Pattern Recognition 52 (2016) 160–173

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A realistic virtual environment for evaluating face analysis systems under dynamic conditions

Mauricio Correa [a], Javier Ruiz-del-Solar [a,b,*], Rodrigo Verschae [a,1]

[a] Advanced Mining Technology Center, Universidad de Chile, Chile
[b] Department of Electrical Engineering, Universidad de Chile, Av. Tupper 2007, 837-0451 Santiago, Chile

ARTICLE INFO

Article history:
Received 14 November 2014
Received in revised form
7 September 2015
Accepted 12 November 2015
Available online 22 November 2015

Keywords:
Face analysis
Face recognition
Face
recognition benchmark
Evaluation methodologies
Virtual simulation environment
Simulator

ABSTRACT

This paper proposes a new tool for the evaluation of face analysis systems under dynamic experimental conditions. The tool primarily consists of a virtual environment where a virtual agent (e.g., a simulated robot) carries out a face analysis process (e.g. face detection and recognition). This virtual agent can navigate in the virtual environment, where one or more subjects are present, and it can observe the subjects' faces from different distances and angles (yaw, pitch, and roll), and under different illumination conditions (indoor or outdoor). The current view of the agent, i.e. the image that the agent observes, is generated by composing real face and background images acquired prior to their usage in the virtual environment. In the virtual environment, different kinds of agents and agents' trajectories can be simulated, such as an agent navigating in a scene with people looking in different directions (mimicking a home-like environment), an agent performing a circular scanning (such as in a security checkpoint), or a camera-based surveillance system observing a person. In addition, during the recognition process the agent can actively change its viewpoint seeking to improve the recognition results. The proposed tool provides to the developer all functionalities needed to build the evaluation scenario: a set of real face images with real background information, a virtual agent with navigation capabilities, a scenario configuration (number, position and pose of the subjects to be observed), an agent trajectory definition, the generation of the simulated agent's view-dependent images, some basic active vision mechanisms, and the ground truth data (e.g. face id and pose for every observation), allowing the evaluation of face analysis methods under realistic conditions. Three usage examples are presented: the study of the robustness of face detection and face recognition methods under pose variations, and the evaluation of an integrated face analysis system to be used by a service robot. The proposed methodology may be of interest for researchers and developers of face analysis methods, in particular in the robotic and biometrics communities.

1. Introduction

Face analysis[2] plays an important role in building computer vision systems, HRI (Human–Robot Interaction) systems, and in general in any system that uses vision to interact naturally with humans or to process information of humans in a given scene. Human detection and human identification based on face information are key abilities of intelligent machines whose purpose is to interact with humans. Face analysis is also very important in surveillance applications in dynamic environment, such as security cameras at airports, and is also being included in consumer electronics, such as face detection and smile detection in cameras. Evaluating face analysis systems for such environments and conditions is not straightforward, in particular, in the cases where the recognition system uses active vision mechanisms to change its viewpoint or position in the scene.

A very important aspect in the development of face analysis methodologies is the use of suitable databases, and reproducible testing and training methodologies. For instance, the well-known FERET database [1], has been very important in the development of face recognition algorithms for controlled environments. However, neither FERET nor other relatively new databases such as LFW [2], CAS-PEAL [3] and FRGC [4,5], among others [6–8], are

* Corresponding author at: Department of Electrical Engineering, Universidad de Chile, Av. Tupper 2007, 837-0451 Santiago, Chile. Tel.: +56 2 2977 1000/ +56 2 2978 4207; fax: +56 2 6720162.
E-mail addresses: macorrea@ing.uchile.cl (M. Correa), jruizd@ing.uchile.cl (J. Ruiz-del-Solar), rodrigo@verschae.org (R. Verschae).

[1] Rodrigo Verschae is now with the Graduate School of Informatics, Kyoto University, Japan.

[2] Face analysis is understood as any computational procedure related to the analysis of faces, with the most common procedures being detection, recognition, alignment and expression recognition, but many other exist, such as super-resolution, gender classification, smile detection, pose estimation, blink detection, kindness recognition, etc.

http://dx.doi.org/10.1016/j.patcog.2015.11.008
0031-3203/

able to provide real-world testing conditions for evaluating face recognition systems that include the use of innovative mechanisms such as spatiotemporal context and active vision, which are required in applications that consider the dynamic interaction with humans in the real world. Even the use of video face databases (e.g. [9–12]) does not allow testing the use of those ideas, because the video sequence is taken using pre-defined viewpoints. The use of a virtual face simulator could allow accomplishing the changes in viewpoints. However, such a simulator would not be able to generate faces and backgrounds that look real/natural enough, which is an important requirement for the realistic testing of face recognition systems.

Nevertheless, the combined used of a simulation tool with real face images and background images taken under real-world conditions could allow accomplishing the goal of providing a tool for testing face recognition systems under uncontrolled, dynamic conditions. In this case, more than providing a database and a testing procedure, the idea would be to supply a virtual environment that offers a database of real face images and real background images, a simulated virtual environment, a virtual agent moving in that environment, active vision mechanisms for the virtual agent, predefined benchmark problems, ground truth data, and an evaluation methodology.

The main goal of this paper is to provide such a virtual environment. In this environment, virtual subjects are located at different positions and with different orientations in a virtual map. Inside the virtual environment, a virtual agent (a virtual entity with the ability to detect, recognize and analyze faces) can navigate and observe face images from different distances and angles (yaw, pitch, and roll). The current view of the agent, i.e. the image that the agent observes, is generated by the virtual environment using real face images previously acquired in indoor and outdoor variable lighting conditions with several pitch and yaw angles (in-plane rotations can be simulated by software), as well as real background images. In the virtual environment, different kinds of agents and agents' trajectories can be simulated, such as an agent navigating in a scene with people looking in different directions (mimicking a home-like environment), an agent performing a circular scanning (such as in a security checkpoint), or a camera-based surveillance system observing a person. In addition, during the recognition process, the virtual agent can actively change its viewpoint seeking to improve the recognition results.

We believe that the proposed methodology and evaluation tool are of interest to researchers involved in development and testing of applications related with the visual analysis of human faces. Its use allows comparing, quantifying and validating face analysis capabilities of agents, and in general intelligent machines, under dynamic working conditions. One of its more relevant features is that it allows repeatability of the experiments. Therefore, it allows the comparison and evaluation of one or more algorithms without damaging the moving agent (e.g. the robot), and with short evaluation times. In the current work we focus on face recognition and detection, although the use of the tool is straightforward in other face analysis problems, such as pose estimation, gender classification, and age estimation.

It is worth mentioning that a special acquisition device was designed and built to acquire face and background images under different view angles, which are essential for the operation of the virtual environment. The simplicity and modularity of the device allows its rapid deployment and use in real-world locations such as streets, gardens, shopping malls, etc.

This article is organized as follows. First, related work on existing face analysis and evaluation methodologies is outlined (Section 2). Afterwards we describe the proposed virtual environment (Section 3), where we give a detailed description of its different modules. Later, we present some usage examples of the proposed system (Section 4), to finally conclude (Section 5).

2. Related work

The availability of standard databases, benchmarks, and evaluation methodologies is crucial for the appropriate development and comparison of face analysis systems. There is a large number of face databases and associated evaluation methodologies that consider different number of subjects, camera sensors, and image acquisition conditions, and that are suited to test different aspects of the face recognition problem such us illumination invariance, aging, expression invariance, etc. (e.g. the surveys and comparatives studies [13–17,46]). An overview and basic information about existing face databases can be found in [7,18]. Although some new databases (e.g. LFW database [2] and Photoface database [19]) are designed to include real-world images, most databases and evaluation protocols (including LFW database [2] and Photoface database [19]) are designed to test methods using images captured by static cameras. Also, similar methodologies are commonly used in face recognition infrared images [20,21].

Out of the existing databases for face recognition, probably the most well known is the FERET database [1] and its associated evaluation methodology, which has become the standard choice for evaluating face recognition algorithms under controlled conditions. Alternative popular databases used with the same purpose are the Yale Face Database [22] and BioID [23]. Other databases, such the AR Face Database [24], ORL database [47] and the University of Notre Dame Biometrics Database [25], include faces with different facial expressions, illumination conditions, and occlusions. However, from our point of view, all of them are far from considering real-world conditions.

The Yale Face Database B [26] and PIE [27] are the most utilized databases to test the performance of algorithms under variable illumination conditions. The Yale Face database contains 5 760 single light source images of 10 subjects, each seen under 576 viewing conditions (9 poses $\times$ 64 illumination conditions). For every subject in a particular pose, an image with ambient (background) illumination was also captured. PIE is a database containing 41,368 images of 68 people, each person under 13 different poses, 43 different illumination conditions, and with 4 different expressions. Both databases consider only indoor illumination.

The LFW database [2] consists of 13,233 face images of 5749 different subjects, obtained from news images by means of a face detector. There are no eyes/fiducial point annotations; the faces were just aligned using the output of the face detector. The images have a very large degree of variability in the face's pose, expression, age, race, and background. However, given that the LFW images are obtained from news, which in general are taken by professional photographers, the images are obtained under good illumination conditions, and mostly in indoors.

FRGC ver2.0 database [5] consists of 50,000 face images divided into training and validation sets of controlled and uncontrolled images. The uncontrolled images were taken under varying illumination conditions in indoors and outdoors. Each set of uncontrolled images contains two expressions, smiling and neutral.

The Photoface database [19] is a database of 3D faces, which consist of 3187 sessions of 453 subjects, captured in two recording periods of approximately six months each. The Photoface device was located in an unsupervised corridor allowing real-world and unconstrained capture. Each session comprises four differently lit colour photographs of the subject, from which surface normal and albedo estimations can be calculated. This allows for many testing scenarios and data fusion modalities. Eleven facial landmarks have been manually located on each session for alignment purposes. Additionally, metadata such as gender, facial hair, pose and expression is available.

