STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVIII**, Number 4, 2013

OPTIMEASURE: A MOBILE IMPROVEMENT ALTERNATIVE TO CLASSIC OPTICAL MEASUREMENTS

ADRIAN SERGIU DARABANT, MIHAI CLAUDIU ISPAS, AND DIANA BORZA

Abstract. This paper presents a novel approach to measure the most important morphological parameters needed for any eyeglasses prescription, with an error less than one millimeter. OptiMeasure is an Android application that successfully replaces the inaccurate and error prone traditional instruments used in optometry. The measurement process is performed in a few quick and fully automated steps, without the need of supplementary optical equipment, other than the tablet itself. We proposed original measurement corrections in order to obtain the results as accurate as possible. The final results of our system were validated against those obtained with a calibrated measurement device.

Key-Words: image processing, optical measurements, mobile applications, optometry

1. Introduction

Optometry is a health care branch of medical ophthalmology concerned with the health of the human visual system and related optical instruments / imaging devices. Modern eye glasses manufacture and prescription require a high precision measurement of some morphological parameters of the patients face and their correlation with morphological parameters [1] of the frame that the patient will wear. The most important are:

- *Interpupillary distance (PD)* - distance between the projections on the cornea of the centers of vision for the left and right eye and *Half Pupillary Distances* - the distance between the projections and the correspondent border of the spectacles;
- *Frame Bridge* - distance between left and right glasses;
- *Frame Boxing* (height and width) for each lens;

Received by the editors: September 28, 2013.

2000 *Mathematics Subject Classification.* 03G10.

1998 *CR Categories and Descriptors.* I.4.6 [**IMAGE PROCESSING AND COMPUTER VISION**]: Segmentation – *Edge and feature detection*; I.4.7 [**IMAGE PROCESSING AND COMPUTER VISION**]: Feature Measurement – *Size and shape*; J.3 [**LIFE AND MEDICAL SCIENCES**]: Medical information systems – .

- *Fitting Height* - the distance between the bottom of the lens and the eye pupil height in the frame as the patient wears the frame;
- *Pantoscopic Angle* - angle between the spectacle plane and the vertical plane.
- *Vertex* - the depth distance between the cornea surface and the spectacles lens surface;
- *Frame/Face Wrap Angle* - the angle between the vertical plane and the lens plane in the spectacles;

Modern lenses are complex and accommodate on the same pair of lenses all types of vision: far, intermediate and near. There is no need to change the spectacles for different types of activities. The above parameters need to be measured thus in two variants: *far vision (FV)* and *near vision(NV).* In FV the patient is looking at infinite with the eyes axes of vision parallel. NV is necessary for measuring the eye behavior in reading position when the gaze converges near towards the reading material.

The classical tools for measuring the morphological parameters are the ruler and the pupilmeter (for the interpupillary distance). However these methods are error prone, so recently several computer aided systems were developed to provide the required measurement accuracy.

Most of the large lens and eyeglasses producer companies [2, 3, 4, 5], develop systems that measure the interpupillary distance with errors tolerances ranging from less than millimeter to a few millimeters.

The measurements are performed on a facial image of the patient. To convert the pixels from the image to real world scale, these methods make use of an additional object. This additional object, called support, is placed on top of the patient‘s eyeglasses rims and is often used as a conversion factor between pixels and millimeters. The support has some markers/patterns that can be precisely identified by the operator or, can in some cases be detected using advanced computer vision algorithms. The real distances between the markers and their relative positions are precisely known.

The eye center of vision corresponding to the pupil centers need to be determined with high precision. Most measurement devices producers on the market usually pick the visible center of the pupil which is not usually accurate enough. The rest of them are projecting a source of light into the eye and capture its reflection on the cornea.

To perform a measurement, the patient stands still in front of the measuring device at a distance of approximately 2m for FV and a picture of its upper body part is taken. Most of the devices have adjustable heights, in order to accommodate to various patient heights [2, 5].

The Smart Centration Diamond System from ACEP [2], can also be used to train the patients, as it can present 3D simulations videos, such as anti-reflective coating simulation and polarized lenses.

The main disadvantages of these systems are their high cost and the lack of portability, due to their large size.

Swiss Optical Group developed Automatic PD Scanner [5], a system that uses 3D technology to accurately measure the interpupillary distance. This method does not require an extra device to be positioned on the frames of the patient. Moreover, movements are allowed during the measurement and it can also be used as an entertaining device, which can play clips for promotional purposes. The manufacturer of the Automatic PD Scanner claim PD measurement errors of less than 0.25 mm, by using advanced 3D technology, but the device is expensive and has a large computational time. Due to its large size, this system can be only used for patients between 130 and 200 cm tall.

Another disadvantage of this device is that there is no way to determine whether the pupils were correctly detected. After the measurement is finalized, the customer only receives a ticket with the measured PD.

