Sparse-MVRVMs Tree for Fast and Accurate Head Pose Estimation in the Wild

Mohamed Selim, Alain Pagani, and Didier Stricker

Augmented Vision Research Group,
German Research Center for Artificial Intelligence (DFKI),
Tripstaddterstr. 122, 67663 Kaiserslautern, Germany
Technical University of Kaiserslautern
{mohamed.selim,alain.pagani,didier.stricker}@dfki.uni-kl.de
http://www.av.dfki.de

Abstract. Head pose estimation is an important problem in the field of computer vision and facial analysis. We model the problem of head pose estimation as a regression problem, where the three rotation angles (yaw, pitch, roll) are functions of the face appearance. We make use of that fact and learn the appearance of the face using a tree cascade of sparse Multi-Variate Relevance Vector Machines (MVRVM). Our method is fast and suitable for real-time applications as it is not computationally expensive. Our method learns the face appearance to estimate the head rotation angles. We evaluated our approach on two challenging datasets, the YouTube Faces and the Point and Shoot Challenging (PaSC) dataset. We achieved results of head pose estimation (yaw, pitch, roll) with mean error less than 5 degrees and with error tolerance less than ± 4 on the PaSC dataset. In terms of speed, one prediction takes around 6 milliseconds, which is suitable for real-time applications and also with high frame rate.

Keywords: Head Pose Estimation, MVRVM, Cascade, YouTube Faces, PaSC

1 Introduction

Due to many potential applications, head pose estimation has become one of the most active and important topics in computer vision [12]. The problem can be considered as a sole problem to be solved and tackled, or as an important part of a bigger system. For example, it can help in gaze estimation problems. Valenti *et al* [17] combined head pose with eye location to solve gaze estimation problem.

As outlined in [12], the problem of head pose estimation has been framed as a crucial factor in the field of facial analysis, in case robustness to pose is required in an application. For example, in implementing a gender classifier, a pose estimator can be an important pre-processing step in the system.

The problem can be addressed as a classification problem, where the system can try to classify the face in one of the main rotations, like left profile, right

profile, semi profile on both sides and frontal face. An SVM could be sufficient in that application. However, if we add more possibilities, the number of classes will be very big in a way a classifier can fail at. Thus, to predict a wide range of angles, we model the problem as a regression problem, where we provide data in the training phase, and our approach can learn the data, and use it to estimate the three rotation angles at a time.

Our approach builds a tree cascade of regressors, where each node in the tree is trained in subset of the training dataset. We estimate the three rotation angles with the cascade tree of Multi-Variate Relevance Vector Machines [16].

Fig. 1. Sample frames from the PaSC dataset [3]. The top images are from the control subset videos (steady camera). The bottom frames are from the hand-held videos. Hand-held have lower quality and resolution. The dataset have videos captured indoor and outdoor. The persons walk during the video, thus we have different, continuous head poses

Although we build over previous work where MVRVM was used for head pose estimation[15], we significantly improve over this work by building a more complex structure of MVRVMs that yields less error. The work in [15] was limited to single subject only. However, with our new tree structure, we generalized the method for faces of unseen subjects. Moreover, We trained MVRVM models with better input angles generated by state of the art head pose estimation algorithm by [2]. Moreover, we validated our new approach for generalization and worked with more challenging datasets, the PaSC [3].

2 Related Work

In recent time, head pose estimation attracted more interest in the computer vision community. Different approaches have been investigated in solving this problem. Some researchers work on 2D facial images [5, 13, 10], and others work on 3D data [4, 8]. For the methods that use AAMs [5] or any specific facial feature, their estimation is dependent on specific features detection, like facial landmarks, thus, making that error prune. In case an error exists in the features detection, it propagates to the head pose estimation.

In the approach proposed by [7], 3D data is used. The 3D data requires special hardware for capturing. In fact, that makes their approach limited to this type of data and cannot be applied on 2D video sequences. Besides, the work done by [11] uses both color data and depth data. They base their estimation on the point cloud data, they achieve very good results. However, comparing to these approaches is not possible as we work with 2D images from video sequences.

