PEOPLE GROUPS ANALYSIS FOR AR APPLICATIONS

M. Mancas[1], *S. Laraba*[1], *A. Bandrabur*[1], *P.-H. De Deken*[1]
K. Hagihara[2], *N. Leblanc*[2], *S. B. Yengec Tasdemir*[3], *B. Macq*[2], *T. Dutoit*[1]

[1] NumediArt Institute, TCTS Lab - University of Mons, Belgium
[2] Belgium, ICTEAM - Université Catholique de Louvain, Belgium
[3] Abdullah Gul University, Turkey

ABSTRACT

Automatically characterizing groups and crowds of people plays an important role in different domains such as psychology, architecture or entertainment. In the engineering field, people grouping is important in social signal processing but also videosurveilance. Our challenge is to introduce such solutions into the field of augmented reality, where people get added content on real groups of people.

As a preliminary work, this paper presents a system which provides a live visual feedback in virtual reality augmenting real groups of people with added information such as their ID, group ID or group coherence. The scene was analyzed with mainly one classical RGB camera and enhanced with a smartphone that a holder points towards the other persons.

This paper goes through the proposed system, which is capable of tracking people, performing people grouping, analyzing groups and augmenting those groups in a virtual world.

The first qualitative results show the feasibility of an augmented crowd environment and provide a set of interesting practical insights on the different modules of this system in the context of real-life scenes.

***Index Terms*—** People behavior, Social Signal Processing, People grouping & tracking, VR, AR.

1. INTRODUCTION & CONTEXT

People groups behavior analysis is crucial in social signal processing needed in several activity areas such as public space management, video surveillance, museums, smart cities, psychology, etc. The current research comes into the context of amusement parks where groups of people (families, friends, ...) are very important to characterize.

However, obtaining the exact position and the behavior of every person based only on non-intrusive video sensors is very complex in real-life scenarios due to noise and uneven illumination, people crossing, complex backgrounds, etc. Without a correct people position, it is very complicated to provide precise group analysis. While previous works [14] [16] [13] [17] [3] provide some hints in this field, results are global and not very precise. The latest deep neural network (DNN)-based developments in people detection such as OpenPose [2], dramatically changed the people tracking and analysis capabilities. OpenPose uses learning to detect body parts and then links those body parts in a skeleton. It does not only work on RGB cameras but also on IR monochrome images. The skeletons are detected by default in 2D; however, later versions can retrieve in 3D by using several cameras. The algorithm is stable and efficient when it is applied on groups of people even in difficult conditions such as crowds. The processing frame rate does not decrease dramatically even with the scene of many people, and the performance of calculating the number of detected persons and 2D skeleton quality is better than the one extracted using the Kinect v2 SDK despite that the latter has additional depth information.

In addition to people detection, multi-target tracking has made very important advances [15] enabling very stable people tracking. Finally, long-range and wide-angle depth cameras [1] have also arrived on the market providing good chances to acquire depth information in complex situations.

These modern technologies allow analyzing real-life situations with several people including crowds in a robust way. This evolution widely opens new research possibilities in social signal processing such as people relative position and group behavior which become accessible for analysis. This paper intends to fill a gap in this area by analyzing people groups based on video input and to extract individuals as well as groups behavior. The use of augmented reality can be a very interesting tool for result interaction and validation.

In the next section, we describe a global overview of the proposed system consisting of different modules. In section 3, we explain our people tracking algorithm in more detail. Section 4 deals with the people grouping and their feature analysis based on the people tracking results and also on face information. Section 5 presents the user feedback and crowd augment applications. Finally a discussion concludes the paper giving also some practical insights learnt from our first qualitative tests.

