FINGER DETECTION IN VIDEO SEQUENCES USING A NEW SPARSE REPRESENTATION

Vasile GUI[1], Gheorghe POPA[1], Pekka NISULA[2], Veijo KORHONEN[2]
[1]*"Politehnica" University of Timisoara,* [2]*University of Applied Sciences Oulu*
Bv. V. Parvan No. 2, +0256403306, Fax. +0256403295, vasile.gui@etc.upt.ro

Abstract: In this paper we propose a new method for finger detection in video sequences. The method is robust and has a low computational cost. From a background subtracted image, we generate a sparse image representation, based on line strip features. We use a robust and adaptive clustering method, related to the mean shift (MS) to detect and track fingers, as well as to extract finger position parameters from validated clusters. In contrast with the traditional mean shift tracker, based on color histogram, we use shape information to detect and track hand fingers. Our experiments prove that the tracker is accurate and robust to extreme noise tests and partial occlusion. We applied the proposed method in a human computer interface designed to control very large public displays.

Keywords: *Mean shift, sparse, finger tracking by detection, human computer interface.*

I. INTRODUCTION

The ubiquitous presence of computers in modern society is a major driving force of research devoted to design efficient, natural and easy to use human computer interfaces. Using keyboards, mouse or even touch screen interfaces to interact with information screens, robots or intelligent domestic appliances is considered inconvenient in more and more circumstances. Human computer interface (HCI) can be based on information extracted from hands, arms, body, facial expression, head motion, eye tracking and so on. Good surveys of the research in HCI can be found in [1]-[3]. In this paper, we restrict our attention to hand gestures. We address the problem from the perspective of a system that uses non-expensive video cameras, like those currently available on laptops or mobile phones. Our system uses finger detection and tracking, to generate position and trajectory information. Index fingertip trajectories define dynamic gestures.

We believe that the performance of a HCI is heavily influenced by a few key factors, like the choice of features and the design of gesture set. The work reported in this paper is focused on the first problem. We propose a new solution for finger detection and tracking. It has been found from early work on hand gesture recognition [4] that fingers can be detected and tracked reliably. To detect and track fingers, we use inexpensive line strip features extracted from segmented foreground images. For both tasks we use robust estimation techniques and design computationally efficient algorithms. In contrast to previous work, we use a MS related clustering algorithm to detect and to track finger *shapes*. Our approach belongs to a recent trend in tracking called tracking by detection [5], [6].

The rest of the paper is organized as follows. Section II reviews relevant results of the research in the approached domain. The proposed method is presented in section III and the results of our experiments are the subject of section IV. A discussion of the results and relevance of the work concludes the paper.

II. RELATED WORK

Color information is perhaps the most widely used feature for hand detection. Remarkably, skin hue has relatively low variation between people and is invariant to moderate illumination change. Unfortunately, hue information is unreliable at low illumination levels and for objects which are achromatic or have low saturations. Background subtraction is a fast and powerful technique used in video segmentation [7]-[10] and can be used effectively with static cameras. This technique can be designed to deal with illumination changes, although problems remain with fast and high amplitude changes. Shape information was used to detect hands in many ways. Edges concentrate well the hand shape information, but many edges can be generated by background objects and hand texture. Post-processing or combined methods which use color or background information can improve significantly the results of edge detection. Hand contours can be obtained from edges or directly from segmentation. Recognizing hands from contours remains challenging even with accurate contours, because of the huge variability of hand postures. Edge detection errors make the problem more difficult. Such errors may often be the result of a motion blur. Advanced hand segmentation methods combine hand segmentation with tracking and recognition in joint approaches [11]-[12].

Fingers and finger tips have more simple and stable shapes and therefore can be detected and tracked more reliably than hands. Their usefulness is best in applications where the hands are used to point [13], since occlusions are less frequent in such cases. Pose information can be extracted from finger data very easily and used to define both static and dynamic

Manuscript received November 15, 2010; revised April 1, 2011

gestures. Mathematical morphology and related approaches are the commonly used approach to detect fingers or finger tips [14]-[16]. It was found in [15] (and our experience confirms) that gestures executed with only one or two fingers can be recognized the most reliably, while allowing a very large set of gestures to be defined and recognized.