Our work deals with 2D facial images. This problem was addressed before in the work done by [19], however, they have a high error tolerance of $\pm 15°$. The problem of head pose was addressed in the work by [2], and they depends on landmark detection. Thus, making head pose estimation depending on the landmark detection. Having this constraint in their approach, they are limited to angles of about -60 to +60 degrees, where enough landmarks are still visible. Our proposed approach doesn't rely on landmarks.

Previously, using MVRVMs in solving the problem in head pose estimation was introduced in [15]. The idea was tested on videos of single subjects from the YouTube faces dataset [18]. It was limited to one subject in the training, in other words, it wasn't generalized to work with any unseen subjects. In this work, we go deeper into the MVRVMs by testing different kernel types. We also work with larger dataset, the Point and Shoot Challenging dataset [3].

3 Methodology

As introduced before in the introduction and related work sections, we build our approach on previous work by [15]. We used MVRVMs for head pose estimation, where it was trained on a single subject. In this work, we want to reduce the error in the estimated head rotation angles and validate the generalization of our approach for unseen faces. For that we introduce the idea of building a more complex structure, that doesn't only have one single MVRVM to make the head pose estimation, but the structure has a tree of MVRVMs. As we build upon previous work, we use the same feature extraction method, which is a vector of normalized pixel intensities extracted from the facial image.

Figure 2 shows an abstract overview of the proposed method. The detected faces are fed into the feature extraction step, then the features and corresponding angles are fed into the Root node of the cascade tree. The cascade is discussed in details in the next subsection.

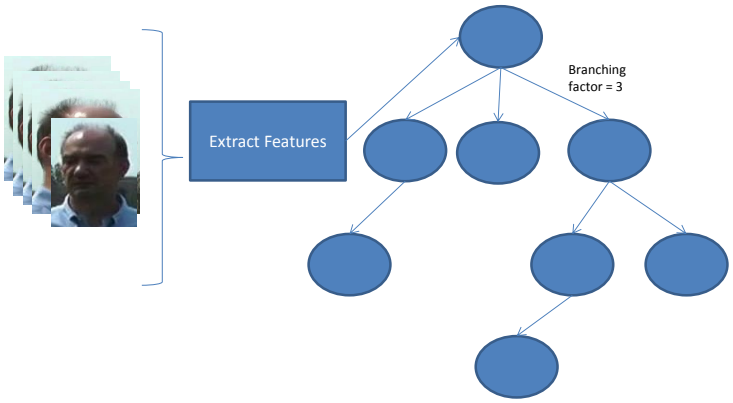

Fig. 2. Overview of the input faces and the cascade tree of sparse MVRVMs

3.1 Cascade of Sparse Regressors

The cascade of the regressors is built in a tree structure. The yaw angle is used in branching the tree, as it is the rotation angle of the head that has the widest range. The yaw angle can start from -90°(left profile face) to +90°(right profile face). At the root node in the tree, the MVRVM is trained on the input samples, which consists of the features of each face and its corresponding three rotation angles. Going to the next level of the tree, the number of children of the node is determined by the branching factor b. If $b = 3$, the yaw angle range is split into three ranges, and the data is filtered such that each node has the samples that lie in the corresponding yaw angle range. The branching goes on untill we reach the maximum depth of the cascade, or a node does not have sufficient data to be learnt.

The resulting tree of MVRVMs, is used in the prediction process. The prediction process starts from the root node. The root node is designed to give a rough estimation of the head pose. Based on the predicted yaw angle, the child node is chosen to be the next node used in the path while traversing the tree of the cascade. The longest prediction path is predicting and improving the estimation by d predictions, where d is the maximum depth of the cascade.

Although, free variables available in using MVRVM for solving the problem of head pose estimation were optimized [15], we now introduce new free variable that needs to be optimized. We carried out experiments to optimize those parameters, the tree branching factor and its depth. We also investigated different MVRVM kernel types.