[1] https://www.stereolabs.com/zed/

978-1-5386-7590-8/18/$31.00 ©2018 IEEE

2. GLOBAL FRAMEWORK

2.1. Setup & scenario description

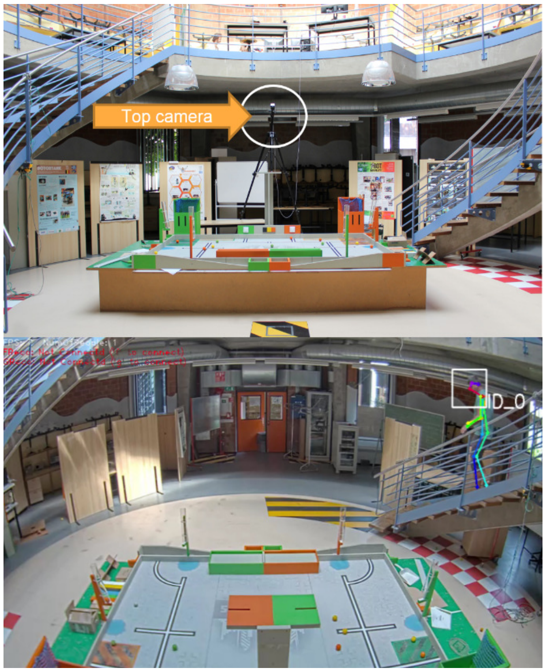

Fig. 1. A top-camera provides a wide-angle view for both people detection and face recognition. A second camera zooming more on the faces was needed for emotion detection.

The area covered by the camera is large enough to contain dozens of people at different distances from the camera (details on Fig. 1). This first attempt focused on the tracking and avoided to introduce further complications due to a multi-camera setup, this is why only one camera is used for people tracking and re-identification and it was placed at 3 m from the ground to have a view which avoids most of the people occlusions but to still see people from the side to be able to use their skeleton. We installed another camera at the level of people faces to extract human emotions which needed a better face resolution. The background is very complex to simulate real-life group interactions.

The scenario of the test was the following :

- Several people move in the scene during 5 minutes and form groups
- A new person comes in with a smartphone and points towards the others. A 3D scene shows dragon-like avatars instead of the people along with their individual and group ID.

The 3D scene on smartphone is interesting both for visual feedback reasons and for validation reasons. This user could indeed take notes of the personal and group IDs of the people he points his smartphone to.

2.2. Pipeline

The framework pipeline is composed of three main modules. The first one takes its input from the top camera (see Fig. 2).

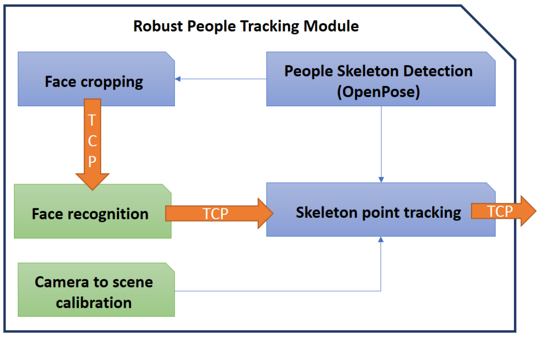
Robust People Tracking Module
Face cropping
People Skeleton Detection (OpenPose)
T C P
Face recognition
TCP
Skeleton point tracking
TCP
Camera to scene calibration

Fig. 2. Module 1: People detection, tracking and face-based re-identification.

From the images taken by the camera, people skeletons are detected via an OpenPose-based submodule, the tracking of a skeleton point is then conducted and the face recognition is used to identify the tracked person. The camera is calibrated in order to project the pixel position of people on input camera image to the real physical position of the scene.

The second module (see Fig. 3) takes its input from the first module. Through messages from the Module 1 to the Module 2 over a network communication (based on TCP), the people ID and position are transfered. The second module will perform people grouping and assign a group ID to each person, which will be appended to the initial message. This module will also compute group features as the group coherence (see section 4.2).

The personal ID, group ID and group characteristics are then sent to the third module which will focus on user feedback (see Fig. 4).

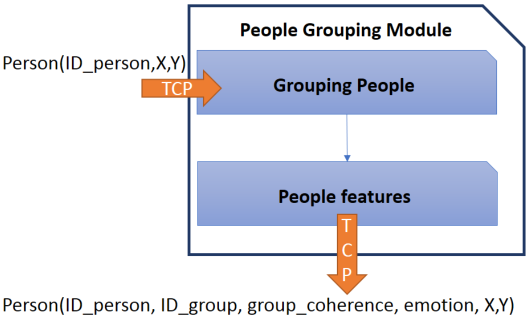

Fig. 3. Module 2: People grouping and feature extraction (position coherence & emotions).

