TOWARDS REAL-TIME AUDIOVISUAL SPEAKER LOCALIZATION

Gianluca Monaci

Video & Image Processing group
Philips Research Eindhoven, The Netherlands
gianluca.monaci@philips.com

ABSTRACT

There is a growing interest in multi-modal signal processing: sets of related signals are jointly processed to extract information that is otherwise hidden when considering the different modalities independently. One popular problem in cross-modal processing is the localization of visual sources synchronous with audio stimuli. Audiovisual source localization allows to pinpoint and extract salient audio-video information from a scene, enabling innovative applications in communication, interaction and gaming. In this paper we aim to achieve cross-modal localization in *real-time* using single camera, single microphone data. Existing works use complex statistical data models or complex representations of audio and video features, limiting their applicability in real-time systems. In this paper we propose a simple yet effective algorithm that allows to detect and localize in real-time synchronous audio-video sources. The proposed approach obtains the best speaker localization performances reported to date on the popular CUAVE database, while running in real-time and without requiring any training.

1. INTRODUCTION

Audiovisual communication systems are becoming increasingly popular and practically useful in many application domains. Video cameras and microphones in laptops are commonly used for video chatting and multi-party video conferencing, while gaming and social network applications might want to integrate the users' faces and voices in users avatars. A key component for these applications is the association, at any given moment in time, of the speech signal in the audio track with the video region containing the corresponding speaker. Audiovisual speaker localization allows to extract the relevant audio-video information (e.g. the speaker's face with his/her speech), while the cost of storing and sending this information is much lower. Besides, relevant audiovisual data can be specifically protected using error protection mechanisms or encrypted for privacy reasons. This information can be easily used to create realistic avatars and it will enable smart videoconferencing tools such as automatic meeting summarization or *virtual director* applications.

Several methods exist to estimate the spatial location of sound sources using multi-microphone systems and stereo triangulation [16, 23]. Here instead we want to achieve this task using common hardware available in laptops and phones, i.e. one camera and one microphone, exploiting the synchrony between audio and video signals. In fact, several studies have demonstrated that humans continuously combine audio-video cues to enhance our understanding of the environment: for example, sounds appear to be produced by motion synchronous with acoustic stimuli [5, 19].

These observations motivated Hershey and Movellan [8] to design an algorithm to locate sounds using audio-video synchrony. The correlation between audio and video was measured using the correlation coefficient between the energy of an audio track and the value of single pixels. Successive studies in the field [4, 6, 7, 9, 14, 17] focused on the statistical modeling of the relationship between simple audio and video features, proposing audiovisual association strategies based on Canonical Correlation Analysis [9], maximization of Mutual Information [4, 6, 14], learning Gaussian Mixture Models [7] or Hidden Markov Models [14, 17]. While these works focus on the modeling of the correlation between basic audiovisual features (audio energy and pixel values), in [11, 12] we propose to model audio and video data using features that capture structural signal properties. Video sequences are expressed as sparse sums of time-evolving visual structures, while audio tracks are expressed as sums of Gabor atoms. Audiovisual correspondences are established by checking the co-occurrence of these audio-video structures in a way that is pretty similar to what we humans do.

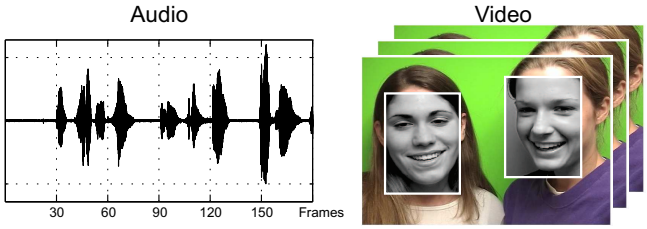

Figure 1: Audiovisual scene featuring two individuals, each one of which may be speaking at any time. In this work we analyze one audio track (left) and several video streams (two here, within the white boxes on the right), each one showing the face of one speaker. Our goal is to associate the audio with the proper video stream in real-time.