4 Evaluation and Results

In this section, we present the evaluation of our approach on challenging datasets of persons captured in different conditions. These datasets have continuous head pose variation. They vary in the background, illumination, indoor and outdoor locations, resolution, etc. We show the results of the experiments on the datasets to optimize the free variables in our approach. In general we optimize the kernel width and type of the RVM. Moreover, we optimize the branching factor of the tree. Finally, we validate our approach for generalization purposes on large subsets of the dataset.

4.1 Datasets

The standard datasets like FERET [14] has discrete specific values for head pose. A continuous angles variation is an important feature that the dataset must have to perform a proper evaluation of our regression-based approach. Moreover, using real data captured in the wild is an important feature to assure the validity of our algorithm on real-life scenarios. The Labeled faces in the wild [9] is a challenging dataset in terms of occlusion, image quality, varying poses, different illumination, etc. However, it does not provide sufficient samples for each subject in different poses. Good candidates to the best of our knowledge are the video datasets, YouTube faces [18] and the PaSC [3].

YouTube Faces Dataset The YouTube faces dataset [18] is a challenging dataset that has 3425 videos of 1595 different people. The authors of the dataset provided the rotation angles of each frame in the dataset. They used face.com API to provide the head rotation angles.

Point and Shoot Face Recognition Challenge (PaSC) In 2013, Beveridge *et. al* produced the PaSC dataset [3]. They used inexpensive "point-and-shoot" cameras. They collected 9376 still images and 2802 videos of 293 people. The videos were recorded in different locations, outdoors and indoors, with varying illumination and backgrounds. The authors provided meta-data with the dataset that contains the face detection in the video frames. The head rotation angle was provided by the PittPatt detector. The scenarios they had in the videos shows the face from the right profile to the left profile in continuous motion, where the yaw angles changes widely along the videos. Two video types were provided in the dataset, hand-held and controlled subsets. In the hand-held videos, the frames are very shaky and challenging. The controlled videos, have a stable background. Both video types are challenging. Figure 1 shows sample images from the dataset.

The rotation angles in the datasets were produced using the face.com API for the YouTube Faces, and PittPatt for the PaSC (yaw angle only) dataset. However, the work done by [2] focuses on facial landmarks detection, and can estimate the head pose. We used their approach to generate estimations of the

head rotation angles. However, even this approach was challenged, as it was unable to detect and track the landmarks in some hard frames of the detected faces in the PaSC dataset. It worked with about 72% of the faces provided by the meta-data in the dataset. However, we don't need all the frames, as our approach learns the head pose from appearance and can estimate it for any detected face.

4.2 Parameters Optimization

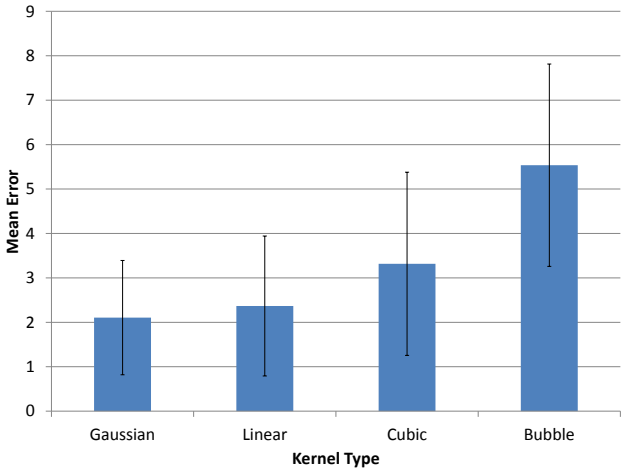

Fig. 3. Comparing different kernel types on the PaSC dataset. Average errors in the yaw angle estimation are shown with the standard deviation among all dataset videos. Gaussian kernel yields the least error.

