Measuring Gaze Orientation for Human-Robot Interaction

R. Brochard*, B. Burger*, A. Herbulot*†, F. Lerasle*†

$*$ CNRS; LAAS; 7 avenue du Colonel Roche, 31077 Toulouse Cedex, France
† Université de Toulouse; UPS; LAAS-CNRS : F-31077 Toulouse, France

1 Introduction

In the context of Human-Robot interaction estimating gaze orientation brings useful information about human focus of attention. This is a contextual information : when you point something you usually look at it. Estimating gaze orientation requires head pose estimation. There are several techniques to estimate head pose from images, they are mainly based on training [3, 4] or on local face features tracking [6]. The approach described here is based on local face features tracking in image space using online learning, it is a mixed approach since we track face features using some learning at feature level. It uses SURF features [2] to guide detection and tracking. Such key features can be matched between images, used for object detection or object tracking [10]. Several approaches work on fixed size images like training techniques which mainly work on low resolution images because of computation costs whereas approaches based on local features tracking work on high resolution images. Tracking face features such as eyes, nose and mouth is a common problem in many applications such as detection of facial expression or video conferencing [8] but most of those applications focus on front face images [9]. We developed an algorithm based on face features tracking using a parametric model. First we need face detection, then we detect face features in following order: eyes, mouth, nose. In order to achieve full profile detection we use sets of SURF to learn what eyes, mouth and nose look like once tracking is initialized. Once those sets of SURF are known they are used to detect and track face features. SURF have a descriptor which is often used to identify a key point and here we add some global geometry information by using the relative position between key points. Then we use a particle filter to track face features using those SURF based detectors, we compute the head pose angles from features position and pass the results through a median filter. This paper is organized as follows. Section 2 describes our modeling of visual features, section 3 presents our tracking implementation. Section 4 presents results we get with our implementation and future works in section 5.

2 Visual features

We use some basic properties of facial features to initialize our algorithm : eyes are dark and circular, mouth is an horizontal dark line with a specific color,...

Then we use sets of SURF to learn a better description of visual features. SURF are extracted with their relative position in order to keep geometrical information about features. We use a similarity measure to compare two sets of SURF (see the probability measures in [10]), we detect a feature as the maximum of this function in image space. This function is defined as:

$$\sigma(S_0, S_1) = \sum_{P \in S_0} \sum_{Q \in S_1} \exp\left(-\frac{\|P-Q\|_{2D}^2}{2\sigma_{2D}^2}\right) \times \exp\left(-\frac{\|P-Q\|_{desc}^2}{2\sigma_{desc}^2}\right) \qquad (1)$$

where S_0 and S_1 are two sets of SURF, $\|.\|_{2D/desc}$ are the Euclidean norms in image space (*2D*) and in SURF descriptor space (*desc*). σ_{desc} defines the local tolerance (descriptor space) whereas σ_{2D} defines the geometrical tolerance (image space). Learning a set of SURF is a matter of selecting SURF which are representative of the feature and putting them into the same set with their relative position (you can set the origin to the center of the feature).

3 Tracking implementation

We use a simple face model described by the position of eyes, center of mouth and nose tip similar to the one in [6]. We use 7 parameters (two absolute coordinates, two angles, spacing of eyes, scale, relative abscissa of nose) filtered by an ICONDENSATION particle filter [1]. Likelihood measures and detection are based on the same probability maps, detected features being the maxima of those maps. A particle likelihood is the product of all features likelihood. Maps information is merged in a way that if there is a high probability to have a feature, it decreases the probability to have a different one. Likelihood is computed using both standard properties of tracked features and similarity of SURF sets. We have about 100 SURF stored for mouth, 20 for eyes, it's varying a lot for nose (from 0 to 40) depending on visibility of nostrils and light saturation. Because SURF similarity can be very small, we add a small constant ϵ (0.01) in order not to get a null likelihood for particles that are not so bad. Since we cannot predict head movement, the dynamic model we use is a random walk model, we only expect head moves not to be too fast.

4 Results

We have taken a video sequence of 1140 frames (Fig. 1) to compare the behavior of the algorithm with and without SURF and how the number of particles in the particle filter influences the result. This sequence shows a single person making head moves : pan and tilt rotations as well as fast head movements, both rotations and translations and a simple "look at" movement. For each test, we ran the algorithm 21 times on the same sequence due to the stochastic nature of particle filters. We ran 3 tests with this sequence. The first test with 250 particles without SURF and only 50 particles with SURF. The second test is with 50 particles for both versions of the algorithm. The last tests with 250 particles both with and without SURF.