Recently, several PD measurement applications were developed for iPad devices that can precisely measure all the optometric parameters with great accuracy. This PD measurement tools are more practical, flexible and less expensive than their counterparts. Moreover, when using a tablet to perform the measurements, the near vision parameters, such as the reading distance, of the patient can also be determined: the patient can hold the device in the same position as when reading a book.

Hoya developed VisuReal portable [4], an iPad application that can accurately measure the pupillary distance. A support object is placed on top of the patients glasses and picture is take in order to perform the measurements. The support markers and the pupils are automatically detected by the software. However, for the pupil detection there is no light source projected into the patient eyes, so the pupils cannot be precisely detected. It also requires a complicated mirror system to be fitted above the camera. The image processing part and the computation are not done on the device itself but uploaded to a central server with more processing power, limiting thus the use for the cases where there is an Internet connection available.

Another portable solution for PD measurement is Activisu‘s iExpert [3], another iPad application for measuring the optometric parameters. Similar to VisuReal portable, the measurements are performed on a picture of the patients that has a support object placed on top of his glasses. This application requires an additional kit used for projecting a light source into the patient eyes, in order to produce the corneal reflex.

There are two applications on Google android market: Pupil Distance Meter [6] and Pupil Meter [7] that claim computing some of the morphological parameters. However, none of them had their results validated by a certified organism or compared to the results of an Optometry certified device. They both use a cred card to establish the photo scale, method that is inaccurate as the only case where this method could work is when the cred card is parallel to the camera plane. The relative positions of the camera plane and card plane cannot be estimated.

2. Contributions

In this paper we present the mathematical model, the architecture and implementation details for an Android tablet application that can accurately measure all of the above optometric parameters needed for *far vision* and *near vision*: the full and half *PDs*, the fitting height used for lens centering, boxing, the bridge, height and the reading distance, vertex, etc.

The optometric parameters are determined in a few quick measurement steps: a support is placed on top of the patient‘s eyeglasses and the optometrist takes a high resolution picture of the subject with the tablet. The pupil centers and the support markers are precisely detected by the application. One of the main advantages of this work is that the optometrist does not need any additional systems to complete the measurements: the only tools necessary for the process are an Android tablet and a support object. Our solution implements all image processing algorithms and computations directly on the tablet. A major contribution is the optimal implementation of all proposed computational methods and object detection algorithms so that they can be run directly on the tablet CPU. The measurement accuracy for the proposed measurements is under 0.75mm. A measurement with an estimated error under one mm is considered accurate.

For the pupil detection, we propose an original method for projecting a source light into the patient’s eyes without using an additional lighting kit: for the far vision we use the tablet’s embedded flashlight, and for the near vision we set the screen brightness to a high level while displaying a constant white pattern during the snapshot.

The remainder of this work is organized as follows: Section 3 presents the computer vision algorithms used to detected the support markers and the pupil centers; in Section 4 we discuss the main challenges we encountered while developing this application. Section 5 presents the computation of the optometric parameters. In Section 6 we describe the experimental results we performed and a comparison between the results obtained using our application and those obtained using a validated device by Essilor Instruments. Finally, in Section 7 we formulate the conclusions and the future work.

3. Eyes and support markers detection

This section presents the computer vision algorithms employed in order to detect the support markers and the pupil centers. The support is an object of known size that is placed on top of the patient eyeglasses and is used in the measurement process (section 5). This object has 3 white round markers with a small black point in the middle.

In the rest of this paper the support markers will be named as follows: support point A is the leftmost marker in screen coordinates, support point B is the rightmost, and support point C1 is the middle support point, placed between support point A and B.

To detect the support marker we employed an iterative method: the same algorithm is applied on binary images obtained by applying various threshold values to the original color image.

First, the algorithm detects all the white blobs in the binary images based on geometry constraints. Next, the detected blobs in all binary copies of the same image are analyzed and repeating blobs are merged together and their reinforcement index (number of repetitions) is computed. A score is assigned to each marker based on its brightness level and its reinforcement index. The support markers A and B are selected from the set of white blobs based on geometry constraints (the slope of the line determined by these two points and their area) and on their score (the pair with the maximum score is chosen).The next step is to select the middle support marker C1, depending on the previously detected markers. First, the ideal position for this marker is computed: the center of the segment between support points A and B.

Support marker C1 is selected from the set of detected markers based on area constrains (its area must be close to the area of support points A and B), position constraints (its relative position to the ideal determined position for this marker) and based on its score.

The last step of the support detection algorithm is to determine the black middle point of the support marker. This is performed by detecting the darkest pixel near the center of the detected blob.

Eyes are detected using the object detection algorithm proposed by Paul Viola and Michael Jones [8] in 2001, which can detect a variety of objects in real time. This method is based on four key concepts: the use of simple rectangular features, called *Haar-like features*, a new image representation: *Integral Image* for fast feature detection, the *Ada Boost* algorithm [9] and a cascaded classifier in order to filter out features efficiently. We used the algorithm and the training data (haarcascade_mcs_eyepair_big.xml) provided by the Open-CV framework.

