Interpretation of divers' symbolic language by using hidden Markov models

Mario Menix, Nikola Mišković and Zoran Vukić

University of Zagreb Faculty of Electrical Engineering and Computing
Laboratory for Underwater Systems and Technologies
Unska 3, Zagreb, Croatia

E-mail: {nikola.miskovic, zoran.vukic}@fer.hr

Abstract - This paper reports the results of a system based on hidden Markov models (HMM) that is used to interpret both static and dynamic divers' hand signals using real time video feed. Two methods of collecting features that describe diver gestures are described and two types of HMMs are investigated: one based on discrete outputs variable distribution and the other based on mixture of Gaussians outputs variable distribution. For 8 basic diver gestures, 800 data samples were collected and the results were analyzed for the purpose of determining the most appropriate *i)* feature vector describing diver gestures, *ii)* HMM parameters (number of states and mixture components), and *iii)* type of the hidden Markov model. Finally, the quality of performance for recognition of each gesture is evaluated and analyzed, and necessary steps for improving the overall system are reported.

I. INTRODUCTION

Divers operate in a very challenging environment where even the slightest unexpected situation can have catastrophic results. Professional divers are usually involved in risky underwater operations some of which include surveying a part of a seabed (search and recovery missions), collecting samples (marine biology missions), documenting a site (marine archaeology missions) or performing inspections of underwater facilities (underwater inspection and maintenance missions). Having in mind the dangers that divers are exposed to during their activities, in the last couple of decades, underwater robots are being put to use more frequently. However, application of autonomous underwater vehicles or remotely operated vehicles still cannot completely replace a human diver, especially during highly sophisticated operations such as intervention, repair, etc.

The FP7 CADDY project (Cognitive Autonomous Diving Buddy) recognizes the necessity for a human diver and has a goal to develop a multicomponent robotic system, consisting of a surface autonomous marine platform and an autonomous underwater vehicle that will help divers during their underwater activities by performing a role of a diver "guide" (leading the diver through the underwater), diver "observer" (monitoring diver behaviour), and diver "slave" (helping the diver perform different set of tasks). In order to achieve these roles, establishing a communication channel between the diver and the buddy robot is of high importance. Since divers usually communicate using a predefined set of hand gestures, it is natural to use the same symbolic language as means of communication between divers and robots underwater.

Gesture recognition is an intriguing research area and different methods such as hidden Markov models (HMM), conditional random fields, particle filtering and condensation, finite-state machine as statistical modeling, optical flow, skin colour, connectionist model, etc. are being investigated, [5]. Among these methods, HMM has proved to be the most frequent tool, [2-7]. It has been successfully applied for spatial-temporal processes such as speech recognition [2], protein modeling [6] and gesture recognition [4].

This paper focuses on the application of the HMMs for interpretation of divers' hand signals using real time video feed. In this paper, we use the output mixed Gaussian distribution in HMM to improve the recognition rate.

The paper is organized as follows. The continuing part of Section II briefly describes the image processing system used to obtain gesture features. Section III focuses on the HMM theoretical background directly related to the applied approach. Section IV gives details on the specific problem of interpreting diver gestures and Section V describes experiments and provides analysis of the results. The paper is concluded with Section VI.

II. IMAGE PROCESSING SYSTEM

The image processing system has been mostly adopted from the publicly free-to-use system described in [1]. The software locates important hand features accurately and increases recognition tolerance to different hand rotations and tilt. A feature vector provided by the software as output includes three components: x and y coordinates of the palm center in the video frame, and the number of fingers that are stretched out and currently visible. All three of the mentioned features are of high importance for gesture recognition.

The research leading to these results has received funding from the European Union Seventh Framework Programme under FP7-ICT project "CADDY - Cognitive Autonomous Diving Buddy" Grant Agreement Number: 611373.

