22

Learning Features for Identifying Dolphins

Luiz Gonçalves, Adelardo Medeiros, and Kaiser Magalde
Universidade Federal do Rio Grande do Norte
Brazil

1. Introduction

Dolphins are mammalians that integrate the cetacean family. They can be generally found in the sea but there are some species that live in rivers. There are some 37 species of dolphins mapped in Brazil. Their life time is about 35 years and the period of gestation is 10 months. They can reach up to 3 meters in length. The most known, of long nose, are about 2 meters long. Dolphins are most known by their abilities of diving and jumping at the water level, wondering people. An important characteristic used for distinguishing different individuals is its dorsal fin. It has no bounds in its composition and its size varies from specie to specie. Most important, its shape varies for individuals of the same specie. Besides this shape peculiarity, the dorsal fin keeps marks made by natural attackers, as sharks and others, or even fights, that may also be used for distinguishing them. These features form a natural signature as a finger tip.

Currently, there are several researches dealing with these animals in partially demarked environments in beaches or rivers. The objectives are such as behavioural studies between groups (Link, 2000), counting statistics, and others. One of the difficulties is the lack of an exact limit of the observed areas. Besides, there is also some difficulty due to human observing capability, which is somewhat avoided due to a constant presence of a same specie in a same region. Other known difficulty is the acquisition of dolphin images. Most species keeps some distance from boats and other aquatic vehicles or even they disappear to deep water. This makes hard the work of researchers who needs to observe a live data or data obtained from photos for posterior identification. The last is known as visual photo-identification and is often practiced by biologists. Photo-identification can give information about motion parameters and population dynamic (Mann, 2000). This method has being applied through the last three decades mainly in the *Tursiops truncatus* (bottlenose) dolphin specie. This method is also employed to identification of *Orcinus Orca* (most known as the killer wale) and other cetacean species (Defran, 1990; Flores and Mann, 2000). As the pictures are generally taken in different conditions (day, time, and weather), photo-identification through purely human observation may be subjected to errors and to human analyzer subjectivity.

The population of Dolphins analyzed in the current work lives in a coastal region of Rio Grande do Norte state known as Dolphin Bay, to the south of Natal (state Capital), between the Tabatinga and Buzios beaches. The work proposal is to build a system able to extract efficient features from digital pictures and to use them for identification of dolphins. This is a challenge mainly because pictures are generally taken by biologists and

Source: Mobile Robots Towards New Applications, ISBN 3-86611-314-5, Edited by Aleksandar Lazinica, pp. 784, ARS/plV, Germany, December 2006

students from distances varying from at least 15 to 40 meters from the animals, being some 30 meters the average. Also, the pictures studied are characterized by the presence of sea water in almost its entire area but the dolphin, which is generally centralized in the photo.

Efficiency refers to the use of several features in a unique, integrated system able to identify the several animals of a family. We remark that there must exist some control mainly in the exactly moment in which the pictures are taken, so a somewhat professional, or trained, photographer is needed. This is in order to take pictures with the dorsal fin completely visible as possible. Photos of joint animals and with boats in it can be treated, but must also be avoided as possible, as well as photos with the dorsal fin partially visible.

We apply two methodologies for identification. The first executes a search in a priority space, attributing scores to statistical measures of trustworthiness. The second method uses an associative memory based on a multi-layer perceptron trained with an algorithm of the type backpropagation (BPNN) (Rummelhart, 1983; Werbos, 1988; Riedmiller and Braun, 1993). As a new contribution, we added a self-growing mechanism that makes easy the inclusion of new individuals eventually discovered in the environment. In this way, feature extraction and abstraction combined with the classification algorithm with self-growing capability are the new ingredients. These are the system key that makes the system robustness, working even in not ideal conditions. For example, as one of the difficulties, the worked images are generally obtained with dolphins in different positioning, in different angles and in different scales. We developed a normalization approach that partially avoids these problems.

We show experiments and results with images from a database of the Biology Department of Federal University of Rio Grande do Norte. The main objective is to define or to learn characteristics for some specific animals from this database and to later recover positive identifications for these animals from digital pictures of them, making easier the biologist work. We remark again that the system deals with already known and also with new individuals eventually found in the working region.

2. State of art

We found a reduced number of works in the literature about analysis of cetacean fins. Between them, we cite the works of Aarabi et al (2000a; 2000b; 2001) in which, respectively, string matching and invariant models for photo-identification of dolphins are evaluated. From the last work, it is concluded that methods based on curvature analysis of the dorsal fin are much more efficient. Differential, semi-differential, and algebra invariants are used. A problem found is that digitalization errors may substantially compromise the results in the case of using an algebra differential model. The use of semi-differential invariance is practically viable, since the number of derivatives as well the number of points used for correspondence be limited in two. Hillman (1998) and Kreho (1999), working together, experiment two possibilities. In the first, the fin curvature is reduced to a set of Fourier coefficients and some similarity invariants are extracted producing as a result invariance measures in relation to camera position, angle, and dolphin distance. The other technique is based on the search for holes using a rigid curve (a stiff snake), through a four degree polynomial that determines an approximation for the fin curvature. Following, the difference from this curve to the real points of the dorsal fin is computed. The last technique has produced a more promissory set of features.