In all existing works, either complex statistical models of audiovisual dependencies have to be estimated [4, 6, 7, 9, 14, 17], or computationally intensive audiovisual data representations have to be computed [11, 12]. These factors limit the applicability of existing approaches, because audiovisual localization has to be performed in real-time for interaction and communication applications. In this paper we propose a simple yet effective algorithm that can detect in *real-time* synchronous audio-video structures, allowing to localize audiovisual sources. Audio and video structures are defined respectively as concentrations of acoustic energy and motion of video regions. We consider audiovisual scenes featuring several persons like the one depicted in Fig. 1, in which each individual may be speaking at any time. Having a one-microphone audio recording of the scene and several video streams, one for each subject in the scene, our objective is

to associate the audio data with the proper speaker (video stream) at any given time. Furthermore, the whole algorithmic chain, comprising face detection and audiovisual association, has to be implementable in real-time. We demonstrate that our method obtains the best performances reported to date on the standard audiovisual CUAVE database [15] for speaker association, while working in real-time and without requiring any labeled training data or any scene or lip model.

2. AUDIOVISUAL SPEAKER LOCALIZATION

In this section we describe the proposed audiovisual speaker localization algorithm. The audio signal is represented using a simple feature described in Sec. 2.1. Let us assume here that the video is pre-processed using a face detector or tracker to extract video streams showing the speakers' faces. The type and amount of motion present in each video stream is estimated using a Block Matching algorithm, that is presented in Sec. 2.2. The way audio-video representations are combined to measure their degree of synchrony is described in Sec. 2.3. Finally, in Sec. 2.4 we describe how we associate the audio track with the correct speaker to localize him/her.

2.1 Audio Representation

As mentioned above, we look for synchrony between audio-video events. An interesting audio event, from our point of view, is the presence of a sound. Thus we estimate an audio feature that assesses the presence or absence of an acoustic event. Finer audio features are unnecessary in this setting, but can be considered to perform more complex tasks.

Here the audio signal, $\mathbf{a}(t)$, is represented using a simple feature that estimates the acoustic energy contained in each frame. The audio feature, $\mathbf{f}_a(t)$, is computed as:

$$\mathbf{f}_a(t) = \text{downsample}(\text{avg}(|\mathbf{a}(t)|^2, winSize), d). \quad (1)$$

The function $\text{avg}(\mathbf{x}(t), winSize)$ computes the running average of the signal $\mathbf{x}(t)$ over a window of *winSize* samples. We choose an averaging window that spans two video frames, i.e. $winSize = 2 \cdot \nu_a/\nu_v$, with ν_a and ν_v respectively the audio and video sampling frequencies. The function $\text{downsample}(\mathbf{x}(t), d)$ decreases the sampling rate of $\mathbf{x}(t)$ by a factor d, keeping every d^{th} sample. Here the feature $\mathbf{f}_a(t)$ is down-sampled such that it has the same temporal sampling rate of the video, thus $d = \nu_a/\nu_v$. The audio feature $\mathbf{f}_a(t)$ is depicted in Fig. 2 (b).

2.2 Video Representation

In this paper we assume that a real-time face detector and/or tracker extracts the face region of the individuals present in the scene. One such detector could be the very popular Viola-Jones detector [18], or one of its variants, which are nowadays ubiquitously present in consumer electronics products. Thus, we have a series of video streams, each one showing the face of a person present in the scene. We estimate the motion in these video streams using the Block Matching Algorithm (BMA). BMA is a popular and effective algorithm that is used in virtually any video coding system [21]. We decided to use this approach because the motion information in the form of *motion vectors* is already embedded in any video encoded data: this means that the motion information can be obtained with very small or no computational overhead. Please note however that for the sake of simplicity, in this paper we re-estimate the video motion using the BMA applied to decoded video data.

