Early Speech Development of a Humanoid Robot using Babbling and Lip Tracking

Jonas Hörnstein[1], Cláudia Soares[1], José Santos-Victor[1] and Alexandre Bernardino[1]
[1]Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa, Portugal
jhornstein@isr.ist.utl.pt

Abstract

In this work we show how a humanoid robot can learn to produce and recognise both vowels and consonants using a unified method for speech production and recognition. The method is inspired by the motor theory and the discovery of mirror neurons. Both auditory and visual information is used and mapped to the robot's articulatory space where the recognition and speech production is performed. A combination of babbling and imitation is used to learn the maps. We find that the visual information can be useful not only to increase the recognition rate of already learnt phonemes, but also to drive the learning of new phonemes.

INTRODUCTION

Language acquisition is a complex and highly social process. To interact with humans using speech a robot need to be able both to produce and to recognize a number of phonemes. Speech production, speech recognition, and learning of phonemes are usually handled by different processes, but here we handle these tasks with a unified approach. This approach is based on our earlier work (Hörnstein and Santos-Victor, 2007), where we map the speech signal to motor representations in the robot's vocal tract and perform both speech planning and speech recognition in motor space. A similar approach is taken in (Kanda and Ogata, 2007). The idea to use motor space rather than directly using the speech signal comes from the Motor Theory (Liberman and Mattingly, 1985). They found that being able to produce a certain sound also increased the possibility to recognize the same sound. In an other work it has been found that there is an increased activity in the tongue muscles when listening to words that requires large tongue movements (Fadiga et al., 2002). Both these works lead us to believe that the motor area is involved not only in speech production, but also in speech recognition.

In this work we further extend and develop our unified approach by including visual input in the form of a lip tracker and a self clustering algorithm that automatically groups learned motor positions into phonemes. We also show how a humanoid robot can use the described approach to learn both vowels and simple consonants during its early speech development. The robot used in this work is the iCub, Figure 1. The iCub is equipped with sensors in the form of microphones and cameras, and can produce sound through a simulated vocal tract. It has no preprogrammed knowledge about language. Instead it has to learn how to speak by exploring its vocal tract and learn its initial sensory-motor maps using babbling. It also has to learn which sounds are useful for communication with humans, group these sounds into phonemes, and to recognize the same phonemes when pronounced by different speakers. The set of sounds considered as useful depend on the cultural environment in which the robot is placed and therefore has to be learned through the interaction with humans. Here we use different types of imitation games to allow the robot to learn new phonemes and gain speaker invariance.

The rest of the paper is organized as follows. In section 2 we give an overview of the architecture used and especially focus on the new parts like the lip tracker and the clustering algorithm. In section 3 we describe the babbling and imitation behavior that the robot uses to develop its speech. In section 4 we show some experimental results and conclusions are given in section 5.

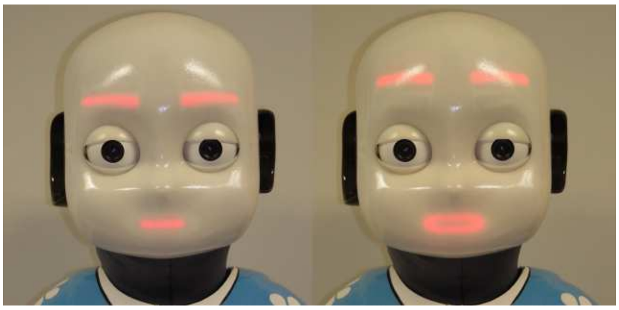

Figure 1: iCub robot learning to speak

System architecture

The architecture used in this work is an extension of the architecture described in (Hörnstein and Santos-Victor, 2007). As in the previous work the architecture consists of a speech production unit, a sensor unit, a sensor-motor map and a speech recognition unit, Figure 2. The main difference compared to the older version is the addition of a visual sensor in the sensor unit and a vision-motor map in the sensor-motor map unit. We have also done some modifications in the position generator that drives the babbling and added a self clustering algorithm in the motor