The EURECOM Kinect Face Dataset [28] consists of multimodal facial images of 52 people (14 females, 38 males) acquired with a Kinect sensor. In each session images are collected according to different facial expressions, lighting and occlusion conditions: neutral,

smile, open mouth, left profile, right profile, occluded eyes, occluded mouth, side occlusion with a sheet of paper and light on. An RGB color image, a depth map, as well as the associated 3D data are provided for all samples. The dataset includes 6 manually labeled landmark positions for every face and information, such as gender, year of birth, ethnicity, presence of glasses, and the time of each session.Some databases, and the corresponding methodologies, have focus on learning issues, such as incremental learning [29], active learning [30], and weakly labeled data [31].

There are also many video face databases that have been proposed in the literature [9,10,12,32–36]. Although these databases allow evaluating non-statics scenarios (e.g. with moving subjects), they are still restrictive and can only be used on very specific scenarios that do not allow considering active vision mechanisms. It is important to stress that none of the mentioned databases allows the evaluation of face analysis systems under fully dynamic conditions.

3. Realistic virtual environment

The evaluation of any method to be used in a real-world system must be done in conditions as close as possible to the ones observed in a real scenario. In this section we describe in detail the proposed virtual environment for evaluating face analysis systems. The tool was developed to evaluate, in a simulated environment, face recognition and detection systems that later will be used by a moving agent, such as a service robot or a pan-till-zoom camera. The moving agent is a virtual entity that can move within the virtual environment and sense it. While moving in the virtual environment, the agent has the ability to detect, recognize and analyze faces. The main advantage of using a simulator is that it allows repeatability of the experiments. Therefore it allows the comparison and evaluation of one or more algorithms without damaging the moving agent (e.g. the robot), and with short evaluation times. The tool can be used to evaluate any existing face analysis system, even if it not meant to be used in robotic applications.

The proposed tool allows testing face analysis systems in uncontrolled conditions (pose, illumination, expression, etc.). More specifically, within a virtual environment, a virtual agent can move and observe images generated by the simulator according to the current agent's position. The observed images can be used by the agent for tasks such as face detection and face recognition. Given that the system allows the agent to navigate in the environment, the agent can observe the faces from various viewpoints (distances and angles). This can allow the agent to improve its recognition performance during the face analysis process, because the agent can actively change its position in the environment.

Two key features of this tool are that: (i) the face images observed by the agent are generated using real face and background images obtained in real conditions, and (ii) one or more environments can be generated, for example considering varying the location of the subjects in the virtual environment's location map.

The tool consists of three main modules (a block diagram of the evaluation tool is presented in Fig. 1):

- *Image Generation*, which generates the realistic images to be observed by the agent,
- *Agent Vision*, which processes and analyzes the generated images, and
- *Agent Navigation and Positioning*, which moves the virtual agent inside the virtual environment.

At every given time, the state of the virtual environment is defined by the pose of the agent and the pose of N subjects in the virtual environment (represented using the *Global Map*). The virtual agent can navigate and make observations inside this virtual scenario.

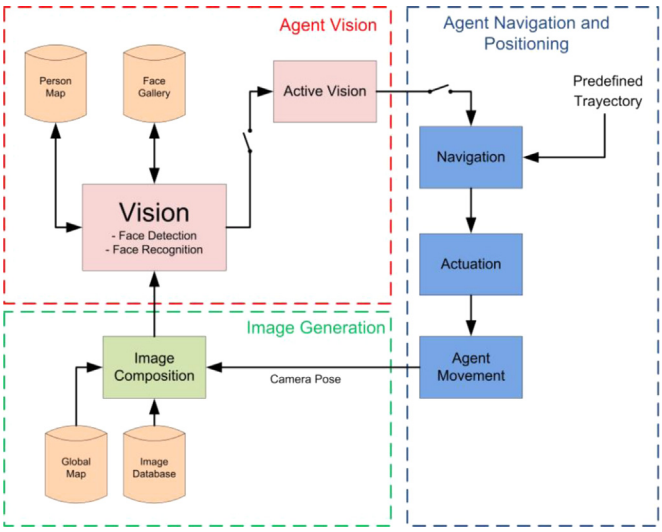

Fig. 1. Diagram of the evaluation tool.

The simulator generates images containing faces seen from different distances and angles, considering different illumination conditions (indoors or outdoors). For this, it uses images that are captured and stored in the *Image Database* in an offline process. During the navigation process, the agent can move inside the map to change its relative viewpoint and distance to the subject using active vision in order to modify its observations, seeking to improve its face recognition results. The *Image Generation* module composes the image observed by the agent by taking the relative positions of the subjects and the agent within the map. The virtual environment offers all the functions that the agent could have in a real scenario: navigation, positioning, and visual sensing by generating the images observed by the agent at a given time according to its current pose.

3.1. Image Generation

In the *Image Generation* module, real face and background images are used to compose the scene as observed by the virtual agent (real face and background images are acquired using the device described in the Section 3.1.1). Every time the agent changes its pose, the simulator generates the corresponding new image as requested by the vision module. For instance, Fig. 2 shows a given set of agent poses, and the corresponding images generated by the simulator.

The *Image Generation* module reads the position and angular pose of the agent (X_A, Y_A, θ_A) from the *Global Map*, with (X_A, Y_A) the position and θ_A the orientation of the agent, as well as the list of positions and angular poses of the subjects in the virtual scene, and then composes an observation (image) for the virtual agent. The images stored in the *Image Database* are used to compose the observed images. The *Image Database* contains two types of images: face images with different out-of-plane rotations and background images. In-plane rotations are generated by the simulator.

3.1.1. Data acquisition and database construction

Real face images of each subject, as well as background images of the same location, are acquired under several yaw and pitch angles using a custom designed acquisition device, which uses a CCD camera mounted in a rotating structure (see Fig. 3(b)). During the acquisition process, the person being scanned is in a still position, while the camera, placed at the same height as the person's face (the camera height is adjustable) and at a fixed distance of 140 cm from the person, rotates in the axial plane.

The acquisition device is manually moved and an encoder placed in the rotation axis calculates the face's yaw angle. The

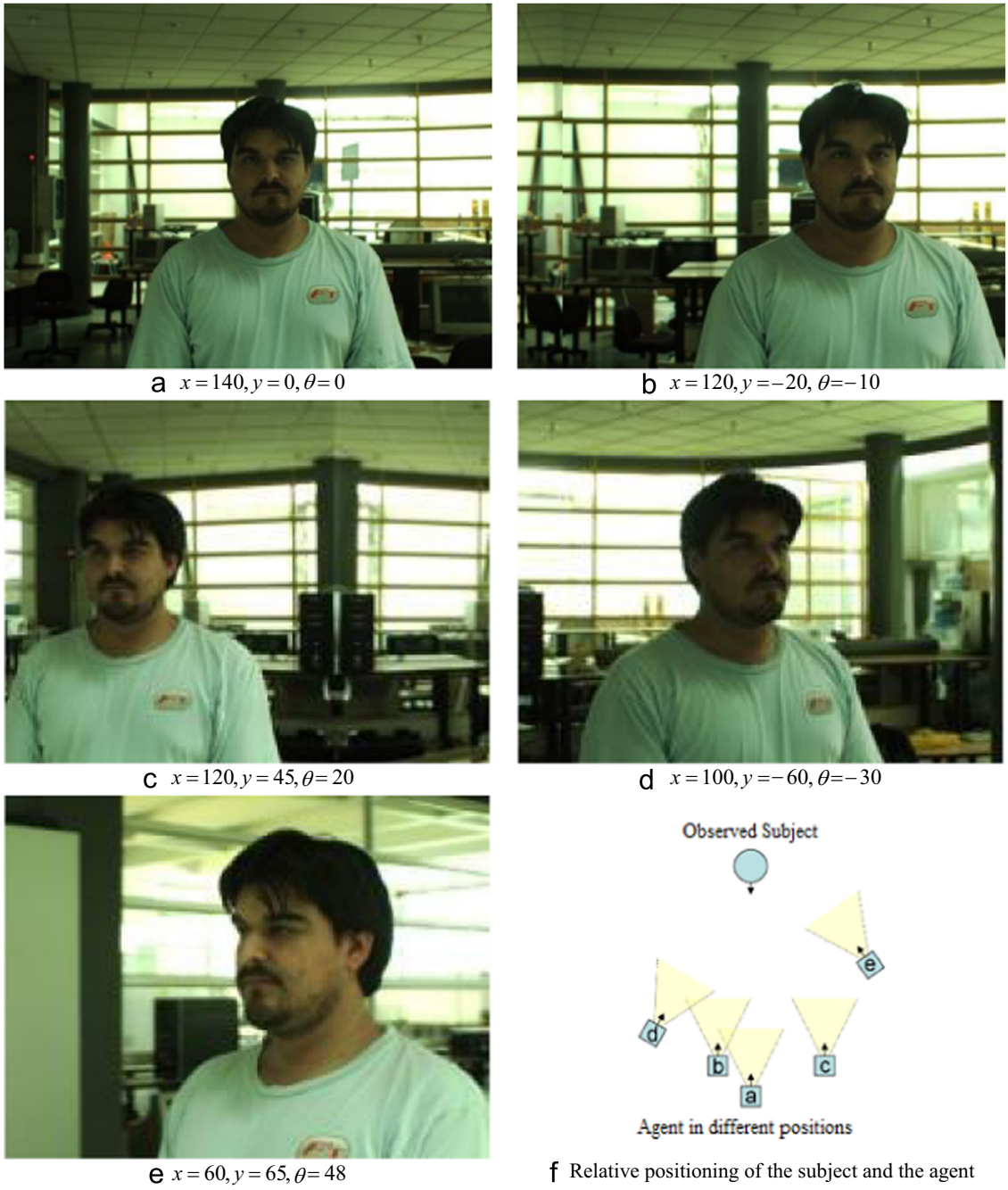

Fig. 2. Example of the agent's positioning and the image generated by the simulator. The relative position of the subject and the agent is shown in (f). The agent is located in five positions (a, b, c, d e), and the corresponding generated images are shown in figures (b)–(e). The arrows in (f) indicate pose, while the x, y and θ values indicate relative translation and rotation (yaw).

system is able to acquire images with a 1° resolution. The scanning process takes 25 s, and we use a 1280×960 pixels CCD camera (DFK 41BU02 model). In a frontal face image, the face's size is about 200×250 pixels. The acquisition device is portable (it does not require any special installation), and therefore it can be used at different places. Thus, the complete acquisition process can be carried out at different locations (streets, laboratory environment, shopping environment, etc.).

