Abstract: Today, in the majority of the shops, customers try the shoes to feel the comfort, look at themselves in a mirror and finally buy the chosen product. However, the trial often encounters difficulties: i.e. the desired shoe doesn’t feet. In EUROShoE Project, methodologies and processes for a custom-made shoe are conceived. A new problem comes out; the customised shoe will not be ready in the shop for the trial. ITIA-CNR is designing an Augmented Reality System where the customer can “wear”, fell and see the just customised shoe: FootGlove and MagicMirror will be the haptic devices will allow this process.
1- Introduction

Today, in the majority of the shops, customers try the shoes to feel the comfort level, looks at themselves in a mirror to see their look, and finally buy the chosen product. In this natural decision process, the customer often encounters different difficulties: there is not the shoe he wants to try (just the copy in the shop window exists), there is not the size of the model he wants to try, because of colour or because of some other aesthetic or fitting characteristic the shoe is not appreciated and the customer is forced or to buy nothing or to search for another kind of shoe.

In the context of the EUROShoE Project a custom-made oriented methodologies and process for a custom-made shoes are conceived, so that the process can be more dynamic and free for the customer. However a new problem comes out; the customised shoe will not be ready in the shop for the trail [1, 2, 3, 4, 5, 6].

In order to avoid the above problems, the EUROShoE Project is creating the shop of the future, different in its own nature: the customer doesn’t have a static pre-built set of samples among which to choose the desired product, but he can customise the product itself and then order its realisation, so making its dream come true.

This kind of approach opens new ways to the shop concept because it moves the focus from the concept of mass production to the concept of mass customisation. By having the opportunity to do a “virtual trial” of a product, the meaning of “store” can be modified because less space is requested for storage than the usual one. By excluding the need of a large storage space in a shop, there is the possibility to open new shops in little small spaces that otherwise could not be utilised.

ITIA-CNR, in the EUROShoE Project is designing the way to achieve this new approach: the system, with a special laser scanner, samples the three-dimensional shape of the customer’s foot. Then the customer, thanks to a dedicated software application (called Cosmic) sets the aesthetic and material parameters based on the three-dimensional model of its foot. Finally, through an Augmented Reality System, the customer wears a special device shaped as a shoe (the FootGlove) and looks himself in a computer monitor (the MagicMirror). In this mirror he will see his image “reflected” with the shoe (previously customised and parameterised) on the foot where the FootGlove is worn, which is a live shoot by a camera. In the meanwhile the FootGlove, a modular physical device reconfigurable during the trial time of the customer, gives him also the feeling of the wearing. So, the aesthetic set-up will be seen in real-time on the virtual shoe worn by the customer and the ergonomic set-up will be felt with the physical configuration of the FootGlove itself. The shoe virtually worn has been built with the characteristics customised on the previously sampled customer’s foot.

Another relevant factor in this process is the presence and the interaction of the customer with the system: he doesn’t just perform some configuration on a commercial web site but he really can see the virtual shoe in the Magic Mirror integrated with his person and his clothes.

Actually, ITIA-CNR is working on the feasibility study of a prototype system that can satisfy the minimal requirements: with this application it’s possible now to see in a computer monitor a shoe’s 3D model rendered in a camera live stream of a last. The last is the simulacrum, the pretence of the FootGlove: by moving the real last, the corresponding virtual shoe seen on the monitor (with the rest of the real environment) moves with 6 degrees of freedom. In this way, for the purposes of the system MagicMirror + camera + FootGlove, the Augmented Reality’s problem of the occlusions of the virtual objects is resolved: for example, if the customer’s clothes cover a little bit the FootGlove, the related part of the virtual shoe is in effect covered; if the customer moves a leg by partially (or totally) covering the foot with the FootGlove, in the Magic Mirror the virtual shoe is partially (or totally) hidden.

Moreover, in the simulation of the trial of the shoe a tactile feedback is necessary. The
FootGlove has the ambition to be the engine of this kind of haptic feeling. This device must fit to all the sizes and with no large number of components to assemble together. The study of the relations that rule the dimensional variation between the sizes of the whole last and its parts actually permits to think about the realisation of a FootGlove model that can be modular and scalable for all the shoes sizes with a restricted number of components.

It has been graphically verified as fitting, through a careful choice of upper’s sections and a translation, the outside volume of a less number to a greater number is possible. Findings confirms that the approach taken was able to satisfy the demands of the FootGlove, the next step has been the creation of a three-dimensional model of the form to see as the outside volume change weighs upon the internal volume of the shoe. The result has been that the internal volume behaves like the outside volume; it increases in proportional way.