Choice of the Kernel and its size As mentioned in section 3, the kernel type is chosen in a way that suits the data provided to the RVM. Kernel type affects the accuracy of the training as it is the metric mapping the input to the output of the RVM. We evaluated four kernel types (Gaussian, Linear, Bubble, and Cubic) on the PaSC dataset subject videos and on the YouTube faces dataset.The Gaussian kernel yields the least error in the yaw angle estimation, hence it is the kernel that we used in the next evaluations. The kernel width has a strong effect on the accuracy of the cascade. In [15], the kernel was optimized for the head pose estimation problem. It was varied starting from value 3 up to 50. The optimal value found was 13.

Grid Size In the work [15], the result of optimizing the face grid size to was reported to be 15×15. Here we build upon these results, and proceed to build the cascade tree for solving the problem of head pose estimation.

4.3 Branching Factor

The tree branching factor affects the accuracy of the estimations. We variate the branching factor of the cascade tree from 2 splits up to 5 splits. Having more than 5 splits makes the range very small in the child nodes. We start the optimization at branching factor of value 2, which is the minimum number of splits possible. When the branching factor was more than 4, the number of input samples decreased quickly in the tree, h ence, resulting in a shallow tree. The branching factor with the least error was 3. We set the root node to have only two children, thus classifying right or left profile faces. Further children in the tree have branching factor of 3. Based on that setup, the leaf nodes of the tree get very small range of angles after depth of 3. Consequently, we set the maximum depth to be 3 levels.

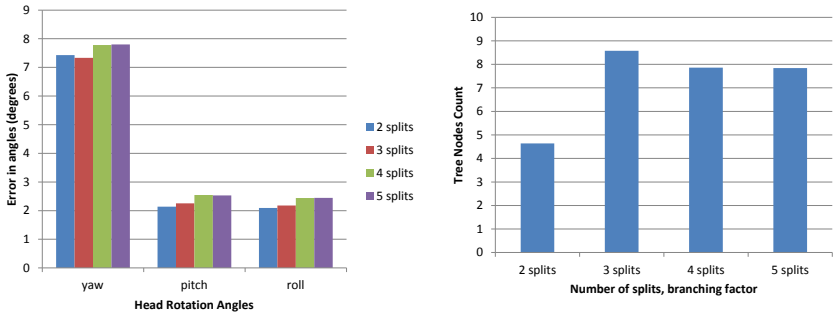

Fig. 4. The effect of the branching factor on the mean error in the head rotation angles (left). The effect of the number of the branching factor on the tree nodes count (right), using the YouTube faces dataset

Figure 4 shows the effect of varying the branching factor on the YouTube dataset. The results are the average of 4-fold cross validation on all the 1595 subjects in the dataset. In figure 4, we see that the 3 splits has the most number of nodes, which means a better representation of the data in the cascade tree. It also follows that the least average error on the main head rotation angle, the yaw, is at 3 splits. Considering the presented evaluations, we optimized the free parameters in our approach by 4-fold cross validation experiments which considered all the videos of one of the subjects. Thus, the next step is validating the approach with as many samples from the dataset as possible, which is discussed in subsection 4.5.

4.4 Single RVM vs. Cascade Tree

It is important to compare the single MVRVM [15] to the Cascade tree of MVRVMs. Table 1 shows the mean accuracy of 4-fold cross validation test on

the PaSC dataset. The cascade tree approach yields smaller errors in all head rotation angles.

Method	Yaw	Pitch	Roll
Single MVRVM [15]	5.4 ±4	5.4 ±4	3 ±2.5
Cascade Tree	**4.6 ±3.32**	**5 ±4**	**2.3 ±2.1**

Table 1. Comparing the Single with the Tree Cascade of MVRVMs. The Cascade shows smaller mean error - PaSC. (Mean error ±std). Validation experiment of chunks of 5000 samples.