Algorithm 1 Eye detection algorithm

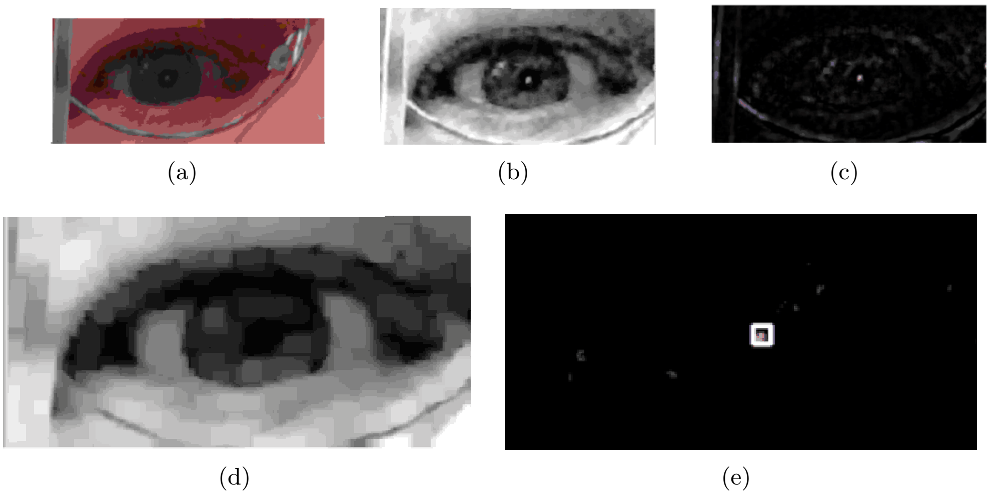
```
procedure EyesDetection(image, training_data)
    Detect all possible eye zones using Viola-Jones algorithm
    Eliminate all the regions that do not have a horizontal pair
    Split the remaining regions into left/right groups with respect to the mean center of all the regions
    Merge all overlapping regions
    Select the best matching group based on the number of eye zones contained and the dimension of the bounding rectangle
end procedure
```

Most of the methods presented in the literature for pupil detection are based on projecting an IR source of light into the patient‘s eyes, as the human pupils have a special behavior when illuminated with an IR source [11]. This method cannot be used on Android devices because the cameras have an IR filter applied. The pupil centers are determined using morphological operators and geometric constraints [12].

The Pupil detection algorithm is presented below:

Algorithm 2 Pupil detection algorithm

```
procedure PupilDetection(image, eye_ROI)
    Separate the green channel of the image
    Determine all the small white reflections: difference between the grayscale image and its morphological opening
    Threshold image using OTSU [10] method
    Compute contours and iterate: for each contour compute maximum intensity pixel and grayscale center of gravity
    Filter out false positives based on size and shape constraints
    Choose the best matching point by position constraints (relative position to the eyes Region of Interest (ROI) center) and brightness intensity
end procedure
```



(a) (b) (c) (d) (e)

Figure 1. Vision center detection steps. **(a)** Eye ROI. **(b)** Eyes ROI grayscale. **(c)** Opening. **(d)** Difference image. **(e)** Selected contour

4. Implementation challenges

This section describes the main challenges we encountered along the implementation of the solution. The measurement precision is influenced by several factors, which are hard to fulfill given the hardware specification of our device.

In the ideal case, for *Far Vision* the patient should stand in front of the device at a sufficiently large distance (around 2 m) so that he has a relaxed position and eyes looking to the infinity.

Most of the Android devices have a wide angle camera, so at this distances the captured human face that is of interest is very small, while the image captures a large area around the subject. As applying zooming lenses on a daily used tablet used is not practically feasible we changed the snapshot camera to subject distance from 2m to 60 cm. Under 60 cm the measurement is not feasible anymore as the patients eyes are converging towards *near vision* specific positions which are not usable. Even at 60 cm the captured face does not contain enough details for automatic processing.

The immediate solution we proposed is to use digital zoom, but we noticed that the quality of the captured image is strongly altered. To overcome this problem, we use the full resolution of the camera 8Mpix and we render a zoomed preview in the viewfinder, but we capture the image without any digital zoom applied. As a result, the zoomed preview does not correspond to the captured image. To address this issue, the displayed image is cropped from the original image based on the same scale transformation as the preview's zoom factor and on the layout width/height ratio such that it matches the preview at the moment of capture [14].

This approach has as additional implication: the detection algorithm is faster without having to process unnecessary information as parts of the image not containing the head are removed. All the measurements are performed on the original cropped image without any zooming applied, which allow us to obtain accurate results as markers are placed on the real image pixels data and not on pixels obtained through interpolation.

Another major difficulty is caused by the fact that both the device and the patient's head have several degrees of freedom. In the ideal case, for the measurements to be precise, the patients sagittal plane and the tablets screen plane should be perpendicular to each other. This is not achievable in practice as the subject should be left in its most natural head posture which is different from person to person. Moreover, the operator that handles the device has its own degrees of liberty in movement and natural positioning. The relative position of the patient to the tablet camera is hard to determine in the conditions where the camera orientation is not precisely known or can vary.