The underlying assumption behind motion estimation is that changes between two consecutive (or close enough) frames are due to motion, and thus every structure in the current frame has a corresponding structure in the previous reference frame. The main idea of block matching is to divide the current frame into a matrix of *macroblocks* (MB) that are then compared with the corresponding block and its adjacent neighbors in the reference frame to create a vector that assesses the movement of a MB from the reference to the current frame. The matching of one MB with another is based on the output of a cost function. The MB that results in the least cost is the one that more closely matches the current block. Various cost functions exist, e.g. mean absolute distance (MAD), mean squared distance (MSD), and normalized cross-correlation (NCC) [20]. Probably the most popular and less computationally expensive, which we use here, is the mean absolute distance:

$$\text{MAD} = \frac{1}{M^2} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} |c_{ij} - r_{ij}|, \quad (2)$$

where M is the side of the macroblock and c_{ij} and r_{ij} are the pixels being compared in current and reference MB, respectively. The search area for matching a MB is constrained up to p pixels on the fours sides of the corresponding MB in previous frame. Larger motions require a larger p, and the larger the search parameter p the more computationally expensive the process of motion estimation becomes.

Another important step of a BMA is the search strategy employed to match a MB from one frame to the other. The exhaustive search (or *Full Search*) algorithm is typically not used because of its computational complexity: it calculates the cost function at each possible location in the search window. Several methods have been developed during the last decades to find a trade-off between motion estimation accuracy and computational complexity [20]. In this work we estimate MB motion using a block matching algorithm called *Adaptive Rood Pattern Search* (ARPS) [13]. This method exploits the observation that motion in a frame is usually coherent. Thus, if the macroblocks around the current MB moved in a particular direction then there is a high probability that the current MB will also have a similar motion vector. This algorithm uses the motion vector of the MB to its immediate left to predict its own motion vector, greatly reducing the search area. In this work we use a slightly modified version of the ARPS algorithm implemented in Matlab that is publicly available at [1].

2.3 Audiovisual Synchrony Assessment

In this section we describe how we estimate the degree of synchrony between motion of visual structures (macroblocks) and acoustic activity. We will then point out in the next section how this synchrony measure is used to localize the speaker in the sequence.

For each video stream, we analyze only the lower half of the image, representing broadly the mouth region. This will help to filter out spurious movements and to speed-up the computation. The mouth region is subdivided into N macroblocks and for each of them, using BMA, we obtain a motion vector describing its motion from one frame to the