A first part concerns the global feedback to the users which will be described in section 5.1. The second part of this module focuses on providing a 3D feedback on a smartphone.

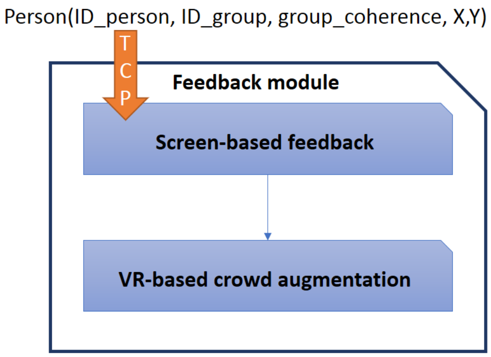
Person(ID_person, ID_group, group_coherence, X,Y)
TCP
Feedback module
Screen-based feedback
VR-based crowd augmentation

Fig. 4. Module 3: Feedback to users.

3. PEOPLE TRACKING

3.1. OpenPose-based tracking

OpenPose [2] is a real-time deep learning-based multi-person system that can jointly detect human body, hands and facial keypoints (in total 135 keypoints) on single images. The algorithm can realize, with a dedicated hardware, multi-person detection on RGB streams in real-time. OpenPose does only make detection frame by frame and does not produce any frame-to-frame tracking allowing to apply a unique ID to each detected person.

In this work, we first have added a tracking layer on top of OpenPose in order to attribute IDs to every detected person. This layer is a simple, yet highly effective object tracking which relies on the Euclidean distance between the neck joint on two consecutive frames. The neck is one of the most stable joints given by OpenPose skeleton. We will keep referring to it as “centroid” though in the rest of the paper.
For every detected person at a frame t we compute the Euclidean distance between each pair of existing centroids and input centroids as illustrated in Fig. 5. The primary assumption of the centroid tracking algorithm is that a given object will move between frames, but the distance between the centroids for frame *t-1* and frame t will be smaller than all other distances. Therefore, we build our people tracker by associating centroids with minimum distances between subsequent frames. In the case where there are more input detections that existing people being tracked, we assign a new ID for the new detected person and we store the position of centroid for the tracking in the next frames.
The main issue of this tracking algorithm is the difficulty to handle occlusions. If two people are crossing their paths, the algorithm may be confused and the IDs may switch. It is also the case when a person crosses an object big enough to hide the neck joint completely (a wall for instance). The algorithm will consider this person when detected again as new person.

For this reason, we use a facial recognition module based on FaceNet [18], to strengthen the tracking and to solve the occlusion problem. This module is detailed in subsection 3.3. The face recognition module is not called at each frame but only every time a new person is detected or when two people are close enough that the tracking gets confused. The given IDs are then limited to the number of people used to train the face recognition module.

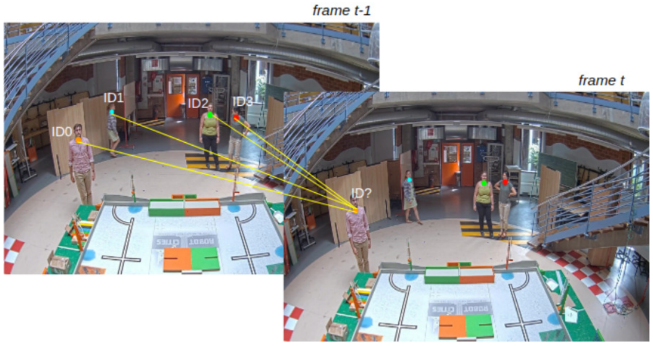

Fig. 5. Illustration of the tracking algorithm. We compute the Euclidean distances between each pair of original centroids at frame *t-1* and new centroids at frame t.

All these modules combined are very resource-consuming, we make use of multi-threading to manage different processes and run them in parallel, particularly video grabbing, detection, tracking and visualization. This way, the speed of processing doubled from 6 to 12 frames per second (FPS).

3.2. From 2D camera to the scene

Once we have detected people, we want to find their position in the scene. We choose to use the 2D projection of a body on the ground as real position.

