Sign Language Recognition using Partial Least Squares and RGB-D Information

Barbara N. S. Estrela*, Guillermo Cámara-Chávez†, Mario F. M. Campos*,
William Robson Schwartz* and Erickson R. Nascimento*
*Computer Science Department
Universidade Federal de Minas Gerais, Belo Horionte – MG, Brazil
Email: {bnibia,mario,william,erickson}@dcc.ufmg.br
†Universidade Federal de Ouro Preto, Ouro Preto – MG, Brazil
Email: guillermo@iceb.ufop.br

***Abstract*—Deaf people use systems for communication based on sign language and fingerspelling. Manual spelling or fingerspelling is based on the alphabet, where each letter is represented by an unique movement of the hand. Hand-shapes corresponding to letter of the alphabet can be characterized using appearance and depth images. The fusion of visual and shape cues is a very promising approach. The recent introduction of fast and inexpensive RGB-D sensors, such as Microsoft Kinect, allow to collect appearance and depth images. In this paper, we present a framework that recognizes the American Sign Language (ASL) Fingerspelling using RGB-D images. The proposed framework is based on the bag of features strategy combined with the Partial Least Squares (PLS) technique in order to create models of the letters in the manual alphabet. It also uses the Binary Appearance and Shape Elements (BASE), a fast and low cost descriptor that combines intensity and shape information. We conduct two experiments, the first is based on classification methods. The SVM and PLS classifiers are evaluated. In the second experiment, SIFT and BASE feature descriptors are appraised. SIFT achieves slightly higher accuracy, while BASE runs faster with low memory consumption. This is desirable condition when the objective is to run in real time. In both experiments PLS classifier outperforms the SVM.**

I. Introdution

Sign languages are present in the human communication history since the beginning. They are used for both deaf and non-deaf people in communication process. Sign languages can combine a set of body signs, such as facial expressions, orientation and movement of the hands. As spoken language, the sign communication is a rich and a complex way to express yourself. Despite its use for hearing and deaf people, it is thanks to latter and their needs that the sign languages provide a complete framework of communication, with complex grammars that can be applied in a conversation to discuss from concrete to abstract topics.

Despite the common misconception, manual alphabets, called fingerspelling, are not part of sign languages. However, they are a common source of extracting new signs. Fingerspelling consists in using hands to create shapes and sometimes combined with movements to represent letters of a writing alphabet. In this work we present a framework to recognize the letters in these manual alphabets using RGB-D images.

Human action and gesture recognition is one of the most active topic in computer vision and algorithms to provide solutions for such problems have a large number of applications. From human robotic interaction and entertainment to sign language learning. Differently from traditional approaches that have been using only color images, more recent techniques have used depth images [1] or combining both color images and depth data [2] to improve to accuracy in action and gesture recognition. Consequently, their methodologies can overcome variation in scene illumination and textureless objects, common issues with real scenes that may dramatically decrease performance of classifiers based solely on the image.

The fusion of visual and shape cues is a very promising approach for object recognition. As far as accuracy is concerned, [3], [4], [5], [2] have already shown that the combined use of intensity and depth information outperforms learning using only one of the two.

The reason that many descriptors have not used shape information can be partially explained by the fact that until recently, object geometry was not easy to obtain, nor quick, so as to be combined with image feature data in a timely manner. Although 3-D sensing techniques have been available, such as techniques based on time-of-flight (Canesta), and projected texture stereo (PR2 robot), they are still very expensive and demand a substantial engineering effort. With the recent introduction of fast and inexpensive RGB-D sensors (where *RGB* implies trichromatic intensity information and *D* stands for depth) the integration of synchronized intensity (color) and depth has become easier and cheaper to obtain.

A RGB-D system outputs color images and the corresponding pixel depth information enabling the acquisition of both depth and visual cues in real-time. These systems have opened the way to obtain 3D information with unprecedented trade-off of richness and cost. One such system is the Kinect [6], a low cost commercially available system that produces RGB-D data in real-time for gaming applications.

In this paper, we present a recognition framework, which using a fast and low cost descriptor that combines intensity and shape information presents discriminative power enabling enhanced and faster classification. Experimental results presented later in the paper show that our framework is a robust and computationally efficient technique.

The remainder of the paper is organized as follows: Sec. II describes our recognition system, with details of the features

and classification scheme used. Experimental results are shown in Sec. III and, finally, Sec. IV presents our conclusions and future work directions.