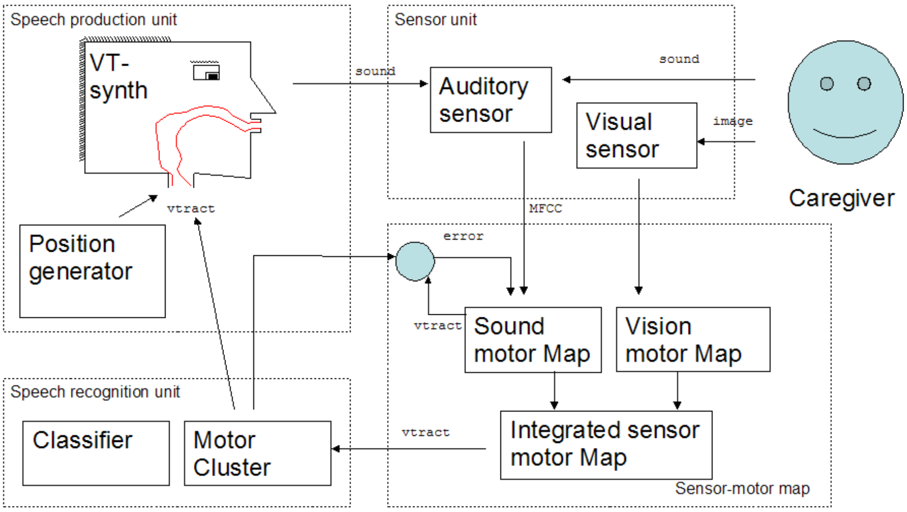

Figure 2: Speech architecture

cluster. In this section we give a short overview of each unit and explain the differences between the current and the previous version in more detail.

Speech production unit

The speech production unit is responsible for moving the lips and producing sound. As in the previous version we do not use a physical model of the human vocal tract, but simulates the vocal tract in a computer model. The model used is vtcalcs developed by Maeda (Maeda, 1990). This model has six parameters that can be used to control the movements of the vocal tract. One parameter is used for the controlling the position of the yaw, one for the protrusion of the lips, one for lip opening, and three parameters for controlling the position of the tongue. A synthesizer converts the vocal tract positions into sound. While the synthesizer works well for vowel-like sounds, it is unable to produce fricatives sounds and can hence only produce a limited set of consonants.

In the new architecture the vocal tract position is also used to control the shape of the robot's lips. Our robot has a very simple lip model consisting of a number of leds that can either show a closed or an open mouth. A simple threshold is used to decide whether the mouth should be shown as open or closed. Examples of the mouth positions are shown in Figure 1.

The most important difference in the speech production unit is the new position generator. While the previous version only created random positions for the vocal tract the new unit offers more advanced babbling behavior. One of the problems with the random position generator was that it created lots of non-humanlike sounds that aren't useful for human-robot interaction and slows down the learning process. In (Soares and Bernadino, 2007), it has been shown that a convex combination of three corner vowels [i], [a] and [u] is able to produce the complete vowel space. The corner vowels represent extreme placements of the tongue and can therefore be considered as known stable points when starting the exploration of the articulatory space.

Thus, in this work we include these corner vowels as starting points in the motor cluster, even though we have previously shown that it is possible to learn those using random babbling. The position generator creates a new sound by picking two positions in the motor cluster and creating a trajectory between those. As shown in (Soares and Bernadino, 2007) we always create tangible speech as long as we stay within the convex envelope of the corner vowels. However, as this would also restrict us to the vowel space we add some noise to the positions before creating the trajectory. This way we allow the robot to also explore the articulatory space beyond the vowel space.

Sensor units

We use two sensors, an auditory sensor unit and a visual sensor unit that extract features from the acoustic and visual spaces respectively. The auditory sensor remains unchanged. A microphone is used to record the sound. The sound is windowed into 30 ms frames and Mel frequency cepstral coefficients (MFCC) (Davis and Mermelstein, 1980) are calculated for each frame. The visual sensor unit is, on the other hand, a complete new unit that has not been presented in the previous work. In the following we explain this sensor unit in more detail.