Any tablet/camera inclination on any of its axes, triggers an affine perspective transformation over the captured image [15] making impossible to compute and reconstruct correct distances and orientations measured on the captured human face.

In a known referential system the patients head position could be determined by using the support orientation. In order to solve this problem we use the device gyroscope and accelerometer.

The device orientation is determined as the dot product between the gravitational vector of the accelerometer and the gravitational field pointing downwards along the Z axis, as illustrated in Figure 2.

$$(1) \qquad G_p \times \left(\begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right) = G_{pz} = |G_p| \times \cos(p) = \frac{G_{pz}}{\sqrt{G_{pz}^2 + G_{py}^2 + G_{px}^2}}$$

We refer the reader to [13] for an explanation of the principle and involved variables. The accelerometer sensor data is filtered in our application (SENSOR_DELAY_NORMAL [14]) in order to eliminate noisy data that could be caused by shaking/trembling of the optometrists hands.

In addition, we apply a high pass filter over the computed acceleration values in order to filter out the noise data:

Considering the following equations, the filtering is realized for $\alpha = 0.75$,

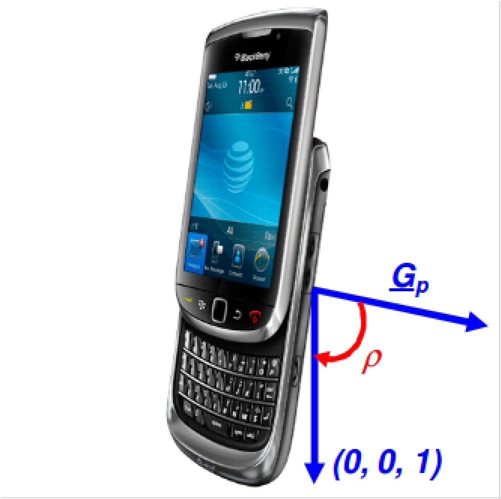


FIGURE 2. Accelerometer axis orientation, after [13]

$$\begin{aligned} x_i &= x_{i-1}\alpha + (1-\alpha)xSensor \\ y_i &= y_{i-1}\alpha + (1-\alpha)ySensor \\ z_i &= z_{i-1}\alpha + (1-\alpha)zSensor \end{aligned} \tag{2}$$

The x_i, y_i, z_i in the above equation are the current values of a series of readings from the sensor and proportionally averaged to eliminate noisy readings and to smooth the transitions.

The device tilt angle is used to accurately determine the pantoscopic angle and fitting heights by compensating the affine transformations on the image.

Another aspect to take into consideration is the precise determination of the eyes center of vision. Without dissecting the eye the only practical approach is to project a flash light into the patients eyes at the moment of the image capture and get the corneal reflex of the flash. Usually the tablets are equipped with a camera flash, but the problem is that this flash has an offset relative to the camera that produces an error margin in the position of the reflection on the image. To compensate this error, the offset is measured in mm and applied in the formulas to make the necessary corrections.

Also the near vision (reading vision) situation exhibits the same need for determining the center of vision. Tablets are not usually fitted with a flash on the front camera, but as the patient is holding the tablet closer to his eyes, by maximizing the screen brightness one can get a visible corneal reflex. Besides this, to simulate a real reading position/behavior for the eye, a red dot with some text is placed right under the camera so that the patients eyes look towards the camera in the reading position.

5. Morphological measurements

This section we highlight the contributions related to eye convergence, scaling, image rotation, pupillary distance, horizontal rotation angle of the head, vertical head inclination (pantoscopic angle).

Figure 3 represents an abstract schema of the measurement process, which takes place in the following manner: The patient (subject) stands in vertical position, the optician is placed in front of the patient at a distance of approximately 60 cm, holding the mobile device in vertical position at the subjects eye level.

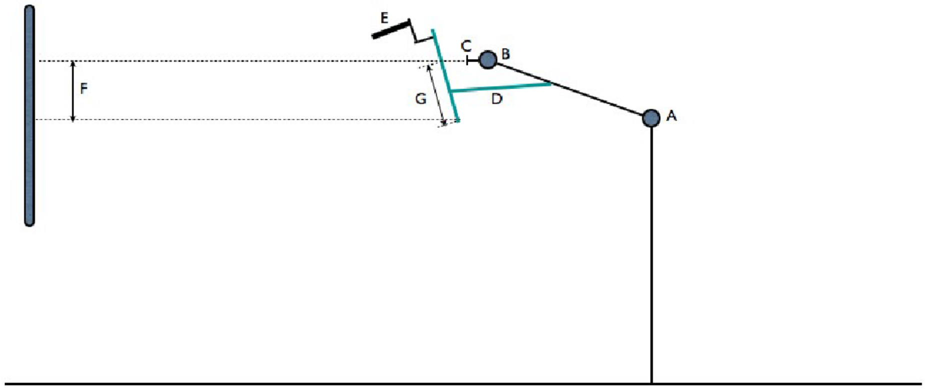


Figure 3. Model of the measurement environment (tablet and patient)

The system participating in the measurement has the following degrees of freedom in vertical plane:

- rotation around axis A: head rotation on the neck
- eyeballs rotation around their centers B (C pupils centers)
- device swivel back and forth
- device movement up and downwards
- device movement left and right (distance between device and customer may change)

The eye B is looking at device with the pupil C. The visible height F is photographed by the device camera. The goal of measurement is to find the fitting height G.