II. Methodology

This section describes the methodology developed to perform sign language recognition from videos captured with depth information. First, in Sections II-A, II-B and II-C, we describe the techniques used in this work, then, in Section II-D, we present the proposed method, which uses such techniques as its building blocks.

A. Descriptor Extraction

SIFT descriptor: SIFT is one of most popular image descriptors algorithms. Thanks to their discriminative power, it became standard for several tasks such as keypoint correspondence and object recognition. Lowe, in his landmark paper [7], presents SIFT to be used in object recognition applications, due to the high discriminative power and stability. In the first step in SIFT descriptor creation, each pixel around the keypoint location has the gradient magnitude and orientation estimated. Then, a region of 16×16 pixels, centred in the keypoint localization, is subdivided in 4×4 subregions. These 16 subregions are rotated relative to the canonical orientation. For each subregion, a histogram with 8 orientation bins is computed. The magnitude values for all gradients inside of the region are weighed by a Gaussian window and accumulated into the orientation histograms. The 8 bins of all 16 histograms are concatenated forming the 128-vector, which after normalization, represents the SIFT descriptor.

Although SIFT brings forth discriminative descriptors, it has a high processing cost. Furthermore, it suffers with slow match and high processing time and memory consumption (vectors with 128 and 64 floats respectively). Hence, it is not feasible in applications where it is necessary to store millions of descriptors or have real-time constraints.

In order to overcome these issues, several methodologies have been proposed. More recently, several compact descriptors, such as [8], [9], [10] and [11] have been proposed employing ideas similar to those used by Local Binary Pattern (LBP) [12]. The strategy adopted by those descriptors is based on using simple intensity difference tests, which have small memory consumption and modest processing time in the creation and matching processes.

The use of binary strings as descriptors has been used with promising results and one successful example of this methodology is the BASE descriptor [11].

BASE descriptor: The new simplified descriptor, called Binary Appearance and Shape Elements (BASE) [11], uses a circular patch with a fix radius of size 24 to select pairs of pixels and normals in the point cloud. In contrast to BRAND[10] and EDVD[13], BASE does not compute the canonical orientation. Similar to BRAND, the gradient information and geometrical features (based on the normal displacements) are combined using function 1:

$$f(\mathbf{x}_i, \mathbf{y}_i) = \begin{cases} 1 & \text{if } \tau_a(\mathbf{x}_i, \mathbf{y}_i) \vee \tau_g(\mathbf{x}_i, \mathbf{y}_i) \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

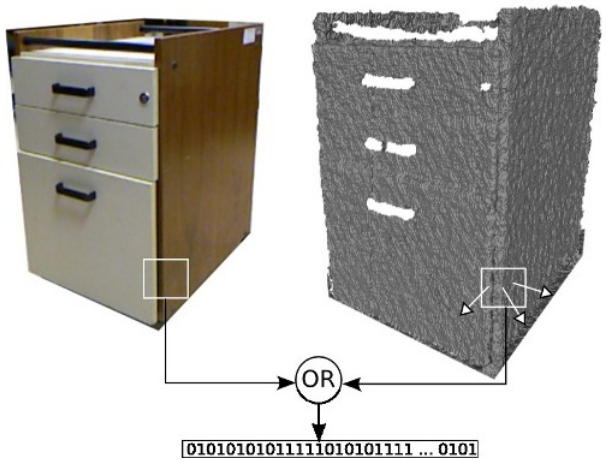


Fig. 1. Creation Diagram of BASE Descriptor. The difference in intensity of pairs of pixels in an image and the degree among the normal points are evaluated to create a binary vector. Diagram was extracted from [11].

where the function $\tau_a(.)$ captures the characteristic gradient changes in the keypoint neighborhood and $\tau_g(.)$ function evaluates the geometric pattern on its surface.

One of the benefits of this version is that it requires modest computational costs, since the steps to compute the canonical orientation and the keypoint scale are not performed. In spite of the simplicity of our descriptor, our experiments have shown robustness against small rotation and scale changes.

B. Bag of Features

We built a recognition system using the Bag of Features (BoF) approach [14] combined with Partial Least Squares (PLS) technique [15]. The main reason for using the BoF approach was that it is not possible to extract keypoints from the same location in differents samples. The choice of PLS, however, was due to good results in several recognition tasks such as human detection [16] and face recognition [17].