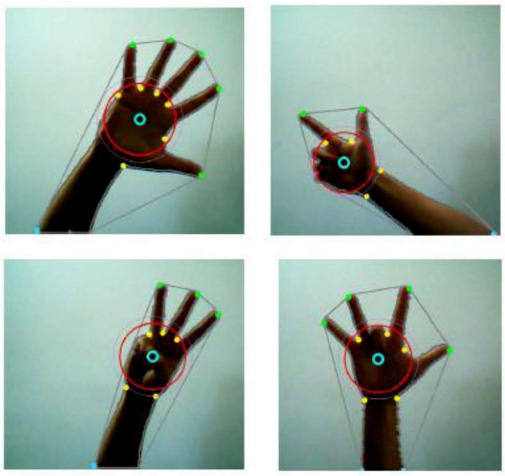

Figure 1. Snapshots from [1] showing hand images as recognized by the image processing system.

The software takes an RGB image captured by a web camera as an input and processes it to generate a binary image which provides enough information of hand contour. The binary image is used to calculate the contour and the convex hull of the contour. The palm position can be initially estimated by the information extracted from convex hull, and the fingertips' position can be detected from the hand contour. Examples of the results obtained from the image processing software are shown in Fig. 1.

III. HIDDEN MARKOV MODELS

Here we briefly review the theory of hidden Markov models (HMM) and some problems related to HMM training and human gesture recognition. Detailed description of the conventional HMM can be found in [2].

HMM is a statistical Markov model in which the system being modelled is assumed to be a Markov process with unobserved (hidden) states. HMM is characterized with *N* (number of states in the model, where individual states are denoted as *S*), *M* (the number of distinct observation symbols by state, where individual symbols are denoted as *V*) and $\lambda = (A, B, \pi)$ where,

$A = \{a_{ij}\}$ is the state transition matrix with

$$a_{ij} = P\left[q_{t+1} = S_j | q_t = S_i\right], \quad 1 \le i, j \le N, \quad (1)$$

$B = b_j(k)$ is the observation symbol probability distribution

$$b_j(k) = P\left[v_{k,t} | q_t = S_j\right], \quad 1 \le j \le N; 1 \le k \le M, \quad (2)$$

and $\pi = \{\pi_i\}$ is the initial state distribution

$$\pi_i = P\left[q_1 = S_i\right], \quad 1 \le i \le N \quad (3)$$

where $S = \{S_1, S_2, \ldots, S_N\}$ is a set of states, q_t is a state at time *t*, and $V = \{v_1, v_2, \ldots, v_M\}$ is a set of symbols.

Given the observation sequence $\boldsymbol{O} = O_1 O_2 \ldots O_T$ and a model $\lambda = (A, B, \pi)$, the task is to efficiently compute the probability of the observation sequence $P(O | \lambda)$ with the given model (Problem 1), and to adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O | \lambda)$ (Problem 2).

The observation symbol probability distribution $P\left[v_{k,t} | q_t = S_j\right]$ can consist of discrete symbols or continuous variables. If the observations are discrete symbols, the observation model can be represented as a matrix in the form

$$B(i,k) = P\left[v_{k,t} | q_t = S_i\right]. \quad (4)$$

If the observations are vectors, it is common to represent $P(O_t | q_t)$ with a Gaussian:

$$P\left[v_{x,t} | q_t = S_i\right] = N\left(\boldsymbol{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \quad (5)$$

$$N\left(\boldsymbol{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = \frac{1}{\sqrt{(2\pi)^d}\sqrt{|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}. \quad (6)$$

A more flexible representation is a mixture of M Gaussians

$$P\left[v_{x,t} | q_t = S_i\right] = \sum_{m=1}^{M} P\left[M_t = m | q_t = S_i\right] \times N\left(x_{m,i} | \mu_{m,i}, \Sigma_{m,i}\right) \quad (7)$$

where M_t is a hidden variable that specifies which mixture component to use and $P\left[M_t = m | q_t = S_i\right] = C(i,m)$ is conditional prior weight of each mixture component. In our approach, we implement both discrete and mixture of Gaussians outputs variable distribution.