III. PROPOSED METHOD

In order to succeed, object detection, recognition or tracking tasks need realistic object models, which suit the applications at hand. Our object model consists of several parameters, characterizing the shape of the fingers. The main finger used in this work to generate gestures is the index. Our finger based human computer interface does not aim at recognizing gestures used in sign language. That problem is much harder and the results of systems addressing sign language recognition are not yet reliable enough for most real world applications. Our hand gesture based human computer interface was designed to work with straight fingers. While this restriction allows natural hand postures, it contributes a lot to make finger detection simpler and, most important, more reliable. A block diagram of our HCI is given in Figure 1.

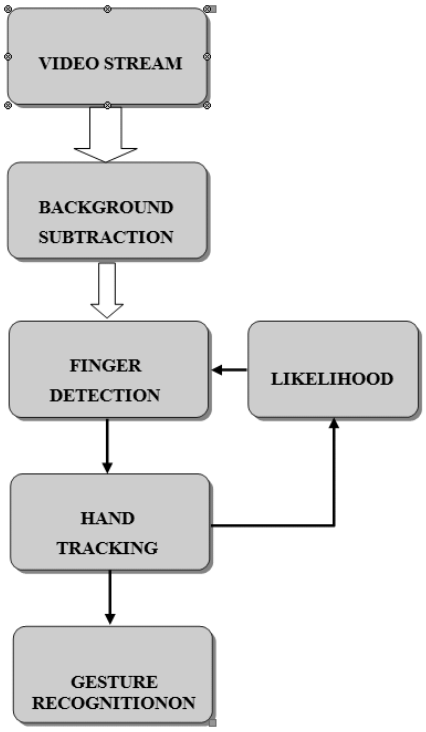


Figure 1. Block diagram of the proposed HCI.

We first extract foreground objects from the video stream by using a background subtraction technique [10]. Background subtraction is an important step towards hand segmentation, resulting in considerable reduction of the processing data.

The most dramatic reduction of data is obtained in the next step, consisting of finger feature extraction. A new sparse image representation, consisting of line strip features, which is able to capture relevant information needed, is used as input to the finger detection layer.

To detect fingers, we use very simple yet thoughtfully chosen features: line strips, obtained by horizontal or vertical scanning of the segmented foreground (Fig.2). Strips belonging to the same finger have similar length and thickness and their centers belong to a curve which can be well modeled as a line segment. Therefore, to find fingers, we look for strips with similar lengths, thickness and with centers on the same line. To find out the slope, φ, of the line connecting two strips, we use the coordinates of their centers, (x, y). For a horizontal strip, we look for a pair strip displaced vertically with its own length, l. Conversely, for vertical strips we look for horizontal neighboring strips. Strips without such neighbors are discarded from further processing. Additionally, strips with positions to far from the current position found by the tracker are discarded. This primary data filtering step is accomplished through the likelihood processing block, placed in the feedback loop. From the remaining strips, the fingers are extracted through a clustering method. Data clustering is the ultimate solution for robust detection, able to break down the 50% outlier data limit of most robust estimators.

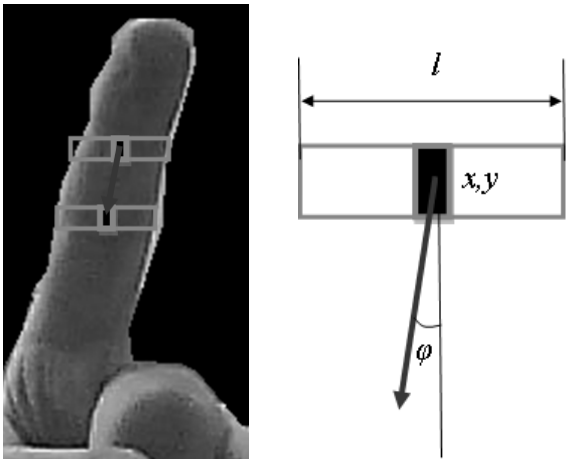


Figure 2. Line strip features extracted for finger detection.