Like other recognition systems based on the BoF approach, our system is composed of four main steps:

1) **Feature extraction**: In this step, we split RGB-D images into a grid with approximately 150 cells and for each cell we compute a descriptor vector;
2) **Codebook creation**: After selecting a set of images of each letter in the data set and doing the feature extraction, we choose randomly K descriptors to be the clusters. Finally, we stack all of the K clusters creating a matrix of K rows. The number of columns is defined by the size of the descriptors, e.g. 32 columns of bytes using BRAND and BASE or 512 columns of bytes using SIFT. The number of clusters used in our experiments to build the codebook was $K = 1000$ for all descriptors;
3) **Bag of Feature vectors extraction**: For every image in the dataset, we computed a histogram of the number of descriptors assigned to each cluster. These histograms are called bag of features vectors and are used to represent each image in the codebook domain;
4) **Learning**: In this last step, we run the PLS algorithm with a set of bag of features vectors to build the

classification model. Since our recognition system uses the one-against-all scheme, we build a model for each class using the remaining samples of other classes as negative samples.

C. Partial Least Squares

The method Partial Least Squares (PLS) estimates predictor variables (latent variables) as a linear combination of the original predictors, represented by a sample matrix X (feature matrix), which contains one sample per row [18]. Vector y contains the responses associated with the samples (class labels for the sign recognition task).

Given an m-dimensional feature space associated with the class label for a sample, a set with N samples is represented by the feature matrix $X_{N\times m}$ and by the vector $y_{N\times 1}$. PLS decomposes the X and y as

$$\begin{aligned} X &= TP^T + E \\ y &= Uq^T + f \end{aligned} \tag{2}$$

where $T_{N\times p}$ and $U_{N\times p}$ stand for latent matrices containing p extracted latent vectors, the matrix $P_{m\times p}$ and the vector $q_{1\times p}$ represent the loadings, and $E_{N\times m}$ and $f_{N\times 1}$ store the residuals from the decomposition. In general, PLS employs the Nonlinear Iterative Partial Least Squares (NIPALS) algorithm to estimate a set of projection vectors $W = \{w_1, w_2..., w_p\}$. These vectors are estimated to maximize the covariance between the predictor and the response variables [19]. Thus, the PLS method focuses on the discrimination among the classes, providing a low dimensional feature space suitable for regression and classification.

To classify samples, we employ the *one-against-all* classification scheme based on PLS [20]. The learning procedure builds models for each of the N classes classes being considered, $C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N\}$. When the i-th class is considered, the remaining samples $C \setminus \boldsymbol{c}_i$ are set as negative examples of the i-th subject. Using the descriptors and the class label information, PLS gives higher weights to more discriminatory features when building each model. In the test, when an unknown sample is presented, its feature vector projected onto each one of these N models. The best match is the one associated with the PLS model with the highest regression response.

D. Proposed Solution

In our solution, for each letter in the manual alphabet, we select 50 images. Then, these images are sample using a uniform grid and a descriptor is extracted for every cell in the grid. Due to the difference of size among the images in the dataset, the number of cells in the image grid varies between 100 and 150 cells. Figure 2 shows an example of image with a grid in yellow lines.

After the extraction of the set of descriptors for the images selected, we perform a random selection of 1000 descriptors to build the dictionary. This dictionary is used to estimate the bag of features for each image.

For all remaining images in the dataset, we define the grid and, as well as for dictionary images, it is extracted a descriptor for each cell. Using the set of descriptors, which represent the grid image, the next step is to compute the codebooks based on the dictionary. A codebook represents an image in the dataset and it is defined as the centers of each clusters. These clusters are defined by the lines that compose our dictionary and each codebook (or bag of features) is represented by a histogram.

At last, the codebooks are used to train a model for every letter in the alphabet. We randomly select 40 codebooks for a letter and we test with all the remaining codebooks.