The purpose of the visual sensor is to provide visual clues on the position of the vocal tract. While there are methods to find the exact contour of the lips, like the usage of snakes or active contour methods (Kass et al., 1987), these methods are typically too complex to use in speech recognition. With no a priori assumption of the shape of the lips the estimation becomes slow and more error prone. Further more, the complexity of the final description makes further data processing costly. For practical applications where we need to track the movements of the lips in real-time, and are interested in some simple feature like the area of the mouth opening rather than the exact contour, we need a compact representation of the lips. In this work we have chosen to represent the lips by an ellipse, which is fitted to the pixels that belong to the lips. The pixels that belong to the lips are found by using color segmentation. The color segmentation can be done in several different ways. It is usual to extract the color from the first frame using the initial position of the lips. In (Tian et al., 1999) the whole color distribution of the lip region is calculated and modelled as a Gaussian mixture and the EM method is used to estimate both the mixture weights and the underlying Gaussian parameters. Here we use a much simpler method and simply model a lip by its redness, where we define the redness as:

$$Redness = R^2/(R^2+G^2+B^2)$$

where R, G, and B are the red, green, and blue value of an RGB-image. If the redness of a pixel is above some threshold we define the pixel as a lip. The threshold can be calculated from the initial frame, but we have chosen a fixed threshold of 0.9. As shown in Figure 3, the threshold seems to work well even for different persons. Of course there are other pixels apart from the lip pixels that are classified as red so we need to know the approximate position of the lips and only use those pixels to fit the ellipse. Here we use a face detection algotithm, based on (Viola and Jones, 2001) and (Lienhart and Maydt, 2002). The face detection algorithm not only gives us an initial estimate for the position of the lips, but also gives us the size of the face which is later used to normalize the area of the mouth opening. However, the face detection algorithm is rather slow so the position and size of the head is therefore only calculated once in the beginning of every experiment and the subject with which the robot interacts is assumed to maintain approximately the same distance to the robot during each experiment.

To fit the ellipse to the lip pixels we use a least square method described in (Fitzgibbon et al., 1999). The result is shown in figure 4. We then use the ellipse to calculate the area of the mouth opening. The ratio between the area of the mouth opening, given by the lip tracker, and the area of the face given by the face tracker, is used as a visual feature and is sent to the vision-motor map.

As said before, the face detection is too slow to be useful for tracking the movements of the lips between two frames in the video stream. We therefore use the method suggested by Lien et. al (Lien et al., 1999). They use Lucas-Kanade tracking algorithm (Lucas and Kanade, 1981) to track the movements of the lips between adjacent frames. One problem with the tracking algorithms is that it is sensitive to the initial feature point selection as most points on the lips have ambiguities around the lip edges. Here we solve this by looking for Harris features (Harris and Stephens, 1988) around the lips and use these as initial points that will be tracked. The result gives us a sufficiently good estimate to maintain an initial estimate of the lip position over the video sequences used in our experiments.