As regards the sole and the insole a solution, which must still be verified, has been proposed. Seeing the FootGlove only through a model produced to the computer is possible at present but a physical model is in the realization phase.

2- The MagicMirror

In this paragraph the results achieved by the feasibility study about the Magic Mirror are presented. The technology that makes possible the realisation of this functionality is the Augmented Reality (AR) \[7, 8, 9, 10\]. The AR is a field of Virtual Reality (VR) where the user sees in a visualisation device both the reality surrounding him and artificially generated visual informations. To obtain this result the scene is live shot and the 3D models are added so that the user, as final result, can see the image of the real scene which contains also the virtual objects.

In the case of MagicMirror the user must see his own real legs and the virtual shoes. In AR, the most utilised technique to render the 3D objects mixed with the real scene is that one of the markers: the markers are sheets with drawn symbols, a vision system that live shoot the scene recognizes them and computes their position and orientation and utilizes this data to place the 3D geometry. Depending on the type of experiment or of application, the markers technique can be convenient. The marker, due to its nature, must be always visible from the camera so that the software can compute its position. It is evident that, in the trial stage of the shoe, the user can turn himself on one side or that he can tilt his foot, and these are operations that, in general, don’t grant the presence of the markers in the camera’s field of view. Moreover, even if this is just a feasibility study, to ask to the user to do not perform some particular move that brings away the markers from the shooting field of the camera it’s very constraining. The solution actually adopted is with a 6 degrees of freedom (DOF) position sensor coupled with the FootGlove so the position is read directly by the hardware instead of the use of algorithms of markers position recognition that overload the computational complexity of the system.

The only disease element is the sensor’s cable but surely it influences less that to constrain the user in a small set of predefined movements.

A difficult aspect of an AR system is also the computation of the depth of the digital objects for the right rendering of occlusions. An object is occluded, partially or totally, by another when it is behind it with respect to the look direction. In reality, this happens in a natural way: the light reflected from the hidden part doesn’t reach the observer because there is an object blocking its trajectory. In general, in the digital environments this must be simulated in a coherent way with the reality’s laws. This is usually done in totally digital environments (for example, VR or 3D CAD). Instead, in AR systems, the occlusions recovery is problematical because every single frame of the live shooting it’s
just an image without any kind of explicit threedimensional information. Hence, in the act of overlaying the digital object to the actual frame, the system doesn’t know which parts are to be occluded. As a result, the digital objects are, in general, always in the foreground. The techniques to solve this problem are still computationally heavy and not suitable for a realtime system at, at least, 20 frame/s unless a specialized high-cost system.

For its own intrinsic nature, the shoe trial stage implies that the user moves and its movements generate partial reciprocal occlusions of foots. Moreover, this implies that is not possible, unless assuming great constraints on the user’s movements, to avoid the occlusions problem. In a first phase, various solutions have been analysed by searching some special camera able to compute in real-time the depth map of every pixel of the live shooting, but they have been discovered as very expensive.

An alternative has been the simulation of the binocular vision with two cameras and then utilizes the epipolar geometry relations to derive the right depth values of the pixels in the live shooting. This solution has been abandoned because it is very computationally heavy: the mathematics is not simple and is necessary the knowledge of the inner parameters of the camera, not always so easy to have. It is necessary, to maintain the real-time performance, that every second 20 frames can be processed and sent to the monitor.

The idea for the solution to the occlusion problem has been found thinking about to the nature of the system itself: the user wears the FootGlove that, thanks to its modularity, is reconfigured in a way to have the right measure of the user’s foot. In any case, the shoe, by considering its shape, it’s an object that can be replaced with the FootGlove. By keeping in to account these factors, therefore we thought to apply to the system the chromakey technique, usually employed in the movies for the digital effects and in the virtual sets of some television broadcast. The chromakey technique, in its classic use, involves the construction of a set with a particular colour, the chromakey, that doesn’t appear in the scene, for example blue or green. Then, the set with the actors and the needed objects for the scene is shot. In the post-production phase, the movie is processed and the regions where there is the chromakey are replaced with the desired image of the scene, digitally generated. After this post-production processing the final sequence with the right set designing is obtained.