This idea is reinforced by Cesar and Costa (1997) that explain how energy curvature diagrams of multi-scales can be easily obtained from curve graphics (Cesar, 1996) and used as a robust feature for morphometric characterization of neural cells. That is, the curvature energy is an interesting global feature for shape characterization, expressing the quantity of energy necessary for transforming a specific shape inside its most down energy state (a circle). The curvegram, which can be precisely obtained by using image processing techniques, more specifically through the Fourier transform and its inverse, provides a multi-scale representation of digital contour curvatures. The same work also discusses that by the normalization of the curvature energy with respect to the standard circle of unitary perimeter this characteristic gets efficient for expressing complex shapes in such a way it is invariant to rotation, translation, and scale. Besides, it is robust to noise and other artefacts that imply in image acquisition.

Just recently, Gope et al (2005) have introduced an affine curve matching method that uses the area of mismatch between a query and a database curve. This area is obtained by optimally aligning the curves based on the minimum affine distance involving their distinguishing points.

From all observed works, we could see that the major difficulties encountered by researchers is in the comparison between several individuals photographed in different situations, including time, clime, and even different photographers. These and other issues insert certain subjectivity in the analysis done from picture to picture for a correct classification of individuals.

Photo analysis is based mainly in the patterns presented by the dorsal fin of the animals. That is, identification has been carried out mostly by identifying remarkable features on the dorsal fin, and visually comparing it with other pictures, following a union of affine features. Effort and time spent in this process are directly proportional to the number of pictures the researchers have collected. Yet, visual analysis of pictures is carried out by marine researchers through use of rules and other relative measurement forms, by hand. This may classify individuals with little differences as the same or else to generate new instances for the same individual. This problem may affect studies on population estimation and others, leading to imprecise results. So, a complete system that starts with picture acquisition, to final identification, even a little helped by the user, is of capital importance to marine mammal researchers.

3. The proposed system architecture

Fig. 1 Fig. 1 shows the basic architecture of the proposed system. It is basically divided into the shown 7 modules or processing phases. Each of these phases will be briefly described in the next.

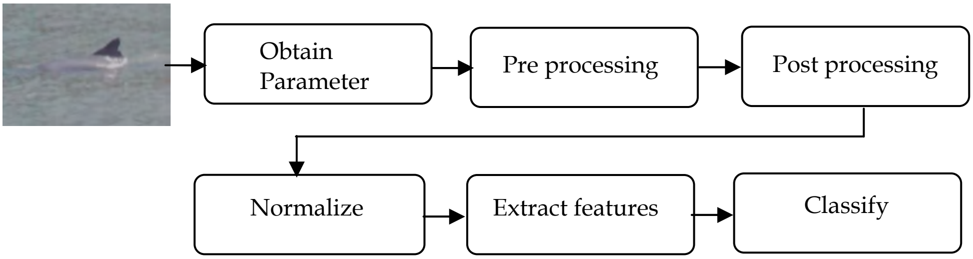
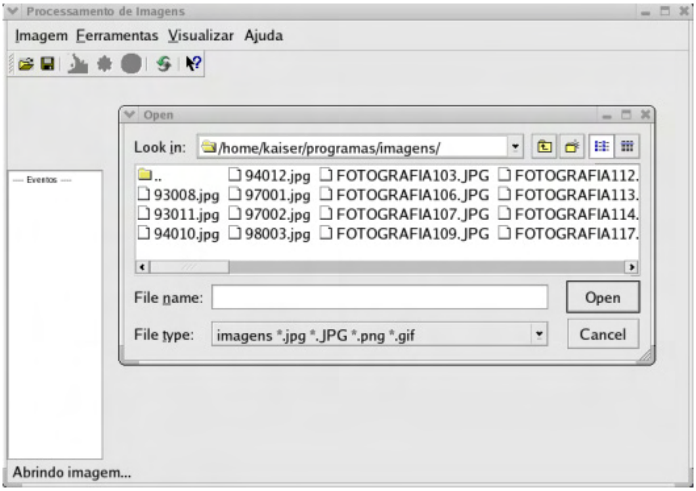


Fig. 1. Architecture of the proposed system.

3.1. Acquiring module

The acquired image is digitized and stored in the computer memory (from now we refer to this as the original image). An image acquired on-line from a digital camera or from a video stream can serve as input to the system. As another option, in the main window of the system shown in Fig. 2, a dialog box may be activated for opening an image file, in the case of previously acquired images.



