cognitive functions. Several variables related to alpha activity have been measured and related to cognition in the human brain. It was noted that alpha peak frequency in the brain increases in response to increasing task demand, with this being demonstrated as power decreases (alpha power desynchronisation) across posterior (parieto-occipital) scalp locations [19]. Taking advantage of this evidence, a group of researchers developed an adaptive VR scenario training for the first time [20]. The idea here is to use EEG alpha band to evaluate to user's workload and consequentially adapt the task difficulty to an ideal level which challenged the participants. The mean of squared signals coming from scalp electrodes were calculated for each epoch and then the mean of last four epochs was computed to get the resultant task load in real time. This information was then sent to Unity for use in managing the VR training application difficulty levels. The main contribution of this paper was conducting a user case which identified the relationship between brain activity and visuomotor performance in a VR training context.

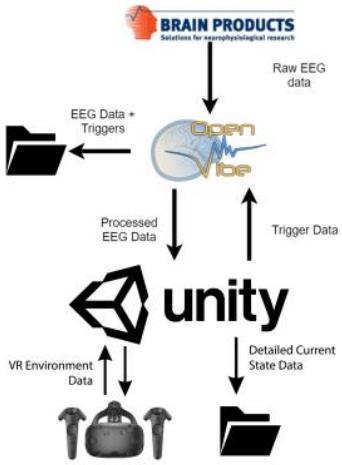


Figure 18 Zhang et al 2017 Adaptive VR system based on EEG activity recording.

Chapter 3

Development of a physiologically driven VRs for Social Anxiety Disorder Therapy

4 Materials and Methods

4.1 Description of the problem

Anxiety disorders are the most common mental illnesses. The last version of The Diagnostic and Statistical Manual of Mental Disorders includes in this category a variety of mental disorders such as specific phobias, post-traumatic stress disorders, panic disorder and social anxiety disorder (SAD). SAD, also called sociophobia, is a persistent fear of one or more social situations where embarrassment may occur, and the fear or anxiety is out of proportion to the actual threat posed by the social situation as determined by the person's cultural norms [26]. According to the ICD-10 classification of mental and behavioural disorders, engaging in feared situations is accompanied by autonomic symptoms of anxiety such as sweating (EDA), trembling or increased heartrate. Traditionally, the treatment of choice for social anxiety is cognitive behavioural therapy (CBT). Treatment is conducted both individually and in group-settings. Exposure therapy

is central to CBT and is very effective in fear reduction [27]. Despite their proven efficiency, exposure therapies are hardly accepted by most people that would benefit from them, with the risk of precluding treatments to the most aggravate patients.

Later in the years innovative therapies were proposed involving stimuli made less aversive by providing subliminal stimuli instead of clearly visible ones, and or by adding further steps to exposure thanks to technological possibilities offered by virtual reality. Speaking of SAD therapy, VR exposure has several advantages compared to the in vivo one: it provides readily available environments, exposure is highly controllable and can be modified to fit the needs of the patient. Moreover, therapy occurs in a safe designated room and thus the threshold for initiating exposure might be lower than for in vivo exposure [28].

In their first implementations for anxiety therapies, VR scenarios were used for combining subliminal and supraliminal stimuli, with the idea of hitting the subject without the latter knowing he was actually hit. This was done to avoid overloading the patient with fear-related elements that were uncomfortable for him, while still providing stimuli in a subconscious way.

In more recent years, VR scenarios were integrated with anxiety related biofeedback in order to create adaptive scenarios that could provide progressive exposure (see section 3.2).

The system proposed in this work is centred on this advanced generation of VR that adapts the virtual scenario to the specific needs and situation of the patient. The virtual administration of stimuli is driven by computational model of physiological signals modelling anxiety disorders, mainly focusing on SAD as clinical model. The idea is that the VR environment will push the subjects towards the targeted stimulus, as requested by exposure therapy, keeping it tolerable but at the same time effective.

The adaptive virtual reality scenario proposed is supposed to be a part of a bigger project. Indeed, the complete system should be composed by three elements:

- **a VR system** running the fame and integrating control algorithm that adapt changes in the virtual scenario in response to the subject's estimated anxiety state
- **a wearable monitoring platform** able to record in real time specific physiological responses of the subject
-**a real time model** able to process the recorded physiological signals and estimate the subjective anxiety level.

In this work, we focused our attention solely on the development of the adaptive scenario and on the study of its possible integration into a larger system. The integration of a sensor for biofeedback measurement was not considered as signal simulations were used. Furthermore, the mechanism that models the patient's anxiety used for our scenario, is extremely trivial and not very subjective. This step would also require the development of a more reliable model of the subject state, perhaps based on a machine learning classifier.

4.2 Scene design

The platform used to develop the VR was Unity [38]. The latter is a cross-platform game engine that can be used to create three-dimensional (3D) and two-dimensional (2D) games, as well as interactive simulations and other experiences. Unity is known for its ability to create games and applications that can run on multiple platforms, including Windows, macOS, iOS, Android, and more. Unity's Asset Store is a marketplace where developers can access a wide range of pre-built assets, such as 3D models, textures, sound effects, and more. These assets can be downloaded and integrated into projects to save development time.

Assets from the Asset Store or custom-made elements can be used to populate the game world or the scene. Objects can be manipulated, rotated, and translated within the Unity Editor to create the desired environment.

While the visual and design aspects of a game or application can be created using the Unity Editor, the core business logic is typically written in C#. This means that developers use C# to define how objects in the scene behave, interact, and respond to user input.

Unity provides a wide array of Application Programming Interfaces (APIs) [37] and components that developers can use to control the behaviour of game objects. This includes physics simulation, animation, sound, and user interface elements.

Since this application was designed with the purpose of exposing SAD affected persons to their phobia, it is logical that the scene should be populated with avatars. The latter where obtained through an open library called Mixamo [39] that provides a huge amount of 3d characters and animations.