4.5 Validation

Based on the findings so far, the kernel width optimal value is 13, and the optimal number of grid divisions is 15. The final step is approach validation. We generated better head pose estimations for the PaSC dataset using a newer method proposed by [2](compared to PittPatt used in the dataset metadata). Their method deals with landmark localization and tracking, and it can be used in head pose estimation. To validate our method for generic use, we first shuffle all video frames from all subjects. Then we divide the frames into sets of 5000 frames each. We run 4 fold cross-validation on each set. The number of validation sets in the PaSC is 25, each having 5000 random frames from different subjects. Table 4.5 shows the average mean error with standard deviation reported on all the sets.

Dataset	Yaw	Pitch	Roll
PaSC [15]	5.4 ±4	5.4 ±4	3 ±2.5
PaSC (Our work)	**4.6 ±3.32**	**5 ±4**	**2.3 ±2.1**
HPEG (Work [6])	4.25 ±3.04	3.83 ±2.72	-
HPEG (75,25)	**2.6 ±1.2**	**1.9 ±2**	-
HPEG (25,75)	**3.45 ±1.9**	**2.15 ±2.25**	-

Table 2. Validation results, reported on the PaSC dataset. Average errors in the angles with the standard deviations are reported.

The MVRVMs can learn the head pose by the appearance of the face with high accuracy. Less the 4.6 °error in the Yaw angle in the validation tests are reported on very challenging uncontrolled datasets. Regarding the pitch and roll angles, the MVRVM reported errors less than 5°on PaSC. Fig. 5 shows the distribution of the errors in the angles on the dataset frames. We can see that the error is below 5° in the yaw angle for about 66% of the data. and

below 10° for about 98% of the data. The errors in the pitch and roll are a bit higher which could be due to the fact that the video frames did not have as big variations as in the yaw angle. Finally, our method is suitable for real-time applications as the time taken by the computation of one single prediction of the three head rotation angles is only 6 milliseconds, with no need of complex landmark detection or model fitting or tracking.

We compared our approach with the work in [6] using the same dataset, the HPEG dataset [1]. The dataset contains 10 video sequences of people rotating their head, the groundtruth angles were provided by the dataset. They acquired it using markers attached to the subject's head (outside the face). The work in [6] requires tracking and doesn't require training, We require training on a subset of the dataset. We used different training and testing percentages of the dataset, either 25% or 75%. Results are shown Table 4.5.

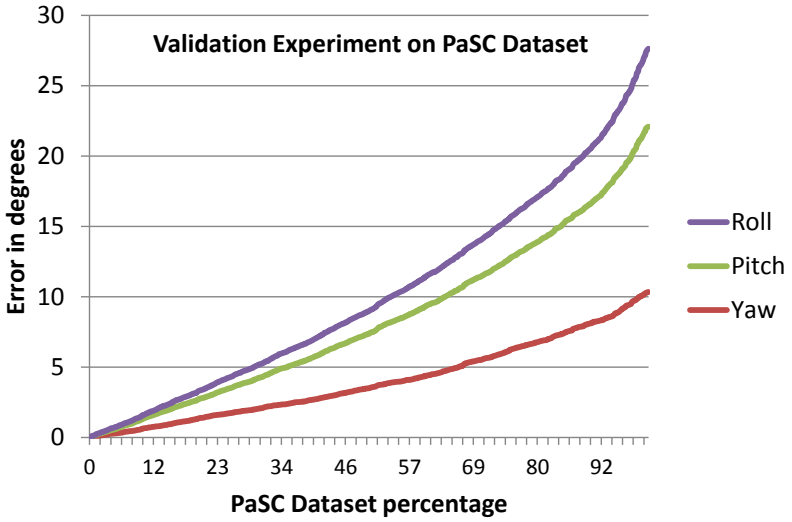

Fig. 5. Results of our new approach in the validation experiment on the PaSC dataset. Yaw error is below 5° in the yaw angle for about 66% of the data. and below 10° for about 98% of the data

The architecture of the machine used in the evaluations is a 6-core Intel Xeon CPU with hyper-threading technology, and 64 GB of RAM. Our evaluation application runs in parallel using the 12 threads provided by the CPU.