Features belonging to the same finger tend to form clusters in the feature angle space. Therefore fingers can be detected by clustering. Among the many clustering methods, the MS algorithm [17] is one of the most powerful and well founded theoretically. Starting from any point in the feature space, the mean shift algorithm makes steps in the direction of the probability density gradient in the feature space. The MS is a gradient ascent algorithm which stops when a density maximum is reached, since the gradient at such points is zero. It is theoretically related to the family of M estimators from the field of *robust statistics*. In this work, we use the mean shift algorithm to find the maximum likelihood estimate of the finger angle as well as to further extract valid strips. To alleviate the risk of being trapped in local maxima, we use our multi-scale version of the algorithm, called MSMF [18].

Starting from an initial center point, $\vec{f}^{(0)}$, the MSMF algorithm iterates the following equation:

$$\vec{f}^{(t+1)} = \frac{\sum_i \vec{f}_i g\left(\left\|\frac{\vec{f}^{(t)} - \vec{f}}{h_{(t)}}\right\|^2\right)}{\sum_i g\left(\left\|\frac{\vec{f}^{(t)} - \vec{f}}{h_{(t)}}\right\|^2\right)} \quad . \tag{1}$$

Each new center point is computed as a weighted average of all available feature points. The weights quantify the influence of each feature point on the location of the next center point and are computed by means of the kernel profile, $g()$. Features are weighted by a scale parameter, h, which controls the smoothing effect of the kernel.

In contrast to the MS algorithm, in MSMF the scale parameter varies during iterations, allowing a course to fine approach which proved to be beneficial in avoiding false maxima. The final point reached is the maximum probability density point. Its *location* is our solution.

We use the clustering algorithm in the φ feature subspace to detect finger strips and to extract finger pose information. Only strips with lengths l_m within a 20% limit from the currently estimated finger thickness are considered as valid data. Since we start from a good initialization, obtained from the angle histogram with bins corresponding to 10^o, convergence can be reached in a few steps. In all our experiments, two steps proved to work well.

Fingers are detected by a cluster validation method. A cluster is valid if the ratio between the hypothetic finger length and thickness fits predetermined limits. For vertical strip clusters, the validation equation is:

$$th_{\min} < \frac{\sum_i g\left(\left\|\frac{\hat{\vec{f}} - \vec{f}_i}{h}\right\|^2\right)}{l_m \cos(\frac{\pi}{2} - \hat{\varphi})} < th_{\max} \tag{2}$$

In equation (2), "^" stands from estimated values at convergence, l_m is the estimated finger thickness and $g()$ is the Epanechnikov kernel profile. We found by experiments that $th_{\min} = 2$ and $th_{\max} = 6$ work well. In the current implementation, the user sees his/her image on the screen and starts by fitting his/her gesturing (index) finger in a starting box. This enables the detector to find a good initialization for the finger thickness and the fingertip position. Later on, these parameters are obtained from the output of the finger detection and used in the finger tracking process.

The main results of finger detection are the parameters of the valid clusters (detected fingers). In our HCI application, the gestures are done with only two fingers: the index and the thumb. Therefore we select the longest two fingers detected. Additionally, clustering serves to define the inliers and the corresponding strips. From this filtered set of strips, we compute the (x,y) coordinates of the index finger tip, its direction, φ and its mean thickness, l_m. The parameters used for gesture recognition are x,y and φ.

IV. RESULTS

We made tests in order to evaluate the stability, accuracy and practical usability of our HCI. All experiments were made with 640×480 resolution web cameras. The stability of our HCI against occlusion is demonstrated in Figure 3, representing image frames 31, 35, 39, 43, 47 and 51 from a test sequence. No noticeable effect of the occluding finger on the detected fingertip position, marked with the ellipse, can be observed.

Figure 3. Occlusion test sequence frames

The accuracy of the finger tracker was tested in static and dynamic circumstances. Static tests were made on a wooden stick, fixed on a pedestal. Table 1 reproduces the standard deviations of the x, y and φ parameters of the stick, while Figure 4 represents a 3D plot of the experimental data. The Z coordinate is the angle. It is close to 90^o. Note that the verticality of the stick was not measured by alternative methods. The space coordinates are expressed in pixel units and the angle in degrees. While the results are suitable for practical purposed, we believe they depend heavily on camera noise and background business. We did not repeat this test with high quality cameras.