Look in: /home/kaiser/programas/imagens/

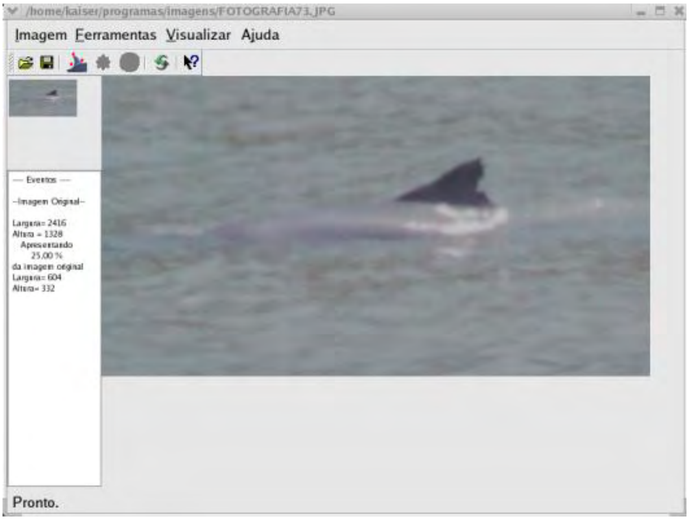
```
..              94012.jpg   FOTOGRAFIA103.JPG   FOTOGRAFIA112.
93008.jpg       97001.jpg   FOTOGRAFIA106.JPG   FOTOGRAFIA113.
93011.jpg       97002.jpg   FOTOGRAFIA107.JPG   FOTOGRAFIA114.
94010.jpg       98003.jpg   FOTOGRAFIA109.JPG   FOTOGRAFIA117.
```

File name:

File type: imagens *.jpg *.JPG *.png *.gif

Abrindo imagem...

Fig. 2. Main window of the proposed system.


/home/kaiser/programas/imagens/FOTOGRAFIA73.JPG
Imagem Ferramentas Visualizar Ajuda
— Eventos —

--Imagem Original--

Largura= 2416
Altura = 1328
Apresentando
25,00 %
da imagem original
Largura= 604
Altura= 332

Pronto.

Fig. 3. Acquired image visualized in lower resolution in the system main screen.

3.2. Visualization and delimitation

In this phase, a human operator is responsible for the adequate presentation of the interest region in the original image. This image is presumably to be stored in a high resolution, with at least a 1500 x 1500 pixels size. The image is initially presented in low resolution only for making faster the control by the operator, in the re-visualization tool. All processing are done at the original image, without loosing resolution and precision. As a result of this phase, a sub-image is manually selected with the region of interest in the original image, containing dolphins, and mapped in the original image. This sub-image may be blurred to the user, so it would be nice to apply a auto-contrast technique for correcting this (Jain, 1989).

3.3. Preprocessing

In this module, the techniques for preprocessing the image are effectively applied. Between these techniques, the Karhunem-Loève (KLT) (Jain, 1989) transform and the auto-contrast are used. The KLT transform is applied in order to make uncorrelated the processing variables, mapping each pixel to a new base. This brings the image to a new color space. The auto-contrast technique is applied in order to obtain a good distribution for the gray level or RGB components in case of colored images, adjusting thus the image for the coming phases.

3.4. Auto-segmentation

A non supervised methodology through a competitive network is applied in order to segment the image based on its texture patterns. This generates another image which has two labels, one for background and another foreground (objects) respectively. In order to label the image, we use the average of the neighbor values of a selected pixel as image attributes for entering the competitive net. It is calculated the mean for each of the components, R, G and B, thus giving three attributes for each pixel.

3.5. Regularization

In this phase, several algorithms are applied to regularize the image provided by the previous phase. This regular, standard image is important for the next phase, mainly for improving the feature extraction process. A clear case of the real necessity of this regularization is the presence of dolphins in different orientations from one picture to another. In the extreme case, dolphins may be pointing to opposite senses (one to left and other to right of the image). For this regularization, first the three ending points of the dorsal fin, essential to the feature extraction process, are manually chosen by the operator. Note the difficulty to develop an automated procedure for the fin extraction, including segmentation, detection and selection of the ending points. We have made some trials, ending up with the hand process, as this is not the main research goal. In order to get the approximate alignment of the dolphin, the system presents a synthetic representation of it (that is, a 3D graphical dolphin) whose pointing direction can be controlled by using keyboard and mouse. This tool is used by the user to indicate the approximate, actual orientation (direction and sense) of the dolphin in the original image. In practice, the user indicates, interactively, the Euler angles (roll, pitch, yaw) relative to the approximate orientation. These angles are the basis for defining the coefficients of a homogeneous transform (a 3D rotation) to be applied in the 2D image in order to approximately conform to the desired orientation in the model. Then the image is ready for feature extraction.

3.6. Morphological operators

Mathematical Morphology techniques tools are used for extracting boundaries, skeletons, and to help improving other algorithms. Mathematical morphology is useful for improving extraction of shape features as the fin, besides being the basis for algorithms for curvature analysis, peak detection, between others.

3.7. Feature extraction

In this module, some features representing the dorsal fin shape are extracted, as: height and width, peak location, number and position of holes and/or boundary cracks, between others. These features will be better presented and discussed next, in Section 4 (Feature extraction).

3.8. Identification

The extracted features are then presented to the classifier, being given as result an answer about the correct identification or not of the dolphin. As a new tool, we added one of the methodologies for classifying with a self-growing mechanism, that can even find new instances of dolphins, never met previously.

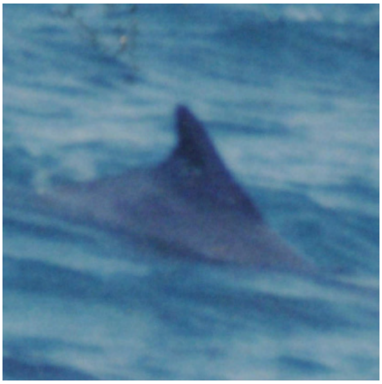


Fig. 4. Dolphin with substantial rotation around Y axis.