The scenario created is an enclosed space that resembles a large office divided into several rooms. The rooms are furnished with simple design elements such as tables,

televisions, sofas, plants and chairs (see fig. 19). The environment, even if indoors is open enough to leave room for the avatars and the user to move freely. This type of design was chosen because projecting sociophobe people into a narrow scenario with avatars could cause them constant high anxiety and the application might not work as intended. A canvas on the top right of the scene continuously displays runtime information regarding the status of the application.

Figure 19 A view of the scene developed.

For the purpose of this work two avatars and their walking animations were downloaded from Mixamo (see Fig. 20) The first one, a mannequin, was used as a low fear-inducing stimulus, while the second one, a realistic girl was used as a second stage stimulus for higher fear induction.

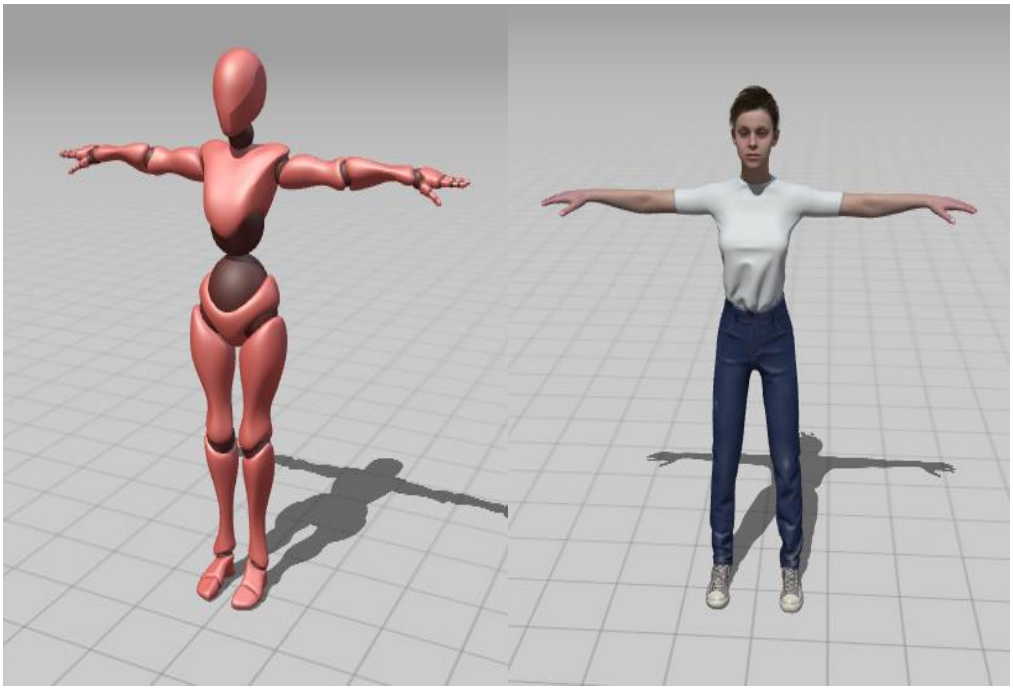


Figure 20 The two avatars used for the application, on the left the stylized mannequin while on the right the more realistic avatar.

4.3 Managing sense of presence

Avatars featured in the scene are in constant motion as if patrolling the area and, an associated animator controller, manages their walking animation. Moreover, each avatar is driven by an ai which calculates the route in such a way that they do not collide with objects and their movement looks realistic. This occurs thanks to a unity built in component called Navigation mesh (Navmesh). A navmesh in Unity is a critical component for implementing pathfinding and artificial intelligence (AI) movement in 3D environments. It is a simplified representation of the game world that defines where agents (e.g., characters or NPCs) can move. The first step in using a navmesh is to generate it. Unity provides tools to automatically generate a navmesh based on the scene's 3D geometry. This includes the terrain, static obstacles, and walkable surfaces. The second step is the definition of an agent by specifying its size, shape, and other characteristics. The agent represents the entity that will navigate the scene using the navmesh. For example, a player character or an enemy NPC might be agents.

Finally, agents move using the Unity's navigation system (NavMesh system). The latter handles pathfinding calculations, considering the agent's size and the navmesh layout to find the most optimal path while avoiding obstacles.

Although it might seem totally random, in reality the movement of the avatars occurs through small movements between well-defined points. Every time the avatars reach a destination the system takes advantage of this moment to carry out checks on the possibility of making adaptive changes. We will go more in the details of this statement in the next section.

Adaptive variables chosen for this work were: distance of movement of the avatar from the user, the number of avatars and the appearance of the avatars.
When the application starts there is only one avatar in the scene. Additional avatars, if needed, are inserted into the scene by cloning existing ones. it is possible to set the maximum number of avatars that can be reached in the unity hub before starting the application. There are two aspects that the avatars can assume: the less anxiety-inducing one, is a mannequin, similar to those that can be found in shop windows, while the other is a girl with sober clothes. Avatars can move in three different areas, each of them contains a cluster of reachable destinations. It was decided that the change of appearance could only occur for the first avatar on the scene and that clones could appear only once the first avatar had assumed the appearance of a person. In this way the change in appearance is the first step towards a more anxious scenario while the increase in the number of avatars is a stimulus that comes into play only later

Previously it was reported that when it comes to virtual reality therapy, it is essential that the application created is as close as possible to reality. Presence and immersion must be maximized because the greater they are, the greater the chance that participants will behave in a VE in a manner similar to their behaviour in similar circumstances in everyday reality.
Obviously in terms of performance and repeatability the best choice would be to make the changes happen immediately. This way the user's state of anxiety would be immediately matched by the stimuli provided in the scene. On the other hand, changes in appearance and the spawn of avatars before the patient's eyes would completely destroy his sense of presence and immersion. The solution to this problem was to have the swaps, spawns and disappearances occur only in places that were not visible to the patient. A direct consequence of this approach is that changes could not occur at a fixed time

distance from the cause that generated them. In facts the only feature that could vary as soon as the state switches was the distance of the avatar's movement from the user, this because the avatar can simply change his destination while walking without seeming strange to the user.