In horizontal plane, the cinematic schema is arranged similarly. The goal of measurement is to find lens centering right and left PDs and bridge.

The support (the additional object placed on top of the frames) is used as the measure of scale to convert pixels to millimeters. In order to precisely measure the optometric parameters, we also took into account that the distance in the image between the support markers (as well as the scale computations) are affected by the head orientation.

The first step in the measurement process is to rotate the image such that the line determined by the support markers A and B is parallel to the horizontal axis. In order to achieve this, the rotation angle is computed using the following formula inferred from the sine theorem:

$$\theta = \arcsin(\frac{ySupB - ySupA}{\sqrt{(xSupB - xSupA)^2} + \sqrt{(ySupB - ySupA)^2}}) \tag{3}$$

Next, the image and all the markers are rotated around the image center with this angle. For each pixel the following transformation is applied, obtaining its new position:

$$\left(\begin{array}{c} x'_{rotated} \\ y'_{rotated} \end{array} \right) = \left(\begin{array}{cc} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{array} \right) \times \left(\begin{array}{c} x' \\ y' \end{array} \right) \tag{4}$$

From now on, the rotated coordinates of the markers will be used in all the formulas.

The first parameter computed from the picture is the horizontal rotation angle of the head. For the beginning, a rough approximation of this angle is computed, followed by a refinement, to obtain a more precise value. The real value of the rotation angle of the head cannot be analytically expressed with information from a single image. It can be numerically approximated though, to a predefined error ratio.

The formula for the initial approximation is computed as follows:

$$\delta = \arctan(\frac{[(xSupB + ySupA)/2 - xSupC] * AC}{[(xSupB + ySupA)/2 - ySupC] * CC'}) \tag{5}$$

Where $xSupA$, $xSupB$ are the x coordinates for support markers A, B. CC is support cone (prominence) length. AC is the distance between support markers A and C.

Using this approximation and additional camera related parameter K, it is possible to obtain a more accurate approximation of the angle. The value of the parameter K is determined using a calibration process, where several pictures of the support are taken at certain distances.

To refine the head horizontal turning angle a sequential choice method is employed, substituting into the derived expressions horizontal head turning angle received as a result of the approximate algorithm and varying it into higher or lower direction with 0, 0005 radian increments.

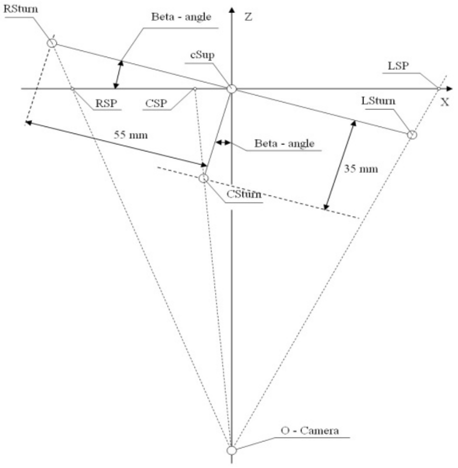


Figure 4. The coordinates of the projections on X axis relative to camera view

$$\begin{aligned} xRSP &= -\tfrac{Dist+AC*\cos\delta}{Dist+AC*\sin\delta} \\ xLSP &= -\tfrac{Dist+AC'*\cos\delta}{Dist+AC'*\sin\delta} \\ xCSP &= -\tfrac{Dist+CC'*\cos\delta}{Dist+CC'*\sin\delta} \end{aligned} \tag{6}$$

Where RSP, LSP, CSP are the projections of the marker points A, B and C on Ox axis.

The imaging distance $Dist$ (which is also named reading distance for a near distance vision measurement), is determined by the quadratic equation resolution algorithm:

$$Dist = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}, \tag{7}$$

where

$$\begin{aligned} A &= K\ times(xLSP - xRSP) \\ B &= -2 \times AC \times\ cos\alpha \\ C &= -K \times (xLSP - xRSP) \times AC^2 \times \sin\alpha \end{aligned} \tag{8}$$

And the final value for the refined head horizontal rotation angle:

$$\delta` = \frac{AC \times \cos\delta}{Dist - AC \times \sin\delta} + \frac{CC` \times \cos\delta \times \sin\delta}{Dist - CC` \times (\cos\delta)^2} - K \times (xLSP - xCSP) \tag{9}$$

Finally, when the patient's head is inclined backward or forward, the fitting heights need to be corrected in order to obtain an accurate value.