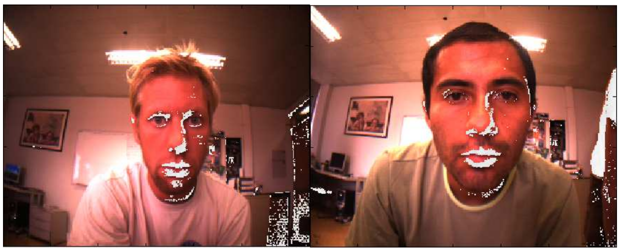

Figure 3: Color segmentation

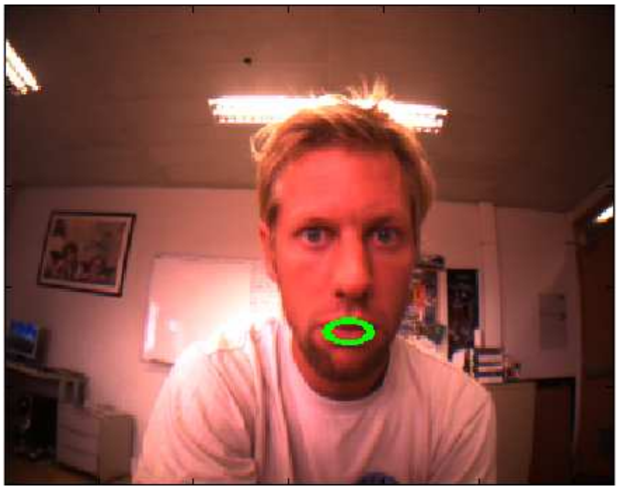

Figure 4: Lip tracking

Sensor-motor maps

The sensor-motor maps are responsible for retrieving the vocal tract position from the given auditory and visual features. We use two separate neural networks to map sound-motor map and the vision-motor map respectively. The sound-motor map is the more complicated of the two, mapping the 12 cepstral coefficients back to the 6 parameters of the vocal tract model. The problem is extra difficult since several positions of the vocal tract results in the same sound, giving several possible solutions for a given set of features. While the position generator described above reduces the risk of producing the same sound from two different positions, we still get some ambiguities that have to be solved through the interaction with a caregiver. For the sound-motor map we use an artificial neural network with 20 hidden neurons.

The vision-motor map is a very simple unit, doing a linear mapping from the mouth opening to the lip height parameter of the synthesizer.

Since the output from both the sound-motor map and the vision-motor map consist of vocal tract positions, the integration of those sensor outputs becomes very simple. Here we simply use a weighted average of the lip height calculated from the two maps. The weight is currently set by hand, but should preferable be set automatically according to the quality and intensity of the visual and auditory stimula.

Speech recognition unit

The speech recognition unit contains a motor cluster and a classifier. In our previous version of the architecture the classifier was a simple dictionary that stored motor positions that were considered useful for the communication with the caregiver. In this new version we have implemented an hierarchical clustering algorithm based on (Hastie, 2001), which starts with creating one cluster for each stored position and then iteratively joins the two clusters with the minimum euclidean distance until we only have one single cluster containing all stored positions.

For each level of the clustering process, we have different relationships between data groupings. So, the question is: what is the "natural" grouping for this dataset? To estimate the number of clusters in a data set we used Gap statistic (Tibshirani et al., 2001). This function compares the within-cluster dispersion of our data with that obtained by clustering a reference uniform distribution. This is to compare the gain of raising the cluster number in a structured data with that arising from adding another cluster to a non-informative and not structured set of points.

We choose the first maximum in the Gap statistic as the optimal number of clusters. Each position within the same cluster is considered to be part of the same phoneme or pseudo-phoneme.

The recognition task is handled by the classifier that compares positions given from the sound motor map with the mean positions of each pseudo-phoneme in the motor cluster and can be configured to use either Euclidean distance or the Mahalanobis distance to find the nearest neighbor.

Babbling and imitation

In this this section we describe the mechanisms used by the robot to learn to vocalize vowels and simple consonants. The methods are inspired by the way children develop their speech through a combination of self-exploration in the form of babbling and through interaction with a caregiver. We separate between two types of interactions, the robot imitating a caregiver, and the caregiver imitating the robot. Both these behaviors can be found in the interaction between a child and its parents.

Here we first describe what happens during babbling, then we explain which maps that are updated as the caregiver imitates the robot, and finally what happens when the robot imitates the caregiver. However, we would like to point out that these activities should be seen as parallel rather than sequencial and that all behaviors are active during the whole development.

Babbling

The babbling behavior is realised by the position generator. As explained in the previous section the position generator randomly takes two positions from the cluster, add some noise to the positions, and then create linear trajectory between the two points. In the beginning the noise level is set relatively high in order to explore as much as possible of the articulatory space. With time, and as more positions are stored in the cluster, the noise level in the babbling is gradually reduced and the babbling is focused on the trajectories between the learnt positions.

Each position in the generated trajectory consists of the 6 parameters in Maeda's model. These are then passed on to the speech production unit that calculates the resulting sound. The sound is then fed into the auditory sensor unit that calculates the MFCC and passes these to the sound-motor-map. The sound-motor-map finally tries to map the MFCC back to the original articulator position vector and compares the result with the output from the position generator. The error between the mapped and the correct positions is the used to update the map using a back-propagation algorithm.