This was a key point in the development of the work as delays introduced for the reasons outlined above could lead to changes occurring too late. For example, it could happen that, while a patient is in a state of excessive anxiety, the system takes too long to reduce the stimuli. Indeed, there are several walls in the scene that are large enough to hide the avatars from different perspective. The more walls there are, the higher the probability that the avatars will be hidden from the patient's gaze and consequently the higher the probability that changes will occur suddenly.

4.4 Logical Scheme for Adaptation

The idea behind the adaptation mechanism was that the scenario could follow the level of anxiety of the user, more precisely, the scenario should help the user to be always around a moderate and controlled state of anxiety.

When the user is in a high level of anxiety the application should try to comfort him by changing in such a way that it stresses him less, for example by lowering the number of avatars. Instead, when the user is quiet the system tries to stimulate him by increasing the anxiety provided by the scenario, for example by increasing the number of avatars.

To achieve the above, the following logical reasoning was adopted.

The scene should be able to evolve by switching between well-defined different states. Each state is represented as a combination of the three adaptive variables. If, during the course of the simulation, the patient is not influenced by the stimulating changes in the scene or, if the patient finds himself in a to high level of anxiety, then the system switches state adopting a new combination. In this way, achieving a certain state is a good indication of the patient's anxiety response to the application.

Following this approach to the letter, the number of plausible states could become too large very quickly: for example, if each variable only had 5 possible configurations the number of states to manage would be equal to 125. Looking to this problem from a

technical perspective it is easy to understand that creating a system capable of managing so many states would be anything but trivial. The following table gives a graphic interpretation to the logic expressed above.

This table is only an indicative representation of how the patient's state of anxiety could vary.

The two states at the extremes of the table were the only ones assumed as fixed and immutable. The state on the extreme left is supposed the most anxiety-generating so, it will be reached only if the patient is comfortable while, on the opposite, the state at the extreme right is supposed to be the less anxiety-generating one and the scene will not move from there if the patient is too anxious. All the other states between the two side ends differ from each other for the variation of a single variable. It was not clear to us which of the three variables could affect the state of anxiety the most, so the "middle" states are interchangeable with each other as long as one of the variables moves towards one of the extremities. It would be interesting to better understand the weights of each variable in terms of anxiety induction. Giving an answer to this would mean to find a function that describes how the state of anxiety varies as a function of the variation of the features. Finding this function would be another whole problem that we decided not to address in this study. Therefore, for the reasons listed, we tried to implement a system that could manage a fewer number of states by handling variations of multiple variables in each states transition.

4.5 Implementation of the adaptive mechanism

4.5.1 Transition among states

The software on which all the VR is based is basically a Finite State Machine (FSM). The states that make the FSM are three and from now on for simplicity we will call them: state 1, state 2 and state 3. The higher the number that corresponds to the state, the more it can be associated with a chaotic and anxiety-producing state. During the development of this thesis, the biofeedback input signal coming from the adaptive control mechanism, was not available. Moreover, our aim was to keep the system valid in general. For this reason, we decided to drive the adaptation mechanism with the heart rate, as it has been proven to be a reliable marker of SAD related anxiety [27]. The application collects the heart rate signal for 30 seconds. If, in this time window, the average of the signal is higher than one hundred then it means that the patient is not able to sustain the state. Such threshold was chosen because it is a good approximation of the average heartrate of a person in a state of anxiety.

At the time of the check, if the average heart rate registered in the previous thirty seconds is above the threshold there are two possibilities. if the current state is above one then it is lowered by one level. Otherwise, if the current state is state 1, since it is the less chaotic, the application waits another 30s. If, the next check, the average is again greater than one hundred, the application quits for precaution.

On the other side if the average is less than the threshold it means that the patient is not in such an anxious state and therefore, we can think of stimulating him a little more by moving on to a higher state.

4.5.2 Adaptive variable management at state switch

Above it was mentioned that each state is defined as a combination of the three adaptive variables. For this reason, to be able to describe the evolution of the states depending on heart rate, it is necessary to give a detailed overview of the adaptation mechanism of each of the variables.

-Avatar appearance: At startup the system is in state 1. In this state the avatar looks like a mannequin. From the moment the state changes to state 2, each time the avatar arrives at his current destination the system "tries" to change his appearance from mannequin to person. The change takes place only if the destination reached by the mannequin is hidden from the eyes of the patient. To perform the change all the renderers of the current avatar are disabled, and a new character object is putted in its place. The switch of aspect occurs the same way every time the state passes from state two to state one, this time going from person to mannequin. If the finite state machine reaches state 3 no changes in appearance will occur anymore and the avatar will remain a person unless the application returns to state one.

Figure 21 Mannequin roaming the scene.

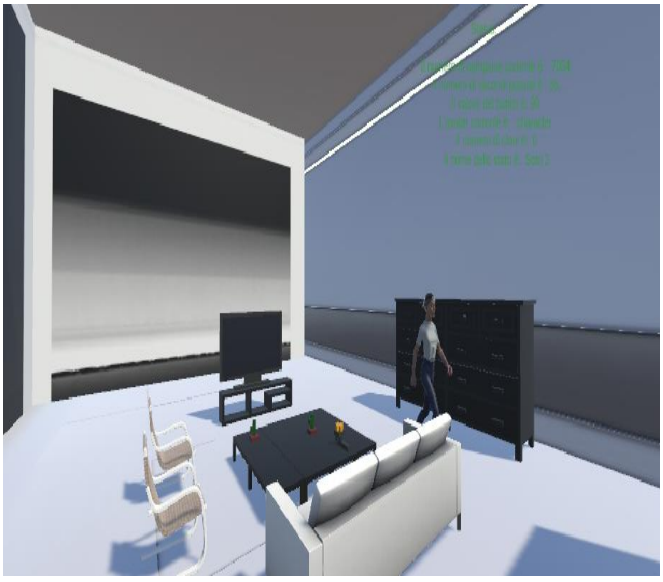


Figure 22 Person patrolling the scene.