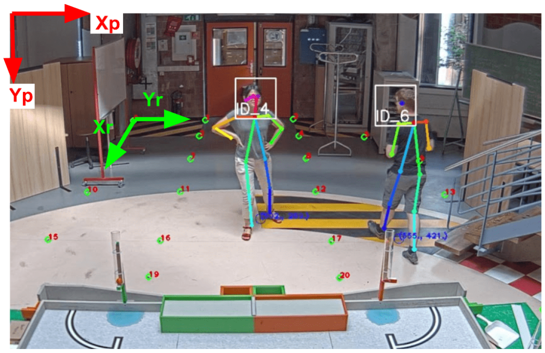

Fig. 6. Camera to scene calibration to convert camera pixels into scene-related coordinates.

The goal is to estimated 2D real coordinates corresponding to the ground plane from 2D pixel coordinates corresponding to the image plane. To realize this step, we use the planar homography [19]. Homography relates the transformation between two planes, this transformation has 9 degrees of freedom.

To estimate the homography matrix, we have marked and measured a dozen of points on the ground in the scene (in green on Fig. 6), and we have associated each of these points to the corresponding pixel coordinates in the image from our camera. Since we have more than 9 points, the coefficients of homography matrix are estimated with a simple least-squares scheme. Our results give a maximal error around 10-15 cm.

3.3. Face recognition

In this work, we have decided to use face recognition to complement the tracking system in order to solve the occlusion problems. It is also relevant to use such a system to keep tracking the same people when they leave and come back to the scene. Indeed, in a standard tracking system, if a person leaves the scene and then comes back, the system will not recognize him/her and will consider it as a new person to be tracked. In our architecture, we propose to use face recognition to re-identify people even when they leave and come back the field of view of the camera. We use in this case the FaceNet framework [18].

FaceNet is a one-shot model, that directly learns a mapping from face images to compact Euclidean space where distances directly correspond to a measure of the face similarity. Once this space has been produced, tasks such as face recognition, verification and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors [18]. One-shot learning aims to learn information about object categories from one, or only a few, training images. The model needs to be first trained on large dataset with many faces that can be publicly available (CASIA-WEBFACE [23], MS-Celeb-1M [7], VGGFace2 [1]. . .), and then reused on our data. In our work, we have collected approximately 30 to 40 face images per participant. We retrain then the FaceNet model, that was pre-trained on VGGFace2 dataset and achieved 99.65% of accuracy on this dataset. VGGFace2 dataset consists of 3.3M faces and 9000 classes.

In our framework, several face snapshots of the person are taken and recognized. A majority-based voting is then used to select the ID which is the most recognized among several snapshots. This leads to a more robust face recognition.

4. PEOPLE GROUPING & ANALYSYS

4.1. People grouping

For grouping individuals, there are different methods such as ones based on pairwise proximity and velocity [6], the trajectories of people in 3D space and head poses [20] and the position of the persons in the scene on the ground plane as well as their body orientation [21].

Following our context, we consider that group members are not constantly close to each other but this assumption is true in long term. Indeed, they tend to follow each others during their activities. As we have inputs of 2D trajectories of individuals in the scene, our approach is based on their distances from each other.

We established three steps analysis for online grouping: detecting spatial groups based on 2D positions of individuals (group detection), associating those groups to the ones estimated previously (group tracking) and then reevaluation of group members by voting process (group reevaluation).

For the group detection, we used MeanShift clustering (Fig. 7) since the number of groups is unknown and MeanShift does not initially require a number of classes.

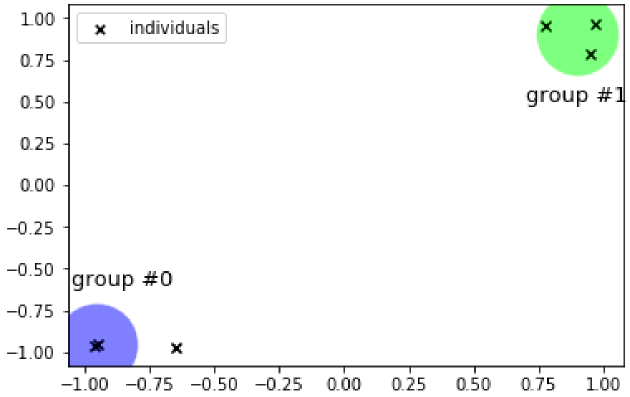

Fig. 7. Spatial grouping result using MeanShift clustering.