There is no update of the vision-motor map during babbling since the robot does not get any visual feedback of its lip position.

Caregiver imitating robot

Having the caregiver imitating the robot is arguable the most important factor in learning both the sound-motor map and the vision-motor map. While the robot can easily learn the map between its own sound and motor positions through babbling, there is a large difference between the speech produced by the robot and normal human speech. The same can be said about a child whose vocal tract is significantly different to that of an adult. Add to that the fact that the sound produced by ourselves is transmitted not only through the air, but also through bone structures in the head which make our own voice sound significantly different compared to the sound produced by others even if we would have exactly the same vocal tract. To compensate for those things we have to interact with other people in our environment and tune the maps according to their voices.

This interaction starts with the robot creating a trajectory in the same way as for the babbling and sending the sound to the speaker. The caregiver then tries to repeat the same utterance with its own voice. It is important that the caregiver repeats the perceived utterance rather than the exact sound produced by the robot. Here we do not handle the problem of deciding whether the person with whom the robot interacts is actually imitating what the robot said or not, but simply assumes that the received response is the same utterance. We also make sure that the utterance has the same length and that it is correctly aligned in time with the utterance of the robot. This is done manually at the moment by selecting some keypoints along the trajectory and finding the same key points in the response of the caregiver. We also extract images from the video stream that match each of the key points.

The maps are then trained using the vocal tract positions of the robot together with the auditory and visual response from the caregiver. The sound from the caregiver is fed into the auditory sensor and the corresponding MFCC are calculated and sent to the sound-motor map. The mapped position is compared to the vocal tract position used by the robot and the map is updated to compensate for the error. In the same way the image of the caregiver is fed into the visual sensor which calculates the mouth opening and sends the result to the vision-motor map. Again the mapped position, this time of the lip height only, is compared to the original position and the map is updated according to the error.

This is repeated for various utterances and preferably with several different caregivers in order to increase the robots posibility to correctly map utterances from other persons to its own vocal tract in order to reproduce the same sound or to recognize what the other person actually said.

Robot imitating caregiver

One problem having the caregiver imitate the robot is that the robot is not very likely to say something meaningful by just doing babbling. In order to get the robot to actually learn some useful phonemes it is better to have the human to make the utterance and let the robot try to imitate. How well the robot will be able to repeat the same utterance depends on how well it has learnt the sensor-motor maps.

If the robot has mostly used babbling and had little or no previous interaction with its caregiver it is not likely to correctly map the sound of the caregiver when the caregiver uses his or her normal voice. In order to direct the robot to the correct utterance the caregiver may therefore need to adapt his or her own voice. This behavior can also be found in the interaction between a child and its parents and has been studied in (de Boer, 2005). When the robot answers with the intended utterance we give the robot positive feedback which causes the robot to store the current articulator positions in its cluster. This reinforcement was given through the keyboard in the current implementation, but more sophisticated methods could be used.

This step is only used insert new positions in the cluster and no training is going on in this step.

Experimental results

We performed three experiments using the architecture with babbling and imitation as described above. In the first experiment we test if the clustering algorithm is able to correctly group the positions it learns for 9 portuguese vowels. In the second experiment we use the learnt vowels and see how well the robot can recognize the same vowels when pronounced by different human speakers. Especially we look at the effect the visual features have on vowel recognition. In the third experiment we teach the robot some simple consonants and again look at the effect of using vision for recognition by studying the well know McGurk effect (McGurk and MacDonald, 1976) which can be expected when combining visual and auditory features.

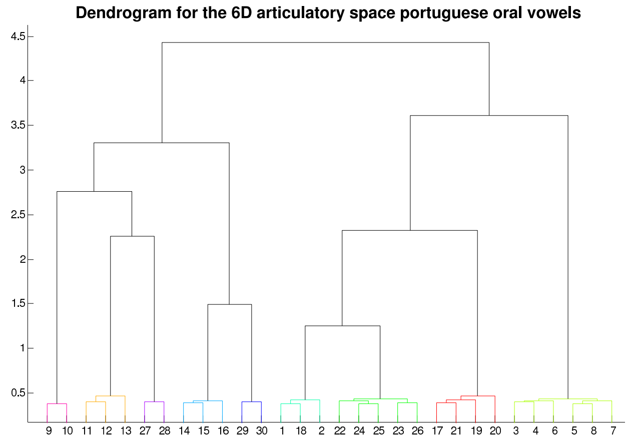

Figure 5: Dendrogram depicting the hierarchical clustering performed by the robot.