Variations in pitch are obtained by repeating the described process with the different pitch angles. In each case, the camera height is maintained, but the person looks at a different reference points located at 160 cm in front of the person at different heights in the vertical axis (see Fig. 3(a)).

For the experiments reported later in Section 4, a database that consists of face images of 50 subjects, captured in indoors (laboratory with windows) and outdoors (in the university campus), was built. During the acquisition process, there were no restrictions on the person's facial expression. For each person, 726 registered face images ($121 \times 3 \times 2$) were acquired and stored. The yaw angle range was [120°, 120°], with a resolution of 2°, which gives 121 images. For each different yaw angle, 3 different pitch angles were considered ($-15°$, 0°, and 15°). In our experience this is enough to represent typical human face variations. For each yaw–pitch combination, one indoor and one outdoor image was taken. Background images (without any subject) for each location, camera-height, and yaw–pitch angle combination are captured with the acquisition device. Using these images the simulator will generate, later on, the

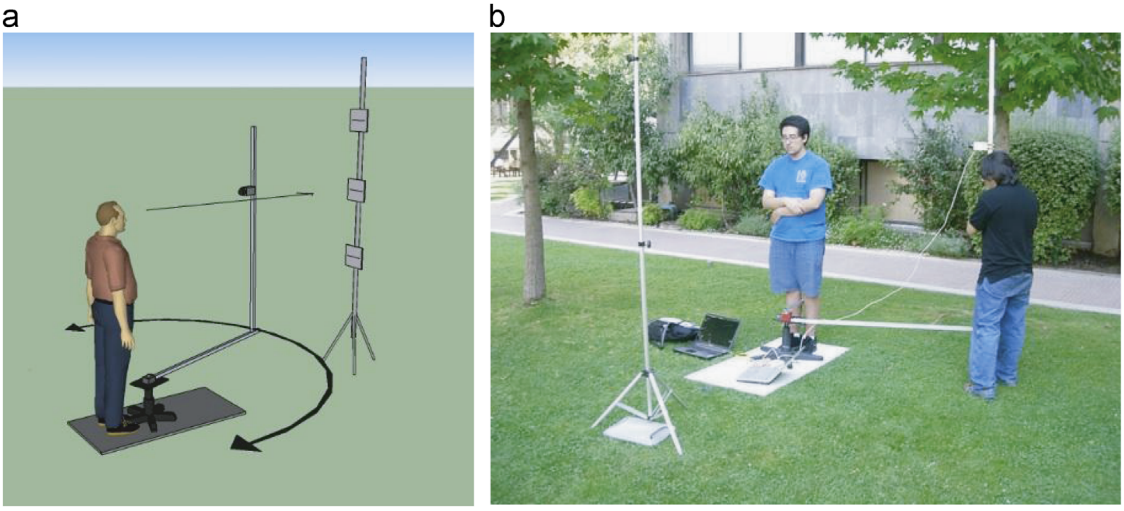

Fig. 3. (a) Diagram of the image acquisition system. (b) The system operating in outdoors.

Fig. 4. Example of images taken using the device in indoors (first row) and outdoors (second row).

images to be shown to the agent. In Fig. 4, some images taken with the described acquisition device are shown. This database will be made available for academic research purposes (upon request at http://vision.die.uchile.cl/databases.php).

3.1.2. Image composition

The image composition process consists basically of 4 steps, as illustrated in Fig. 5: first, the closest subject in the field of view is select; second, the image corresponding to the relative pose between the agent and the subject is selected from the database; third, additional background information is added to the image, and finally the image is rescaled, translated and cropped to obtain the composed image. In addition, it is possible to add an occluding object (see Fig. 6). The details of the complete process are presented in the following.

In order to generate the image observed by the virtual agent, the simulator first estimates which subject, if any, is in the field of view of the agent. If more than one subject is in the field of view of the agent, the closest one within its field of view is selected.

First, the simulator calculates the relative pose from the agent to every subject located in the *Global Map*. Then, the closest subject within the field of view of the agent is selected. Let us define

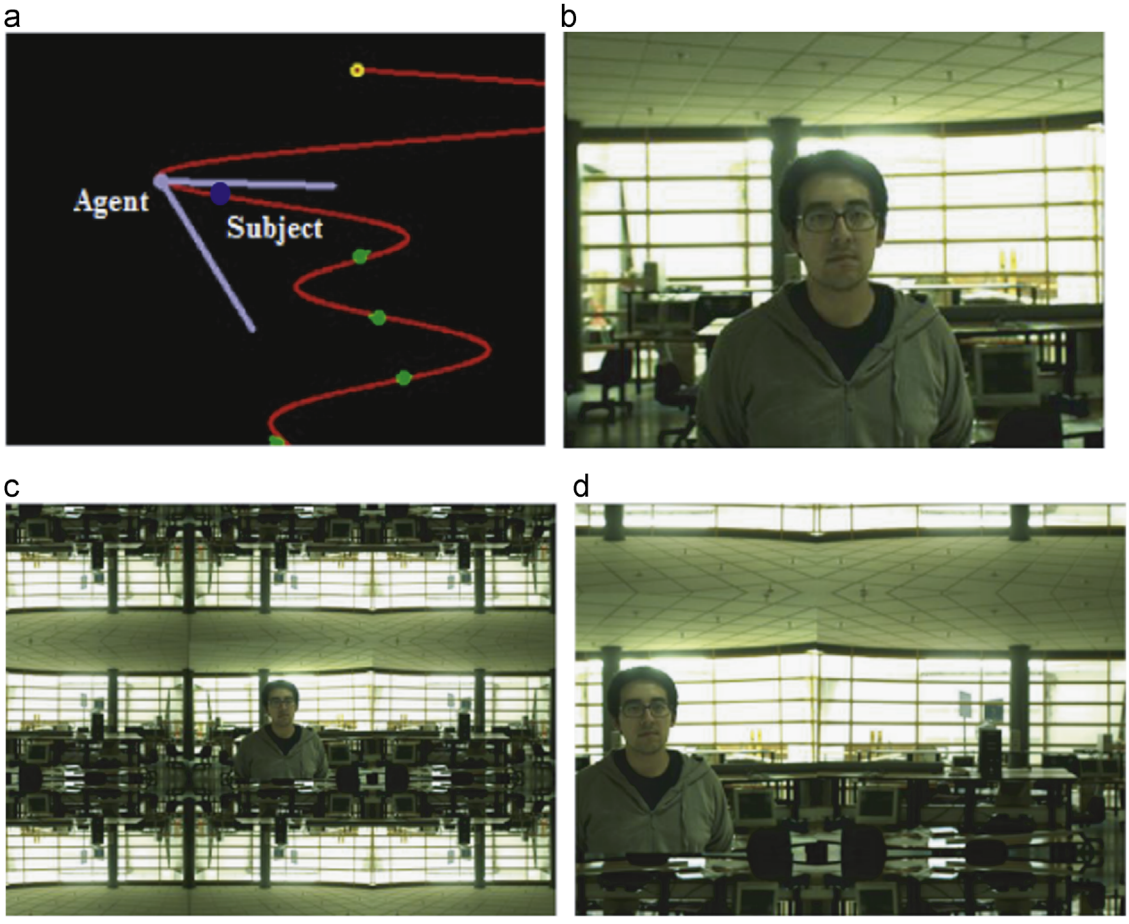

Fig. 5. Example of the image generation process. (a) Position of the agent (magenta) and the observed subject (blue) in the global map. (b) Image for the corresponding relative pose (retrieved form the DB). (c) Image with added background information. (d) Composed image after rescaling and translating (image observed by the agent). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

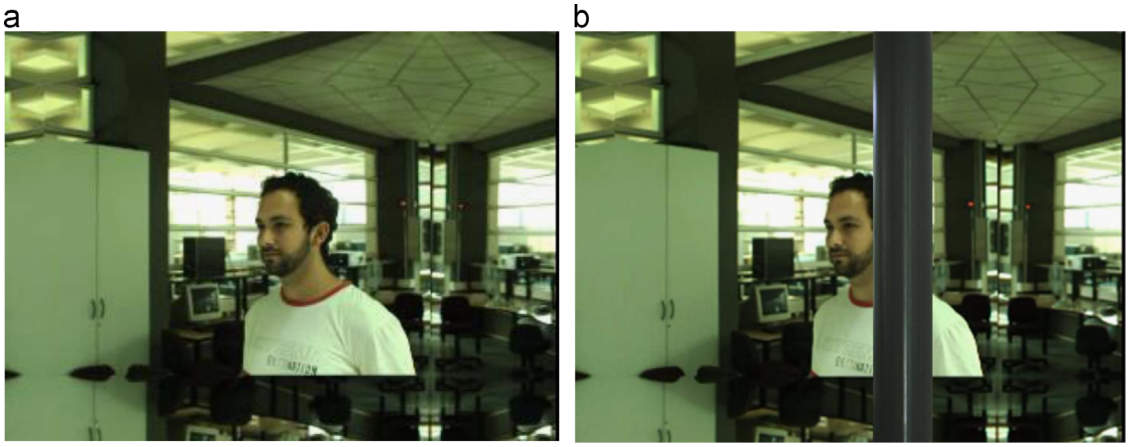

Fig. 6. Occlusion example. (a) Generated image without occlusion and (b) generated image with occlusion.