It is slower to compute particles likelihood with SURF because of similarity calculations involved in each likelihood evaluation. This is the reason why the first test does not compare the two versions of the algorithm with the same number of particles. Without optimizing likelihood calculations the algorithm runs 200 SURF particles as fast as 7000 particles without SURF. An interesting point to note is sets of SURF can learn the shape of a feature as it is varying, as a direct consequence if you close your eyes it can still track them, this is useful when the person is blinking.

The average standard deviation for the 3 tests with SURF is 10.9°for tilt, 14.9°for pan and 4.4°for roll, without SURF we get 10.5°for tilt, 15.8°for pan and 8.09°for roll. We get a smaller standard deviation over several runs with SURF and a small amount of particles than without SURF and with lots of particles. Nose tracking is not working properly yet and all the results are obtained with an ambiguity on the sign of angles which is not always solved properly (the model we use requires nose position to solve this sign ambiguity) which decreases greatly the quality of measures. We ran a few tests with a better nose detector (which uses SURF too) and already got much better results with SURF due to learning being based on the result of the particle filter.

We also ran the algorithm on the Yale B face database [7] on images with frontal light. The Yale B database contains gray scale face images from 10 different persons with various lighting conditions and poses. With SURF we get an average error (over the 10 different faces) of 3.15°for tilt, 10.3°for pan, 1.41°for roll and without SURF we get 4.06°for tilt, 11.1°for pan, 1.5°for roll.

Our current implementation adds a noticeable overhead per particle but this can be optimized in order to have the same overhead per particle as the implementation without SURF. It would be a bit slower only because of the need to extract SURF from the image which takes approximately 8ms during our tests.

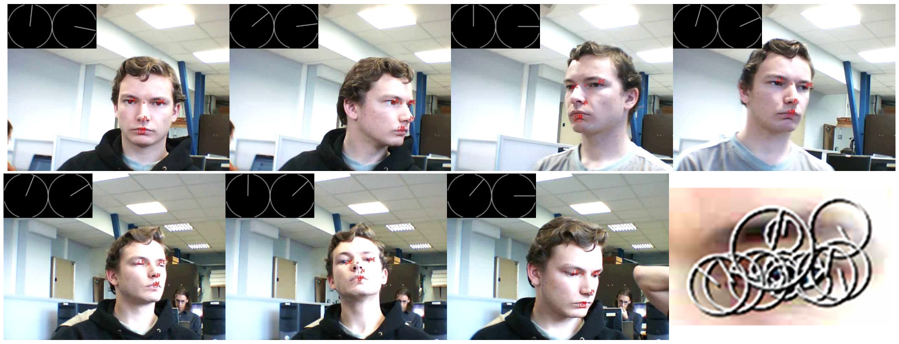

Fig. 1. Video sequences. Last image shows SURF extracted from an eye.

5 Conclusion and future works

Using sets of SURF to detect and track face features makes the process more robust and stable under various poses. It allows detecting and tracking features with varying shapes. Our current implementation of SURF similarity calculations is slow. With some optimization using SURF adds a small overhead and we could use more SURF. Taking face geometry (features relative position) into account could also improve detection robustness. The main idea is to use sets of SURF to learn something to track and learning changes in its shape as it moves. We are currently integrating our algorithm on a robot and have planned to use both human focus of attention and gesture recognition [5] to achieve better human-robot interaction.

Acknowledgments

This work was partially conducted within the EU STREP Project Commrob funded by the European Commission Division FP6 under Contract $FP6-045441$ and the french ANR project AMORCES.

References

1. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2001.
2. H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded-Up Robust Features. In *European Conference on Computer Vision*, pages 404–417, Graz, Austria, 2006.
3. B. Benfold and I. Reid. Colour invariant head pose classification in low resolution video. In *British Machine Vision Conference*, September 2008.
4. L.M. Brown and Ying-Li Tian. Comparative study of coarse head pose estimation. In *Workshop on Motion and Video Computing*, pages 125–130, December 2002.
5. B. Burger, G. Infantes, I. Ferrané, and F. Lerasle. Dbn versus hmm for gesture recognition in human-robot interaction. In *Int. workshop on Electronics, Control, Modelling, Measurement and Signals*, pages 59–65, Mondragon, Spain, July 2009.
6. A. H. Gee and R. Cipolla. Determining the gaze of faces in images. *Image and Vision Computing*, 12:639–647, 1994.
7. A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.
8. J. Luo, C. W. Chen, and K. J. Parker. Face location in wavelet-based video compression for high perceptual quality videoconferencing. In *International Conference on Image Processing*, volume 2, pages 583–586, October 1995.
9. M. Pantic and L. Rothkrantz. Automatic Analysis of Facial Expressions: The State of the Art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, December 2000.
10. H. Zhou, Y. Yuan, and C. Shi. Object tracking using SIFT features and mean shift. *Computer Vision and Image Understanding*, 113(3):345–352, 2009.