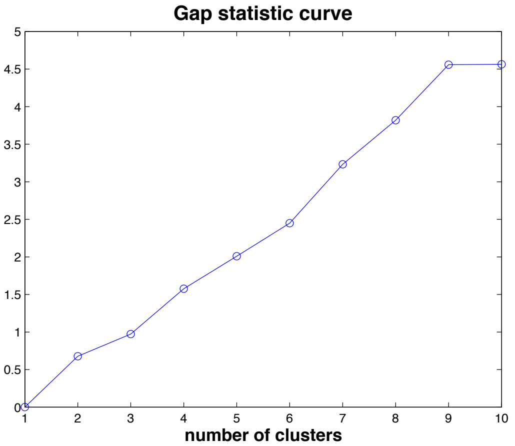

Figure 6: Gap statistic versus number of clusters. The growth of the curve stops at nine clusters.

Learning vowels

To create a sufficient number of valid training vowels for the robot, we created a dataset with 900 vowels, and then submitted them to the evaluation of 16 native speakers, so that they rejected or approved each vowel as a valid portuguese vowel and — for those that were approved — agreed or not in their phonological classification. From these 900 vowels, 281 were considered appropriated.

The original dataset was generated from nine prototype vowels in the 6D articulatory space, added with 10% of white noise.

Applying agglomerative hierarchical clustering to the present vowel dataset originated some good results, as we can see in figure 5. The nine vowel groupings depicted in different colors are clearly visible.

The dendrogram shown can be seen as a summary of the data structure that was detected by our simple dissimilarity measure: euclidean distance between 6D vectors and average dissimilarity between groups.

In ten performed trials, the Gap statistic consistently pointed to nine as the most natural number of clusters. One example of this result is presented in figure 6.

Vowel recognition

To be able to compare the results obtained in this work with the results obtained in (Hörnstein and Santos-Victor, 2007), we actually do not use the vowels positions learned by the cluster above, but instead use the positions learnt in the referred work. There the robot first learned its own sound-motor map by doing a completely random babbling. A caregiver then taught the robot nine Portuguese vowels by having the robot imitate the vowels and storing those that were successfully pronounced. As seen in Figure 7 the articulator positions used by the robot are similar to those used by a human speaker.

Next, the 14 speakers (seven males and seven females) were recorded while reading words that included the same nine Portuguese vowels. Each speaker read the words several times, and the vowels were hand labeled with a number 1 to 9. The amplitude of the sound was normalized and each vowel was then divided into 30 ms windows with 50% overlap. Each window was then treated as individual data which resulted in a training set of 2428 samples, and a test set of 1694 samples.

In addition to the original data we also extracted images from the video sequences that corresponded to each person pronouncing the vowels. Only one image for each person and vowel was extracted creating a training and test set of 63 images each. The images were then processed by the visual sensor in order to calculate the mouth opening in each image.

After the learning of the maps using random babbling the recognition rate for the human vowels in the test set were as low as 17.5%. We then used the data from the seven persons in the training set to imitate the robot's vowels to allow the robot to further train both the auditory-motor and the visual-motor maps. After the interaction with the persons in the training set, the recognition rate for the persons in the test set became 63.3%. If the robot was just presented with auditory input and was not allowed to see the person the recognition rate became 57.7%.

Learning consonants

We have also done some initial experiments with teaching the robot consonants using the methods described above. Each consonant is here modelled with a single target point in the articulatory space. It should be noted that the point by itself cannot reproduce the consonant. To reproduce the consonant we create a trajectory between two vowels that passes through the target point.