-Distance from the patient: When the application is running the avatars move going from one point to another. These points are not random or decided at runtime, but they are predefined in a strategic way: the indoor explorable area of the scene is basically divided into 3 invisible sections, each section in turn contains four positions, so in total there are twelve possible locations divided into three clusters.

Each time the avatar reaches the location towards which it is moving the system makes a calculation to decide which one will be the next destination. The locations clusters are sorted according to their distance from the patient. After that, depending on the current state of the state's machine, the system selects a cluster:

- if the current state is state one the most distant cluster will be selected.
- if the current state is state two the medium distance cluster will be selected.
- if the current state is state three the closest cluster will be chosen.

Finally, a random point among those belonging to the chosen cluster is selected as next destination.

This schema is not applied to additional clones spawned in state 3, which move completely randomly between the various clusters.

Returning to the main purpose of this application we can say that a sociophobic person is more anxious when people are close to him rather than far away. Based on how the finite state machine is defined and using the movement pattern just described, the avatar will move close to the patient only after some time that the latter is in a calm state (state 3).

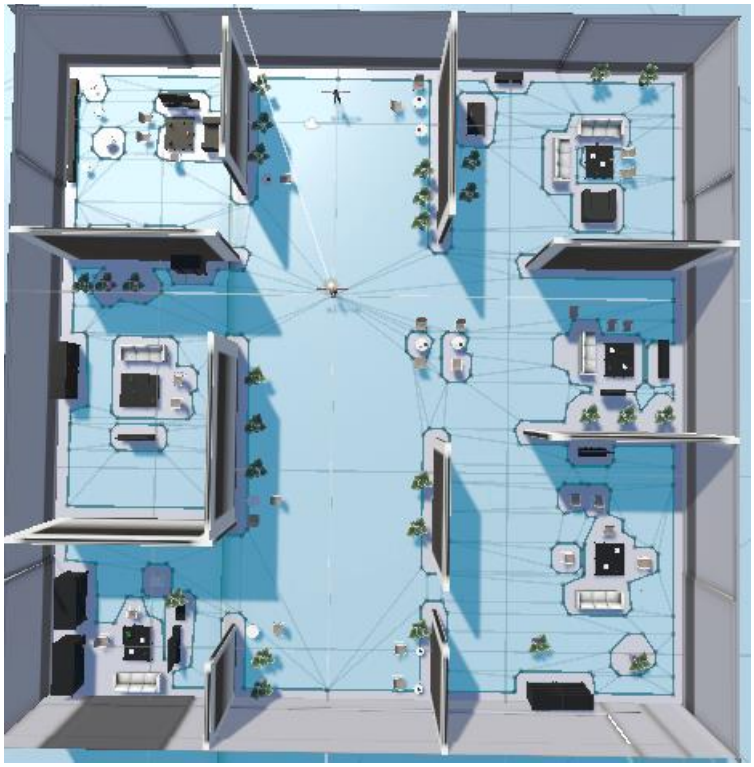


Figure 23 A top view of the virtual scene. From this perspective it is possible to better recognize the three different clusters area. The horizontal walls on the sides conceptually spear the scene in 3 areas.

-Number of Clones: During states one and two there is only one avatar inside the scenario. When the finite state machine goes up to state three if, the avatar's appearance is that of a person, one clone spawns in the scene. However, if, for some reason, the avatar still looks like a mannequin, then the system waits and as soon as the avatar reaches a new location where it is not visible it changes appearance and one avatar spawns.

As previously said state 3 is the maximum state reachable by the system. If the state machine is in state three and in the last thirty seconds the heart rate average did not pass one hundred then, instead of increasing again the state, the system spawns a clone. This occurs until the number of clones roaming in the scenario matches the maximum number previously settled, in that case nothing changes until the state drops to two.

To make the user experience more realistic the spawning of the clones is done with some tweaks: the system chooses a random location between the twelve available (see "distance from the patient"), if this location is not visible by the user the spawns occur

otherwise another location is chosen. This mechanism is repeated recursively until a hidden location is found.

As soon as the state switches from state three to state two, the process that leads to the destruction of the clones begins. Once a clone arrives to its current destination a check is made on the visibility of the location by the user: if the location is not visible the clone is destroyed otherwise it keeps going until it arrives to a not visible location. In case the state changes to state one and there are still clones, the system gives priority to the destruction of the clones: only once all the clones have been destroyed will the original avatar look like a mannequin again.

All changes that require the avatars to be invisible to user are managed exploiting a method provided by the Unity API. Using this method, a ray is generated starting from patient's position towards the direction that connects the latter with the destination of the avatars. If, during its path to the avatar's destination, the ray encounters an object belonging to a specific layer, then it means that the avatar is hidden. Obviously, to ensure that the avatar is effectively hidden, not all objects in the scene are considered by the ray. For this reason, there are large walls in the scene capable of entirely covering the avatar when the latter arrives at destination. Walls are the only objects to which the ray is susceptible.

Figure 24 Four avatars, with realistic appearance, roaming the scene during a simulation. This stage is reached only if the patient is really calm during the experience.

5 Results and discussions

5.1 Testing with simulated heart rates

Given the intrinsic high variability of the application's behaviour associated to the movement of the user (see section 4.5.2), it was not so trivial to decide how to structure objective assessments. The effectiveness of the developed system should be tested in the context of a user evaluation study, which is out of the scope of this thesis.

We ended up with the idea of collecting data that could give a general idea of the application's responsiveness to the heart rate changes. This means to quantify the temporal distance between when the finite state machine switches state and when adaptive variable effectively changes in the screen. Obviously, bio signal controlling the application was simulated but still, a short delay in the adaptive responsiveness, would means that very likely the application is feasible for human exposure. It is also worth noting that such delays are not depending on the actual biofeedback used, but on the devised switching strategy.