The detected groups could be irrelevant because the corresponding individuals might be just crossing each other or approaching to reach different destinations. This problem will be addressed by a temporal voting process.

The group tracking is realized by matching their members with the groups previously estimated. And finally, we perform group reevaluation in order to choose the most likely group for each individual in long term by voting process. The voting process is carried out over the set of assigned group labels over time, which are examined at the group tracking step. If the person is mainly in a given group during time, this person will be assigned to this group assuming that he spent more time close to his own group than close to the others.

Fig. 8 represents the result of this method from simulated trajectories of six individuals, which start at random positions and move to make two groups at the top-right and the bottom-left corners of the observed scene. At the beginning, group 2 is assigned to two individuals crossing each other but thanks to the voting process, those individuals got relevant group IDs 0 and 1 at the last phase.

4.2. Groups features

4.2.1. Groups coherence

The group collective coherency appears from the local interaction and position between the individuals. Coherent groups tend to move with stable relative positions [24] [22].

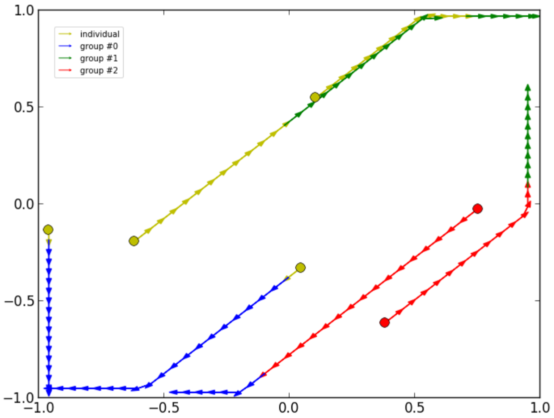

Fig. 8. Grouping from simulated trajectory of individuals.

To measure the coherency of the groups, first of all, the distance between the different persons in a group is computed. The distance information is categorized as personal space (close range), social space (mid-range) and public space (distant range) using people proxemics [8]. For each frame this process is repeated.

Finally, the information of the coherence is extracted by looking at the closeness information variation for each group. For instance, if a group tends to have a very variable intra-personal distance, the group is labeled as not coherent. If the inter-personal distances are stable in time, then the group is labeled as coherent.

For non-coherent groups, no other feature can be extracted, while for coherent groups, the average proxemics can be extracted in addition: close-interaction groups, mid-interaction groups and far-interaction groups. Thus, groups can be either not coherent or coherent with close/mid/far-interactions.

4.2.2. *People emotions*

The purpose of this section is to check the emotional level of a group of people and check its coherence. Previous work [11] already shown how to evaluate group happiness, we will focus here on the group coherence in terms of emotions.

Automatic emotion detection is based on facial expression recognition. The main idea starts from the measurement of the intensity of the facial muscles, so called action units (AU), in order to discriminate between different types of facial actions. There are seven basic emotions that we tried to detect: happy, disgust, contempt, anger, fear, surprise, sad. In order to detect those emotions, we used a deep residual network with 34 layers, ResNet34 [9], which is implemented in fast.ai pytorch framework [2].

This model was already trained on ImageNet database [4]. We fine-tuned it by retraining only the last layer using emotioNet database [5] (Fig.9). The following annotated Action Units provided with emotioNet were used in order to recreate the basic emotions: AU1, AU2, AU4, AU5, AU6, AU9, AU12, AU26. Data augmentation was also performed by applying several random transformations: rotation, zooming, lightning, dihedral, to increase the training dataset and the robustness of the model.

We set the learning rate schedule, to find the optimal learning rate, proposed by Leslie N. Smith [10]. The ADAM optimization algorithm [12] which uses different learning rates for every parameter and momentum was set. To add regularization, dropout in all layers was used, with smaller percentages in the early layers and bigger dropouts in the later layers, to make available as much information from the early layers to the later ones. To avoid over-fitting and for faster training, batch normalization and rectified linear units were performed.