Table 1. Standard deviations of finger parameters.

	x	y	Φ
Standard deviation	0.6776	0.5830	2.1245

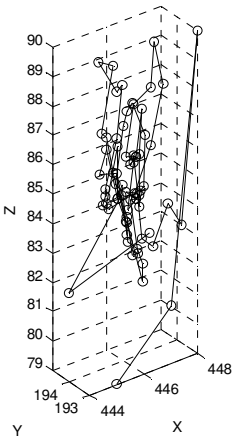


Figure 4. Static test finger parameter x, y and φ data plot.

To test the tracker accuracy in dynamic conditions, with background truth available, we generated a synthetic test sequence with a randomly moving rectangular "finger" on a noisy background. An image from the test sequence is shown in Figure 5. The speed of the stick was changed randomly within the interval [0-20] pixels. Horizontal and vertical noise strips with randomly generated *sizes* have the same color with the finger and cover 50% of the background. This kind of disturbance is much more difficult for a tracker than a random white noise.

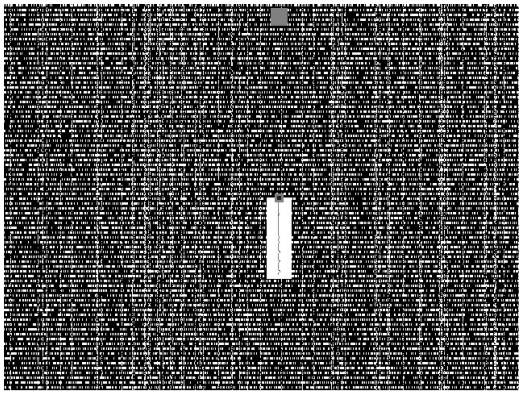
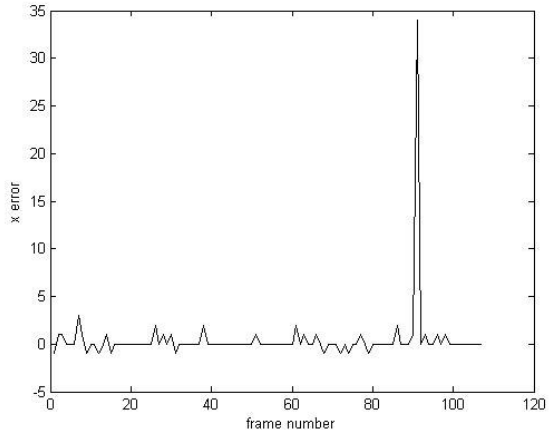


Figure 5. An image frame from the dynamic accuracy test sequence

The results of this test are shown in Figure 6 and Table 2. To our best knowledge, no other trackers have been tested in so heavy noise conditions. Yet, the target is lost only for a very short time and recovers in a few frames. Note that the vertical coordinate error is zero in all tracked frames (except during target loss). The overall accuracy for the tracked frames is at the sub-pixel level.

Table 2. Standard deviations of the tracking errors in all frames and in first 80 frames of the dynamic test sequence.

	X	*Y*
Standard deviation all frames	3.3374	7.0572
Standard deviation frames 1-80	0.7004	0.0000



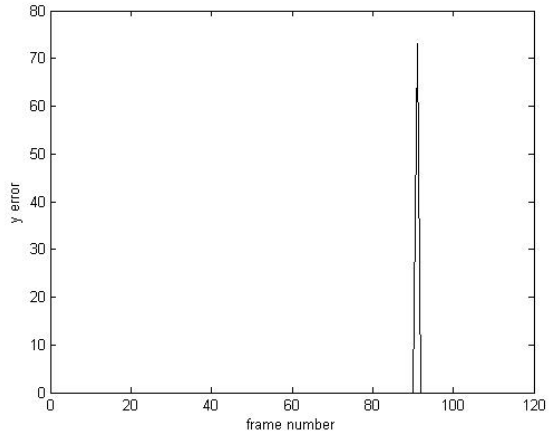


Figure 6.Dynamic test sequence error plot

We do not include in this study results of our experiments with the histogram based mean shift and CAMSHIFT trackers. While these trackers are very robust in following the hand motion, the trajectories generated cannot be controlled accurately by the users. In our opinion, the reason is not a fault of these trackers, rather than a consequence of tracking a *color distribution* instead of a *shape*. As the centre of the ellipse approximating the tracked region depends on too many variables, such as hand tilt and motion of all fingers. Actually disappointing results of these initial experiments motivated us to develop the reported work. In fact, to disambiguate the hand pose problem, people usually point with one finger.