To do this, the pantoscopic angle (the head vertical inclination) is determined using the following formula:

$$y = \Delta + DeviceAngle + CameraVA \tag{10}$$

Where Δ is the angle between the frontal plane and the camera plane determined using the vector rotation rules of the coordinates of the center support point(CS1, CS2), by the formula:

$$\Delta = \arctan(\frac{yCS2 \times xCS1 - xCS1 \times yCS1}{xCS2 \times xCS1 + yCS2 \times yCS1}) \tag{11}$$

The angle of the device ($DeviceAngle$) is computed using the information from the sensors of the device (accelerometer) (as presented in Section 4).

$CameraVA$ represents the device camera viewing angle, i.e. the angle between the camera and the device. For most of the devices this angle would be equal to zero, but there are some cases when this angle is nonzero.

The pupillary distance is computed as shown in Equation 12.

$$\begin{aligned} LeftPD &= xLeftEye - xCenterSup \\ RightPD &= xCenterSup - xRightEye, \end{aligned} \tag{12}$$

where $xLeftEye$ is the x coordinate of the left eye, $xRightEye$ is the x coordinate of the right eye, $xCenterSup$ is the x coordinate of the center support marker.

In the ideal case, for far distance vision, the eyes must be looking in parallel to infinity. However, in our case, the patient looks at the Android tablet from a finite distance and, as a consequence, it is necessary to apply some corrections to the measured values of the pupillary distances, taking into account the computed angles of rotation of the head, the angle of the device and the angle of rotation of the image.

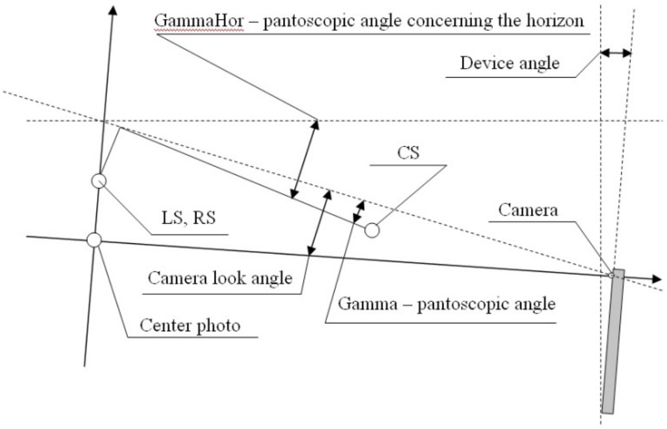

GammaHor – pantoscopic angle concerning the horizon
Device angle
CS
LS, RS
Camera
Camera look angle
Gamma – pantoscopic angle
Center photo

Figure 5. Pantoscopic angle relative to horizon

$$LeftPD' = \frac{LeftPD}{\cos\alpha} \qquad RightPD' = \frac{RightPD}{\cos\alpha} \tag{13}$$

These formulas are applied successively for each angle that influences the measured value.

However, this is a rough approximation of the pupillary distance. A more realistic correction uses the vertex distance and eyeball size. But since the exact values of Vertex Distance and eyeball size are unknown at this point, the following assumptions are used, which represent the average values for a human eye:$VertexDistance$ = 14 mm and $EyeRadius$ = 12,5 mm.

Then a new refinement of the final value for the pupillary distance can be applied after the previous one, using the formulas:

$$\begin{aligned} &adjust = (VertexDistance + EyeRadius) \times \tan\alpha \\ &LeftPD' = LeftPD + adjust \\ &RightPD' = RightPD + adjust \end{aligned} \tag{14}$$

6. Experimental results

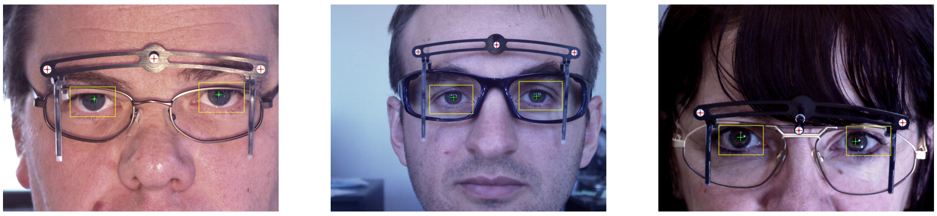
In this section we present the experimental results of our feature detection algorithms and of the measurement process.

We tested our detection algorithms on several databases, with over 2000 facial images. The images were captured in real life conditions (in optician stores) by an ACEP Smart Mirror System [2]. In table 1 we present the detection rates of the algorithms on a set of 100 facial images.