For this experiment the robot started with the three corner vowels [i], [a] and [u], and did an initial babbling by creating 1000 trajectories with 10 points along each trajectory.

In the second step we let the caregiver imitate the robot. We only created the straight trajectories [i] to [a], [a] to [u] and [u] to [a] as the alignment between the robot and the human utterances had to be made by hand, but these were sufficient to give the robot initial sensor-motor maps for the auditory and visual features of the caregiver.

The last step was to let the robot imitate the caregiver. We wanted to teach the robot three new phonemes /b/, /d/, and /g/ by having it imitating the utterances ba-ba, da-da, and ga-ga. This was done by feeding the last hearable sound before reaching the goal position of the consonant to the auditory sensor along with an image of the lip position at the goal position. The sound and the image were extracted automatically when the sound got below a threshold.

Teaching the robot a /b/ was pretty straight forward as it the robot could easily extract the main position from the visual feature. The latter two demanded a little more patience from the caregiver. The task got extra difficult since the synthesizer used does not create any clear consonants so we actually needed to inspect the resulting vocal tract position of the robot in order to decide if we were happy or not with the result. As we got happy with an utterance we stored the position in the motor cluster. The learnt positions can be seen in Figure 8.

Once the robot had learnt the positions we again switched roles and let the caregiver imitate the robot. After doing that the robot could easily recognise and reproduce the correct consonant. However we only did this experiment with a single caregiver so we do not expect the robot to generalize and correctly classify the same consonants when uttered by another speaker.

Finally we did a simple experiment were we tried to reproduce the McGurk effect by feeding the auditory sensor with ba-ba while feeding the visual sensor with ga-ga. Depending on the weight we put on the visual sensor relative to the auditory sensor the robot classify the utterance as either a ba-ba or da-da.

Conclusions

We have demonstrated how a humanoid robot can develop speech by using a combination of babbling and imitation.

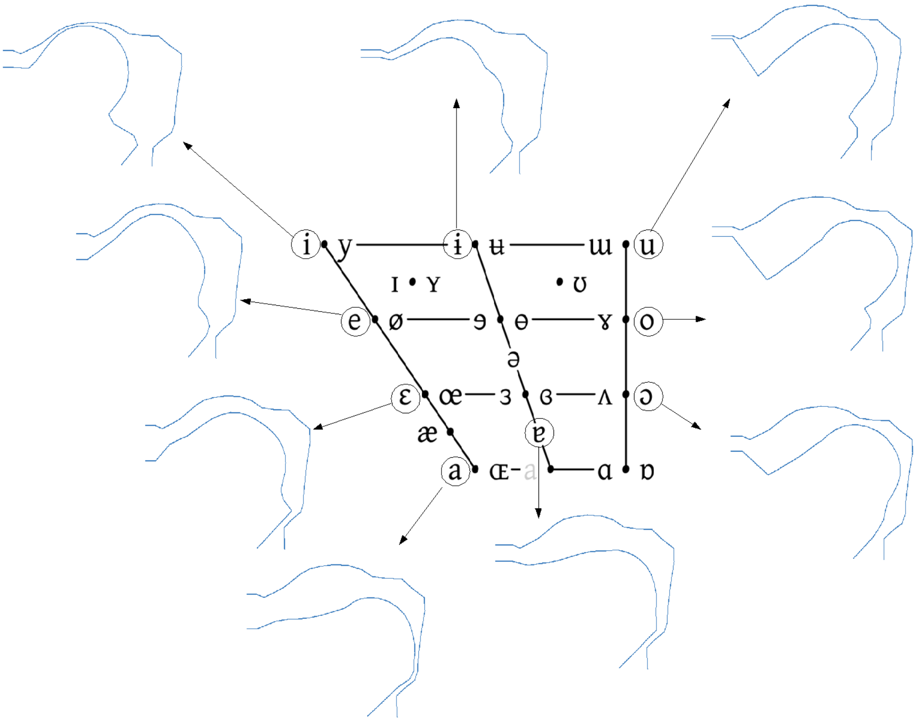

Figure 7: Articulator positions used by the robot for the Portuguese vowels. In the center we show the positions of the vowels in the International Phonetic Alphabet (IPA). The vertical axis in the IPA corresponds to the vertical position of the tongue and the horisontal axis to the front-back position when the vowel is pronounced by a human speaker. For the simulated articulator positions used by the robot the upper line corresponds to the soft palate and the lower line to the tongue. There is a good correlation between how the robot and a human articulate the vowels.