Only two of the possible variations in the scene are not immediate. and can consequently be included in a responsiveness analysis: the destruction of the avatars and the change in appearance. Instead, avatar's movement distance from the user starts to change linearly without being noticed as soon as the state switches. Also, avatar's spawn is immediate since it is mediated by a recursive algorithm that continues to be called until a covered waypoint, not visible to the user, is found. This likely happens in a matter of milliseconds and can therefore be considered an immediate process.

The application was tested with fictitious heartbeats created using MATLAB. These signals are sets of step functions which assume different values every ten seconds (the values oscillated in a possible range for a physiological heartbeat). Signal lasted three minutes and, to make everything as smooth as possible, their frequency was chosen equal to the update frequency of Unity (128 Hz).

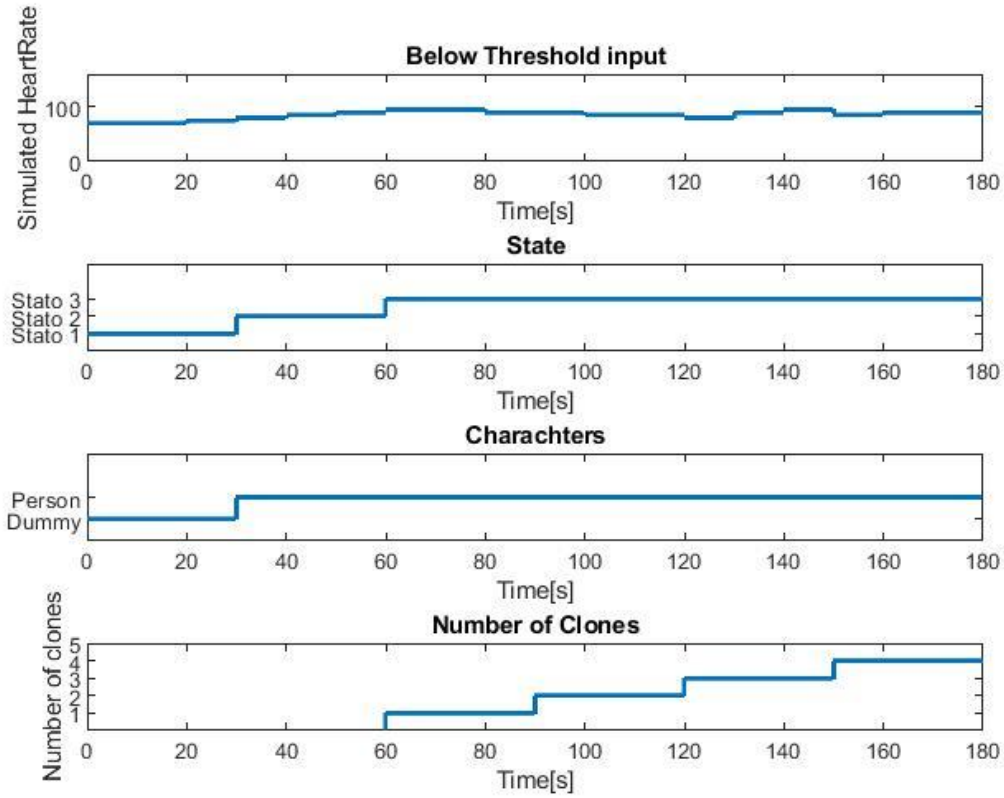
To evaluate the behaviour of the application in different plausible situations, each signal has different slope and pace between each other: crescent, below threshold, random and so on.
Since it was not possible to test the application on head mounted display, runtimes were simulated by launching the application with Unity on a PC. The position and the movement of a possible patient were substituted by the ones of the camera controlled with mouse and keyboard.

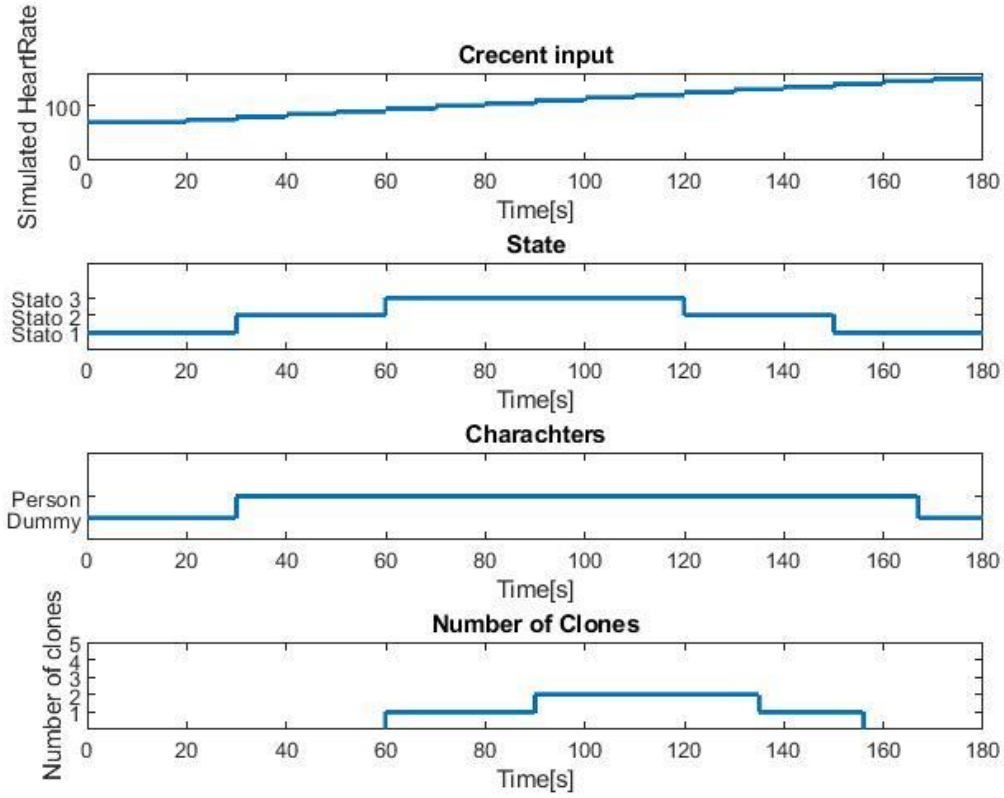
Six signals were used for assessment purposes. Three of them was created with a pace that could resemble heart rates of patients with different fear levels:
-A signal below the threshold of one hundred BPM for its entire duration, resembling a patient in a calm state.
-A crescent signal starting below threshold that could simulate a patient that could not face his fear and gets more anxious while the application runs.
-A crescent-decrescent signal that could remind the heart rate of a SAD patient that is able to adapt during the course of the application.