A. Solution to Problem 1

In order to calculate the probability of the observation sequence $\boldsymbol{O} = O_1 O_2 \ldots O_T$ with the given model $\lambda = (A, B, \pi)$ the most straightforward way of doing this is through enumerating every possible state sequence of length *T* (the number of observations), but fortunately there is more efficient procedure called *forward-backward* procedure. Solving this problem will provide us with the answer how to successfully recognize which trained HMM model (trained gesture) is the closest match to our observation sequence (gesture).

The general approach will be to iteratively update the weights. We do this by defining

$$\alpha_t(i) \triangleq P(\boldsymbol{o}_1 \boldsymbol{o}_2 \ldots \boldsymbol{o}_t, q_t = S_i) \text{ and} \quad (8)$$

$$\beta_t(i) \triangleq P(\boldsymbol{o}_{t+1} \boldsymbol{o}_{t+2} \ldots \boldsymbol{o}_T | q_t = S_i) \quad (9)$$

where $\alpha_t(i)$ is the probability that model is in state q_i at step *t* having generated the first *t* elements of sequence $\boldsymbol{O}$, and $\beta_t(i)$ is the probability that the model is in state $q_i(t)$ and will generate the remainder of the given target sequence, i.e., from $t+1 \to T$.

We can solve both for $\alpha_t(i)$ and $\beta_t(i)$ inductively (initialization – (10) and (12), induction – (11) and (13), and termination – (14)) :

$$\alpha_1(i) = \pi_i \cdot b_i(\boldsymbol{o}_1), \quad 1 \le i \le N \tag{10}$$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\boldsymbol{o}_{t+1}), 1 \le t \le T-1, 1 \le j \le N \tag{11}$$

$$\beta_t(i) = 1, \quad 1 \le i \le N \tag{12}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_t(j), \ 1 \le t \le T-1, 1 \le j \le N \ . \tag{13}$$

We can also derive

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = \sum_{i=1}^{N} \beta_1(i) = \sum_{i=1}^{N} \beta_t(i) \alpha_t(i) . \tag{14}$$

B. *Solution to Problem 2*

The most difficult task is to adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O \mid \lambda)$ because given any finite observation sequence as training data there is no optimal way of estimating the model parameters. This is the main issue when training specific HMM models for a specific gesture. But we can choose the model parameters so that $P(O \mid \lambda)$ is locally maximized using the *Baum-Welch* procedure (part of the expectation-maximization method).

First we define the probability of being in state S_i at time *t*, and state S_i at time *t+1* , given the model and observation sequence, i.e.,

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | \boldsymbol{O}, \lambda) . \tag{15}$$

From definition of the forward-backward algorithm, we can rewrite $\xi_t(i,j)$ in the form:

$$\begin{aligned} \xi_t(i,j) &= \frac{P(q_t = i, q_{t+1} = j | \boldsymbol{O}, \lambda)}{P(\boldsymbol{O}|\lambda)} = \\ &= \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)}{P(\boldsymbol{O}|\lambda)} = \\ &= \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)} \end{aligned} \tag{16}$$

and the probability of being in state S_i at time *t* :

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{17}$$

It has been proven by Baum et al. in [11] that maximization of $Q(\lambda, \overline{\lambda})$ over $\overline{\lambda}$ leads to increased likelihood (19) what leads to the estimation of an HMM.

$$Q(\lambda, \overline{\lambda}) = \sum_{Q} P(Q|O, \lambda) \log\left[P(Q, O|\overline{\lambda})\right] \tag{18}$$

$$\max_{\overline{\lambda}}[Q(\lambda, \overline{\lambda})] \Rightarrow P(O|\overline{\lambda}) \ge P(O|\lambda) \tag{19}$$

A set of reasonable re-estimation formulas for A, B and π are:

$$\overline{a}_{ij} = \frac{\text{no. of transitions from state } S_i \text{ to } S_j}{\text{no. of transitions from state } S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{20}$$

$$\overline{b}_j = \frac{\text{no. of times in state } j \text{ and symbol } \boldsymbol{v}_k}{\text{no. of times in state } j} \tag{21}$$

where $\overline{\pi}(i) = \gamma_1(i)$ is expected frequency in state S_i at time $t = 1$.