	Detected	**False positives**	**Not detected**
Eye detection	98%	0%	2%
Pupil detection	90%	8%	2%
Support detection	95%	3%	2%

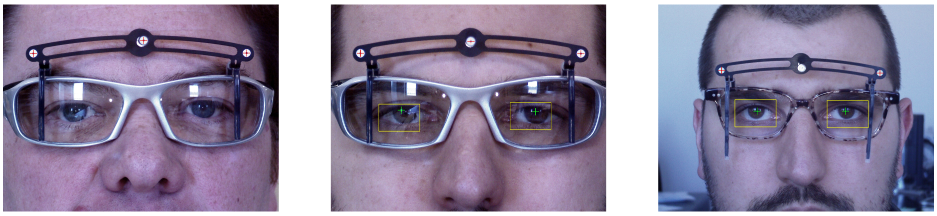
Table 1. Algorithms detection rate

Figure 6 shows the results of pupil and support markers detection, and Figure 7 presents some failure cases.



(a) (b) (c)

Figure 6. Pupil and markers detection results in various conditions.



(a) (b) (c)

Figure 7. Some failure cases. (**a**) Failure to detect the eyes; (**b**)Failure to detect the pupils; (**c**) Failure to detect the support markers

The support detection rate is very high: 98%. However, there were 2 cases where support point C1 (the middle support marker) was not detected; because a source of light is projected into the patient's eyes in order to detect the pupil centers, the area surrounding this support marker is overexposed, so the white blob cannot be detected.

The eye region was not detected in two images, where the lenses of the eyeglasses contained multiple reflections. The pupil detection doesn't yield accurate results when there are multiple reflections on the surface of the eyeglasses lens. This is because we perform pupil detection based only on edge information and geometrical constraints.

Far Vision	Measurement 1			Measurement 2			Measurement 3			Measurement 4		
	Android	Ipad	Error	Android	Ipad	Error	Android	Ipad	Error	Android	Ipad	Error
left PD	31.76	31.7	0.06	32.96	32.6	0.36	32.24	30.6	0.64	32.78	32.5	0.28
right PD	33.74	32.6	1.14	34.26	34	0.26	34.08	33.3	0.78	35.01	34.2	0.81
left height	23.44	24.3	0.86	20.83	21.5	0.67	20.95	21.7	0.75	20.9	22.4	1.5
right height	22.91	24.3	1.39	20.19	20.8	0.61	21.76	22.5	0.74	20.68	21.8	1.12
lens width	46.73	49.6	2.87	46.68	47	0.32	47.43	47.3	0.13	46.45	46.5	0.05
lens height	36.99	36.3	0.69	35.5	35.5	0	37.79	36.8	0.99	35.75	35.4	0.35
bridge	18.36	17.8	0.56	18.55	17.8	0.75	18.28	18	0.28	18.5	18.1	0.4
pantoscopic angle	-6.84	-3.3	3.54	-3.67	-0.4	3.27	-11.23	-7.9	3.33	-0.87	3.9	4.77

Table 2. Measurement comparison for Far Vision

Next, we present the results of our measurement process. We compared our results with a validated iPad measurement application from ACEP [2].

For the testing procedure, we asked the subject to stay still while two alternative pictures were taken: one with the iPad and one with an Android tablet. The person taking the picture maintained the same relative position to the subject.

We tested our system on more than 30 measurements of people wearing different types of glasses. Next we present the four most different results we obtained after the measurement process. Table 2 shows the measurements comparisons for *far vision* and table 3 shows the measurements comparisons for *near vision.*

Figure 8 and Figure 9 graphically show the variations of the half interpupillary distances, measured with an Android tablet and an iPad device.

Near Vision	Measurement 1			Measurement 2			Measurement 3			Measurement 4		
	Android	Ipad	Error	Android	Ipad	Error	Android	Ipad	Error	Android	Ipad	Error
left PD	29.57	29.5	0.07	29.28	29.3	0.02	28.01	29.3	1.29	30.15	29.4	0.75
right PD	30.75	30	0.75	31.71	31	0.71	30.89	30.2	0.69	31.1	30.7	0.4
reading distance	29.29	26.5	2.79	28.47	26.4	2.07	27.23	26.6	0.63	30.12	26.6	3.52
left height	13.94	15.8	1.86	11.38	8.9	2.48	15.33	14.7	0.63	10.98	10.1	0.88
right height	13.73	15.3	1.57	10.25	8.3	1.95	15.84	15.2	0.64	10.77	9.9	0.87
bridge	18.39	18.4	0.01	17.77	17.9	0.13	18.06	18.5	0.44	17.77	18.1	0.33

Table 3. Measurement comparison for Near Vision

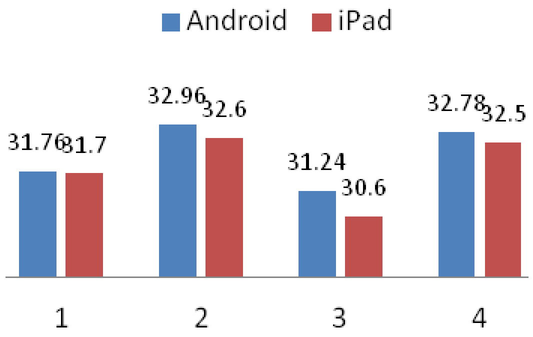


Figure 8. Variation (mm) of the left PD in different measurements between the validated device and our software

7. Conclusions and future work

In this paper we proposed an original solution for one of the current problems in modern optometry/ophthalmology. The precise measurement of morphological spectacles properties, human face morphological characteristics and their correlation is achieved in our case by implementing a pervasive optometric application.