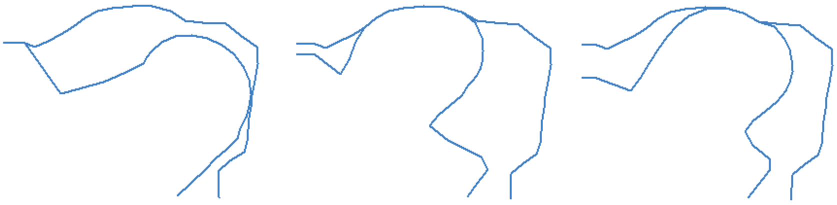

Figure 8: Learnt positions for the consonants /b/, /d/ and /g/.

While babbling make it possible for the robot to learn the map between its own sound and motor positions, interaction with a caregiver is more important for learning to map and understand human speech.

By letting the robot and the caregiver take turn in imitating each other it is possible both to teach the robot reproduce utterances made by the caregiver and learning which utterances that are useful for communication.

We have also shown that visual features can be helpful both to increase the recognizion rate of already learned phonemes and for learning new phonemes.

Acknowledgement

This work was partially supported by EU Project CONTACT and by the Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the POS Conhecimento Program that includes FEDER funds.

References

Davis, S. B. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, speech, and signal processing*, ASSP-28, no. 4.

de Boer, B. (2005). Infant directed speech and evolution of language. In *Evolutionary Prerequisites for Language, Oxford: Oxford University Press*, pages 100–121.

Fadiga, L., Craighero, L., Buccino, G., and Rizzolatti, G. (2002). Speech listening specifically modulates the excitability of tongue muscles: a tms study. *European Journal of Neuroscience*, Vol 15:399–402.

Fitzgibbon, A., Pilu, M., and Risher, R. B. (1999). Direct least square fitting of ellipses. *Tern Analysis and Machine Intelligence*, 21.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

Hastie, T. (2001). *The elements of statistical learning data mining inference and prediction*. Springer.

Hörnstein, J. and Santos-Victor, J. (2007). A unified approach to speech production and recognition based on articulatory motor representations. In *IROS07*, pages 3442–3447.

Kanda, H. and Ogata, T. (2007). Vocal imitation using physical vocal tract model. In *IROS07*, pages 1846–1851.

Kass, M., Witkin, A., and Terzopoulus, D. (1987). Snakes: Active contour models. *International Journal of Computer Vision*.

Liberman, A. and Mattingly, I. (1985). The motor theory of speech perception revisited. *Cognition*, 21:1–36.

Lien, J. J.-J., Kanade, T., Cohn, J., and Li, C.-C. (1999). Detection, tracking, and classification of action units in facial expression. *Journal of Robotics and Autonomous Systems*.

Lienhart, R. and Maydt, J. (2002). An extended set of haar-like features for rapid object detection. *IEEE ICIP*, pages 900–903.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop*, pages 121–130.

Maeda, S. (1990). Compensatory articulation during speech: evidence from the analysis and synthesis of vocat-tract shapes using an articulatory model. In *Speech production and speech modelling, W. J. Hardcastle and A. Marchal, esd.*, pages 131–149. Boston: Kluwer Academic Publishers.

McGurk, H. and MacDonald, J. (1976). Hearing lips and seeing voices. *Nature*, (264):746–748.

Soares, C. and Bernadino, A. (2007). Mapping the vocal tract with a 2d vocalic articulatory space: applications to developmental robotics. In *Symposium on Language and Robots*.

Tian, Y., Kanade, K., and Cohn, J. (1999). Multi-state based facial feature tracking and detection. In *technical report, Robotics Institute, Carnegie Mellon University*.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2).

Viola, P. and Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. *IEEE CVPR*.