In other words if we define the reestimated model $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$, computed using (20) and (21), we have found a new model from which the observation sequence is more likely to have been produced. If we iteratively use $\overline{\lambda}$ instead of λ and repeat the estimation calculation, we can then improve the probability of the observation sequence from the model until some limiting point is reached.

IV. SYSTEM IMPLEMENTATION

In this section, we present details of our proposed model, applied to recognize 8 prime diver hand signals, as shown in Fig. 2.:

"Look at me" – static gesture with two fingers straight up pointing to eyes,

"I have a cramp" – dynamic gesture with the fingers bending to and from the palm,

"Going up" – static gesture with a thumb pointing up,

"Running out of air" – dynamic gesture with an open palm facing down and moving left and right,

"Turn around" – dynamic gesture with one finger pointing up and making a circle in the horizontal plane

"Danger" – extended arm with open palm moving left and right,

"Slow down" – dynamic gesture with an open palm facing down and moving up and down

"Stop" – static gesture with an open palm and fingers together pointing up.

A simplified scheme of the envisioned approach is given in Fig. 3. The first step is to detect the hand from the sequence of image frames. Using the image processing system the hand is recognized and the hand feature vector is obtained for the training and recognition process. The image processing system only extracts hand features when the hand is detected, therefore reduces the computation load when the hand is not detectable.

In this paper we observed HMM models that use two different feature vectors. In the first case, the feature vector, an observation in HMM, includes 3 components:

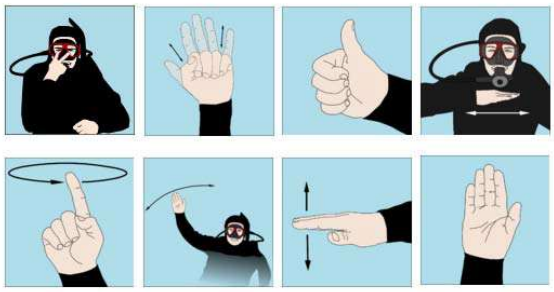

Figure 2. Diver hand signals used in the paper: "Look at me", "I have a cramp", "Going up", "Running out of air", "Turn around", "Danger", "Slow down" and "Stop" (source: Wikipedia)

$$O = (x, y, \text{number of fingers})$$

where x and y are coordinates of the palm centre in the video frame and the third component is the number of fingers stretched out and currently visible. This feature vector is given directly from the image processing system. In the second case, the feature vector includes two components:

$$O = (\text{orientation, number of fingers})$$

where the second component is the same as the third component in the first feature vector and the first component is orientation feature. The orientation is determined between two consecutive points from hand gesture path, using

$$\theta_t = \arctan\left(\frac{y_{t+1} - y_t}{x_{t+1} - x_t}\right) \quad t = 1, 2, \ldots T-1 \qquad (22)$$

where T represents the length of gesture path. The orientation θ_t is quantized in segments of 20° in order to generate codewords from 1 to 18.

As a part of research presented in this paper, we implemented three different training models, to test their efficiency and, based on the experimental results, choose the best among them. The models are chosen based on the form of the feature vector (three or two component feature vectors) and also on the type of output (and input) for the HMM (discrete or mixture of Gaussians outputs variable distribution). The models that have been taken into account are listed in Table I.

V. Experimental Results

The experiments were conducted in such a way that for each of the 8 diver gestures reported in the previous section, 100 video samples were taken and used for the training dataset. For each sample, a total of 36 images were taken with the camera during a period of around four seconds (camera has a rate of 8.6 fps), meaning that every gesture is described using 36 feature vectors extracted from the images. All the experiments were implemented in Visual C++ (using OpenCV library) and Matlab R2009a. The used camera was a Truevision webcam (640x480 pixel image resolution).