Fig. 9. Facial Expressions from emotioNet. [5]

The first impediment was the image resolution - if for a face recognition a 20x20 pixel-wide face was enough, for emotions we need a minimum of 60x60 pixels for the face. We used progressive image resizing, with small resolutions at the start of the training, and we increased them gradually to 224x224 until the training ends. In this way, we succeed to have a module able to detect emotions at different resolutions.

A model was created for each basic emotion, summing up to 7 models and it combined their results to have one module which was able to detect all basic emotions and their confidence scores. The obtained results are displayed in table 1.

Emotion	Accuracy	Samples	AUs
Happy	95%	4000	6, 12
Disgust	95%	1140	9
Contempt	95%	14000	12
Anger	89%	6000	4
Fear	91%	1200	2, 5
Surprise	86%	2000	1, 26
Sad	87%	2000	1, 4

Table 1: Display the results for each emotion, the number of samples used for training and the Action Units used to recognize specific emotions.

We can see that for Disgust, which had a quite small dataset, of only 1140 samples (50% positive and 50% negative) we had a small accuracy 87% as we encoded only the Action Unit AU9, which represents the nose wrinkle, while the complete

[2] http://www.fast.ai

encoding of Disgust is composed by much more action units (AU2,AU4,AU9,AU15,AU17). For Contempt due to the big dataset 14.000, even though we encoded just one Action Unit AU12 we had an accuracy score of 95%.

Although the dataset contains images with facial expressions of emotion in the wild, when we tested the module in real-time on small and bad quality images, the only correctly detected emotion was happiness. This shows the difference between the current emotion research which are based on datasets and real life with its difficult head positions and noise. In our case, as in [11] or in the Kinect SDK people emotions recognized in the wild only focus on the binary distinction between happiness or not.

5. USER FEEDBACK

5.1. Screen Feedback

As the scenario (see section 2.1) involves a 5 minutes time to form groups, we realized that this is not a simple task for people who are not necessarily familiar. A screen-based feedback of people position and interactions was thus proposed to give people a common information basis so that they can interact and more easily form their groups.

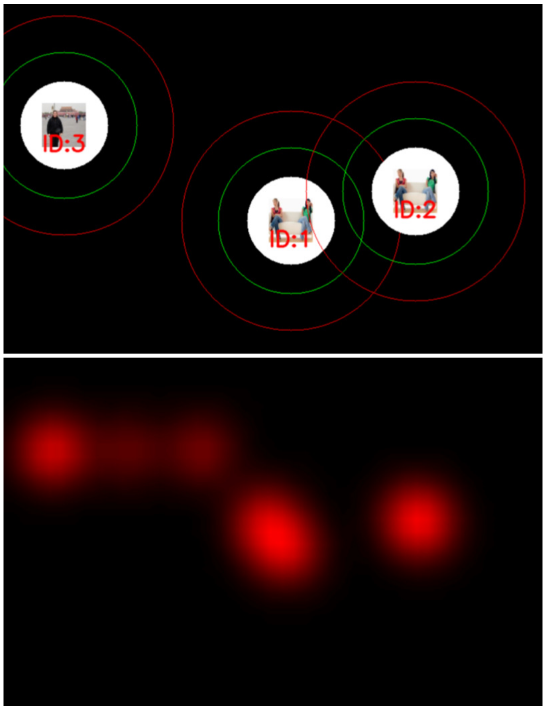

Fig. 10. Top: 3 people with their IDs and information about their personal space (one is far from the two others which personal spaces overlap). Bottom: Spatial occupation heatmap (evolves in time).

As it can be seen in Fig. 10 and 11, two visualizations are provided. The first one shows people proxemics and interpersonal distances depending on their position with a different image in the middle showing personal distances, social distances or public distances. The second visualization is a heatmap of scene occupation showing the areas the most attended by people.