(X_i, Y_i, θ_i) to be the pose of subject i in the virtual map, with $i \in \{1, \ldots, N\}$, and (X_A, Y_A, θ_A) the current pose of the agent in the map. Then, the angle of the subject i with respect to a reference axis X fixed to the agent, φ_i^x, and the distance between the agent and the subject i, d_i, are calculated as:

$$\varphi_i^x = 180^\circ + \tan^{-1}\left(\frac{X_i - X_A}{-Y_i + Y_A}\right) - \theta_A \tag{1}$$

$$d_i = \sqrt{(X_i - X_A)^2 + (Y_i - Y_A)^2} \tag{2}$$

The closest subject satisfying the field of view constraints is obtained as:

$$\begin{aligned} & i^* = \text{argmin}_{i \in \{1,\ldots N\}}(d_i) \\ & s.t. \quad d_i < d_{max} \\ & |\varphi_i^x| < \varphi_{max}^x, \end{aligned} \tag{3}$$

with d_{max} and φ^{x}_{max} the linear and angular parameters that define the field of view of the agent.

If there is no subject who satisfies the constraints, an image containing no faces is generated using background images. If a subject is found, the yaw angle of the subject relative to the agent is calculated as follows:

$$\theta^{*}_{yaw} = \tan^{-1}\left(\frac{X_{i^*} - X_A}{-Y_{i^*} + Y_A}\right) + 90^{\circ} - \theta_{i^*} \tag{4}$$

with $(X_{i^*}, Y_{i^*}, \theta_{i^*})$ the position and orientation of the closest subject i^*, in the virtual map.

Next, using the distance d^{*}_{i} between the agent and the subject, a scale factor SF_S is calculated as:

$$SF^{*}_{S} = \frac{d_c}{d^{*}_{i}} \tag{5}$$

where d_c is the distance between the subject and the camera at which the images in the database were taken (in our case this is equal to 140 [cm]). Since the database has only 3 different pitch angles (-15°, 0° and 15°), the pitch angle of the image is estimated as:

$$\theta^{*}_{Pitch} = \begin{cases} 0^{\circ} & if \quad |(H_{i^*} - H_A)| < \frac{\tan(15^{\circ}) * d^{*}_{i}}{2} \\ 15^{\circ} & if \quad (H_{i^*} - H_A) > \frac{\tan(15^{\circ}) * d^{*}_{i}}{2} \\ -15^{\circ} & \sim \end{cases} \tag{6}$$

with H_A the height of the agent and H_{i^*} the height of the selected subject.

Using this information, the image of subject i^*, containing a face with the rotation angles θ^{*}_{yaw} and θ^{*}_{Pitch} is selected from the database of captured images. The image is resized using the scale factor SF^{*}_{S}. When composing the image, background information is added to ensure that the generated image has the required resolution after rescaling and translating it. Once the image has been rescaled, the rotations Δ_{yaw} and Δ_{pitch}, produced by the relative pose between the agent and the observed subject, are calculated as follows:

$$\Delta_{yaw} = 180^{\circ} + \tan^{-1}\left(\frac{Y_A - Y_{i^*}}{X_{i^*} - X_A}\right) - \theta_A \tag{7}$$

$$\Delta_{pitch} = \tan^{-1}\left(\frac{H_{i^*} - H_A}{d^{*}_{i}}\right). \tag{8}$$

Finally, the translation in the image plane Δ_x and Δ_y, produced by the relative pose between the agent and the observed subject, is calculated as follows:

$$\Delta_x = \frac{\Delta_{yaw} * I_W}{FOV_H} \tag{9}$$

$$\Delta_y = \frac{\Delta_{pitch} * I_H}{FOV_V} \tag{10}$$

with FOV_H and FOV_V the horizontal and vertical FOV (Field of View) parameters of the camera used to capture the database (inourcase$FOV_H = 56.3^{\circ}$ and $FOV_V = 42.3^{\circ}$).

Having occluding subjects in the generated image is also possible. In particular, we added the option of simulating pillars (cylindrical columns) in the environment, as illustrated in Fig. 6. The position of a pillar is predefined for a particular scenario, and the pillar's size in the generated image depends on the relative position of the pillar with respect to the agent. For simulating the occlusion, four images of real pillars at a distance d_p of 100 [cm] were captured using the device described in Section 3.1. The image of the occluding pillar to be used in a particular scenario is chosen at random.

Given (X_o, Y_o), the location of the occluding pillar in the global map, and (X_A, Y_A, θ_A) the current pose of the agent in the global map, the distance d_o between the agent and the obstacle, and the scale factor SF_o are calculated as follows:

$$d_o = \sqrt{(X_o - X_A)^2 + (Y_o - Y_A)^2} \tag{11}$$

$$SF_o = \frac{d_p}{d_o} \tag{12}$$

Then, the translation of the obstacle in the image plane, Δ_{x_o}, with respect to the agent, is calculated as follows:

$$\Delta^{\theta}_{yaw} = 180^{\circ} + \tan^{-1}\left(\frac{-Y_o + Y_A}{X_o - X_A}\right) - \theta_A \tag{13}$$

$$\Delta_{x_o} = \frac{\Delta^{\theta}_{yaw} * I_W}{FOV_H} \tag{14}$$

3.2. Agent Navigation and Positioning

The *Agent Navigation and Positioning* module allows the virtual agent to move inside the virtual environment. Specific movements are allowed depending on the scenario being simulated. The simulator provides four basic commands that are used to move the agent from its current pose (X_A, Y_A, θ_A):

- MoveAgent$(\Delta x, \Delta y)$: It moves the agent relative to its current pose. The final pose of the agent is:
$$(X_A + \Delta x, Y_A + \Delta y, \theta_A) \tag{15}$$
- TurnAgent$(\Delta\theta)$: It rotates the agent relative to its current orientation. The final pose of the agent is:
$$(X_A, Y_A, \theta_A + \Delta\theta) \tag{16}$$
- SetAgentPosition(x, y, θ): It set the agent's absolute pose. The final pose of the agent is:
$$(x, y, \theta) \tag{17}$$
- NextPosition(): It moves the agent to the next point $(\tilde{X}_k, \tilde{Y}_k, \tilde{\theta}_k)$ in a defined trajectory (see more on trajectories below).

These commands allow the movement of the agent, and therefore the use of active vision mechanisms to improve the recognition results. In order to have realistic conditions, the virtual environment can simulate uncertainties in the movement and odometry of the agent (robot in this case).

The virtual environment provides three kinds of navigation modalities:

- *Constrained* navigation: in this modality the agent moves under constrained conditions. The agent has a predefined trajectory for approaching every subject. After this trajectory is executed, the agent moves to the next subject.
- *Predefined* navigation: in this modality the agent follows a predefined trajectory. However, at any position the agent can decide to move out of the trajectory to explore and change its perceptions.
- *Free* navigation: in this modality the agent does not have a predefined trajectory and it moves freely inside the virtual environment without any restriction or guidance. Thus, in this modality the agent controls its movement using the observed images as the only information source.

A trajectory **T** is defined as a sequence of K triplets $\{(\tilde{X}_1, \tilde{Y}_1, \tilde{\theta}_1), \ldots, (\tilde{X}_k, \tilde{Y}_k, \tilde{\theta}_k), \ldots, (\tilde{X}_K, \tilde{Y}_K, \tilde{\theta}_K)\}$, where $(\tilde{X}_k, \tilde{Y}_k, \tilde{\theta}_k)$ indicates the *k-th* position $(\tilde{X}_k, \tilde{Y}_k)$ and pose angle $\tilde{\theta}_k$ of the agent

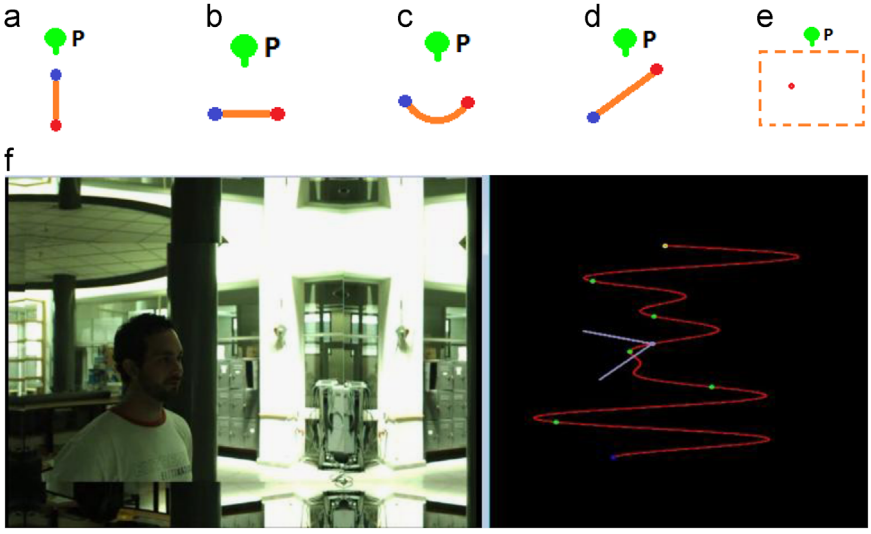

Fig. 7. Examples of the implemented trajectories. See main text for details.

in the Global Map. The following types of trajectories are defined for the *constrained* and *predefined* navigation modalities:

