ROBUST RGB-D HAND GESTURE RECOGNITION SYSTEM USING MULTITHRESHOLDING

S.Sangeetha[1] | R.Sujitha[2]

[1](UG Student, Christ the King Engineering College, sangeesaro3@gmail.com)
[2](Assistant Professor, Christ the King Engineering College, srisuji14@gmail.com)

Abstract— *Gesture analyzing procedure seems to be more interesting domain nowadays. The problems to face respect other tracking algorithms are mainly two: the high complexity of the hand structure which translate in a very large amount of possible gestures, and the rapidness of the movements we are able to make when moving the hand or just the fingers. Recent approaches try to fit a 3D hand model to the observed RGB-D data by an optimization function that minimizes the error between the model and the data. However, these algorithms are very dependent of the initialization point, being unpractical to run in a natural environment. Here we introduce some new algorithm that is enabled to solve all the problems related with hand gesturing processes. Segmentation based multi threshold systems are introduced. We present an accelerometer based smart ring and a similarity matching based extensible hand gesture recognition algorithms. Existing scenarios do arise some drawbacks. To solve these kind of problems, it is common to use an offline dataset with pre-learnt gestures that will serve as a first rough estimate. In concrete, we present an algorithm that uses an articulated ICP minimization function, that is initialized by the parameters obtained from a dataset of hand gestures trained through deep learning framework. This set up has two strong points. First, deep learning provides a very fast and accurate estimate of performed hand gesture. Second, the articulated ICP algorithm allows to capture the possible variability of a gesture performed by different person or slightly different gesture. Our proposed algorithm is evaluated and validated in several ways. Independent evaluations for deep learning framework and articulated ICP are performed. Moreover, different real sequences are recorded to validate our approach, and finally quantitative and qualitative comparisons are conducted with state-of-the-art algorithms.*

Keywords— Hand Gesture, Hand Recognition, Tracking, Deep Learning, Iterative Closest Point Algorithm

1. INTRODUCTION

Hand gesture tracking has attracted many researchers in recent years in order to obtain a reliable hand tracking system that could provide a more intuitive way to interact with computers or other devices. However, the problems to face in order to complete such system are not minor and have a lot of common in retrieving 3D structures . Some of these problems are: high degrees of freedom, self-occlusions, processing speed, uncontrolled environments, and rapid hand motion. Most of these problems are common to any tracking system, but high degrees of freedom and rapid motions are exclusive to hands, which makes hand gesture still a challenge as of today . Scheme for the proposed algorithm. The algorithm can be divided into four different components: 1) The input: a synchronized color and depth images provided by Intel depth camera, 2) The learning module: that estimates the hand gesture from the input depth image and gives a parameter approximation for the hand pose, 3) The registration module: that aligns the hand mesh model with the observed cloud of points of the hand and 4) The output: a hand mesh model that corresponds to the input hand gesture.

In a normal interaction between persons or communication with a computer we usually move hands in a quite fast manner, especially the fingers, which makes hand gesture tracking very difficult. A first approach to address this issue is using Discriminative (appearance-based) methods, where the problem is formulated as a database indexing problem. Despite that these approaches are usually fast and also able to deal with rapid hand motions, they fail to retrieve unseen gestures that are not pre-recorded in the dataset . A second approach is the use of Generative (model-based) methods, which can better deal with high-dimensional data and uncontrolled environments by fitting a 3D model to the observed data. However, they are usually slow and have problems tracking rapid hand motions . The last approach is the Hybrid methods which generally combine the best of the discriminative, used as parameter initialization, and generative methods.

Hence, our proposed work develops a method for integrating the advantages of both the discriminative and generative methods also in a Hybrid fashion. the proposed hybrid method has a two- stage cascaded architecture, where a discriminative method of the first stage is exploited to give a reliable parameter approximation of the hand pose in an input image. This approximation facilitates the initialization of the generative method of the second stage to obtain an accurate tracking result. Technically, we propose to combine the convolutional neural network (CNN) with the iterative closest point (ICP) algorithm for constraining the parameter prediction. Overall, the novelty of the proposed idea responds to address a key issue of assuring a good and reliable independent parameter initialization for the ICP algorithm, which can be problematic under two circumstances: 1) the hand gesture contains occlusions and hence the initial parameters are not good enough and 2) the hand pose changes largely from frame to frame such that parameters from the previous frames are not indicative enough of the current pose parameters. Compared to the state-of- the-art hand gesture tracking algorithms, our proposed approach is able to handle the hand and finger motion better. Tracking loss and tracking errors can be recovered shortly because independent parameter initialization is applied for each individual frame. In addition, partially-occluded hand gestures can also be correctly detected by including occluded training samples in the learning stage.