Three other random signals were added to obtain additional data for the analysis.
To keep data collection homogeneous between each testing simulation, it was decided to keep the camera still in a corner facing towards a wall. This because, when the avatars land on a destination target, depending on the position of the camera some changes may or may not occur (see section 4.5.2). With the camera standing in the location chosen, adaptation should be as fast as possible since in that position very few destination points could be visible to the camera.

5.2 Results Analysis

The plots below show (see fig. 25) the of the simulations driven with simulated heartbeat.


Below Threshold input
Simulated HeartRate
100
0
0 20 40 60 80 100 120 140 160 180
Time[s]
State
Stato 3
Stato 2
Stato 1
0 20 40 60 80 100 120 140 160 180
Time[s]
Charachters
Person
Dummy
0 20 40 60 80 100 120 140 160 180
Time[s]
Number of Clones
Number of clones
5
4
3
2
1
0 20 40 60 80 100 120 140 160 180
Time[s]


Crecent input
Simulated HeartRate
100
0
0 20 40 60 80 100 120 140 160 180
Time[s]
State
Stato 3
Stato 2
Stato 1
0 20 40 60 80 100 120 140 160 180
Time[s]
Charachters
Person
Dummy
0 20 40 60 80 100 120 140 160 180
Time[s]
Number of Clones
Number of clones
5
4
3
2
1
0 20 40 60 80 100 120 140 160 180
Time[s]

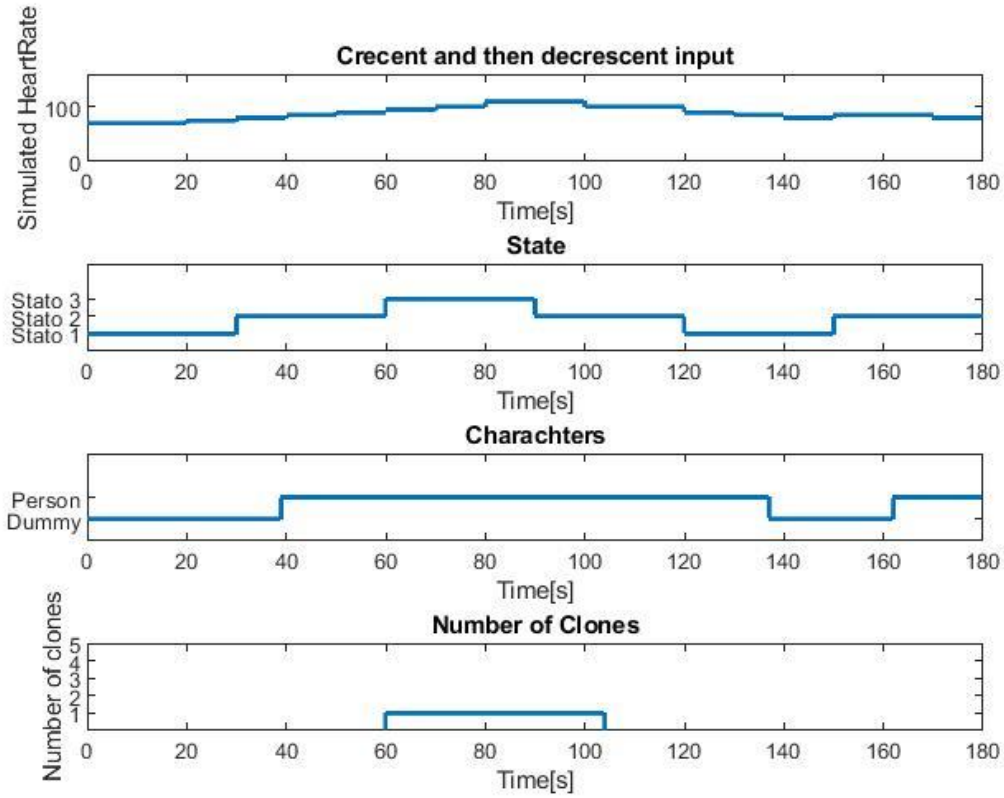


Figure 25 Plots showing results of the simulations using different input signals. From left to right: below threshold, crescent, crescent and then decrescent. The first row shows simulated heart rate, the second shows current state of the application, the third shows the appearance of the avatar and the fourth shows the number of clones roaming in the scene.

As said in previous sections (e.g., see section 4.5.2), the scenario implements a switching logic that, to avoid annoying changes for the patient, postpones the state updates. From now on we will name "delays" these waiting times for update changes. It is important to note that this term do not indicate delays due to application, hardware, software or data communication problems.

The experiments assessed a delay of 9.56 (e.g., see section 5.1) seconds for switching aspect and 13(5.74) seconds for clones' destruction. A remarkable standard deviation is the result aspects that assess time delays of the application very unpredictable. The main cause could be the distance of the avatar from its destination at the moment of the state's switch. It could happen both when the avatar is very far from its destination or very close to it. Furthermore, it must be said that the Unity API that checks the visibility of the user is not so precise. Sometimes it can happen that, in borderline situations, a variation is not allowed while in reality the avatar is hidden or vice-versa. This problem could cause

significant delays because, in order to perform the change, the system should wait until the avatar reaches the next location.

Another aspect to take in consideration, especially for further works, is that the translation speed of the avatars was chosen arbitrarily. However, in order to maintain a realistic scene, avatar's speed in the environment should match the speed of the walking animation so, translation speed cannot be modified to much if you want to achieve a decent sense of presence.