- *Constrained* navigation trajectories: Five variants of these trajectories are provided (see Fig. 7(a–e)):
  a) *Frontal*: the agent approaches the subject while having a frontal view of him. The agent is looking directly at the subject in every point of the sequence.
  b) *Side-to-side*: the agent moves perpendicular to an imaginary line coming from the observed subject. The relative yaw angle and the distance changes. The agent is looking perpendicular to the trajectory in every point of the sequence.
  c) Circular: the agent moves around the subject at a fixed distance. The relative yaw angle changes, but the distance remains constant. The agent is looking directly at the subject in every point of the sequence.
  d) *Strafe*: this movement is a combination of *Frontal* and *Side-to-side*. The agent moves with respect to an imaginary line that is not perpendicular to the frontal view of the subject. The movement does not maintain a fixed distance between the agent and the subject because the agent approaches the subject. The agent is looking in a fixed angle relative to the trajectory in every point of the sequence. The angle can be defined by the user.
  e) *Random*: the agent is placed by the simulator in front of the subject, at a random position within a region defined in front of the subject (rectangle in Fig. 7(e)). The agent always looks directly at the subject.
- Predefined navigation trajectory: Given a trajectory **T**, the agent will visit all the positions (X_k, Y_k) defined by the trajectory. At each point in the trajectory, the angle θ_k is defined as parallel to the tangent of the trajectory. The agent moves from one location to the next one as defined by the trajectory, but in addition, the agent can move freely from and around each point on the trajectory. After the exploration is complete, the agent can continue to the next point on the trajectory. This allows the agent to visit each subject and also explore around it, but without the need to implement sophisticated navigation mechanisms. The implemented trajectories pass near every subject in the map as the agent moves through each of the points (X_k, Y_k, θ_k) of the trajectory (see example in Fig. 7(f)). Given that the agent can freely move out of the predefined trajectory to explore and capture specific views of the scene, it can easily make use of the active vision mechanisms implemented in the *Active Vision* module.

The different navigation modes and trajectories allow simulating different types of application scenarios, such as (i) a moving agent (robot) in an indoor environment (using *free* navigation), (ii) a security camera observing people passing in front of it (*constrained* navigation with *strafe* movements), (iii) a scanning device performing a circular moving around a subject (*constrained* navigation with *circular* movements), or (iv) a simple frontal view of the subjects (*constrained* navigation with *frontal* movement), among many others.

3.3. Agent vision

The agent's vision system receives as input the current view of the scene as generated from the *Image Generation module*. The agent processes these images using the selected vision functionalities (face detection, recognition, etc.). After the virtual agent has finished processing the image, it will move in the virtual scenario and request a new image to the simulator. In case the agent's active vision module is used, the agent's request to move within the virtual scenario can be outside the predefined trajectory, with the requested new position being determined from the sensed visual information. The particular active vision mechanism to be used must be implemented by the user in the *Active Vision* module, so that the agent makes decisions to move using visual information. The request to move to the new position will be processed by the *Agent Navigation and Positioning* module, which will estimate the new position of the agent in the scene.

3.3.1. Vision

Given that the main goal of the evaluation tool is to analyze methods related to the analysis of faces, the evaluation tool provides some functionality to the agent. First, it includes basic face analysis modules, such as face detection, face recognition, eye detection, face alignment, face cropping, and gender classification, modules that can be used as baseline methods for comparison or replaced with other implementations (the user can use its own algorithms). For example, to evaluate a face recognition method, the user can make use a "perfect face detector" that uses the ground truth, or he can choose a more realistic scenario and make use of the OpenCV face detector [37]. Also, the system provides the required data structures to store a set of detected faces, which can be used as gallery/training images for future recognition. It also provides a map, where the agent can store its own estimated map of the location of observed subjects. Therefore, there are two databases to aid the implementation of the agent's vision system:

- *Person Map Database*: The virtual environment provides to the virtual agent a *Person Map* database to store information of the subjects detected in the environment. The agent incrementally adds the pose of the every detected person, and then uses that information to determine if a set of detections corresponds to the same subject or not.
- *Face Gallery Database*: The virtual environment provides a *Face Gallery* database that the agent uses to store information of the subjects seen so far (and the corresponding ID in the Person Map) in order to perform the recognition. The information can be stored before the agent starts to navigate the virtual environment, thus this gallery can be built online or offline depending on the experiment being performed. The stored information corresponds to a set of face images for each subject, the corresponding ID in the *Person Map* and the face representation (features).

The output of the face detection is the position of the face in the image $(\mathrm{FD}_x, \mathrm{FD}_y)$, and the size of the bounding box that frames the faces (F_w, F_H). Then, using the output of the face detector, the distance $CfRe^{\frac{1}{2}}$ from the subject to the agent and its relative angular pose $\left(\theta_x^F, \theta_y^F\right)$ are estimated as:

$$d_i^{\mathrm{e}} = \frac{\tau * \rho * I_F}{F_w}, \tag{18}$$

$$\theta_x^F = \frac{(\mathrm{FD}_x - I_W)}{I_W}\frac{\mathrm{FOV}_H}{2}, \tag{19}$$

$$\theta_y^F = \frac{(\mathrm{FD}_y - I_H)}{I_H}\frac{\mathrm{FOV}_V}{2}, \tag{20}$$

with (I_W, I_H) the image size, τ the mean size of a face (in pixels) at a distance of ρ [cm], and I_F a scaling factor of distance estimation (relative to an image of resolution $I_{WB} \times I_{HB}$):

$$I_F = \frac{I_W}{I_{WB}}, \tag{21}$$

In our database the values of these parameters are: $\tau = 75$ [pixels], $\rho = 100$[cm], $I_{WB} = 320$[pixels], and $I_{HB} = 240$[pixels].

Given that the estimation of d_i^{e} can have errors, before adding a new face to the *Person Map*, this error (E_i) is estimated and used to determine if the new detected face is added to the *Person Map* or not. The error is calculated using a linear equation:

$$E_i = \alpha + \beta d_i^{\mathrm{e}}, \tag{22}$$

where the values of parameters α and β ($\alpha = 50$ and $\beta = 1/3$) were estimated using several images for which the faces where at a known distance.

The construction of the *Person Map* is done in the following way. The *Person Map* is first empty. Then, every time a face is detected the *Person Map* is updated using the information of the detected face. The information of the faces already in the map is used to determine if the new detected face needs to be added to the map.

The detected face's information $(d_i^{\mathrm{e}}, \theta_x^F, \theta_y^F, E_i)$ and the agent's pose (X_A, Y_A, θ_A) are used to estimate the position and orientation of the subject in the person map:

$$X_i = d_i^{\mathrm{e}} * \cos\left(\theta_A + \theta_x^F\right) + X_A \tag{23}$$

$$Y_i = d_i^{\mathrm{e}} * \sin\left(\theta_A + \theta_x^F\right) + Y_A \tag{24}$$

$$\theta_i = 180 + \theta_A \tag{25}$$

The subject's orientation θ_i is estimated under the assumption that the agent is facing the subject. A face pose estimator can be used to improve the angular pose estimation.

After the position of the detected subject has been estimated, the closest subject in the person map within a radius D_{Min} is determined (N_A is the number of subjects stored in the person map):

$$j_{min} = \operatorname{argmin}_{j \in \{1,..,N_A\}}(D_{ij})$$

$$\text{s.t.} \quad D_{ij} < D_{Min}$$

$$\text{with } D_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \tag{26}$$

If the estimated error (E_i) associated to the new subject is smaller than the estimated error of the actual subject in the *Person Map* $(E_{j_{min}})$, then the subject in the map is replaced by the new, detected one. If there is no subject in the person map within a radius D_{Min}, then the new person is stored in the *Person Map,* and the counter of subjects, N_A, is increased by one.

3.3.2. Active vision

As mentioned, during the face recognition process, the agent can move in the map, and change its point of view and distance from the subjects in order to actively modify its observations and improve the performance of the vision system. Thus active vision mechanisms can be implemented.

Given a trajectory **T**, the agent starts positioned at the beginning of the defined trajectory (X_1, Y_1, θ_1). Then from each location (X_k, Y_k, θ_k) the agent is allowed to move freely and modify its observations. After the exploration is complete, the agent can continue along the next point on the trajectory. For example, given a detected face, the agent could use a face pose estimator to estimate the pose of the detected subject (yaw angle θ_i^y), and then to move in order to change its view angle. The agent can repeat this process as many times as necessary to fulfill the conditions (e.g. yaw rotation less than 15°) before applying a face recognition algorithm). After the agent finishes this process, it can return to the trajectory and continue traversing the scenario.

4. Example applications of the evaluation tool

In order to validate the applicability of the evaluation tool, three experiments were carried out. In the first experiment, three state-of-the art face detection methods are compared under different yaw and pitch face angles. In the second experiment, five different face recognition algorithms are analyzed under different yaw and pitch face angles, and under simulated incorrect face alignment. In the third experiment, a simulated scenario where a virtual humanoid robot navigates, detects and recognizes the subjects in the scene, three unsupervised face recognition methods are compared under different viewing conditions, with and without the use of active vision mechanisms. This last evaluation would be very difficult to perform using existing databases that do not support the use of active vision mechanisms.

4.1. Evaluation of face detectors under out-of-plane rotations

The virtual scenario is first used to evaluate face detection methods in a static setting using a *constrained navigation* trajectory with *random* movements (see details in Section 3.2). The agent is placed by the system at a fixed distance of 100 cm in front of each subject (initial pose). The scenario contains $N = 20$ subjects chosen at random order. The experiment was repeated 10 times, with variations in the positioning of each subject (position, angle, pose, etc.). The agent's camera and the observed face are at the same height, and the agent cannot move its head independently of the body. The agent processes the generated images seeking to detect human faces. The following variations in the agent's relative position and viewpoint are incorporated before the agent starts recognizing subject *i*:

Table 1
Detection rates under different maximal yaw and pitch angles of the observed face ($\theta^y_{max}, \theta^p_{max}$). DR: detection rate. FP: number of false positives. See main text for details.