2. RELATED WORKS

In this section, we introduce relevant hand tracking papers as well as discuss the advantages/disadvantages of Discriminative, Generative and Hybrid methods.

2.1 Discriminative Methods

These algorithms are often formulated as a database indexing problem. Generally, each database instance is labeled with a gesture and orientation of the hand respect to the camera and the nearest neighbor search will give the most approximated gesture

and orientation parameters respect to the observed image. These methods are typically limited by the training data, which often does not generalize very well being unable to capture all the variability of the poses of the hand. For example, proposes to segment the hand into several regions and matches the different segmented parts by using a regressor to vote the best candidates. At the same time, for each segment, the correlation between the different hand parts is explored. Other examples are using neighbouring search, or using convolutional networks.

2.2 Generative Methods

These algorithms can be grouped into two categories. The first category is the methods using some pre-learned gestures to match different observed cues, typically optical flow combined with edges and silhouettes. Similar approaches are using belief propagation as a structure to find the gesture match, based on sequential Monte Carlo techniques and using hierarchical Bayesian filter tracking, which encodes a tree structure to accelerate the matching and rapidly discard gestures which are distant to the observed one. Despite the pre-learnt information and methods used to accelerate the recognition process, the execution time is still slow (several seconds per iteration). All these methods, like the methods mentioned before, are also limited by the database size, which usually, contains few gestures, thus recognition accuracy is high for gestures similar to the registered ones in the database, lower otherwise.

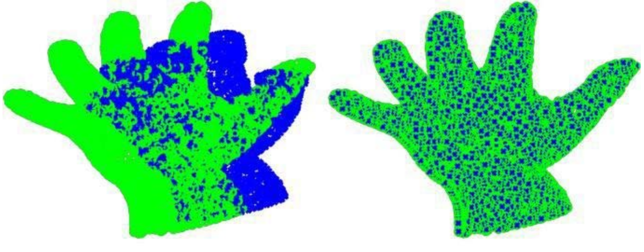
The second category is the methods formulated as an objective function minimization, that can deal better with not pre-learnt gestures. As an example, retrieves the 3D hand pose, hand texture and illuminant for monocular images by minimizing an objective function. The hand model is deformed according to an underlying skeleton model and the hand parameters are estimated to produce a synthetic image that best matches the hand observation. However, in this method, as well as all aforementioned methods, the lack of depth information results in ambiguity problems for some gestures. To solve this problem, uses multiple camera setups. The algorithm detects some salient points together with other visual cues as edges and optical flow, that are computed in a pre-processing stage. In this particular case, the minimization function tries to estimate the parameters of two interacting hands with 35 degrees of freedom (DoF) per hand. Tracking two interacting hands, but with a depth sensor. Here the minimization function used is a particle swarm optimization (PSO) method, which tries to fit the observations from skin color segmented image with a high dimensional model for the two hands. The work is an extension of from same authors where only a single hand is tracked. A real time hand gesture tracking is achieved by . However, in this method, no objective function is minimized, but a simple part matching method is applied to fit each hand polyhedron to the point cloud data. A major problem of these methods is how to obtain a good initialization point for the algorithm to converge. The most common procedure is to manually set up in the first frame the initialization and for the consecutive frames, parameters from previous frames are simply used for initialization. This approach assumes temporal coherence and movement smoothness, but as discussed previously, it is not necessarily true in the case of hand gestures.

2.3 Hybrid Methods

These methods combine the properties of discriminative and generative methods, which is useful, especially in the case of articulated bodies where complexity is high and advantages of a hybrid method is more noticeable. From a monocular image, pose estimation for the articulated body is formulated as a statistical inference problem. Similarly, has estimated also the shape. In most of the cases, the search range for estimating the model parameters is too large and discriminative methods are used to constrain the parameter prediction. In work of depth images are used, which leads to better pose estimations because of that depth data are less prone to ambiguities than 2D images. More recently, a multicamera approach focusing specifically on hand gesture recognition . The approach uses a voting scheme from the discriminative and generative part.