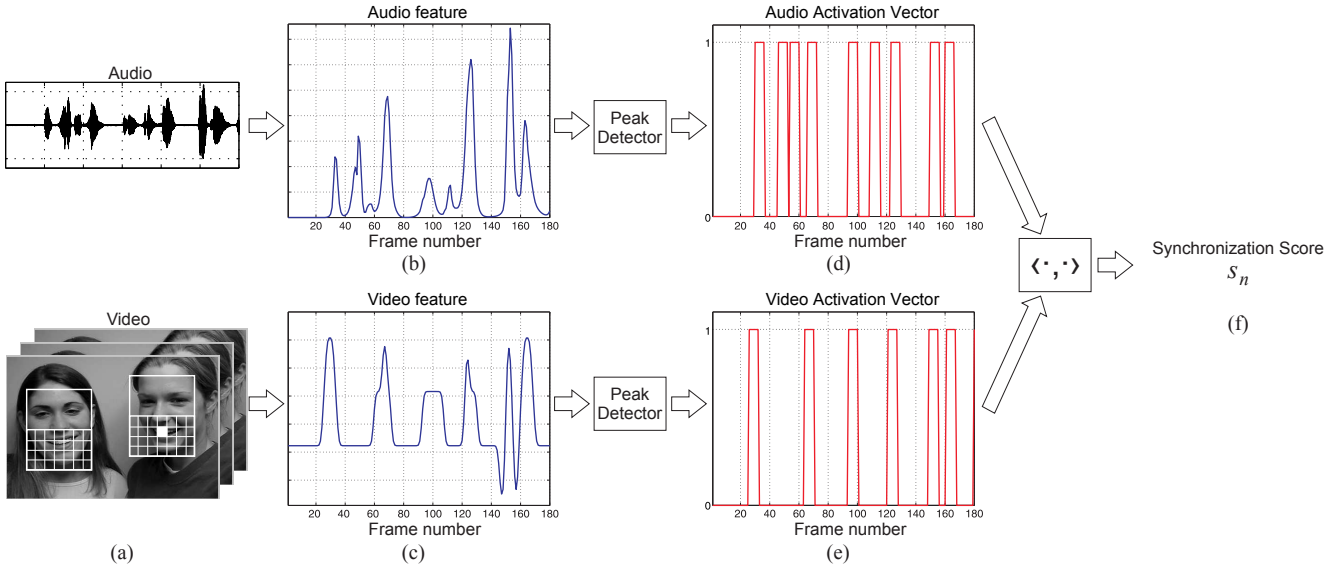

Figure 2: Scheme of the computation of the synchronization score s_n for one MB with $\Delta = 180$ frames. Starting from the original audiovisual sequence (a), we compute the audio feature $\mathbf{f}_a(t)$, shown in (b). Faces are detected in the video signal (white boxes around the individuals' faces) and the lower halves are subdivided into macroblocks. For each MB, the motion feature $\mathbf{f}_{v_n}(t)$ is computed. The feature associated with one MB next to the speaker's mouth (highlighted in white on the right person) is depicted in (c). From these signals we extract the audio energy peaks and the displacement peaks and the activation vectors $\mathbf{y}_a(t)$ and $\mathbf{y}_{v_n}(t)$ are built (d–e). The synchronization score s_n is constructed by computing the scalar product between the two activation vectors (f).

other. Please note that the motion vectors are not estimated on the original images, but they are computed on the mouth region of the *aligned* faces, thus filtering out head movements. The motion of each MB is characterized by a vertical and an horizontal component. For each MB we compute a feature, $\mathbf{f}_{v_n}(t), n = 1,\ldots,N$, which is the absolute value of the vertical displacement. We consider only the vertical motion since we consider a speaker localization task and typical mouth movements occur along the vertical direction.

The considered video feature reflects the movement, from frame to frame, of the image structures present in the MB. The audio feature indicates the acoustic energy content at a given time instant. Peaks in such signals suggest the presence of an event, e.g. the movement of the lips in the video and the presence of a sound. If those audio and video peaks occur at time instants that are temporally close, we are in the presence of an audiovisual event that reflects two expressions (acoustic and visual signals) of the same physical phenomenon (production of a sound). For a given feature vector $\mathbf{f}_x(t)$, $x = a, v$, we build an *activation vector* $\mathbf{y}_x(t)$ which is based on the information about the peaks locations. First, we detect the peaks in the audio feature and in each of the N video features, obtaining vectors which equal 1 where peaks occur and 0 otherwise. Then, such vectors are filtered with a rectangular window of size W. The filter models delays and uncertainty, since it rarely happens that activation peaks occur exactly at the same time instant in both acoustic and video feature vectors. In our experiments with videos recorded at 29.97 frames per second (fps), we have obtained similar results using values of W between 3 and 9. All the results presented in the following are obtained with $W = 7$.

We end up with one activation vector for the audio, $\mathbf{y}_a(t)$, and N activation vectors $\mathbf{y}_{v_n}(t)$, one for each MB. The scalar product between $\mathbf{y}_a(t)$ and $\mathbf{y}_{v_n}(t)$ constructed over a given observation time slot $\Delta = [T_1, T_2]$ counts the number of times the audio and video activation vectors are 1 at the same time and thus gives an estimate of the degree of synchrony between the audio track and the motion in the MB. We define a simple audiovisual co-occurence measure, the *synchronization score* s, as

$$s_n = \langle \mathbf{y}_a(t), \mathbf{y}_{v_n}(t) \rangle, \text{ with } t \in \Delta, \tag{3}$$

where $\langle \cdot, \cdot \rangle$ indicates the scalar product between vectors. The higher the value of the synchronization score, the more synchronous are the video structures with the audio track. Fig. 2 summarizes the computation of the synchronization score s_n for one MB with an analysis window $\Delta = 180$ frames.