Detector	$\theta^p_{max}=0$															
	$\theta^y_{max}=0^\circ$				$\theta^y_{max}=20^\circ$				$\theta^y_{max}=40^\circ$				$\theta^y_{max}=60^\circ$			
	Indoor		Outdoor		Indoor		Outdoor		Indoor		Outdoor		Indoor		Outdoor	
	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP
HaarCascade	97.4	2	78.3	5	92.1	3	87.0	3	71.1	4	65.2	8	47.4	1	82.6	4
NestedCascade1	89.5	4	100	0	86.8	5	100	0	71.1	6	100	0	55.3	9	69.6	3
NestedCascade2	100	0	100	0	97.4	0	100	0	73.7	1	95.7	0	50.0	2	73.9	1
	$\theta^p_{max} \in [-15^\circ, 15^\circ]$															
	θ^y_{max}:=0				$\theta^y_{max} \in [-20^\circ, 20^\circ]$				$\theta^y_{max} \in [-40^\circ, 40^\circ]$				$\theta^y_{max} \in [-60^\circ, 60^\circ]$			
	Indoor		Outdoor		Indoor		Outdoor		Indoor		Outdoor		Indoor		Outdoor	
	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP	DR	FP
HaarCascade	94.7	2	95.7	1	84.2	4	69.6	7	65.8	6	78.3	5	60.5	6	65.2	8
NestedCascade1	94.7	2	100	0	84.2	4	100	0	60.5	11	100	0	42.1	12	69.6	4
NestedCascade2	97.4	0	100	0	94.7	0	100	0	63.2	1	95.7	0	52.6	3	82.6	1

- The initial pose of the agent is modified by adding uniformly distributed random values Δx_i, Δy, and$\Delta\theta_i$. The maximal variation in each axis, $(\Delta x_{max}, \Delta y_{max}, \Delta\theta_{max})$, are simulation parameters.
- The pose of the face of subject i is set at θ^y_i (yaw angle), θ^p_i (pitch angle), and θ^r_i (roll angle). The maximal allowed rotation value in each axis, $(\theta^y_{max}, \theta^p_{max}, \theta^r_{max})$, are simulation parameters.

Three face detection methods are compared. First, the standard Haar cascade detector provided by OpenCV (OpenCV's *HaarTraining*). This detector implements the cascade detector described in [38], which is an extension of the classical Viola & Jones face detector [39], and that uses the Gentleboost training algorithm, Haar-like features, and decision stumps as weak classifiers. Second, a face detector proposed by Verschae et al. [40], which uses nested cascades of classifiers, the real Adaboost boosting algorithm, and domain-partitioning based classifiers. Third, a face detector proposed by Verschae [41], which improves its former detector by using a coarse-to-fine approach when training the cascades. These three detectors are called *HaarCascade*, *NestedCascade1*, and *NestedCascade2*, respectively, in the experiments.

The detection rate (DR) and the number of false positives (FP) of the three detectors are compared for each different viewpoint conditions; the yaw angle of the observed faces is uniformly sampled in the range $[-\theta^y_{max}, \theta^y_{max}]$, as well as the pitch angle is uniformly sampled in the range $[-\theta^p_{max}, \theta^p_{max}]$. Other simulation parameters are kept unchanged $(\Delta x_{max} = \Delta y_{max} = \Delta\theta_{max} = \theta^r_{max} = 0)$.

In Table 1, the obtained results for θ^y_{max} values of 0°, $[-20^\circ, 20^\circ]$, $[-40^\circ, 40^\circ]$ and $[-60^\circ, 60^\circ]$, and θ^p_{max} values of 0° and $[-15^\circ, 15^\circ]$, under indoor and outdoor illumination conditions are shown. Main conclusions of these experiments are: (i) *NestedCascade2* has better performance than the other two detectors, having a very small number of false positives, (ii) in most of the experiments the performance of the methods decrease in outdoors, and (iii) when large out-of-plane rotations are considered, the detection rate of all detectors is decreased in about 40%.

This kind of experiment shows how the proposed system can be used for evaluating the robustness of face detection methods under rotations. It allows characterizing the response of the algorithms (in this case face detectors) to yaw and pitch rotations. For example, from the results of this particular case, it is clear that all detectors have good detection rates for frontal faces, but their performance is very different when larger yaw rotations are considered, with the NestedCascade2 detector having lower false positive rates for similar detection rates for most yaw and pitch rotations. This experiment also shows that it is possible to simulate errors in the acquisition process (through Δx_i, Δy_i, and $\Delta\theta_i$). This kind of analysis cannot be easily done when using existing databases.

4.2. Evaluation of face recognition methods under out-of-plane rotations

The scenario contains $N=20$ subjects. A *constrained navigation* trajectory with *random* movements (see details in Section 3.2) is used. The agent is placed by the system at a fixed distance of 100 cm in front of each subject (initial pose). The evaluation of face recognition algorithms is tested in several cases, taking into account more variability in the pose of the faces.

The agent processes the generated images seeking to recognize the faces. Three local-matching face recognition methods are evaluated: histograms of LBP (Local Binary Patterns) features [42], Gabor-Jet features with Borda count classifiers [43], and histograms of WLD (Weber Local Descriptor) features. The first two methods have shown a very good performance in comparative studies of face recognition systems [13,43]. The third method is proposed in [44], and it is based on the use of WLD features [45]. In all cases, the methods' parameters were selected using standard face datasets [13], and not using the face images included in the proposed virtual environment.

Following the results reported in [13], two different flavors of the histograms of LBP features method are used, one using the histogram intersection (HI) similarity measure, and one using the Chi square (XS) measure. In both cases, face images are scaled to 81×150 pixels and divided into 40 regions to compute the LBP histograms. The two implemented face recognition systems are called LBP-HI-40 and LBP-XS-40. The implemented Gabor-based method uses 5 scales and 8 orientations, and face images scaled to 122×225 pixels, as reported in [13]. Finally, in the case of the WLD based method, following [13] and the results obtained in the FERET, BioID and LFW databases, the following parameters were selected: histogram intersection and Chi square similarity measures, face images scaled to 93×173 pixels and divided into 40 regions to compute the WLD histograms, 2 dominant orientations ($T=2$), and 26 cells in each orientation ($C=26$).

Table 2
Top-1 recognition rates [%] under different maximal yaw angles of the observed face (θ^y_{max}). The other parameters are not varied($\Delta x_{max} = \Delta y_{max} = \Delta\theta_{max} = \theta^p_{max} = \theta^r_{max} = 0$).

Method	θ^y_{max}								
	5°	10°	15°	20°	25°	30°	35°	40°	60°
LBP-HI-40	100	100	100	100	95	95	80	85	55
LBP-XS-40	100	100	100	95	95	95	85	75	30
GJD-BC	100	100	100	95	85	85	75	80	35
WLD-HI-40	100	100	95	95	90	90	85	70	45
WLD-XS-40	100	100	90	90	95	90	75	70	45

In a first set of experiments, the recognition rate of the different methods is compared under different viewpoint conditions; the yaw angle of the observed faces is uniformly selected (random value) in the range $[-\theta^y_{max}, \theta^y_{max}]$. The other simulation parameters are kept unchanged ($\Delta x_{max} = \Delta y_{max} = \Delta\theta_{max} = \theta^p_{max} = \theta^r_{max} = 0$). In this experiments no active vision mechanisms are used, a simulated face detection rate of 100% is considered (ground truth of the face and eye positions is used) and a simulated pose estimator is considered (ground truth is used). Table 2 shows the obtained results. Main conclusions of these experiments are: (i) LBP based methods that use the Chi square similarity measure are more robust to yaw rotations than Gabor and WLD based methods, and (ii) all methods have good performance under yaw rotations within the range $\pm 30°$.

In a second set of experiments, the recognition rate of the different methods is compared under more uncontrolled conditions:

1. The yaw angle of the observed faces is randomly sampled (uniform distribution) in the range $[-45°, 45°]$, and the pitch angle in $[-15°, 15°]$. The roll angle is not modified ($\theta^r_{max} = 0$).
2. The position of the observer agent is modified in each axis, by a random value uniformly selected in the range $[-20,20]$ or $[-40,40]$ cm (The agent places itself randomly in front of the subject within inside a region defined in front of the subject, see Fig. 7(e)). The agent is not rotated ($\Delta\theta_{max} = 0$).

The following face detection and pose estimation conditions are considered: (i) a face detection rate of 80% with no false positives; (ii) a face pose estimation with an error, *pe*, uniformly selected (random value) in the range $\pm 40\%$ or $\pm 80\%$ of the estimated value; and (iii) active vision mechanisms: given a detected face, a face pose estimator is used to estimate the pose of the subject (yaw angle θ^y_i); the agent's position (X^*_A, Y^*_A, Θ^*_A) is used to ensure that the agent is positioned facing the person. Finally the face is recognized. Table 3 shows the obtained results. Main conclusions of these experiments are: (i) LBP based methods are more robust to the defined uncontrolled conditions than Gabor and WLD based methods, (ii) the agent's initial position error has a low influence on the final performance of the recognition systems, (iii) a maximal error of $\pm 15°$ in the pitch angle reduces in $\sim$5% the face recognition rate, and (iv) a pose estimation error increase from 40% to 80% reduces in $\sim$5% the recognition rate.