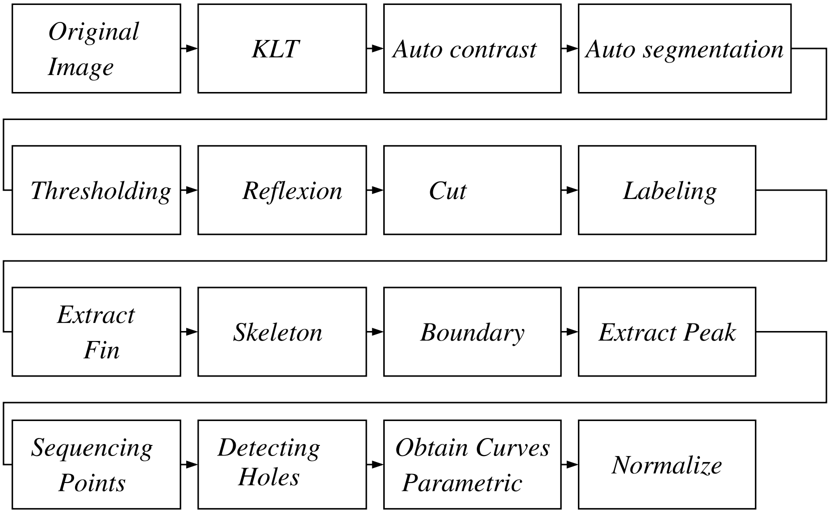


Fig. 5. Proposed sequence for pre-processing images.

4. Feature extraction

Water reflection makes texture features vary substantially from a picture to another. That is, the quantity of intensity is not uniform along the fin, thus even changing local characteristics extracted in relation to other positions in the fins. In this way, the fin shape, number and positioning of holes/cracks, are less sensitive to noise and intensity variations. Of course, there are also restrictions. Lets consider a left handed 3D system with its origin in the centre of projection of the camera (that is, with positive Z axis pointing through the picture). Fig. **4** shows an animal that represents a relatively large rotation in relation to the *Y* axis. In this case, even if the fin appears in the picture with enough size and has several holes, most of them may not be detected. So, curvature features extraction may be strongly affected. In this way, it would be necessary to apply a series of processing that preserves and enhance them at mostly. The processing sequence proposed to be applied for this is shown in Figure 5.

4.1. Preprocessing

As introduced previously, the image is initially processed with the application of KLT transformation to enhance desired features. Results of KLT is an image with two levels of intensity, we adopt 0 and 50. Using these values, the resulting image is further binarized (to 0 or 1 values). The image is then cut (pixels are put to 0) in the region below the line defined by the two inferior points of the fin entered by the user. Note that we assume the *y* axis of the camera in vertical alignment, then the interest region of the fin must be to the top of that line. So the points below it are considered background. The image is reflected (in 3D) around the *y* axis, that is, only reconfiguring the *x* coordinates. This is already a first action towards regularization for feature extraction. Subsequent process of image labelling is fundamental for identification of point agglomeration. Vector quantization is used for splitting the image in classes. Due to noise present on it after the initial phases, it is common the existence of several objects, the fin is the interest object. A morphological filtering is realized to select the biggest one (presumably the fin) to the next processing phase.

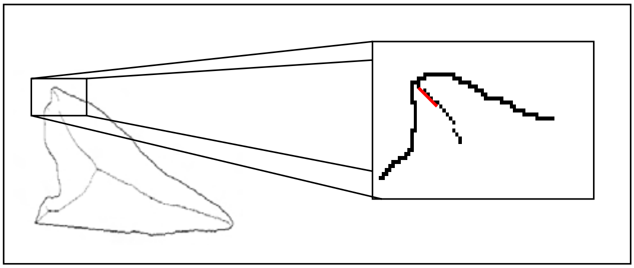


Fig. 6. Fin peak extraction from the image.

At this time, a binary image with the fin (not yet with rotation corrected) is present. Border and skeleton images are then extracted using image processing algorithms. These images will be used for extraction of the peak using the algorithm graphically depicted in Fig. 6. Initially, it extracts two sparse points on the top portion of the skeleton, some few pixels apart of each other. These points define a line segment which is prolonged up until it intersects the fin boundary. The intersection point is taken as the peak. Besides simple, it showed to be a very good approximation to the peak, as the superior shape of the skeleton naturally points to the peak.

4.2. Sequencing the fin points

In this work, the features used for identification can be understood as representations of the fin shape. So, before feature computation, we first extract and order pixels on the fin boundary. Border pixels in a binary image can be easily extracted using morphological morphology. The boundary is given by border points following a given sequence, say counter-clockwise. So, we have just to join the points finding this sequencing. The algorithm starts from the left to the right of the image. Remember during user interaction phase the approximated initial (p_1), peak (p), and final (p_2) points are manually given. Based on the orientation given by these points, if necessary, the image is reflected in such a way the head of the dolphin points to the right of the image. A search on the border image is then performed from p_2 to p_1 to find the boundary, that is, in crescent order of x. If a substantial change is detected on the y actual value in relation to the previous one, this means the y continuity is broken. So the search is inverted, taking next x values for each fixed y. When the opposite occurs, the current hole has finished, the algorithm returns the initial ordering search. In this way, most boundary points are correctly found and the result is a sequence of point coordinates (x and y) stored in an array (a 2D point structure). Results are pretty good, as can be seen in Fig. 7. It shows the boundary of a fin and the sequence of points obtained and plotted.