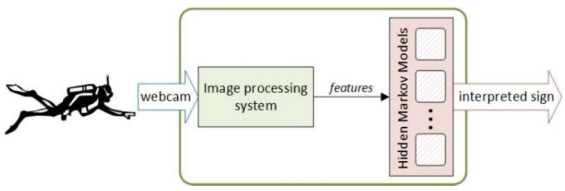

Figure 3. System scheme

TABLE I. HMM types used in analysis

Model short name	No. of components in feature vector	Output type
MG-2	2	mixture of Gaussians outputs variable distribution
D-3	3	discrete outputs variable distribution
D-2	2	discrete outputs variable distribution

The HMM parameters are derived through the training process. The training loop was programmed to end either when the maximum number of iterations has been achieved (5 in this particular case) or when the logarithm of likelihood does not increase.

This extensive set of collected data was used to determine the most reliable HMM model that can be used to determine diver gestures. To evaluate our approach overall, we carried out the following analysis from the obtained data:

- The first analysis aims to determine whether the recognition rate is higher using a model based on 3 or 2 component feature vector.
- The second analysis is to determine the best number of states and/or mixed Gaussians and to compare the recognition rate of the models in regards of their outputs variable distribution, whether it is mixture of Gaussians or discrete.

A. *First analysis – choosing the feature vector*

In order to determine whether the proposed feature vector containing two (orientation of motion and number of fingers) or three (hand position and number of fingers) components gives better performance, models D-2 and D-3 with discrete outputs variable distribution were observed. For the sake of generalization, multiple experiments were analyzed for the cases with different number of states Q. The results are shown in Fig. 4.

As the experiment shows, regardless of the number of states, better recognition is achieved using the D-2 model. The recognition rate using the D-2 model is around 50%,

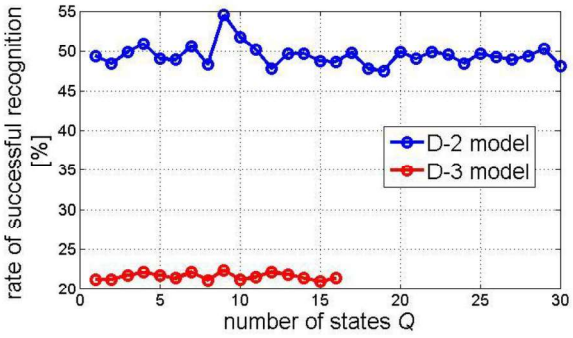

Figure 2. Rate of successful recognition for D-2 and D-3 models.

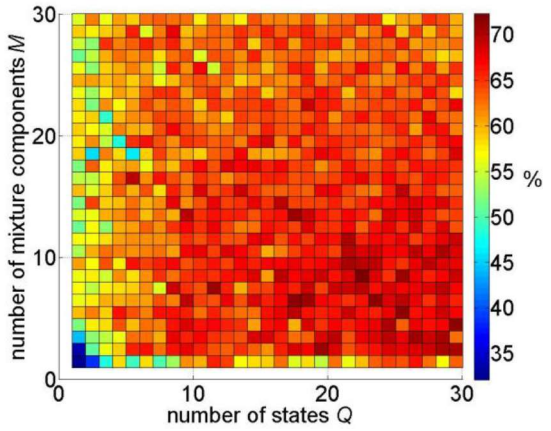

Figure 3. Rate of successful recognition for the MG-2 model.

peaking at 54% with a HMM with 9 states, where for the same number of states the recognition rate for the D-3 model is barely above 22%, with an average 21.5%.

The conclusion raised from this experiment is that models using two-component feature vector give significantly better results.

B. *Second analysis – choosing the number of states and/or number of mixture components, and outputs variable distribution*

This experiment was performed in order to determine which model has better performance relative to the output variable distribution, the one that uses discrete or mixture of Gaussians. Models D-2 and MG-2 were tested. These two models significantly depend on the model parameters.