2.4 Active Speaker Localization

Considering an analysis window Δ, the total synchronization score, S, of a given video stream over Δ is $S = \sum_{n=0}^{N-1} s_n$. The signals are analyzed using a sliding window Δ that is shifted along the sequence with a step δ that is typically equal or smaller than Δ. This is done to guarantee a certain temporal continuity. The active speaker is detected every δ frames as the one in the video stream exhibiting the highest value of total synchronization score S over the analysis window. This detection criterion has been chosen here for the sake of simplicity. Clearly the synchronization scores can be compared to a threshold, as e.g. in [12], so that if several synchronization scores (or none of them) are above the threshold, several speakers (or none) are detected as active.

The proposed algorithm is summarized in Algorithm 1. For simplicity, in Algorithm 1 we describe a system that analyzes a video file of length T frames, but clearly the procedure can be performed in real-time on a video stream of unknown duration.

Algorithm 1: Pseudo-code of the proposed audiovisual speaker detection system.

Input: Video V of duration T frames.
Output: Active speaker $\hat{k}_t$ at each time step t.
for $t = 1 : \delta : T$ **do**
  1. Detect K_t faces;
  2. Estimate audio energy;
  3. Compute audio activation vector;
  for $k_t = 1 : K_t$ **do**
    4. Divide lower half of stream k_t into N MB;
    for $n = 1 : N$ **do**
      5. Estimate motion vectors using BMA;
      6. Compute video activation vectors;
      7. Compute s_n using Eq. (3);
    end
    8. Compute S_k for stream k_t : $S_k = \sum_n s_n$;
  end
  9. Detect stream $\hat{k}_t$ containing the active speaker: $\hat{k}_t = \arg\max_k S_k$.
end

3. EXPERIMENTS

We show here how the proposed system is used to localize the source of an audio signal in real-world video sequences. Given a single audio track and several video streams, one for each speaker in a scene, we want to determine who, if anyone, is speaking at any given time instant.

The proposed algorithm is tested on the CUAVE corpus [15], a multiple speaker audio-visual corpus of spoken connected digits. A frame from one of the sequences of the database is shown in Fig. 3 (left). We use the 22 clips from the *groups* set in which two speakers take turns reading digits and then proceed to speak simultaneously. In order to compare our results to [4, 7, 14, 17] we consider the section of alternating speech and use the ground truth speaker segmentation from [3]. To evaluate the performances of our speaker localization algorithm we adopt the recommendations in [3]: a detection is considered correct if the detector's output for a given time window matches the most present label in the corresponding ground truth window. Siracusa and Fisher extracted the speakers' faces from each clip of the database and they made them publicly available, together with the ground truth labels, at [2]. The faces are grayscale and normalized to 75×50 pixels. A face detector and correlation tracking of the nose region is used to extract aligned faces. The faces extracted from the sequence in Fig. 3 (left) are shown in Fig. 3 (right). The dataset consists of videos sampled at 29.97 fps with corresponding audio tracks at 44 kHz, which are re-sampled here at 8 kHz.

The video streams showing the speakers' faces are analyzed using macroblocks of size $M = 8 \times 8$ pixels. The search parameter p is kept fixed to a value of 4 pixels. In these rather static sequences, the increase of the value of p does not bring any benefit and only increases the computational complexity. Concerning the audiovisual association algorithm, experiments have been carried out varying δ and Δ between the values 15, 20, 30, 40 and 60 frames. The optimal values for these variables are scene-dependent: in a scene showing a very dynamic conversation, it could be beneficial to decide often who is speaking, while in a slowly paced conversation, the decision can be taken less often using more data to provide a more reliable outcome.