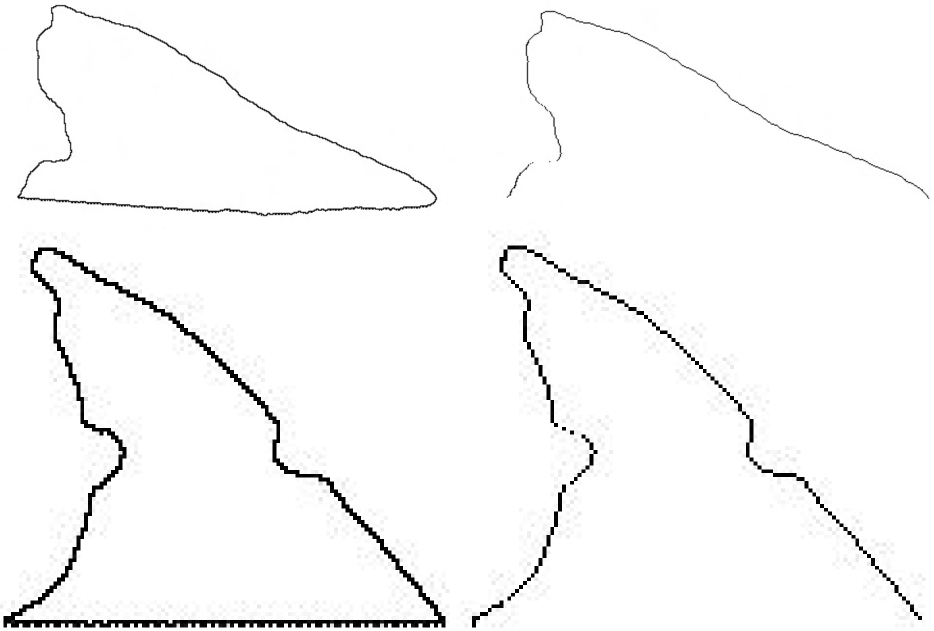


Fig. 7. Extracting fin boundary points from border pixels of the image (borders are in the left, boundaries are in the right of Figure). The method may loose some points.

4.3. Polynomial representation of the fin

For curvature feature extraction, we propose a method that differs from those proposed by Araabi (2000), Hillman (1998), and Gopi (2005), cited previously. Our method generates two third degree polynomials, one for each side of the fin. The junction of these two curves plus the bottom line would form the fin complete shape. This method has proven to be robust, with third degree curves well approximating the curvature. Further, we can use these polynomials to detect possible holes present on the fin in a more precise way. Fig. 8 describes the parameterization of the curves. Note that the two curves are coincident for λ=*0*. This point is exactly the fin peak. The parametric equations of the curves are expressed as:

$$Curve1\begin{cases} x_1(\lambda) = a_{x_1}\lambda^3 + b_{x_1}\lambda^2 + c_{x_1}\lambda + d_{x_1} \\ y_1(\lambda) = a_{y_1}\lambda^3 + b_{y_1}\lambda^2 + c_{y_1}\lambda + d_{y_1} \end{cases}$$

$$Curve2\begin{cases} x_2(\lambda) = a_{x_2}\lambda^3 + b_{x_2}\lambda^2 + c_{x_2}\lambda + d_{x_2} \\ y_2(\lambda) = a_{y_2}\lambda^3 + b_{y_2}\lambda^2 + c_{y_2}\lambda + d_{y_2} \end{cases}$$

$$\mathrm{Re}\,strictions\begin{cases} x_1(0) = x_2(0) \Rightarrow x_1(0) = d_{x_2} \\ x_2(0) = d_{x_2} \Rightarrow d_{x_2} = d_{x_2} \end{cases}$$

In this way, one needs only to solve the above system using the found boundary in order to get the parametric coefficients in *x* and y. A sum of squared differences (SSD) approach is used here, with the final matrix equation simply given by $\Lambda C_x = X$. An equivalent approach is also adopted for the second curve.

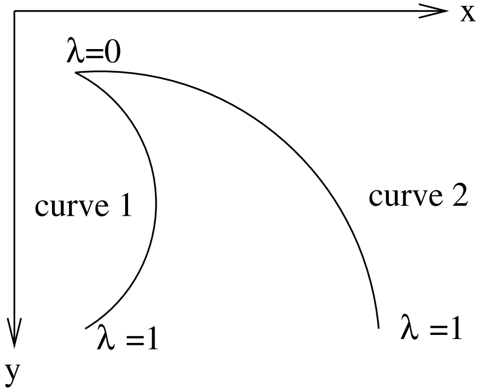


Fig. 8. Graphical model of the parametric curves used.