The D-2 model depends on the number of states Q. As it was shown in the previous experiment (see Fig. 4), the best recognition rate of 54% was obtained with Q=9.

The MG-2 model, on the other hand, depends on two parameters: M – the number of mixture components and Q – the number of states. Both parameters were varied in the region from 1 to 30. The results of recognition rate are shown in Fig. 5. The MG-2 model has produced a recognition rate of 72%, improving the recognition rate of the D-2 model by almost 20%. It is clear from Fig. 5 that better results are acquired for a higher number of states and the number of mixtures being around 10.

C. *Full diver symbol verification*

The previous analysis has shown that the best results are obtained using a mixed Gaussian HMM model with M=8 and Q=22. For the sake of full verification of the system, we performed verification of the model for all 8 gestures. The comparison between the D-2 and MG-2 model is summarized in Table II and Table III where rows represent the commanded gesture, and the columns are the interpreted gesture. For the sake of clarity, we entered only the values for the dominant interpretation.

In the case of D-2 model, the gesture "Slow down" appears most often as an interpretation of the commanded gesture. The reason for this is that during this gesture the whole palm is not visible, therefore the image processing system cannot accurately calculate the position of the hand nor the number of fingers. Gestures "Cramp" and "Stop" are very similar when observing the component orientation. Their main distinguisher should be the component number of fingers, which is, as explained in the image processing system, a component that is not

TABLE III. RATE OF SUCCESSFUL RECOGNITION PER GESTURE FOR THE D-2 MODEL.

		INTERPRETED GESTURE							
		"Look"	**"Cramp"**	**"Going up"**	**"Out of air"**	**"Turn around"**	**"Danger"**	**"Slow down"**	**"Stop"**
COMMANDED GESTURE	**"Look"**							95%	
	"Cramp"		67%						
	"Going up"			33%				40%	
	"Out of air"				50%		37%		
	"Turn around"					70%	28%		
	"Danger"						90%		
	"Slow down"							80%	
	"Stop"		28%					28%	40%

TABLE II. RATE OF SUCCESSFUL RECOGNITION PER GESTURE FOR THE MG-2 MODEL.

		INTERPRETED GESTURE							
		"Look"	**"Cramp"**	**"Going up"**	**"Out of air"**	**"Turn around"**	**"Danger"**	**"Slow down"**	**"Stop"**
COMMANDED GESTURE	**"Look"**	~ 45%							
	"Cramp"		>95%						
	"Going up"			~ 60%					
	"Out of air"				>95%				
	"Turn around"					>95%			
	"Danger"						>95%		
	"Slow down"							~ 70%	
	"Stop"		~ 70%						13%

being accurately calculated and therefore in need of improvement. The gesture “Out of Air”, like the gesture “Danger”, are the only two gestures with left-right movement, and so the model has trouble distinguishing between them. The similar scenario appears for the gesture "Turn Around", since the movement of the hand in a circle trajectory is transferred to a left-right movement on the 2D camera frame. Again, the main distinguisher should be the component number of fingers.

In the case of the MG-2 model, the situation has significantly improved for all gestures except for the "Stop" gesture where the accuracy of interpretation decreased to 13% with the most cases of interpretation as the "Cramp" gesture.

To sum up, while the DP-2 gives somewhat reliable interpretation of only 4 out of 8 trained gestures, the MG-2 model gives high chances of correct recognition for 6 gestures and misinterprets only one gesture.

D. *Comments on the experiments*

From the presented results, it is clear that the MG-2 model gives the best results, compared to other tested models. However, a few comments should be stated.

During our research, we also covered the ergodic and LR (left-right) HMM. For the ergodic model, a connection can be made between any two different states, while in the LR model only a forward connection can be made, i.e. once a state has been passed it could not be achieved again. Our experiments showed no significant difference between these two models so all above mentioned experiment results are ones achieved using the ergodic model.