However, the most common problem for hand gesture tracking is to find a proper initialization point in the objective function that minimizes the error between the 3D hand model and the observed data. Many studies have proposed different methodologies that attempt to find the best way to initialize the objective error function. Based on the proposed work suggests to keep the PSO minimization technique but uses a decision binary tree pre-trained with several hand gestures to find the best hand pose candidate for the observed data. Similarly, combines the ICP algorithm with a PSO to first find a rough estimation of the pose to later refine the gesture pose. Since again the minimization function is very dependent from the initialization point, it proposes to detect the fingertips of the hand and uses that as the initial position. However, fingertip detection might not be so easy for those gestures which contain occlusions or difficult hand orientations. Therefore, proposes to detect a color wristband as a first alignment and then uses several cues such as, PCA priors to estimate the space of hand gestures, the silhouette of hand, temporal priors among others to finally minimize an energy function that estimates the pose to the observed data. The use of external cues for initialization can be sometimes impractical. For this reason, most of the previous studies concentrate on exploiting priors from the given image data and then among those priors, select the most plausible hand pose.

Apart from the above explained state-of-the-art approaches, some other approaches have been evolved during the recent years which have focused on the dimensional hand tracking for improved human-computer interactions. In one of the work by Tang et al. it has improved the drifting issues in hand movement by identifying the motion in 3D space. The method is heuristic- based and provides robust hand detection. In Taylor et al. , it has used the Pose and Correspondences to improve the hand coordination for precise hand tracking. Quantitative assessments are provided on the data set traced using the efficient depth cameras. Chi et al. have illustrated the entire body movement using the visually refined parameters. They have also illustrated the 3D joint displacement in their developed approach. However, lack of proper and robust modeling is the weak part of this approach which can be overcome by the inclusion of advanced visual technologies. One of the applications of these approaches can be the provisioning of hands-free systems which facilitate the fast access to major of the user applications . Apart from these, Choi et al. have developed a 21 degree of freedom model for hand pose tracking. A step by step approach is used



(a) Before Registration (b) After Registration

(a)starting from a source (blue) and target (green) 3D point cloud, the registration algorithm tries to minimize the distance between the two point clouds. The result of the process is shown in (b). Note that we have to account for rotations, translations and deformations of the hand by the authors to overcome the ambiguity in the hand tracking. Piece-wise planar hand model is used to generate the motion of the hand. Incorporation of the 3D facilities can provide further improvement in this approach.

It can be observed from the existing approaches that these might provide efficient hand tracking, but are not robust enough to provide correct readings in case of drift or change of motion. Thus, efficient algorithms and strategies are required to overcome these issues and provide an accurate, precise and robust hand tracking. The work presented in this paper have some points in common and also some differences to the recent state-of-the-art algorithms. We propose to use an articulated ICP algorithm as a method to fit a 3D hand model with the 3D point cloud obtained from a short-range depth sensor. However, as an algorithm to initialize our minimization function we differ from that uses a decision forest, or that uses a PCA, but we use instead of deep learning framework to model the different hand gestures. This deep learning framework provides the necessary parameters for an independent per-frame initialization of the ICP algorithm, which allows us to handle rapid hand motions and hand gestures with occlusions. Due to the machine learning nature, the main limitation of the proposed hand gesture initialization is produced when the user performs a hand gesture that is not pre-trained by the deep learning framework and it is not similar to any of the pre-trained ones. In that case, the ICP algorithm is not able to converge to the correct hand gesture.

2.4 Tracking hand gestures

The hand gesture tracking can be presented as a minimization of an objective function, which tries to register the observed data (in our case a 3D point cloud) to a 3D mesh hand model. As it is well known, algorithms that solve such problems are prone to get trapped in local minima due to nonconvexity and non-linearity of such functions. Traditionally this problem is overcome using manual initialization and assuming temporal coherence for the rest of the frames. The most common way to fit a 3D point cloud to a mesh model is the ICP algorithm . This method can solve the registration problem for rigid bodies in a closed form solution. For articulated or non-rigid bodies, the algorithm needs to be modified. In the articulated object is treated as a composition of separated rigid objects, and minimization is carried out to fit each individual part. In contraposition, the approach used in , all parameters of the model are minimized at once.