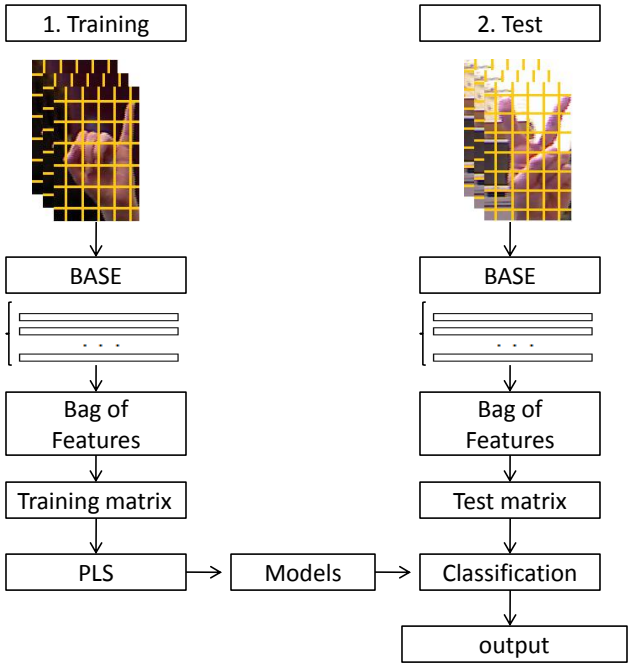


Fig. 2. Diagram illustrating the main stages of the proposed approach. It is depicted both training and test phases.

III. Experiments

In this section we show the performance of our recognition system using an American Sign Language (ASL). Our experiments were performed using the RGB-D fingerspelling data set presented in [21]. This data set contains 24 categories (it does not contain the letters J and Z), for a total of 5 subjects. The images were acquired with a Microsoft Kinect [6]. The color

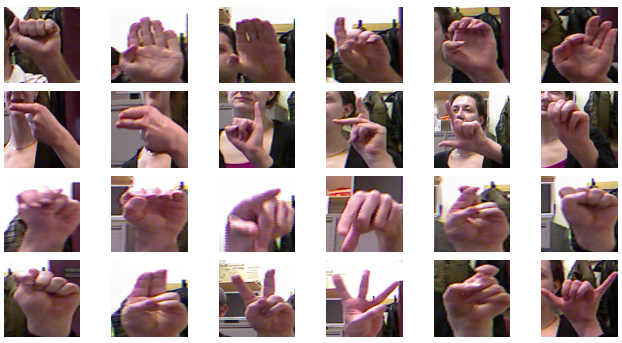
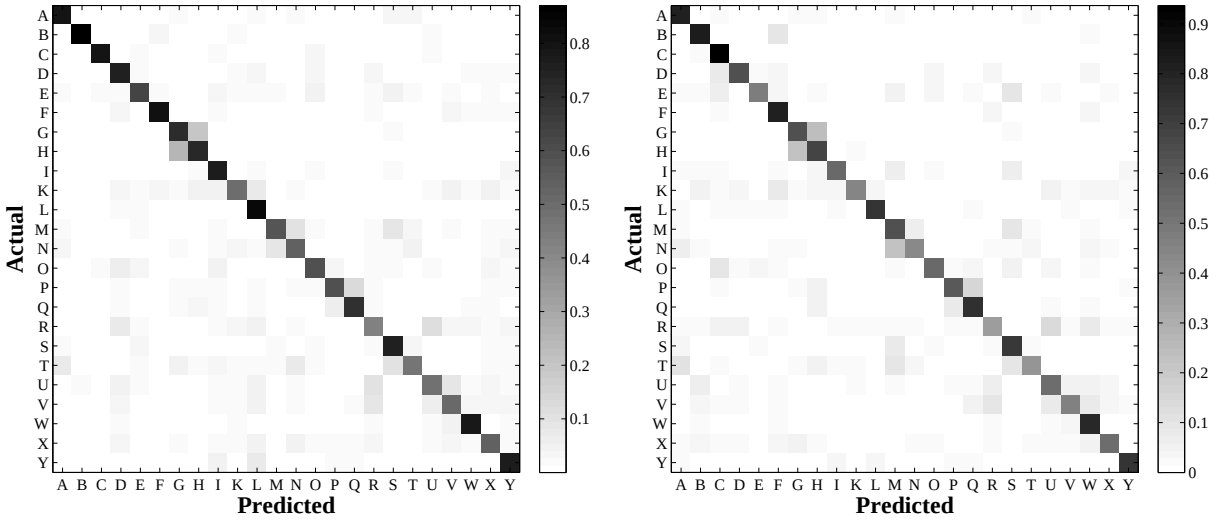
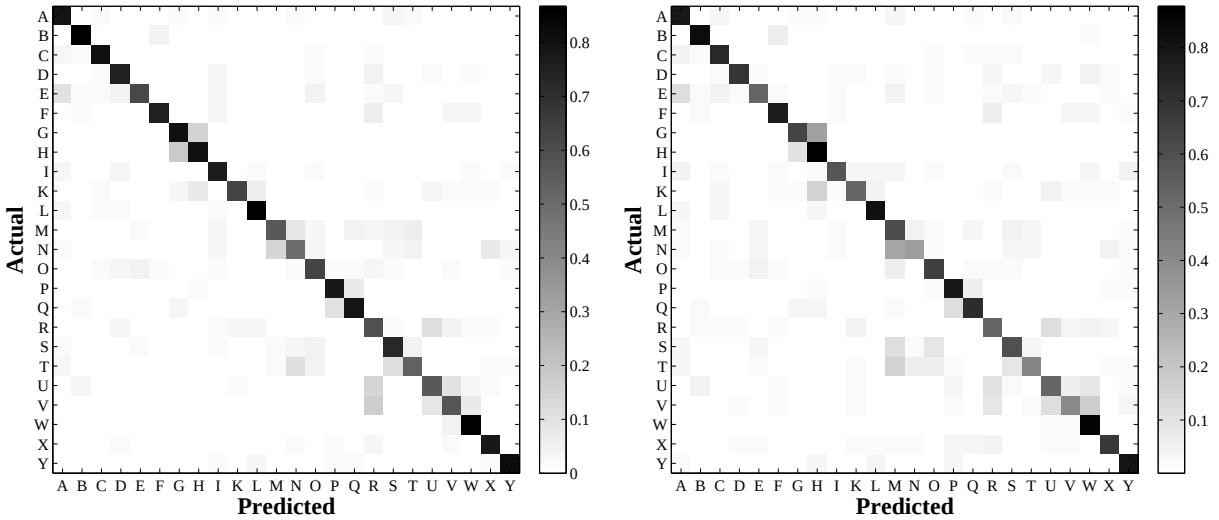