Despite the high standard deviation, it can still be said that an average delay around ten seconds for both the delays is not a bad result. Keeping in mind that the application switches state each thirty seconds, this means that the patient has more than twenty seconds for benefitting the adaptive changes before the next switch.

5.3 Further Developments

In the Introduction of this work, it was mentioned that this work is part of a bigger project. In fact, this virtual reality scenario alone is not self-sufficient to function on its own as an adaptive system for the therapy of people suffering from SAD. The system designed in the parent project expands the VR scenario with the addition of:

- a wearable platform able to record in real time specific physiological responses of the subject.

-a real time model able to process the recorded physiological signals and estimate the subjective anxiety level.

In this work, the environment was tested with simulated heart beats and not with real patients' signals. For this reason, it was not necessary to create a complex model to estimate the patient's anxiety state. A very simple idea was adopted, namely that of carrying out a simple thresholding on a signal window preceding the moment of estimation. However, these thresholds are absolute. In a real case scenario, the correct choice is to calculate subjective thresholds for each patient. A good idea might be to record the heart rate for a period in a calm situation, extract a baseline and calculate thresholds in terms of variations compared to this baseline.

Moreover, heart rate alone may not be enough to make an estimate. The anxiety generated by Social Anxiety Disorder or anxiety in general, results in more symptoms such as changes in EDA values and anomalies in electroencephalographic activity. A

more well-rounded system should perform classification based on more than one biofeedback. It would be interesting to structure a study on how to integrate the different anxiety-related physiological signals to create an accurate classifier. Using multiple signals can also be a good backup in case there are noise or communication problems for one of the signals.

Moreover, the developed scenario should be integrated on HMD. This would require verifying that the dimensions of the objects present in the scenario look the same as those in the simulations and, if they were not, they would need to be adapted.

Integration of the scenario on HMD is a crucial step in order to test its effects on people and collect their feedback. In fact, as mentioned before, it is not clear what weight each of the adaptive variables has in terms of anxiety generation. VR should be tested on people suffering from SAD and they should be subjected to questionnaire providing questions aimed to understand which changes are most influential for their state of anxiety. This way state transitions could be adapted to create a very efficient and robust system. Instead, for this scenario alone, state transitions are only results of assumptions made by the authors.

Furthermore, some of VR works for SAD therapy mentioned the adoption of interactive scenarios. This because anxiety generated by SAD is more likely to manifest in scenarios where the patient is tested/subjected by/to other people. A future revisitation of this work could lead to the addition of interactive elements in the environment, in particular between the user and the avatars (e.g., avatars looking at the patients and/or talking with them).

Bibliography

1. Sherman, W. R., & Craig, A. B. (2018). *Understanding virtual reality: Interface, application, and design*. Morgan Kaufmann.

2. Wiley, V., & Lucas, T. (2018). Computer vision and image processing: a paper review. *International Journal of Artificial Intelligence Research*, *2*(1), 29-36.

3. Sobota, B., & Mattová, M. (2022). 3D Computer Graphics and Virtual Reality. In *Computer Game Development*. IntechOpen.

4. Dhar, E., Upadhyay, U., Huang, Y., Uddin, M., Manias, G., Kyriazis, D., ... & Syed Abdul, S. (2023). A scoping review to assess the effects of virtual reality in medical education and clinical care. *Digital health*, *9*, 20552076231158022.

5. Ma, M., McNeill, M., Charles, D., McDonough, S., Crosbie, J., Oliver, L., & McGoldrick, C. (2007). Adaptive virtual reality games for rehabilitation of motor disorders. In *Universal Access in Human-Computer Interaction. Ambient Interaction: 4th International Conference on Universal Access in Human-Computer Interaction, UAHCI 2007 Held as Part of HCI International 2007 Beijing, China, July 22-27, 2007 Proceedings, Part II 4* (pp. 681-690). Springer Berlin Heidelberg.

6. Pandita, S., & Won, A. S. (2020). Clinical applications of virtual reality in patient-centered care. In *Technology and health* (pp. 129-148). Academic Press.

7. Donga, J., Gomes, P. V., Marques, A., Pereira, J., & Azevedo, J. (2020, August). Application of Adaptive Virtual Environments Through Biofeedback for the Treatment of Phobias. In *Proceedings* (Vol. 54, No. 1, p. 42). MDPI.

8. Baker, C., & Fairclough, S. H. (2022). Adaptive virtual reality. In *Current Research in Neuroadaptive Technology* (pp. 159-176). Academic Press.

9. Zahabi, M., & Abdul Razak, A. M. (2020). Adaptive virtual reality-based training: a systematic literature review and framework. *Virtual Reality*, *24*, 725-752.

10. Ćosić, K., Popović, S., Kukolja, D., Horvat, M., & Dropuljić, B. (2010). Physiology-driven adaptive virtual reality stimulation for prevention and treatment of stress related disorders. *CyberPsychology, Behavior, and Social Networking*, *13*(1), 73-78.

11. Vega, A. V., Madrigal, O. C., & Kugurakova, V. (2021, March). Approach of immersive adaptive learning for Virtual Reality simulator. In *Anais do III Workshop on Advanced Virtual Environments and Education* (pp. 1-8). SBC.

12. Britannica, T. Editors of Encyclopaedia (2017, April 18). psychophysics. Encyclopedia Britannica. https://www.britannica.com/science/psychophysics

13. de Souza, V. C., Nedel, L., Kopper, R., Maciel, A., & Tagliaro, L. (2018, October). The effects of physiologically-adaptive virtual environment on user's sense of presence. In *2018 20th symposium on virtual and augmented reality (SVR)* (pp. 133-142). IEEE.

14. Petrescu, L., Petrescu, C., Mitruț, O., Moise, G., Moldoveanu, A., Moldoveanu, F., & Leordeanu, M. (2020). Integrating biosignals measurement in virtual reality environments for anxiety detection. *Sensors*, *20*(24), 7088.