In this work the hand model used is taken from [41] and corresponds to a mesh model of about 70k triangles controlled by a skeleton. The skeleton has 19 joints and each joint has 3 degrees of freedom (DoF). Despite the hand model can have some configurations that are physically impossible for a human hand, we do not apply any constraints on the parameters in our work.

Let us define a set of Nd data points d ∈ R3 given by a depth camera sensor and a set of Nm model points m ∈ R3 that are the mesh vertices of the 3D hand model defined by [41]. This hand model is a mesh that its deformations are controlled by an underlying skeleton of Nn joints, each one with three degrees of freedom (DoF). Therefore, the mesh hand model gesture and position is defined by the world rotation RW and translation TW and the deformation parameters.

2.5 Learning hand gestures

The algorithm described in Section III-A needs to be initialized with the model parameters φ to start all the process. Usually, an initialization for the current frame uses the parameters in the previous frame. Since a hand tends to have constantly fast movements and is easy to get lost of the track, the parameters obtained in the previous frame might be not good enough. Therefore, we use an independent initialization for each frame given by pre-learnt gestures from a recorded.

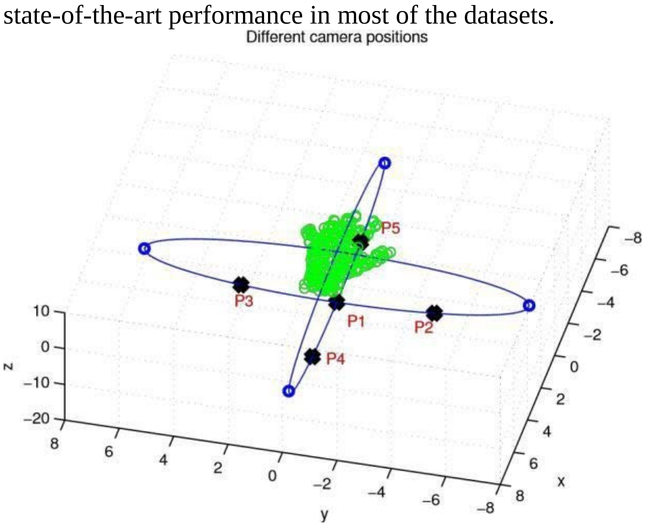
A good dataset that captures most of the usual gestures that user can perform is important to later provide reliable initialization parameters to the optimization algorithm.

These gestures should not be very close to each other, but at the same time not too far if we want that the optimization algorithm produces a reasonable hand gesture close to the observed one after convergence. Exists extense literature on how a hand dataset should be designed to comprise the most common hand gestures done by a person. Conventionally, these hand gestures are designed in a way that a human subject will perform the gesture orienting the hand in a perpendicular position with respect to the camera. While this assumption can be true in the case of sign language, it is not very common for humanmachine interaction, where the user expects to move the hand with his/her body in a more natural manner. To this end, the selected dataset defines 27 basic gestures (see Fig.

3) and two rotated counterparts about x and x,z axes for each basis gesture, which makes a total of 81 different classes. It is recorded with an Intel depth camera, which is a short-range camera and its operating distance goes from approximately 15 cm to 90 cm. The camera provides synchronized RGB images of VGA resolution and the corresponding depth images of QVGA resolution. Each class is then recorded by 10 subjects (5 males and 5 females) and each of them provides 300 images per class by repeating the same hand gesture with slight movements. The total size of the dataset is of 243,000 (10×81×300) tuples of images constituted by a color, a depth and a mask of the hand region.

2.6 Convolutional Neural Networks

Convolutional neural network (CNN) is one of the methods that can be comprised with the term of deep learning. These kinds of methods have been discovered to be as efficient as SVMs, HMMs or graph modeling with a high potentiality to distinguish into a much larger number of classes than just a few dozen of gestures using SVMs. CNN is one of the most suitable methods to recognize hand gestures. One of the major advantages is that this method does not need handcrafted features, being able to select automatically the best features during the learning process. It is demonstrated by different experiments that it reaches state-of-the-art performance in most of the datasets.



2.7 Experiments with real data

The purpose of these experiments is to evaluate the algorithm in real conditions and verify that the independent frame initialization is useful in case of rapid movements to avoid loss of track and if this happens, to be able to be on track again after some frames without re-initializing the whole system. Real sequences are evaluated quantitatively and qualitatively and compared with the same algorithm assuming temporal coherence and with two existing state-of-the-art algorithms. We first introduce briefly the datasets used to perform the quantitative and qualitative evaluations and then we present the results obtained in those evaluations.