Figure 3: Frame extracted from one sequence of the CUAVE audiovisual database (left) and extracted faces of the people present in the video (right).

		Δ				
		15	20	30	40	60
δ	15	86%	89.1 %	90.1%	88.4%	84.8%
	20	86.7%	90.6%	91.7%	89.4%	85.9%
	30	84.6%	90.8%	**93.4%**	91.6%	88 %
	40	82.7%	88.8%	91.5%	90.4%	88.5%
	60	74.7%	82.8%	86.2%	90.1%	86.8%

Table 1: Average detection accuracy on the CUAVE database for different values of δ and Δ expressed in frames.

Table 1 summarizes the results obtained by the proposed method in term of percentage of test points at which the actual speaker is correctly detected for different combinations of values of δ and Δ. The percentages represent the average accuracy over the whole *groups* section of the CUAVE dataset. Our approach achieves very high accuracy with all settings. As it might be expected, accuracy degrades when the analysis window Δ is too long or too short (60 or 15 frames) and when the values of δ and Δ are very different. The best performances are obtained for $\delta = \Delta = 30$ frames (i.e. 1 second), with a correct detection rate of 93.41%. This result compares extremely favorably with existing audiovisual speaker localization methods that have been tested in the same database. Nock *et al.* [14], Gurban and Thiran [7] and Besson *et al.* [4] consider only the last 12 sequences of the database (*g11* to *g22*) for testing because these methods require training and the first 10 sequences of the corpus are used for this purpose. Furthermore, the algorithms in [4, 7, 14] exclude silent frames using dedicated silence detectors. Nock *et al.* [14] achieve 76% average accuracy, Gurban and Thiran [7] 87.4% and Besson *et al.* [4] 85%. Siracusa and Fisher [17] instead use the whole database for testing as they adopt an online learning method, achieving 88.11% accuracy. The results are summarized in Table 2.

In the current experimental setting, 24 motion vectors have to be estimated per speaker every frame, thus 48 in total for the two individuals in the test sequences. In Matlab on a 2.33GHz processor with 2Gb RAM, the computation of the motion vectors takes 0.0155s per frame while the audiovisual association and speaker detection algorithm employs 0.0013s per image. The whole processing in Matlab, exclud-

Method	Mean Accuracy
Nock *et al.* [14]	76 %
Gurban and Thiran [7]	87.4 %
Siracusa and Fisher [17]	88.11 %
Besson *et al.* [4]	85 %
Proposed	**93.41 %**

Table 2: Average speaker localization results on the CUAVE corpus. Our approach outperforms all existing methods tested on the dataset

ing face detection, runs at 60 fps, more than real-time. Furthermore, as already underlined motion vectors do not have to be re-computed as they could be extracted directly from the compressed video sequence, saving a great amount of processing time. Although we do not know the computational complexity of the face detector employed to extract the faces in [17] and used for these experiments, common face detector implementations run typically at about 20 fps, and fast hardware-efficient implementations are being developed that can achieve much higher performances ($>$ 100 fps) [10, 22]. Since we do not need to detect the speakers' faces every frame (2-3 times every second are sufficient to guarantee a smooth behavior), we can safely claim that the proposed system is implementable in real-time on commonly available hardware.

4. DISCUSSION

In this paper we have introduced a method to detect and localize speakers in audiovisual scenes using only the data from one microphone and one camera. The proposed system first detects the faces of the individuals present in the scene and then computes motion within the face regions using a Block Matching Algorithm. The co-occurrence of relevant motion and audio energy peaks is used to associate the audio content to the corresponding speaker at any given time instant, thus localizing the active speaker. Our method is simple enough to run in real-time on common processors and achieves a speaker localization accuracy of 93.41% on the standard CUAVE audiovisual dataset, outperforming all published algorithms tested on this corpus. In contrast with most of the existing methods, no training on labeled data nor silence detector or scene model are required, which makes the proposed approach flexible and general.