By extrapolating the chromakey technique and by contextualizing it in the considered system, the FootGlove plays the role of the chromakey background and the 3D model of the virtual shoe plays the role of the digital set designing. In this case, instead, the technique is utilized in real-time and without post-production additional operations. The system is defined as follows: the FootGlove is a green object, so that it’s a colour difficult to confuse with other colours in the camera’s shooting field. The camera live shoots the user’s foots that wear the FootGlove and send the computer the video. A dedicated software searches in real-time in the frame the chromakey pixels, then it does the rendering, in the frame just received, of the virtual shoe regions corresponding to the chromakey points. The obtained result is that, in the frame, the FootGlove’s image is replace by the 3D model’s image. This technique resolves the occlusions problem for this application case: if a hand is in front of the FootGlove, only a part of its chromakey surface is visible and, consequently, only a part of the 3D model is represented in the frame and, in the live video, the user sees the digital shoe occluded by the real hand.

2.1- The First Experimental Software

Under the considerations and the first analysis performed, a first experimental software has been realised with the aim to improve the applicability of the concepts and of the algorithms. Let’s call this software, merely, MM, MagicMirror. It is a OpenGL software written in C++, composed both by from scratch code and by integrating existent
libraries. It is structured in many parts, everyone of them identifies a logical stage of the
process: live video acquisition from the camera, chromakey processing, virtual shoe
rendering, loading of the 3D geometries, real-time management of the virtual shoe’s
geometry, management of the position sensor coupled with the FootGlove.

MM is subdivided in two parallel threads of execution: one thread manages the position
sensors, the other one performs the processing also by using the updated data of the
sensors thread. The two threads access a shared memory region where, with a
producer-consumer process, respectively, write the updated sensors position and read
the sensors position. A semaphore is employed to synchronize the access to the shared
memory area.

ARToolkit, that is a free development library for AR applications, performs the real-time
acquisition of the live video shot by the camera in the MM. It does calls into ARToolkit
to obtain the frames from the camera live shooting and to transfer them in a dedicated
memory buffer for the further processing.

When the single frame is stored in the buffer, the green pixels are searched. In this
process the image is scanned pixel by pixel and the Red, Green, Blue (RGB)
components are checked. If the RGB is in a specified range (that corresponds to the
desired green) the pixel is saved in a dedicated list, (called ChromakeyPointsList),
otherwise the pixel is discarded. After this operation a rendering pass (Pass A) of
the frame just acquired from the camera is done. So, in this moment, in the OpenGL
visualisation buffer (Colour Buffer) the image of the frame is stored.

Figure 1: The two threads of execution and the main tasks
In the next step, the OpenGL Stencil Buffer is utilised to set up the rendering of the virtual shoe overlayed to the green pixels. The Stencil Buffer is by default reset to 0-values at every frame, and then the pixels saved in the ChromakeyPixelsList are rendered (Pass B) on it. The chromakey pixels leave a 1-value in their corresponding (x, y) position in the Stencil Buffer, a 0-value is stored in the corresponding position of the other pixels. Finally, with the Stencil Test enabled, the virtual shoe is rendered (Pass C) and it’s overlayed on the existent frame’s image in the Color Buffer (with Pass A). In this way, the pixels (of the Pass C) of the virtual shoe in the (x, y) position with a corresponding 1-value at the (x, y) position in the Stencil Buffer are overlayed on the existent ones in the Color Buffer. The other pixels (of the Pass C), related to the 0 values, are not visualized. So the remaining pixels (of the Pass A) remains. The Color Buffer configuration obtained is finally sent to computer monitor. The user sees on the monitor the 3D geometry only in the regions that were chromakey pixels in the original frame live shot by the camera, in the other regions he sees the real world, live shot from the camera.
A important element of the MagicMirror is the virtual shoe. The main tasks that can be done in the choice process are to try different measures of the same shoe model and to change the aesthetics of the model itself. These operations are managed by the C++ class VirtualShoe which allows the MM the low level management of the virtual shoe’s geometry.

If the user wants to try a different measure of the same model, the visualised shoe must increase or decrease its dimensions. The VirtualShoe class allows the editing of the dimensions of the 3D models, by rescaling its vertices, in an independent way in its height, width and length Cartesian dimensions. The scale factor depends on the adopted reference system: French, American and so on. In order to keep the process parametric and general, every scale factor and every geometry modification of the 3D model is independent in the \((x, y, z)\) axes. In this way, besides satisfying the main scaling functionality, the system is set to real-time scaling about independent axes.