Another important parameter in the training of the HMM is initial value of the parameter matrices that have been set to random values in the experiments presented here.

It should also be mentioned that in some cases (e.g. when training the D-3 model with 17 or more states) matrices describing the HMM would become singular and model parameters could not be calculated.

The covariance matrix is another parameter that should be taken into consideration when analyzing the quality of the HMM. There are three different type of covariance matrix: diagonal, spherical and full. Experiments have shown that using the diagonal covariance matrix with more mixture components produced better results in comparison to the full covariance matrix with fewer mixture components. As a result, the diagonal covariance matrix was used and its value was kept constant during the experiments.

VI. CONCLUSIONS

This paper describes a HMM based system for recognizes diver signals from colour video image sequences. Our database contains 100 video sequences for each isolated gesture, as well as 100 test video sequences for each of the 8 prime diver signals.

We have shown that the HMM with the mixture of Gaussians outputs variable distribution and a two component feature vector gives the best performance among the tested models. The results show an average of 72% accuracy.

Although the system showed some good features there is room for improvement. The system is suitable for real-time application, however some modifications and additional research effort has to be invested in order to apply it to the underwater.

In the conducted experiments, the camera was fixed in from of the test subject. Since this methodology has to be applied on a diver, by observing her/him from a moving underwater vehicle, the approach to recognizing gestures had to be changed. Some research on compensation of the observing vehicle movements has been reported in [12]. Our future research will focus on measuring hand motion relative to the divers head, i.e. in the divers coordinate system. It is expected that this would solve the problem of a moving observation underwater vehicle.

Further on, the presented approach will be extended to the detection of dynamic gestures that are performed using both hands, thus extending the interpretation of the diver vocabulary.

In addition to that, we will investigate the possibility of using stereo-cameras and high resolution sonars to obtain the highest possible hand reconstruction quality, thus improving the image processing algorithms.

REFERENCES

[1] Chen, Wei-chao, "Real-Time Palm Tracking and Hand Gesture Estimation Based on Fore-Arm Contour." Master thesis, University of Science and Technology, Taiwan, 2011

[2] Rabiner, L. et al., *Fundamentals of speech recognition,* Prentice Hall, New Jersey, 1993

[3] Ćurković, P., Prepoznavanje govora korištenjem skrivenih Markovljevih modela. University of Zagreb, Zagreb, 1999.

[4] Elmezain, M. et al., "A hidden markov model-based continuous gesture recognition system for hand motion trajectory." Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. IEEE, 2008.

[5] Nguyen-Duc-Thanh, N. et al. "Two-stage Hidden Markov Model in Gesture Recognition for Human Robot Interaction." Int J Adv Robotic Sy 9.39 (2012).

[6] Krogh, A. et al. "Hidden Markov models in computational biology: Applications to protein modeling." Journal of molecular biology 235.5 (1994): 1501-1531.

[7] Chen, Weiying, et al. "Real-time 3d hand shape estimation based on image feature analysis and inverse kinematics." Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on. IEEE, 2007.

[8] Chang, Y.H. & C.M., Automatic Hand-Pose Trajectory Tracking System using Video Sequences., Master Thesis, Department of Information and Computer Engineering, Chung-Yuan Christian University, Chung-Li, Taiwan, 2010

[9] Yin, Xiaoming, and Ming Xie. "Finger identification and hand posture recognition for human–robot interaction." Image and Vision Computing 25.8 (2007): 1291-1300.

[10] Nam, Yanghee, and K. Wohn. "Recognition of space-time hand-gestures using hidden Markov model." ACM symposium on Virtual reality software and technology. 1996.

[11] Shotton, Jamie, et al. "Real-time human pose recognition in parts from single depth images." Communications of the ACM 56.1 (2013): 116-124.

[12] Buelow, Heiko, and Andreas Birk. "Gesture-recognition as basis for a human robot interface (hri) on a auv." OCEANS 2011. IEEE, 2011.