The test bed application that generated in part the work reported in the paper is a hand gesture based control unit for interaction with large public display screens, a part of the UBI (UrBan Interactions) Finnish national research program. An image with our prototype finger pointing information browsing HCI is shown in Figure. 7. The selected image is actually the previously used information unit, with touch screen user interface.

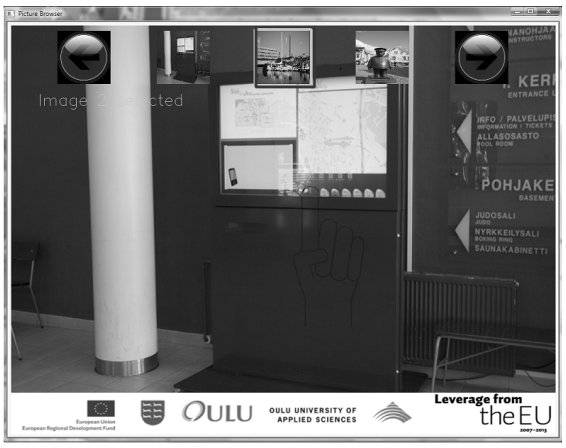


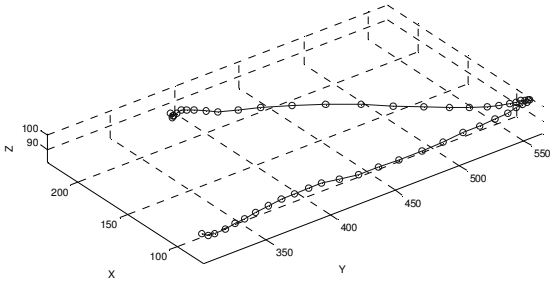
Figure.7. Computer screen of our HCI based information browser

A dynamic gesture 3D data plot obtained with our HCI for gesture "ONE" and its trajectory angle histogram are given in Fig. 8. The trajectory was smoothed with an averaging filter of length 5. This gesture is defined by two strokes with different angles. Each stroke generates a mode in the histogram. Extracting the significant clusters of strokes from this histogram by mode detection and subsequent classification is a trivial task.

On an Intel Core 2 Duo P8400 computer with 2.26 GHz clock, on 640×480 pixel images, our program use on average 5.9 ms per image frame. The most time consuming part is the background subtraction, with 4.5 ms, while the finger estimation and tracking layers use only 1.4 ms. Note however that we did not restrict background subtraction for the tracking ROI, so depending of the ROI size, further computing time can be saved.

V. DISCUSSION AND CONCLUSION

We proposed a finger detection and tracking method, based on a new sparse representation, which is robust, computationally efficient and has a high accuracy. The robustness comes from the clustering approach. Clustering is one of the most powerful approaches coping with data uncertainty. Robustness also comes from the careful selection of the finger features and from using the finger shape as the main information in finger detection and tracking.



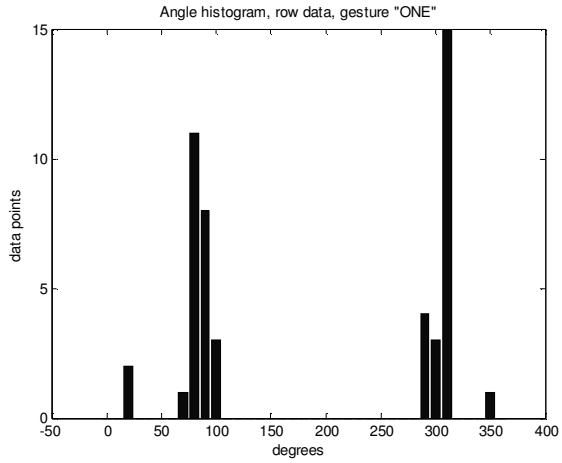


Figure.8. (a) 3D plot of finger pose parameters for gesture "ONE", (b) Trajectory angle histogram for gesture "ONE".