ACKNOWLEDGMENTS

The author would like to thank Prof. Pierre Vandergheynst for fruitful discussions that inspired this work.

REFERENCES

[1] www.mathworks.com/matlabcentral/fileexchange/8761.

[2] people.csail.mit.edu/siracusa/avdata.

[3] P. Besson, G. Monaci, P. Vandergheynst, and M. Kunt. Experimental framework for speaker detection on the CUAVE database. Technical Report EPFL-ITS 2006-003, Lausanne, 2006.

[4] P. Besson, V. Popovici, J.-M. Vesin, J.-P. Thiran, and M. Kunt. Extraction of audio features specific to speech production for multimodal speaker detection. *IEEE Trans. Multimedia*, 10(1):63–73, 2008.

[5] J. Driver. Enhancement of selective listening by illusory mislocation of speech sounds due to lip-reading. *Nature*, 381(6577):66–68, 1996.

[6] J. W. Fisher III and T. Darrell. Speaker association with signal-level audiovisual fusion. *IEEE Trans. Multimedia*, 6(3):406–413, 2004.

[7] M. Gurban and J.-P. Thiran. Multimodal speaker localization in a probabilistic framework. In *Proc. EUSIPCO*, 2006.

[8] J. Hershey and J. Movellan. Audio-vision: Using audio-visual synchrony to locate sounds. In *Proc. NIPS*, 1999.

[9] E. Kidron, Y. Schechner, and M. Elad. Cross-modal localization via sparsity. *IEEE Trans. Signal Processing*, 55(4):1390–1404, 2007.

[10] H.-C. Lai, M. Savvides, and T. Chen. Proposed FPGA hardware architecture for high frame rate ($\gg$100 fps) face detection using feature cascade classifiers. In *Proc. IEEE BTAS*, pages 1–6, 2007.

[11] A. Llagostera Casanovas, G. Monaci, P. Vandergheynst, and R. Gribonval. Blind audiovisual source separation based on sparse redundant representations. *IEEE Trans. Multimedia*, 12(5):358–371, 2010.

[12] G. Monaci and P. Vandergheynst. Audiovisual gestalts. In *Proc. IEEE CVPR Workshop*, 2006.

[13] Y. Nie and K.-K. Ma. Adaptive rood pattern search for fast block-matching motion estimation. *IEEE Trans. Image Processing*, 11(12):1442–1449, 2002.

[14] H. J. Nock, G. Iyengar, and C. Neti. Speaker localisation using audio-visual synchrony: an empirical study. In *Proc. ACM CIVR*, pages 488–499, 2003.

[15] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy. Moving-talker, speaker-independent feature study, and baseline results using the CUAVE multimodal speech corpus. *EURASIP J. Adv. Sig. Proc.*, (11):1189–1201, 2002.

[16] P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proc. IEEE*, 92(3):495–513, 2004.

[17] M. R. Siracusa and J. W. Fisher. Dynamic dependency tests: Analysis and applications to multi-modal data association. In *Proc. AIStats*, 2007.

[18] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Computer Vision*, 54(2):1442–1449, 2004.

[19] M. T. Wallace, G. E. Roberson, W. D. Hairston, B. E. Stein, J. W. Vaughan, and J. A. Schirillo. Unifying multisensory signals across time and space. *Exp. Brain Res.*, 158(2):252–258, 2004.

[20] Y. Wang, J. Ostermann, and Y. Zhang. *Video Processing and Communications*. Prentice Hall, Signal Processing Series, 2002.

[21] J. Watkinson. *MPEG Handbook*. Focal Press, 2004.

[22] W.-S. Wong, C.-R. Chen, and C.-T. Chiu. A 100Mhz hardware-efficient boost cascaded face detection design. In *Proc. IEEE ICIP*, pages 3237–3240, 2009.

[23] H. Zhou, M. Taj, and A. Cavallaro. Target detection and tracking with heterogeneous sensors. *IEEE J. Sel. Topics Signal Processing*, 2(4):503–513, 2008.