4.4 Obtaining images with regularized (standard) dimensions

The camera image can be assumed as a mapping from the world scene in a plane, a 2D representation. Also, we must consider that the dolphin may be rotated in the world, we use Euler angles (*x*, *y*, and *z* rotations are considered). Besides, we can do some important simplifications. The first is not to consider *x* rotation as in practice we noted its maximum value is less than 10 degrees. Lets consider the dolphin body (its length) is distributed along the x_g axis, with its head points to positive direction. The width of the animal is distributed along z_g, with its center at the origin $z_g = 0$. The height of the dolphin is along the y_g axis, with the fin peak pointing to positive values of this axis. The second simplification is that, considering the animal at some distance, the *y* coordinate of a point *P*, say P_{yr} rotated around the *y* axis and projected at the image plane will be the same as if the transformations does not happen (P_y). It is easy to understand and verify this assumption if we analyze the effects on an object close enough and far enough of an observer. If the object is close, a rotation θ around y would affect the y coordinate (the closer to $\pi/2$, the bigger change in y). But, if the object gets far from the observer, the displacement in y decreases. The perspective can be neglected and/or assumed as a rigid body transformation. Ideally at infinity, the y change would be zero. As in this work pictures are taken at a distance varying from at least 15 to 40 meters (average 30), the change in y is so small in comparison with the dolphin size and can be neglected. Note that a rotation and projection is performed here.

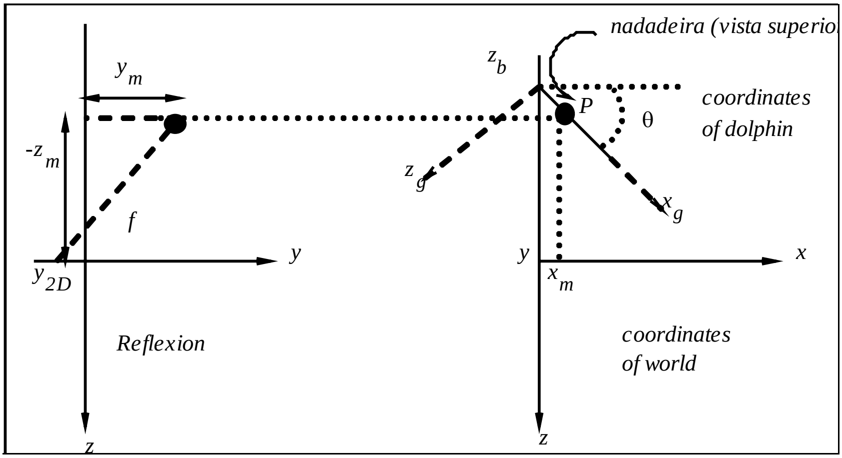


Fig. 9. Mapping between 3D and 2D

Given a picture in the XY plane with $z = 0$, the goal is to get an approximation for the coordinates of the real points of the fin on the world, mapped back to 2D. We now can consider that the fin is parallel to the observer plane without rotation. Fig. 9 can be used to better understand the model. Given the Euler angles (rotation) and assuming the simplifications suggested above, equations can be easily derived for determining the mapping between a generic image and the one on a regularized configuration. These angles are interactively given by the user when he rotates the graphical dolphin shown in a window to conform with the configuration seen on the image.

We remark that these transformations are applied to the parametric equations of the two polynomials that represent the fin. In this way, the equations representing the ideal fin (without any transformation, aligned to the x axis) are found. After these transformations, it is necessary to normalize the equations as varied sized images can be given, as explained next.

4.5. Polynomials normalization

After all pre and post-processing, a normalisation is carried out on the data. This is of fundamental importance for extracting features that are already normalized, making analysis easier, as the images of a same animal coming from the data base may be so different. As stated above, the normalization is done at the coefficients that describe the two parametric, cubic curves. This normalization process is depicted in Fig. 10.

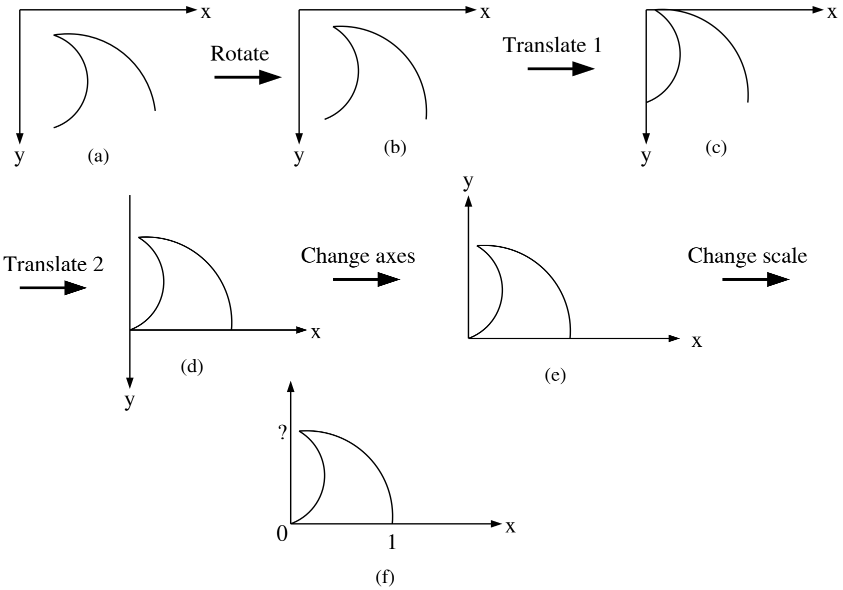


Fig. 10. Sketch of the algorithm for normalization of the polynomials.