Fig. 3. Samples from ASL dataset [21] (only showing the RGB images, which had their aspect ratio changed to fit in a squared region for better visualization).



(a) PLS classifier (b) SVM classifier

Fig. 4. Confusion matrices (rows-normalized) among 24 classes from the ASL RGB-D Dataset [21] achieved using different classifiers with the BASE feature descriptor.



(a) PLS classifier (b) SVM classifier

Fig. 5. Confusion matrix (rows-normalized) among 24 classes from the ASL RGB-D Dataset [21] achieved using different classifiers with the SIFT feature descriptor.

and depth informations were simultaneously recorded. Figure 3 shows some samples of the signs present in the dataset.

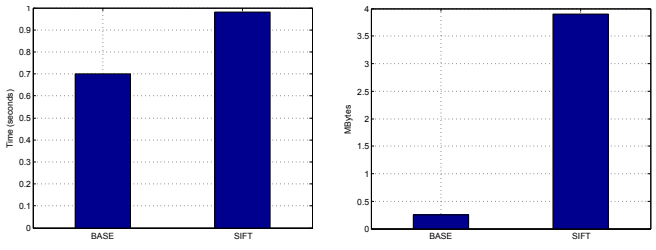
To perform the experiments, we follow the protocol defined in [2]. We split the dataset into two subsets: the training set, which contains 40 samples for each category and the test set with the remaining samples (approximately 460 samples per category per subject).

First, we compare the application of different classifiers using the BASE feature descriptor. Figure 4 shows the confusion matrices obtained for different classification methods, PLS and SVM. According to the results, the application of the PLS classifier achieved an accuracy of 66.27% while the SVM achieved an accuracy of 62.85%.

The second experiment compares the application of different feature descriptors. We have chosen to use the SIFT descriptor since it is widely used by the community. For this comparison, the setup is the same as in the previous experiment, the only difference is the feature descriptor employed. The achieved accuracy with the PLS and SVM were 71.51% and 65.55%, respectively. Comparing the results obtained by both feature descriptors, the SIFT-based approach achieved higher accuracy. The reason for that is the quality of the point clouds which represent each letter. We check and a large number of them present holes in the clouds, due to infrared shadows or environment illumination. Such lack of tridimensional information decrease the quality of normal estimation that are crucial for BASE descriptors.

Finally, we compare the usage of SIFT and BASE descriptors with respect to the computational cost and memory consumption. Figure 6 show the results. According to these values, even though the results achieved using SIFT present

are slightly higher accuracy, the use of BASE reduces the computational cost and memory consumption significantly, which is desirable when the goal is to run in real live videos, which requires real-time classification.



Computational cost Memory consumption

Fig. 6. Average processing time (over all keypoints) and memory usage to build the dictionary.