MM supports two formats for 3D models: ASE and 3DS, both exported from the 3D modeller 3DStudioMax. They are parsed and stored into memory by a dedicated free library, opportunely rearranged and generalised. They can be uploaded and downloaded on-demand by the application in expectation of the decision of the user to try different shoe models. Obviously, the choice of the format to be officially adopted for to come prototype has to be done.
2.1.1 Resizing the Virtual Shoe

In the real modeling of a shoe shape there is a point in the last which is the reference of the size transformations of the last itself to change from a measure to another. This point is called Development Centre (DC). It is located inside the bounding rectangle of the insole. So, in the FootGlove, the DC is the point the expansion of the physical device is referred to; in the MM the DC is the point the expansion, coherent with that one of the FootGlove, of the virtual shoe is referred to. We defined that the DC is located at 65 mm from the heel of the insole, in the direction of the toe. The position sensor of the FootGlove should be mounted in this DC. The 3D geometry of the virtual shoe must have its reference frame in this DC so that it is the identity of the transform from one size to another. In fact, the 3D model’s transform is performed by modifying the coordinates of the vertices of its polygons. If the point with respect to the coordinates are transformed is not the reference frame of the geometry, besides a resizing also a rigid translation in the space is obtained. This is because also the reference frame of the geometry is transformed. If the DC and the reference frame are coincident, only the resizing is obtained and the 3D model stays correctly in the place tracked by its sensor.

2.1.2 SW Architecture

The MM experimental software, as stated above, has been realized by integrating dedicated designed code with freeware libraries.

The part for the acquisition of the live shot frames from the camera is based on the ARToolkit library of the Human Interface Technology Lab of the Washington University. The main functionalities of this library allow the development of a AR application with the live stream acquisition, the marker detection and their position computation, the geometry rendering. ARToolkit uses the free DirectShow and Vision Microsoft libraries for the low-level interfacing with a camera.

![Figure 4: The structure of the SW libraries of the MM](image)

The part of the sensors management is based on the Polhemus_lib library, whose original free code has been extended and readapted.

The shoes 3D models loading part is based on the ASELoader and 3DSLoader libraries, whose original free code has been extended and readapted.

The part of the multi-pass rendering and, more in general, the visualisation part, is OpenGL library based.

2.1.3 HW Architecture

The HW architecture actually adopted by the system is the following:

The camera is a Agfa CL 20, 1 Megapixel, plugged to the computer with USB port. The
sensors are Polhemus Fastrack II Long Range, plugged to the serial port. The computer is a dual-CPU 2.4 GHz Intel Xeon, with Nvidia Quadro 4 980 XGL, 128MB RAM, graphic board. With this configuration the whole system runs at 60Hz frame-rate with live video stream acquisition at 640x480 pixel and then doubled to 1280X960 in the visualisation stage to the monitor.

Figure 5: HW configuration

3- The FootGlove
FootGlove is a system able to simulate the shape, the dimensions and able to reproduce the contact points of the footwear as regards the foot. Therefore, the FootGlove must return the user the sensation the shoe, which is wanted to buy, would supply to the user if he wears it, going to replace in "virtual" way the test of the footwear. The characteristics of the FootGlove are to suit the change of length, Calzata and footwear typology. This paragraph is concerning the possibility of obtaining from a basic form, with opportune transformations, all the shoe sizes. The greatest difficulties in the feasibility study have been discovered because of the handmade in the production of the lasts and of know-how handed down in oral way. Today the last is handmade from artisan and then acquired by a 3D scanner to obtain a 3D model built by polygons. This has involved the study of dynamics of production and an in-depth research of personal experience and oral know-how (even if consolidate and international) to make the techniques of development of last emerge. From the studies it has emerged that all the lasts are resized respect a point called Development Centre (DC) according to predefined parameters that change depending on the measurement system taken into account (following table).

<table>
<thead>
<tr>
<th></th>
<th>Calzata</th>
<th>Width</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>6 mm</td>
<td>2,2 mm</td>
<td>8,46 mm</td>
</tr>
<tr>
<td>French</td>
<td>4-4,5 mm</td>
<td>1,2/1,4 mm</td>
<td>6,66 mm</td>
</tr>
</tbody>
</table>
Figure 6: The complexity of the FootGlove system

The first step is the DC’s definition, it is not a fixed point, but it must remain inside the insole’s bounding box (Fig.7).

Figure 7: The Development Centre

To obtain all lasts numbers radial scaling as regards CDS is needed (Fig.8).

Figure 8: Resizing a last
By now, the lasts scaling as regards the 3D model of a basic last is defined by software that, by using the parameters of length’s variation and calzata, produces the last of next size. The scaling has been decomposed in slice’s translation of the basic form for the single axes (x, y, z).

These slices have been generated and translated to be super-imposable and comparable with the last one generated by CDS's scaling.