15. Chiossi, F., Welsch, R., Villa, S., Chuang, L., & Mayer, S. (2022). Virtual reality adaptation using electrodermal activity to support the user experience. *Big Data and Cognitive Computing*, *6*(2), 55.

16. Kritikos, J., Alevizopoulos, G., & Koutsouris, D. (2021). Personalized virtual reality human-computer interaction for psychiatric and neurological illnesses: a dynamically adaptive virtual reality environment that changes according to real-time feedback from electrophysiological signal responses. *Frontiers in Human Neuroscience*, *15*, 596980.

17. Rosa, P. J., Luz, F., Júnior, R., Oliveira, J., Morais, D., & Gamito, P. (2020). Adaptive non-immersive VR environment for eliciting fear of cockroaches: A physiology-driven approach combined with 3D-TV exposure. *International Journal of Psychological Research*, *13*(2), 99-108.

18. Zhang, L., Wade, J., Bian, D., Fan, J., Swanson, A., Weitlauf, A., ... & Sarkar, N. (2017). Cognitive load measurement in a virtual reality-based driving system for autism intervention. *IEEE transactions on affective computing*, *8*(2), 176-189.

19. Haegens, S., Cousijn, H., Wallis, G., Harrison, P. J., & Nobre, A. C. (2014). Inter-and intra-individual variability in alpha peak frequency. *Neuroimage*, *92*, 46-55.

20. Dey, A., Chatburn, A., & Billinghurst, M. (2019, March). Exploration of an EEG-based cognitively adaptive training system in virtual reality. In *2019 ieee conference on virtual reality and 3d user interfaces (vr)* (pp. 220-226). IEEE.

21. Du, K., & Bobkov, A. (2023). An Overview of Object Detection and Tracking Algorithms. *Engineering Proceedings*, *33*(1), 22.

22. You, S., Zhu, H., Li, M., & Li, Y. (2019). A review of visual trackers and analysis of its application to mobile robot. *arXiv preprint arXiv:1910.09761*.

23. Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (2000). Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings* (pp. 298-372). Springer Berlin Heidelberg.

24. Knop, M., Ressing, C., Mueller, M., Weber, S., Freude, H., & Niehaves, B. (2022). Virtual Reality Technologies in Health Care: A Literature Review of Theoretical Foundations.

25. Choppin, A. (2000). EEG-based human interface for disabled individuals: Emotion expression with neural networks. *Unpublished master's thesis*.

26. National Collaborating Centre for Mental Health (UK. (2013). Social anxiety disorder: recognition, assessment and treatment.

27. Ørskov, P. T., Lichtenstein, M. B., Ernst, M. T., Fasterholdt, I., Matthiesen, A. F., Scirea, M., ... & Andersen, T. E. (2022). Cognitive behavioral therapy with adaptive virtual reality exposure vs. cognitive behavioral therapy with in vivo exposure in the treatment of social anxiety disorder: A study protocol for a randomized controlled trial. *Frontiers in Psychiatry*, *13*, 991755.

28. Morina, N., Kampmann, I., Emmelkamp, P., Barbui, C., & Hoppen, T. H. (2023). Meta-analysis of virtual reality exposure therapy for social anxiety disorder. *Psychological Medicine*, *53*(5), 2176-2178.

29. Chard, I., & van Zalk, N. (2022). Virtual reality exposure therapy for treating social anxiety: a scoping review of treatment designs and adaptation to stuttering. *Frontiers in digital health*, *4*, 842460.

30. https://en.wikipedia.org/wiki/Feature_(computer_vision)

31. https://encyclopedia.pub/entry/28619#:~:text=Feature%20detection%20is%20a%20low,feature%20present%20at%20that%20pixel.

32. Sánchez, J., Monzón, N., & Salgado De La Nuez, A. (2018). An analysis and implementation of the harris corner detector. *Image Processing On Line*.

33. Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, *30*(11), 3212-3232.

34. Fleet, D.J., & Weiss, Y. (2006). Optical Flow Estimation. *Handbook of Mathematical Models in Computer Vision*.

35. Shah, S. T. H., & Xuezhi, X. (2021). Traditional and modern strategies for optical flow: an investigation. *SN Applied Sciences*, *3*, 1-14.

36. Bruhn, A., Weickert, J., & Schnörr, C. (2005). Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International journal of computer vision*, *61*, 211-231.

37. https://docs.unity3d.com/ScriptReference/

38. https://en.wikipedia.org/wiki/Unity_(game_engine)

39. https://www.mixamo.com/#/

40. https://www.vive.com/us/

41. https://www.oculus.com/rift-s/?locale=it_IT

42. https://www.meta.com/it/quest/products/quest-2/

43. Wiederhold, B. K., Jang, D. P., Kaneda, M., Cabral, I., Lurie, Y., May, T., ... & Kim, S. I. (2001). An investigation into physiological responses in virtual environments: an objective measurement of presence. *Towards cyberpsychology: Mind, cognitions and society in the internet age*, *2*, 175-183.

44. Riva, G., Davide, F., & IJsselsteijn, W. A. (2003). Being there: The experience of presence in mediated environments. *Being there: Concepts, effects and measurement of user presence in synthetic environments*, *5*.

45. Jang, S., Vitale, J. M., Jyung, R. W., & Black, J. B. (2017). Direct manipulation is better than passive viewing for learning anatomy in a three-dimensional virtual reality environment. *Computers & Education*, *106*, 150-165.
46. Marín-Morales, J., Higuera-Trujillo, J. L., Guixeres, J., Llinares, C., Alcañiz, M., & Valenza, G. (2021). Heart rate variability analysis for the assessment of immersive emotional arousal using virtual reality: Comparing real and virtual scenarios. *PloS one*, *16*(7), e0254098.
47. Peterson, S. M., Furuichi, E., & Ferris, D. P. (2018). Effects of virtual reality high heights exposure during beam-walking on physiological stress and cognitive loading. *PloS one*, *13*(7), e0200306.