IV. Conclusions and Future Work

We proposed a framework based on bag of features strategy. This framework uses the method Partial Least Squares (PLS) to create models of the letters in the manual alphabet. We also extract features taking into account appearance and geometry from RGB-D images, which brings robustness to different illumination conditions.

The experiments have shown that the proposed approach is promising for sign language classification based on RGB and depth information. According to the results comparing the employment of SIFT and BASE feature descriptors, on the one hand SIFT achieves higher accuracy, but on the other hand the BASE allows the approach to runs faster and with low memory consumption. Further investigation is required to obtain the combination of feature descriptors, classifier and data type that achieves the best results in a low computational cost.

There are several possibilities of research in order to continue the work developed in this paper. First of all, strong results shown in the experiments have demonstrated the importance of using an appropriate strategy to combine texture and geometrical information. We show that even using a descriptor represented by 256 bits it was possible to obtain a similar accuracy when using descriptors with a 128 bytes of size.

One problem in the data used was holes (lack of 3D information) in the point clouds. As future work we intend to apply methodologies to tackle these holes filling or smooth their regions.

Another future work will be related to extend our recognition method to work with dynamic scenes, where the semantic is related to a sequence of movements and shape along the time. Several 4D features have been proposed in action recognition literature from 2D video and more recently there are some for tridimensional data that can be useful for sign language recognition.

Acknowledgment

The authors would like to thank the financial support of FAPEMIG, CAPES and CNPq.

References

[1] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. M. Campos, "STOP: Space-Time Occupancy Patterns for 3D Action Recognition from Depth Map Sequences," in *Iberoamerican Congress on Pattern Recognition (CIARP)*, 2012.

[2] X. Zhu and K.-Y. K. Wong, "Single-frame hand gesture recognition using color and depth kernel descriptors," in *International Conference on Pattern Recognition (ICPR)*, Nov., pp. 2989–2992.

[3] K. Lai, L. Bo, X. Ren, and D. Fox, "Sparse distance learning for object recognition combining rgb and depth information," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

[4] ——, "A large-scale hierarchical multi-view rgb-d object dataset," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

[5] E. Nascimento, G. Oliveira, M. Campos, and A. Vieira, "Improving Object Detection and Recognition for Semantic Mapping with an Extended Intensity and Shape based Descriptor," in *IROS Workshop on Active Semantic Perception*, USA, sep 2011.

[6] Microsoft, "Microsoft kinect," February 2011.

[7] D. G. Lowe., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision (IJCV)*, pp. 91–110, 2004.

[8] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *European Conference on Computer Vision (ECCV)*, September 2010.

[9] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[10] E. Nascimento, G. Oliveira, M. Campos, A. Vieira, and W. R. Schwartz, "BRAND: A Robust Appearance and Depth Descriptor for RGB-D Images," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[11] E. Nascimento, W. R. Schwartz, G. L. Oliveira, A. W. Vieira, M. Campos, and D. Mesquita, "Appearance and Geometry Fusion for Enhanced Dense 3D Alignment," in *Proceedings of Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, 2012.

[12] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51 – 59, 1996.

[13] E. Nascimento, W. R. Schwartz, and M. Campos, "EDVD – Enhanced Descriptor for Visual and Depth Data," in *International Conference on Pattern Recognition (ICPR)*, 2012.

[14] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[15] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," in *Subspace, Latent Structure and Feature Selection Techniques, Lecture Notes in Computer Science*. Springer, 2006, pp. 34–51.

[16] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis, "Human Detection Using Partial Least Squares Analysis," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 24–31.

[17] W. R. Schwartz, H. Guo, and L. S. Davis, "A Robust and Scalable Approach to Face Identification," in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, vol. 6316, 2010, pp. 476–489.

[18] H. Wold, "Partial Least Squares," in *Encyclopedia of Statistical Sciences*. New York, NY, USA: Wiley, 1985, vol. 6.

[19] R. Rosipal and N. Kramer, "Overview and Recent Advances in Partial Least Squares," vol. 3940, pp. 34–51, 2006.

[20] W. R. Schwartz, H. Guo, J. Choi, and L. S. Davis, "Face Identification Using Large Feature Sets," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2245–2255, 2012.

[21] N. Pugeault and R. Bowden, "Spelling It Out: Real-Time ASL Fingerspelling Recognition," in *Proceedings of the 1st IEEE Workshop on Consumer Depth Cameras for Computer Vision, jointly with ICCV'2011*, 2011.