A. Datasets

Dexter dataset. This dataset consists of 7 sequences recorded with a multicamera setup. In concrete, there are 5 RGB cameras, 1 TOF camera and 1 Kinect camera. The sequences are about 250 frames, with each sequence from a single actor performing several gestures at different speed motions. NYU Hand dataset. This dataset contains around 80,000 images of hand gestures performed by two actors in front of 3 Kinect cameras. Ground truth for the hand articulated skeleton is provided. Fast hand gestures dataset. Two sequences are recorded by three different persons. The first sequence lasts 20 seconds and the actor performs several hand gestures in a slow fashion. The hand gestures are chosen freely. The goal of this sequence is to evaluate the independent initialization given by the learning module. The second sequence lasts 10 seconds and, the actor counts from five to zero with the fingers closing the hand very fast in every transition between the numbers. In this sequence the actors are instructed on how to perform the gestures. The purpose of this sequence is to test the algorithm when rapid hand movements occur. Additionally, this sequence allows us to evaluate independent frame to frame initialization against initialization with the previous frame parameters.

All sequences are recorded at 25 fps (frames per second) with the Intel depth camera, which incorporates already the calibration parameters in the software allowing to extract the 3D information directly.

B.Quantitative Results

For the Dexter dataset we define a pose error as the Euclidean distance between fingertips and the palm center with respect to the given ground truth positions. The error is measured in millimiters (mm) and used to evaluate and compare our algorithm with other methods. In Table II we give the error mean values for each one of the sequences of the Dexter dataset.The last two methods are proposed by the same authors and one of the main differences between them is the number of cameras used in the recording setup. The latter is a multicamera setup, which makes it possible to exploit multi-view information and obtain a better performance than all the other methods.For the NYU hand dataset, we obtain a mean error of 11.1 when run for the first 2,000 frames of the test data as in Taylor

If we take as reference the results by Taylor, our results are still quite competitive. They obtain an error lower than 10 for 60% of the frames, while Tagliasacchi and Tompson obtain an error of 11 and 23, respectively. We have an error of 10.6 for 60% of the frames.

we show the per-frame error of Dexter dataset and NYU hand dataset. In the Dexter dataset we can observe some error peaks due to misclassification of our learning

3. ADAPTIVE BACKGROUND SUBTRACTION

Background subtraction, also known as foreground detection, is a technique in the fields of image processing and computer vision wherein an image's foreground is extracted for further processing (object recognition etc.). Generally an image's regions of interest are objects (humans, cars, text etc.) in its foreground. After the stage of image preprocessing (which may include image denoising, post processing like morphology etc.) object localisation is required which may make use of this technique.

Background subtraction is a widely used approach for detecting moving objects in videos from static cameras. The rationale in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called "background image", or "background model". Background subtraction is mostly done if the image in question is a part of a video stream. Background subtraction provides important cues for numerous applications in computer vision, for example surveillance tracking or human poses estimation.

Background subtraction is generally based on a static background hypothesis which is often not applicable in real environments. With indoor scenes, reflections or animated images on screens lead to background changes. Similarly, due to wind, rain or illumination changes brought by weather, static backgrounds methods have difficulties with outdoor scenes.

A robust background subtraction algorithm should be able to handle lighting changes, repetitive motions from clutter and long-term scene changes.

4. MULTI THRESHOLDING

A novel algorithm is proposed for segmenting an image into multiple levels using its mean and variance. Starting from the extreme pixel values at both ends of the histogram plot, the algorithm is applied recursively on sub-ranges computed from the previous step, so as to find a threshold level and a new sub-range for the next step, until no significant improvement in image quality can be achieved. The method makes use of the fact that a number of distributions tend towards Dirac delta function, peaking at the mean, in the limiting condition of vanishing variance. The procedure naturally provides for variable size segmentation with bigger blocks near the extreme pixel values and finer divisions around the mean or other chosen value for better visualization. Experiments on a variety of images show that the new algorithm effectively segments the image in computationally very less time.Thresholding is an important technique for image seg-mentation. Because the segmented image obtained from thresholding has the advantage of smaller storage space, fast processing speed and ease in manipulation, compared with a gray level image containing 256 levels, thresholding techniques have drawn a lot of attention during the last few years. The aim of an effective segmentation is to separate objects from the background and to differentiate pixels having nearby values for improving the contrast. In many applications of image processing, image regions are expected to have homogeneous characteristics (e.g., gray level, or color), indicating that they belong to the same object or are facets of an object, implying the possibility of effective segmentation.