In the proposed solution we use high definition images captured with tablet cameras in order to capture patients most natural position when relaxed (FV) or reading (NV). The images are processed with our detection algorithms that are able to run directly on the tablets limited CPU.

Our computation methods are robust enough to compensate the liberty degrees of movement of both operator (tablet) and patient. The patients' eye convergence in FV is compensated so that the outcome result is as close to

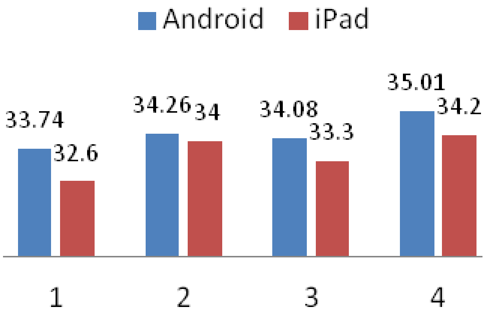


Figure 9. Variation (mm) of the right PD in different measurements between the validated device and our software

as possible to a *gaze to infinity position*. There is no need for any additional system in order to achieve a full *centration* process.

We developed robust marker and eye detection methods with high detection ratios even for images in varying lighting conditions. The measurement results variation, compared with results of existing large scale (fixed) measurement devices, is less than 1 mm.

As a future work we plan to develop a more robust algorithm for pupil detection, which does not yield false positives when there are multiple reflections on the surface of the eyeglasses‘ lens. Another plan is to use two facial images of the patient in order to accurately determine depth related morphological parameters, such as the vertex distance and the wrap angle.

References

[1] Clifford, W. Brooks, *Understanding Lens Surfacing*; Butterworth-Heinemann: United Kingdom, Oxford, 1991.

[2] ACEP, http://www.smart-mirror.com/en/produits.php?id=4. [Online] [Seen on: 20/05/2013.]

[3] ACTIVISU - iExpert, http://www.activisu.com/. [Online] [Seen on: 20/05/2013.]

[4] HOYA - Visu Real, http://www.hoya.eu/index.php?SID=519b2d521a3f7818638341&page_id=20173. [Online] [Seen on: 20/05/2013.]

[5] SWISS Optical Group - Automatic 3D PD Scanner, http://www.swissopticalgroup.com/instore_products/automatic_3D_pd_scanner. [Online] [Seen on: 20/05/2013.]

[6] Vistech Projects, Pupil Distance Meter, https://play.google.com/store/apps/details?id=com.vistechprojects.pupildistancemeter&hl=en. [Online] [Seen on 20/10/2013].

[7] Rob Dewhurst, Pupil Meter, https://play.google.com/store/apps/details?id=air.PupilMeterAnd&hl=en [Online] [Seen on 20/10/2013].

[8] Viola, P.; Jones, M. Robust Real-time Object Detection. In Proceedings of the Second International Workshop on Statistical and Computational Theories of Vision—Modeling, Learning, Computing, and Sampling, Vancouver, Canada, 13 July 2001.

[9] Freund, Y.; Schapire, R.E. A Decision-theoretic Generalization of On-line Learning and an Application to Boosting. In Proceedings of the Second European Conference on Computational Learning Theory (EuroCOLT '95), Barcelona, Spain, 13-15 March 1995; pp. 23–37.

[10] Otsu, N. A threshold selection method from gray-level histograms. IEEE Transactions on System, Man and Cybernetics, 1979, Vol. 9.

[11] Ebisawa, Yoshinobu. Unconstrained pupil detection technique using two light sources and the image difference method. Available from: http://www.sys.eng.shizuoka.ac.jp/ ebisawa/Seika/Videa95/PROC.PDF

[12] Haw-Jye SHyu, RogeR HillSon. Using Morphological Filters for Pupil Detection in Infrared Videos Filters. 2010. NRL/MR/5508–10-9233.

[13] Pedley, Mark. Tilt Sensing Using a Three-Axis . 2013. AN3461.

[14] Google. Android Developer - Sensor Manager [Online] [Seen on: 03/06/2013.] http://developer.android.com/reference/android/hardware/SensorManager.html# SENSOR_DELAY_FASTEST.

[15] P.J. Schneider, D.H. Eberly. Geometric Tools for Computer Graphics. s.l.: Morgan Kaufmann Publishers, ISBN: 1-55860-5940, 2003.

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Cluj-Napoca, Romania

E-mail address: `dadi@cs.ubbcluj.ro`

Dept. of Automation and Computer Science Technical University of Cluj-Napoca, Romania

E-mail address: `ispasmihaiclaudiu@yahoo.com`

E-mail address: `diana.borza@tvarita.ro`