5 Conclusion

In this paper, we present a cascade tree of sparse regressors to solve the problem of head pose estimation. This work is built upon the work in [15]. We use the face appearance as the only input, and generate normalized pixel features for training a cascade of MVRVMs. The simple features used are inexpensive to compute on a CPU.

We significatly improve the work in [15] by building a more complex structure that can handle more input data and improve the accuracy of the head pose estimation. Moreover, we make further analysis of the MVRVMs kernels. We also, use a more challenging dataset for training the cascade and validating our method. Finally, we generalize our method where we train using different subjects and not only one subject. Our new proposed approach works on unseen faces.

We tested our approach on two challenging datasets, the YouTube faces dataset and the PaSC dataset. Although, if the values provided by the datasets or the Chehra library [2] are not the absolute correct head rotation angles, we show that we can learn these numbers without the need of model fitting or complex landmark localization. Besides our extensive cross validation experiments which we ran on hundreds of thousands of images from the datasets, we compared our approach to another model-free one by [6], and we show that we significantly reduce the average error on the HPEG dataset [1].

Acknowledgments. This work has been partially funded by the University project Zentrums für Nutzfahrzeugtechnologie (ZNT), and the European project Eyes of Things (EoT) under contract number GA643924.

References

1. Stylianos Asteriadis, Dimitris Soufleros, Kostas Karpouzis, and Stefanos Kollias. A natural head pose and eye gaze dataset. In *Proceedings of the International Workshop on Affective-Aware Virtual Agents and Social Robots*. ACM, 2009.
2. Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Incremental face alignment in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1859–1866. IEEE, 2014.
3. J.R. Beveridge, P.J. Phillips, D.S. Bolme, B.A. Draper, G.H. Givens, Yui Man Lui, M.N. Teli, Hao Zhang, W.T. Scruggs, K.W. Bowyer, P.J. Flynn, and Su Cheng. The challenge of face recognition from digital point-and-shoot cameras. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pages 1–8, Sept 2013.
4. Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, 2003.
5. Timothy F Cootes, Gavin V Wheeler, Kevin N Walker, and Christopher J Taylor. View-based active appearance models. *Image and vision computing*, 20(9):657–664, 2002.

6. Stefania Cristina and Kenneth P. Camilleri. *Computer Analysis of Images and Patterns: CAIP 2015, Valletta, Malta*, chapter Model-Free Head Pose Estimation Based on Shape Factorisation and Particle Filtering. 2015.
7. Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real time head pose estimation with random regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 617–624. IEEE, 2011.
8. Lie Gu and Takeo Kanade. 3d alignment of face in a single image. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1305–1312. IEEE, 2006.
9. Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.
10. Michael Jones and Paul Viola. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*, 3:14, 2003.
11. S. S. Mukherjee and N. M. Robertson. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Transactions on Multimedia*, 17(11):2094–2107, Nov 2015.
12. Erik Murphy-Chutorian and Mohan M Trivedi. Head pose estimation in computer vision: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):607–626, 2009.
13. Alex Pentland, Baback Moghaddam, and Thad Starner. View-based and modular eigenspaces for face recognition. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 84–91. IEEE, 1994.
14. P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1090–1104, October 2000.
15. Mohamed Selim, Alain Pagani, and Didier Stricker. Real-time head pose estimation using multi-variate rvm on faces in the wild. In *Computer Analysis of Images and Patterns*. 2015.
16. Arasanathan Thayananthan, Ramanan Navaratnam, Bjrn Stenger, PhilipH.S. Torr, and Roberto Cipolla. Multivariate relevance vector machines for tracking. In *Computer Vision ECCV 2006*, volume 3953, pages 124–138. Springer Berlin Heidelberg, 2006.
17. R. Valenti, N. Sebe, and T. Gevers. Combining head pose and eye location information for gaze estimation. *Image Processing, IEEE Transactions on*, 21(2):802–815, Feb 2012.
18. Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 529–534. IEEE, 2011.
19. Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.