The systematic evaluation of face recognition algorithms, when considering realistic scenarios, is not an easy task. For example, having a good estimation of the performance of a face recognition algorithm under large rotations can be important in many applications, but it is often neglected. The proposed framework allows evaluating and characterizing the performance of face recognition methods under different rotations, and in particular to simulate pose estimation errors. Pose estimation and face alignment algorithms are part of the standard pipeline in face recognition methods, and in many cases their effect is not taken into account when evaluating face recognition systems. The ability to simulate pose estimation errors in the proposed framework allows having a better idea of the robustness of face recognition algorithms to these errors.

4.3. Active vision in face analysis for a Humanoid Robot Platform

In this experiment a humanoid robot moving inside a room with *N* subjects is simulated, and its goal is to detect and recognize all subjects. The robot has a *predefined navigation* trajectory (as described in Section 3.2) that allows him to pass near each of the subjects in the room, however not all subjects are facing directly the robot. In this experiment the use of active vision is evaluated, thus the robot can navigate freely and it can change its pose relative to the subject when the active vision module is activated.

In order to recognize faces properly, the virtual agent uses the following modules:

1. *Face Detection*: The agent detects a face (i.e. the face region) in a given image.
2. *Face Pose Estimation*: The agent estimates the face's angular pose in the lateral, sagittal and coronal plane.
3. *Active Vision*: Using information about the detected face and its pose, as well as information observed in the input images, the agent can take actions in order to change its viewpoint for improving face's perception.
4. *Face Recognition*: The identity of the person contained in the face image is determined. The module can include abilities such as face alignment or illumination compensation.

In this experiment the OpenCV Viola&Jones face detector, a LBP-based face recognition method, a perfect face and eye detection (using the ground truth), and a face pose estimation (also provided using the ground truth) are evaluated. The OpenCV Viola&Jones face detector, and the LBP-based face recognition method and well-known methods commonly used in this kind of applications. Given the results obtained in Section 4.2, LBP with histogram intersection (HI) as similarity measure was used.

The number of subjects (*N*) in the map was set to 10 subjects in the first group of experiments, while 20 subjects were used in the last two experiments. It is important to recall that the same subject may be observed in many frames when the robot is navigating the scenario. Also, each experiment is run 10 times, each time with a different subset of subjects out of the 50 subjects of the database and using several robot trajectories (see Fig. 4(f) for an example). Note that the same samples (same subjects and trajectories) are used to evaluate all considered variants (e.g. with and without active vision).

The height of the agent is fixed and equal to the base height of the subjects (160 cm). The height of subjects follows a uniform distribution in [136, 184] cm, i.e. a 15% variation around the agent's height.

With respect to how the gallery of faces for recognition is constructed, we consider two modes:

1. *Offline Gallery*: The virtual environment provides a face gallery before the recognition process starts. The gallery contains one image of each person to be recognized. The gallery's images are frontal pictures (no rotations in any plane), and are taken under indoor illumination conditions. This is the standard operation mode.
2. *Online Gallery*: There is no offline gallery. The agent needs to navigate through the virtual scenario twice. In the first round, the agent should create the gallery online, i.e., face detection, and pose estimation are needed to build the database. In the second round, all subjects change their position in the scene,

Table 3
Top-1 recognition rates [%] under different maximal pitch angles of the observed face (θ^p_{max}), $\theta^y_{max} \in [-45°, 45°]$, different maximal agent's positioning errors ($\Delta x_{max}, \Delta y_{max}$) and variable face pose estimation error (pe).

Method	$\theta^p_{max}=0$ $\Delta x_{max}=20$ $\Delta y_{max}=40$ $pe=40\%$	$\theta^p_{max}=0$ $\Delta x_{max}=40$ $\Delta y_{max}=40$ $pe=40\%$	$\theta^p_{max}\in[-15,15]$ $\Delta x_{max}=20$ $\Delta y_{max}=20$ $pe=40\%$	$\theta^p_{max}\in[-15,15]$ $\Delta x_{max}=40$ $\Delta y_{max}=40$ $pe=40\%$	$\theta^p_{max}\in[-15,15]$ $\Delta x_{max}=20$ $\Delta y_{max}=20$ $pe=80\%$	$\theta^p_{max}\in[-15,15]$ $\Delta x_{max}=40$ $\Delta y_{max}=40$ $pe=80\%$
LBP-HI-40	85	85	80	75	75	70
LBP-XS-40	85	85	80	80	80	75
GJD-BC	85	80	75	70	70	65
WLD-HI-40	80	85	70	65	70	65
WLD-XS-40	80	85	70	65	70	65

Table 4
Evaluation of a face recognition system based on LBP features [42].

(*) It does not use Person Map. (**) 20 subjects are present in the scene; otherwise there are 10 subjects.

Face detection	Active vision	Gallery database	Subjects added to the gallery	Subjects correctly detected [%]	Recognition [%] (out of all subjects in the scene)	Recognition [%] (out of the detected subjects)
Viola&Jones	No	Offline	10.0	84.0%	–	78.4% (*)
Viola&Jones	No	Offline	10.0	84.0%	73.0%	86.8%
Viola&Jones	No	Online	14.8	84.0%	59.0%	70.2%
Viola&Jones	Yes	Offline	10.0	84.0%	78.0%	92.9%
Viola&Jones	Yes	Online	12.2	84.0%	73.0%	86.9%
Ground Truth	Yes	Offline	10.0	100.0%	92.0%	92.0%
Ground Truth	Yes	Online	10.0	100.0%	90.0%	90.0%
Viola&Jones	No	Offline	20.0 (**)	91.8%	83.0%	90.5%
Viola&Jones	Yes	Offline	20.0 (**)	91.8%	87.5%	95.7%

and the agent must detect and recognize them using the gallery already built. In both rounds, the agent observes the person's faces at variable distance and angles, in indoor or outdoor illumination conditions. The subjects pose in the scenario, and the illumination conditions are randomly chosen in all cases.

As mentioned, the case of *predefined trajectories* is considered. With respect to how the agent moves, two cases are considered: with and without the use of active vision. In both cases the use of online and offline gallery DB is taken into consideration, which gives 8 cases in total. When active vision is used, the algorithm works as follows: if a face is detected, the face pose estimator is used to determine whether the detected face has small rotations (relative angles smaller than 15°). If this is the case, the face is either stored in the database (first time detected) or used for recognition. If the rotation is larger than 15°, the next agent's position $(X^*_A, Y^*_A, \theta^*_A)$ is estimated to ensure that the agent is positioned in front of the person and the face rotation is minimal. Given this position, the agent moves, gets new observation, and detects faces again. If the new detected face still has a rotation angle larger than 15°, this process is repeated, otherwise the agent returns to the original trajectory.

Table 4 shows the obtained results. The first, second and third columns show which face detector was used, whether the active vision module was used or not, and how the gallery database was built, respectively. The fourth column displays the number of subjects added to the gallery database (normally when the gallery is built online, there are some false detections); the fifth column shows the rate of number of subjects correctly detected while traversing the environment; the sixth column shows the face recognition rate out of all subjects in the scene; and the seventh column shows the face recognition rate considering only the detected subjects.

From the first two rows of the table we can see that when the *Person Map* module is used, the recognition rate improves from 78.41% to 86.77%. This is because the same subject is not added several times to the map, and at the same time there are less false detections. In all the other results (remaining rows in the table) the *Person Map* is used. From the table it can also be observed that using the active vision module allows, among other things, to build a better gallery database (when the gallery database is built online), and to improve the recognition rate. When the database is built offline, using active vision improves the recognition rate from 86.77% to 92.92% on average.

As it can be observed, the systematic evaluation of a face analysis system for a service robot under real world conditions can be done using the proposed evaluation tool. In this particular experiment, the simulation tool was used to evaluate: (i) the effect of using active vision mechanics during the gallery building process and during the recognition process, (ii) the use of a perfect face detector (using the ground truth) versus the use of a realistic face detector, and (iii) the use of simple (person) mapping mechanisms such as the *Person Map*. As shown in the experiments, such mechanisms are very important in real world applications, as they can have a high impact in the performance of a face analysis system.

It must be stressed that performing this kind of evaluation using standard face databases and methodologies is not possible, while performing such a systematic quantitative evaluation with a robot in a real scenario is very difficult and time consuming. However, the proposed evaluation tool enables the systematic evaluation of the complete pipeline of algorithms in a robot's face analysis system, and in particular using changing scenarios (e.g. gallery database building) with dynamic moving agents (e.g. active vision mechanisms).

5. Conclusions

An evaluation tool for testing face analysis systems under uncontrolled conditions is proposed. The tool combines the use of a simulator with real face and background images taken under

real-world conditions. Inside the virtual environment, a virtual agent navigates and observes face images (from different distances and angles), and with either indoor or outdoor illumination. During the face detection and recognition process, the agent can actively change its viewpoint and relative distance to the faces in order to improve the detection and recognition results.

The applicability of the proposed tool is validated in three scenarios: evaluation of the robustness of face detection methods to pose changes, evaluation of face recognition methods when an active vision mechanisms is used, and simulation of a robot moving in an environment with several subjects. In this last case the face recognition performance was evaluated using online and offline procedures for building the gallery databases, as well as when active vision mechanisms instead of static trajectories are used.

The reported experiments show that the proposed system can be used for evaluating the robustness of face detection methods under rotations, and that it allows characterizing the response of the algorithms (in this case face detectors) to yaw and pitch rotations. For example, from the reported results it can be concluded that although the analyzed detectors have good detection rates for all frontal faces, their performance is very different when larger yaw rotations are considered. This kind of analysis cannot be easily done when using existing databases.

In the case of face recognition methods, the use of the proposed tool allows evaluating and characterizing the performance of face recognition methods under different rotations, and in particular to simulate pose estimation errors. Pose estimation and face alignment algorithms are part of the standard pipeline in face recognition methods, and in many cases their effect is not taken into account when evaluating face recognition systems. The ability to simulate pose estimation errors in the proposed framework allows having a better understanding of the robustness of face recognition algorithms to these errors.