The proposed HCI was used in a data browsing application for public data displays. It can be a solution to replace touch screen interfaces or other interfaces requiring different gadgets. The trajectory data extracted from the finger tracker is smooth and can be easily used to define dynamic hand gestures. The proposed solution needs modest computational power. Therefore, it may be considered for implementation on simpler processors, like those working in mobile applications.

Acknowledgement: this work was supported by grant national ID 931, contr. 651/19.01.2009.

REFERENCES

[1] V. Pavlovic, R. Sharma, and T. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 677-695, 1997.

[2] A. Jaimes and N. Sebe, "Multimodal human computer interaction: a survey", *Computer Vision and Image Understanding*, 108(1-2), pp 116-134 , 2007.

[3] F. Karray, M. Alemzadeh, J. A. Saleh, Mo N. Arab, „Human-Computer Interaction: Overview on State of the Art," *International Journal on Smart Sensing and Intelligent Systems*, Vol. 1, No. 1, pp 137-159, 2008.

[4] J. Crowley, F. Berard, and J. Coutaz,, "Finger Tracking as an Input Device for Augmented Reality," *Proc. IEEE Int'l Workshop Automatic Face and Gesture Recognition* (FG 95), IEEE Press, Piscataway, NJ, pp195-200, 1995.

[5] S. Gu, Y. Zheng, C. Tomasi, "Efficient Visual Object Tracking with Online Nearest Neighbor Classifier". *The 10th Asian Conference on Computer Vision, ACCV* Queenstown, New Zealand,2010.

[6] Santner, J., Leistner, C., Sa_ari, A., Pock, T., Bischof, H., "PROST Parallel Robust Online Simple Tracking", IEEE CVPR. (2010).

[7] C. Stauffer, W. Eric and L. Grimson, "Adaptive background mixture models for real-time tracking," In *Proc. IEEE ComputerVision and Pattern Recognition (CVPR)*, Ft. Collins, USA, pp 2246-2252, June 1999.

[8] A. Elgamal, R. Duraiswami, D. Harwood, L. S. Davis, "*Background and foreground modeling using nonparametric kernel*

density estimation for visual surveillance ", Proceedings of the IEEE, Vol. 90, No.7, pp. 1151-1162, 2002.
[9] C. N. Ianăşi, V. Gui, C. I. Toma, D. Pescaru, "*A fast algorithm for background tracking in video surveillance using nonparametric kernel density estimation*", Facta Universitatis, Niš, Serbia and Montenegro, Series Electronics and Energetics, Vol. 18, No.1, pp. 127-144 , 2005.
[10] R. Stolkin, I. Florescu, G. Kamberov. "An adaptive background model for CAMSHIFT tracking with a moving camera". *Proc. 6th International Conference on Advances in Pattern Recognition*, World Scientific Publishing, Calcutta, pp 261-265 , 2007.
[11] B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical Bayesian filter. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(9), pp 1372-1384, 2006.
[12] L. Gui, J.P. Thiran and N. Paragios, "Joint object segmentation and behavior classification in image sequences", *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* Minneapolis 2007.
[13] L. Song and M. Takatsuka. Real-time 3D finger pointing for an augmented desk. In *Australasian conference on User interface*, volume 40, Newcastle, Australia, pp 99-108, 2005.
[14] C. von Hardenberg, F. Bérard, "Bare-Hand Human-Computer Interaction", *Proceedings of the ACM workshop on Perceptive User Interfaces*, Orlando, Florida, USA, Nov. 15-16, 2001.
[15] K. Oka, Y. Sato and H. Koike, "Real-Time Fingertip Tracking and Gesture Recognition", *IEEE Computer Graphics and Applications*, pp 64-71, 2002.
[16] S. Malik, C. McDonald, G. Roth, „Finger detection via blob finding flood fill: Hand Tracking for Interactive Pattern-based Augmented Reality" *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02)*, 2002.
[17] D. Comaniciu, P. Meer, "*Mean shift: A robust approach toward feature space analysis*", IEEE Trans. Pattern Anal. Machine Intell., Vol. 24, No. 5,pp 603-619, 2002.
[18] V. Gui, "Edge preserving smoothing by multiscale mode filtering". *European Conference on Signal Processing, EUSIPCO'08,* Lausanne, August 25-29, 2008.