4.5.1 Shape features (descriptors)

After polynomial normalization, shape features are then extracted. One of the most important features is the fin format, so a notion of "format" must be given. Two features are extracted for capturing the format notion: the curvature radius of the two curves and the indexes of discrepancy to a circumference. These indexes measure the similarity between one of the sides of the fin (the curve) and a circumference arc.

These curvature features are based on the determination of the center of the arc of circumference that better fits the polynomials at each side of the fin. Supposing that the circumference center is at (x_c , y_c) and that n points are chosen along the curve, the center is calculated in such a way to minimize the cost of the function $J(x_c , y_c)$ defined by:

$$J(x_c, y_c) = \sum_{i=1}^{n} l_i^2 - l_{i-1}^2$$

where

$$l_i^2 = (x_i - x_c)^2 + (y_i - y_c)^2 .$$

By solving this equation, the following matrix equation can be found:

$$AX=B$$

This can yet be expanded to:

$$\begin{bmatrix} 2(x_0 - x_1) & 2(y_0 - y_1) \\ 2(x_1 - x_2) & 2(y_1 - y_2) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} (x_0^2 - x_1^2) & (y_0^2 - y_1^2) \\ (x_1^2 - x_2^2) & (y_1^2 - y_2^2) \\ \vdots & \vdots \\ (x_{n-1}^2 - x_n^2) & (y_{n-1}^2 - y_n^2) \end{bmatrix}$$

The Equation is solved applying a SSD approach. After obtaining the center (x_c,y_c), the Euclidean distances from it to each one of the *n* points of the considered fin side are calculated. The radius is given by the mean value of these distances. Fig. 11 shows two fins that differ in their radius sizes. This can be even visually observed mainly for the left side curves, the right one is smaller, so this is proven to be a relevant feature.

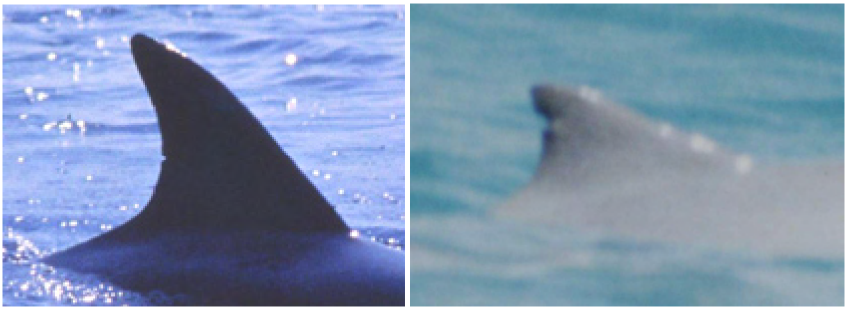


Fig. 11. Two distinct fins that can be identified by the difference in the radius (see text for explanation).

The discrepancy index is given by the sum of squared differences between the mean radius and the distance to each of *n* points considered. Note that for fin formats exactly as a circumference, this index would be zero. The two fins of Fig. 11 have indexes close to zero on the left side, besides different radius. Note that two fins with almost the same calculated radius may have different discrepancy indexes, as for the ones on right sides of Fig. 11.

Peak X	0.0958149			
Peak Y	0.835369			
Number of Holes				
NumHoles 1	0.4			
NumHoles 2	0.2			
Location 1				
Holes1 1	0.129883	0.546875	0	0
Location 2				
Holes 2	0.640625	0	0	0
Radius 1	0.357855			
Radius 2	0.436765			
Index 1	0.034204			
Index 2	0.010989			

Table 1. Features calculated for a given image of a dolphin.

4.5. The feature extraction process for identification

After all processing presented at previous sections, features for identification are extracted using the two polynomials. Between all measures (values) that can be used as features, we have found 16 relevant features:

a) Two coordinates (*x* and *y*) of the peak;

b) Number of holes/cracks in each curve (from observations in the population, a number between 0 and 4 is considered) normalized (that is, multiplied by 2/10);

c) The locations of holes/cracks for each curve, at most 4, being given by the curve parameter value at the respective position (this parameter is already between 0 and 1);
d) Radius of curvature, corresponding to each parametric curve;
e) Discrepancy indexes to the circumference for each curve;

Table 1 shows an example of features extracted for a given animal image, which are already normalized. The number of holes (NumHoles = 0.4 and 0.2) are normalizations given by 2(2/10) and 1(2/10), respectively, and two locations for Holes1 and one for Holes2 are defined, the others are zero. Also note that the discrepancy index of the second side (Index2) is smaller. This indicates the shape of that side is closer to a circumference arc.

5. Identification and recognition

In the context of this work, identification means the determination of a specific animal instance by definition or observation of specific features from the images of it. That is, features extracted from the picture compared with a model (using a classifier) would give as result this specific animal. In a general sense, we say that a specific object can be identified by a matching of its specific features. Recognition means the current observation of a previously seen animal or group of animals. In this sense, recognition complains with a more general classification context, in which a given class can be recognized by way of matching some features of it. So in this work we might be more interested in identification than recognition, besides the last would also be required somewhere.