5. SEGMENTATION

In computer vision , image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels , also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color , intensity , or texture. Adjacent regions are significantly different with respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms likeMarching cubes .

6. CLASSIFICATION

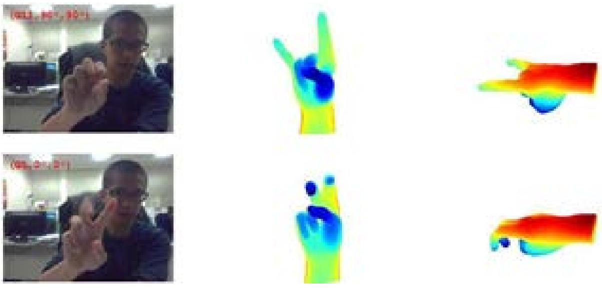
In computer vision , image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels

, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image (see edge detection). Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color , intensity , or texture. Adjacent regions are significantly different with respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms likeMarching cubes .

7. CONCLUSION

We presented an algorithm for hand gesture tracking that faces the difficulties to match a highly complex model (a high degree of freedom, i.e. DoF > 64) to 3D point cloud data provided by a depth sensor. To perform the matching, the articulated ICP algorithm needs good initialization parameters, which are obtained by pre-learning a complete set of gestures offline. The pre-learned hand gestures are done using deep learning framework, where a high average recognition rate is achieved. The proposed algorithm is evaluated quantitatively in synthetic images and finally tested on real and challenging sequences, where a comparison with state-of-the-art methods are also conducted. We demonstrate that the articulated ICP with good initialization point can provide reasonable performance results. Moreover, the idea of using deep learning framework to obtain initial parameters for the optimization algorithm is effective in the sense that it allows



REFERENCES

[1] M. Wang, Y. Gao, K. Lu, and Y. Rui, “View-based discriminative probabilistic modeling for 3d object retrieval and recognition,” IEEE Trans. Image Processing, vol. 22, no. 4, pp. 1395–1407, 2013.

[2] M.-C. Hu, C.-W. Chen, W.-H. Cheng, C.-H. Chang, J.-H. Lai, and J.-L. Wu, “Real-time human movement retrieval and assessment with kinect sensor,” IEEE Transactions on Cybernetics, vol. 45, no. 4, pp. 742–753, 2015

[3] ICPR2016. Joint contest on multimedia challenges beyond visual analysis. [Online]. Available: http://gesture.chalearn.org/icpr2016 contest

[4] J. Romero, H. Kjellstrom, and D. Kragic, “Monocular real- time 3D¨ articulated hand pose estimation,” in Humanoids, 2009.

[5] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, “Markerless and efficient 26-dof hand pose recovery,” in ACCV, 2010.

[6] J. Sanchez-Riera, Y.-S. Hsiao, T. Lim, K.-L. Hua, and W.-H. Cheng, “A robust tracking algorithm for 3D hand gesture with rapid hand motion through deep learning,” in 2014 IEEE International Conference on Multimedia and Expo Workshops, 2014

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient- based learning applied to document recognition,” in Proceedings of the IEEE, 1998, pp. 2278–2324.

[8] A. W. Fitzgibbon, “Robust registration of 2D and 3D point sets,” in BMVC, 2001

[9] H. Liang, J. Yuan, and D. Thalmann, “Resolving ambiguous hand pose predictions by exploiting part correlations,” IEEE Trans. Circuits Syst. Video Techn., vol. 25, no. 7, pp. 1125– 1139, 2015.

[10] V. Athitsos and S. Sclaroff, “Estimating 3D Hand Pose from a Cluttered Image,” in CVPR, 2003.

[11] N. Shimada, K. Kimura, and Y. Shirai, “Real-Time 3D Hand Posture Estimation Based on 2D Appearance Retrieval Using Monocular Camera,” in ICCV Workshop (RATFG-RTS), 2001.