In the case of the robot scenario, the proposed evaluation tool enabled the systematic evaluation of the complete pipeline of face analysis algorithms. In particular, it was possible to evaluate the effect in the recognition performance of the gallery database generation mode (offline versus online), of the employed active vision mechanisms, and of the use of person maps containing information of previously detected faces. The reported results show how the proposed evaluation tool enables the systematic evaluation of face analysis systems for a robot. Performing such an evaluation in a real scenario would be very difficult and time consuming, and implementing it using standard face databases and methodologies not possible.

Possible future research lines and improvements include the development and evaluation of more sophisticated active vision mechanism, the use of several detected faces in the recognition of the faces of the same subject, and the evaluation of other face analysis tasks, such as age and gender classification.

Conflict of interest

None declared.

Acknowledgment

This research was partially funded by the FONDECYT-Chile Grants 3120218 and 1130153.

References

[1] P.J. Phillips, H. Wechsler, J. Huang, P. Rauss, The FERET database and evaluation procedure for face recognition algorithms, Image Vis. Comput. J. 16 (5) (1998) 295–306.
[2] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: a database for studying face recognition in unconstrained environments (Technical Report 07-49), University of Massachusetts, Amherst, 2007.
[3] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, D. Zhao, The CAS-PEAL large-scale chinese face database and baseline evaluations, Trans. Syst. Man Cybern. Part A 38 (1) (2008) 149–161.
[4] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, W. Worek, Overview of the face recognition grand challenge, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. – CVPR 2005 1 (2005) 947–954.
[5] Face Recognition Grand Challenge, Official website public site (Available on June 30th, 2010): ⟨http://www.frvt.org/FRGC/⟩.
[6] A.F. Abate, M. Nappi, D. Riccio, G. Sabatino, 2D and 3D face recognition: a survey, Pattern Recognit. Lett. 28 (2007) 1885–1906.
[7] Face Recognition Home Page (Available on June 4th, 2012): ⟨http://www.face-rec.org/databases/⟩.
[8] BeFIT-Benchmarking Facial Image Analysis Technologies home page (Available on July 5th, 2012): ⟨http://fipa.cs.kit.edu/412.php⟩.
[9] R. Goh, L. Liu, X. Liu, T. Chen, The CMU face in action (FIA) database, in: W. Zhao, S. Gong, X. Tang (Eds.), Analysis and Modelling of Faces and Gestures. Springer Berlin Heidelberg, 2005, pp. 255–263.
[10] B. Martinkauppi, M. Soriano, S. Huovinen, M. Laaksonen, Face video database, in: Proceedings of Society for Imaging Science and Technology Conference on Colour in Graphics Imaging, and Vision, 2002, pp. 380–383.
[11] L. Wolf, T. Hassner and I. Maoz, Face recognition in unconstrained videos with matched background similarity, in: Proceedings of IEEE Conference onComputer Vision and Pattern Recognition (CVPR), 2011, pp. 529–534.
[12] A. Mian, Online learning from local features for video-based face recognition, Pattern Recognit. 44 (5) (2011) 1068–1075.
[13] J. Ruiz-del-Solar, R. Verschae, M. Correa, Recognition of faces in unconstrained environments: a comparative study, EURASIP J. Adv. Signal Process. (2009) 19184617 (Recent Advances in Biometric Systems: A Signal Processing Perspective).
[14] H. Han, S. Shan, X. Chen, W. Gao, A comparative study on illumination preprocessing in face recognition, Pattern Recognit. 46 (6) (2013) 1691–1699.
[15] X. Zhang, Y. Gao, Face recognition across pose: a review, Pattern Recognit. 42 (11) (2009) 2876–2896.
[16] J. Ruiz-del-Solar, J. Quinteros, Illumination compensation and normalization in eigenspace-based face recognition: a comparative study of different preprocessing approaches, Pattern Recognit. Lett. 29 (14) (2008) 1966–1979.
[17] A. Samal, P.A. Iyengar, Automatic recognition and analysis of human and facial expressions: a survey, Pattern Recognit. 25 (1) (1992) 65–77.
[18] R. Gross, Face databases, in: S. Li, A.K. Jain (Eds.), Handbook of Face Recognition, Springer-Verlag, 2005, pp. 301–327.
[19] S. Zafeiriou, M. Hansen, G. Atkinson, V. Argyriou, M. Petrou, M. Smith, L. Smith, The photoface database, in: Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2011, pp. 132–139.
[20] G. Hermosilla, J. Ruiz-del-Solar, R. Verschae, M. Correa, A comparative study of thermal face recognition methods in unconstrained environments, Pattern Recognit. 45 (7) (2012) 2445–2459.
[21] R.S. Ghiass, O. Arandjelović, A. Bendada, X. Maldague, Infrared face recognition: a comprehensive review of methodologies and databases, Pattern Recognit. 47 (9) (2014) 2807–2824.
[22] Yale University Face Image Database public site (Available on June 5th, 2012): ⟨http://cvc.yale.edu/projects/yalefaces/yalefaces.html⟩.
[23] BioID Face Database public site (Available on June 5th, 2012): ⟨http://www.humanscan.de/support/downloads/facedb.php⟩.
[24] AR Face Database public site (Available on June 30th, 2010): ⟨http://cobweb.ecn.purdue.edu/~aleix/aleix_face_DB.html⟩.
[25] P.J. Flynn, K.W. Bowyer, P.J. Phillips, Assessment of time dependency in face recognition: an initial study, in: J. Kittler, M.S. Nixon (Eds.), Audio-and Video-Based Biometric Person Authentication, Springer, Berlin, Heidelberg, 2003, pp. 44–51.
[26] Yale Face Database B. Public site (Available on June 30th, 2010): ⟨http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html⟩.
[27] PIE Database. Basic information in (Available on June 30th, 2010): ⟨http://www.ri.cmu.edu/projects/project_418.html⟩.
[28] T. Huynh, R. Min, J.L. Dugelay, An efficient LBP-based descriptor for facial depth images applied to gender recognition using RGB-D face data, in: Proceedings of Computer Vision-ACCV 2012 Workshops, Springer, Berlin, Heidelberg, 2013, pp. 133–145.
[29] K. Choi, K.-A. Toh, H. Byun, Realtime training on mobile devices for face recognition applications, Pattern Recognit. 44 (2) (2011) 386–400.
[30] S. Chakraborty, V. Balasubramanian, S. Panchanathan, Generalized batch mode active learning for face-based biometric recognition, Pattern Recognit. 46 (2) (2013) 497–508.
[31] D. Rim, M.K. Hasan, F. Puech, C.J. Pal, Learning from weakly labeled faces and video in the wild, Pattern Recognit. 48 (3) (2015) 759–771.
[32] C. Sanderson, B.C. Lovell, Multi-Region Probabilistic histograms for robust and scalable identity inference, Lect. Notes Comput. Sci. 5558 (2009) 199–208.

[33] C. McCool, S. Marcel, A. Hadid, M. Pietikäinen, P. Matějka, J. Černocký, N. Poh, J. Kittler, A. Larcher, C. Lévy, D. Matrouf, J.F. Bonastre, P. Tresadern, T. Cootes, Bimodal person recognition on a mobile phone: using mobile phone data, in: Proceedings of IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2012, pp. 635–640.
[34] Y. Wong, S. Chen, S. Mau, C. Sanderson, B.C. Lovell, Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2011, pp 74–81.
[35] M. Demirkus, J.J. Clark, T. Arbel, Robust semi-automatic head pose labeling for real-world face video sequences, Multimed. Tools Appl. 70 (1) (2014) 495–523.
[36] A. Hadid, M. Pietikäinen, Combining appearance and motion for face and gender recognition from videos, Pattern Recognit. 42 (11) (2009) 2818–2827.
[37] P. Viola, M. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[38] R. Lienhart, A. Kuranov, V. Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, Lect. Notes Comput. Sci. 2781 (2003) 297–304.
[39] P. Viola, M. Jones, Fast and robust classification using asymmetric adaboost and a detector cascade, in: T.G. Dietterich, S. Becker and Z. Ghahramani (Eds.), Advances in Neural Inform. Processing System 14, MIT Press, 2002.
[40] R. Verschae, J. Ruiz-del-Solar, M. Correa, A unified learning framework for object detection and classification using nested cascades of boosted classifiers, Mach. Vis. Appl. 19 (2) (2008) 85–103.
[41] R. Verschae, Object detection using nested cascades of boosted classifiers: a learning framework and its extension to the multi-class case (Ph.D. thesis), Universidad de Chile, 2010. Available at ⟨http://rodrigo.verschae.org/files/Thesis_Rodrigo_Verschae_final.pdf⟩.
[42] T. Ahonen, A. Hadid, M. Pietikäinen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern. Anal. Mach. Intell. 28 (12) (2006) 2037–2041.
[43] J. Zou, Q. Ji, G. Nagy, A comparative study of local matching approach for face recognition, IEEE Trans. Image Process. 16 (10) (2007) 2617–2628.
[44] M. Correa, J. Ruiz-del-Solar, I. Parra-Tsunekawa, A virtual environment for realistic testing and training of face detection and recognition systems, in: Proceedings of IEEE RO-MAN, 2010, pp. 69–75.
[45] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikäinen, X. Chen, W. Gao, WLD: a robust local image descriptor, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1705–1720.
[46] M. Bereta, P. Karczmarek, W. Pedrycz, M. Reformat, Local descriptors in application to the aging problem in face recognition, Pattern Recognit. 46 (10) (2013) 2634–2646.
[47] ORL face database. AT&T Laboratories, Cambridge, U.K. [Online]. ⟨http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html⟩.