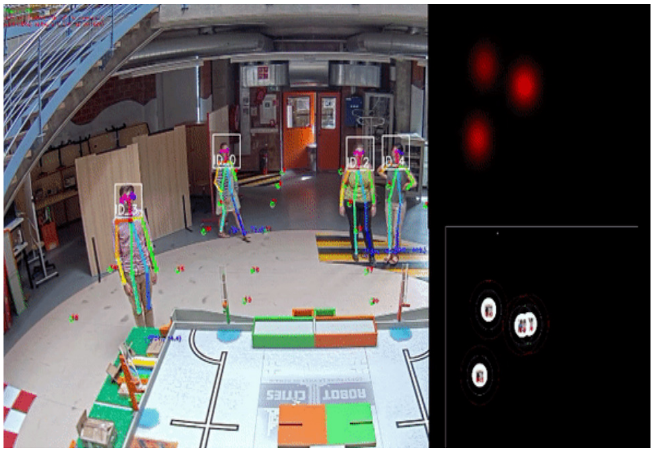

Fig. 11. Real scene view from camera (left), top-view after calibration of occupation heatmap and personal spaces configuration.

5.2. VR Feedback

We created with Unity [3] which is a widely used 3D engine a virtual reality (VR) application embedded on a smartphone. This application is a feedback as well as a validation tool for all the previous analysis pipeline. The user that looks through the smartphone pointing to a precise person is able to see a virtual scene that mimics the real world. At the location of the targeted person, the application shows 3D avatars (here a dragon-like avatar was implemented). All localization data (position of the different persons and the smartphone's holder position) are received from the tracking module through a network TCP connection. Thanks to this positioning data the point of view can be set inside the application (See VR scene in Fig. 12).

It is possible to position the user at the right place inside the application but to point adequately the virtual scene along the Euler's angles, the smartphone's gyroscope was used to get the direction of the gravity vector. From it, the pitch and roll of the smartphone can be computed. Regarding the smartphone yaw, the data obtained from its compass is used.

In that way the smartphone holder is able to point any person and see his dragon-like avatar close to which he can see some additional data such as its personal and group IDs. This is interesting as feedback but also to validate the tracking as this person can take notes of when the ID of a person changes due to tracking issues.

[3] https://unity3d.com/

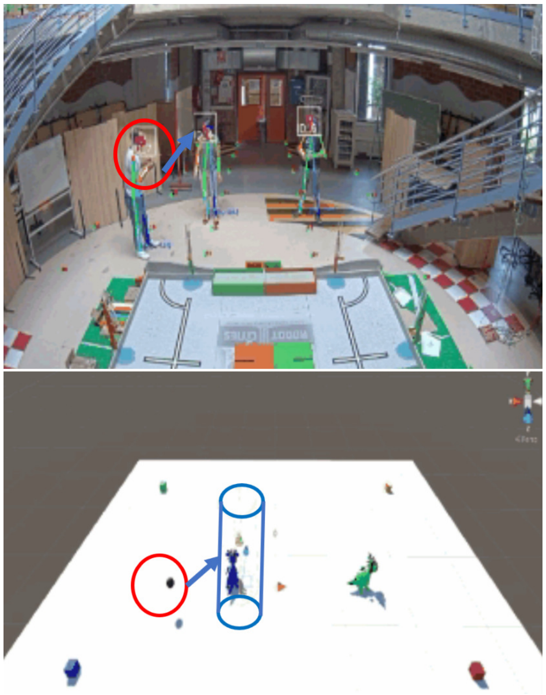

Fig. 12. Real scene with a user having a smartphone (on top). Simulation of the smartphone direction (bottom).

6. CONCLUSION & DISCUSSION

In this paper, a complete framework for augmenting people groups was presented. Based on one camera, the people tracking and re-identification provides solid data for people grouping and analysis. This data is then shown in a virtual world close to each person whom a smartphone is pointed towards. Three different modules are described here to show all the process from image acquisition to the 3D scene augmentation.

The first qualitative results show some interesting insights:

- People detection based on deep learning is now very efficient.
- Face-based re-identification is good enough by using only around 30 images for training which is about some seconds face recording in real-life scenarios (including face not visible, changing frame rates, etc.).
- Face-based re-identification is better when comparing several face snapshots on different frames and using a majority voting system. This avoids miss-recognition due to noisy face snapshot in real-life scenarios.
- Emotion identification shows difficulties in real-life scenarios. Only happiness was detected on small images and a minimum of 60x60 pixels face size is necessary to get reasonable results on the other emotions.
- If the personal tracking is correct, the grouping and coherence are quite easy to extract in an efficient way.
- It is possible to use together the position of a smartphone and people around along with the smartphone compass and gyroscope to correctly make a correspondence between a VR scene on the real scene.