To create the digital models of slices and of insole it is necessary a complex set of passages. The 3D model that has been used in this study is a 3D scanning of a last and its original file format is IGES. Converting the format has been necessary to make it usable with our software (AUTOCAD, 3DSMax) and to reduce the number of polygons of the model.

The way to use the last model with 3DSMax is the conversion of the file from IGES to VRML using Pro-Engineering. The WRML obtained, is composed from several triangles (one layer for all triangles) and to simplify it, the file has been exported in DWG (AutoCad format) where all triangles have been put in a single layer.

![Figure 9: Process to obtain a usable 3D model](image)

The scaling of the last in x and y happens towards the insole, so it is necessary open the file with several triangles and delete those ones that aren’t part of the insole.

![Figure 10: The insole and last alignment](image)
Once the insole is obtained, the last is imported on a single layer and aligned with the insole itself (Fig. 10). The alignment is important in order to cut the last towards development centre, on the insole, in a right way. To generate a 2D model from the 3D insole is necessary to set to zero the z-values (Fig. 11).

![Image: Insole transformation](image)

**Figure 11: Insole transformation**

Now, importing the 2D insole into AutoCad, the bounding box can be generated and its CDS. From CDS the lines that intersect the insole perimeter has been drawn and the slices that will be translated to scale the last has been created (Fig. 12).

![Image: Slices definition](image)

**Figure 12: Slices definition**

Moreover, the translation lines has been created in order to move the slices: the middle point (C) of line joining the points of intersection of the lines drawn from CDS (A, B) and the CDS (Fig. 13).
The number of slices is defined by necessity of a good overlap between insole's arc (G), created by lines drawn from CDS (E, F), with the same arc of insole of bigger numbers (H) (Fig. 14), better the overlap is request bigger will be the number of slices.

See now what happens in 3D. To create the slices plans of cut (perpendiculars to plane of insole) has been generated through the lines that identify the arcs of insole (Fig. 14; E, F) and then the model of last has been cut (Fig. 15).

To verify if the choice of number of slices is correct, it has been necessary to compare the model generated with slices translation with the model generated by last scaling. It has been measured in AutoCad that the surfaces are superimposed in a very precise way (the error is of the order of $10^{-4}$ – $10^{-5}$ meter) in x and y-axis, while for z-axis an error of the order of $10^{2}$ has been measured, in fact has not been considered that increasing x and y also z needs to be increased.

To create the sections, in 3DSMax, plans of section (perpendiculars to plane of the insole) have been generated through the lines that identify the translation direction (Fig. 17).

With the creation of sections it is now possible to put the last's model in the origin that will be revealed very useful in the management of the positioning of slices for the
translation in z-axis, because the intersection between slices identifies the last's CDs (see a set of pictures to explain the sequence).

Figure 15: the 3D slices

Figure 16: Superimposing of last generated with scaling towards CDS (red last) and last generated from slices translation

Figure 17: Plane of section
All the sections has been reoriented in an orthogonal way to the observer and divided in slices. 3D Slices for the z-axis are the result of a cut between the slices generated for the translation towards x and y-axis and cones generated in CDS (Fig. 19). The dimension of cones (radius and height) has been founded for all slices from subdivision in slices of section (Fig. 20).

When all cuts have been done the FootGlove looks like in Fig. 21. To verify, like for translation in x and y-axis, if the choice of number of slices is correct the model generated with slices translation has been compared to the model generated by last scaling. Also in this case the superimposed error between the lasts (translated and scaled) is of the order of $10^{-4} - 10^{-5}$ meter.
Figure 20: Subdivision of section to find the dimension of the cones

Figure 21: FootGlove after cuts for translation in z-axis

Figure 22: Superimposing of last generated with scaling towards CDS (blue last) and last generated from slices translation in x, y and z-axis
4- Conclusions
The authors know that the studies presented are just the starting point for the creation of the first two prototypes of the devices and lots research activities has to be performed. The different technologies of the MagicMirror need to be further investigated and integrated in a single application. The FootGlove device has to be conceived and designed, and then the engineering will take place. Moreover, some of the input information will come from different new software tools under development in the EUROShoE project (the foot 3D scanner and the shoe-configurator) and will be ready just next year. After that also the data integration should be taken into account. However the partial results obtained encourage ITIA-CNR to go ahead for reaching in 3 to 5 years commercial products.

References
http://www.euro-shoe.net/
J. P. Mellor, Seth Teller, Tomas Lozano-Pérez, Dense Depth Maps from Epipolar Images, Nov. 1996, A.I. Memo No. 1593, MIT, AI Laboratory.