Future work consists first in the framework quantitative validation based on observation of the smartphone holders to check people ID, group ID and group features. Second, other features can be added to group coherence such as group energy or group attention direction (motivation). Finally a better knowledge about people emotions, age or sex can help in characterizing them in a more precise way.

7. ACKNOWLEDGMENTS

This research was carried out in the context of eNTERFACE'18 workshop and the Parkar/DGO6 project (#1610130) funded by the Région Wallonne WALINNOV and Alterface S.A.

8. REFERENCES

[1] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pages 67–74. IEEE, 2018.

[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *2017, IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] Allan De Freitas, Lyudmila Mihaylova, Amadou Gning, Donka Angelova, and Visakan Kadirkamanathan. Autonomous crowds tracking with box particle filtering and convolution particle filtering. *Automatica*, 69:380–394, 2016.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[5] C Fabian Benitez-Quiroz, Ramprakash Srinivasan, and Aleix M Martinez. Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5562–5570, 2016.

[6] Weina Ge, Robert T. Collins, and Barry Ruback. Automatically detecting the small group structure of a crowd.

2009 Workshop on Applications of Computer Vision (WACV), 2009.

[7] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016.

[8] Edward T Hall, Ray L Birdwhistell, Bernhard Bock, Paul Bohannan, A Richard Diebold Jr, Marshall Durbin, Munro S Edmonson, JL Fischer, Dell Hymes, Solon T Kimball, et al. Proxemics [and comments and replies]. *Current anthropology*, 9(2/3):83–108, 1968.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Javier Hernandez, Mohammed E Hoque, and Rosalind W Picard. Mood meter: large-scale and long-term smile monitoring system. In *ACM SIGGRAPH 2012 Emerging Technologies*, page 15. ACM, 2012.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Matei Mancas. Attention-based dense crowds analysis. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, pages 1–4. IEEE, 2010.

[14] Matei Mancas, Nicolas Riche, Julien Leroy, and Bernard Gosselin. Abnormal motion selection in crowds using bottom-up saliency. In *2011 18th IEEE International Conference on Image Processing (ICIP)*, pages 229–232. IEEE, 2011.

[15] Anton Milan, Seyed Hamid Rezatofighi, Anthony R Dick, Ian D Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *AAAI*, volume 2, page 4, 2017.

[16] Nicolas Riche, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. 3d saliency for abnormal motion selection: The role of the depth map. In *International Conference on Computer Vision Systems*, pages 143–152. Springer, 2011.

[17] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and Jean-Yves Audibert. Density-aware person detection and tracking in crowds. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2423–2430. IEEE, 2011.

[18] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[19] Richard Szeliski. Computer vision: Algorithms and applications. pages 29–60, 2010.

[20] K. N. Tran, A. Gala, I. A. Kakadiaris, and S. K. Shah. Activity analysis in crowded environments using social cues for group discovery and human interaction modeling. *Pattern Recognition Letters*, 44:49–57, 2014.

[21] Sebastiano Vascon, Eyasu Z. Mequanint, Marco Cristani, Hayley Hung, Marcello Pelillo, and Vittorio Murino. Detecting conversational groups in images and sequences: A robust game-theoretic approach. *Computer Vision and Image Understanding*, 143:11–24, 2016.

[22] Sebastiano Vascon, Eyasu Z Mequanint, Marco Cristani, Hayley Hung, Marcello Pelillo, and Vittorio Murino. Detecting conversational groups in images and sequences: A robust game-theoretic approach. *Computer Vision and Image Understanding*, 143:11–24, 2016.

[23] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[24] Bolei Zhou, Xiaoou Tang, and Xiaogang Wang. Coherent filtering: detecting coherent motions from crowd clutters. In *Computer Vision–European Conference on Computer Vision (ECCV) 2012*, pages 857–871. Springer, 2012.