5.1. Template matching

Two techniques are used in this work for identification/recognition. The first uses simple template matching to determine the distance of the feature vector extracted from a given image to a previously calculated feature vector, a model existing in an internal population table (like a long term memory). That is, for each vector in the table, the difference is calculated for all of its components, being given a score for it. At the end, the highest scored vector is assumed to be the winner, so the specific dolphin identified. Being f_i a feature value, m_i a value of the corresponding feature in the model (table), n the number of features in the vector, the Equation for calculating the score for each pattern in the table is given by:

$$s = 10 - \frac{10}{n}\sum_{i=1}^{n} f_i - m_i$$

In the case of a total matching, the resulting score would be 10. Because all features are normalized, the summation would be up to n (number of features). In this work, number of features are 16). In the worst case (no matching at all), the score would be zero. Note that the above matching procedure ever produces a winner no matter its feature values are stored or not in the table. So, it would be hard to determine if a new individual is found by looking the results, in the case a minimum threshold for a positive identification is not set. Of course, by hand (visual comparison) this would be harder, or even impracticably. We thought in establishing a threshold in order to verify for new individuals, but that shows to be hard too. The problem here is how to define the most relevant features that may separate one individual to another. That varies from dolphin to dolphin. For example, the curvature of the fin may be the key for separating two dolphins, but the number of holes may be the key for other two. Then, we decided to find another, more efficient technique for identification. Besides, this technique initially allowed a qualitative analysis of the extracted features, verifying them as good features.

5.2. Using a backpropagation network

In a previous work (Garcia, 2000) we used a multi-layer perceptron trained with the backpropagation algorithm (BPNN) (Braun, 1993). We decided to verify its application to this work case. So in a second implementation, the BPNN model is applied. Basically, the BPNN is a fully connected, feed-forward multi-layer network. That is, activation goes from the input to the output layer. Knowledge is codified at the synapses (links with weights), in each connection between the layers, at unities that are called knots or neurons. In the training phase, the synapse weights are adjusted by a process which has basically two steps (or phases). In the first step, called forward, the algorithm calculates the forward activation in the net using current weights. In the second step (the back propagation step), the error in the last layer is calculated as the difference between the desired and calculated outputs. This error is propagated to back, to intermediate layers which have their weights approximated corrected following an adjusting function. Then, another input is chosen and the forward step is done again for this new instance, and so on until the weights reach a stability state. Ideally, random instances are to be chosen at each step, and the net would be trained at infinity to produce exactly matches. Of course, a time limit is also set to stop it, but so after a good approximation is reached. After the net gets trained, if an input is presented to it, activation flows from the input to the output layer, which represents the correct answer to the input pattern.

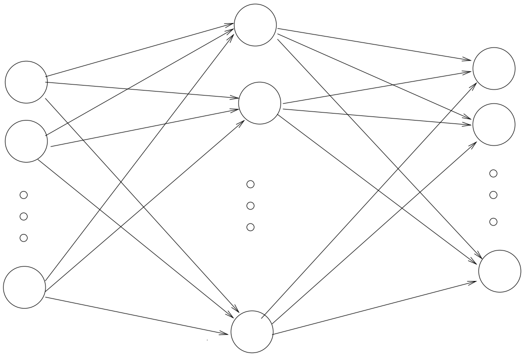


Fig. 12. Topology of backpropagation network used.

The BPNN algorithm and its variations are currently well defined in the literature (Rumelhart, 1986; Werbos, 1988; Braun, 1993; Riedmiller, 1993} and implementations of it can be found in a series of high-level tools (as WEKA, Mathlab, etc). We implemented our own code, using C++ language.

Understanding that the net must act as a (short) memory, learning automatically new features, we have proposed an extension in the BPNN original structure in such a way new instances of dolphins can be added without needs of user intervention. That is, the system would find itself a new dolphin instance by looking at the output layer results, acquire the new feature set for this new input, re-structure the net, and retrain it in order to conform to the new situation (producing as result a new dolphin added to the memory). We named this extension as a self-growing mechanism.

Fig. 12 shows structure of the used in this work. It has one input for each feature above described (totaling 16 inputs). The number of nodes in the hidden layer changes according to the number of, actually known, instances of dolphins (self-growing). This number was determined empirically, 1.5 x the number of nodes in the last layer produced good results. The number of nodes in the last layer changes dynamically. They are the number of dolphins already known. So a new output node is added for each new instance of dolphin detected in the environment. This is a very useful feature, avoiding human intervention for new instances. Note that each one of the extracted features (see Table 1) is normalized (values between 0 and 1). This is due to imposed restrictions on the BPNN implementation, in order for the training procedure to converge faster. If an input feature vector representing an individual already in the memory is presented, the activation for the corresponding output would be ideally 1, and all the others 0. So the question is how to define a new individual, in order to implement the above mentioned self-growing mechanism. A simple threshold informing if an instance is new (or not) was initially set experimentally. For a given input, if its corresponding output activation does not reach the threshold, it could be considered as a possible new instance. In addition all the not activated outputs also must be very close to 0. In fact, this simple procedure showed some problems and we decided for another way, defining a weighting function, using the net errors. The novelty of an input vector is given by a function that considers the minimum and maximum error of the net, given in the training phase, and the current activations in the last layer. Considering o_a as the value of the current most activated output, the current net error ξ is given by $\xi = 1/2(1-o_a + 1/(n-1)\Sigma o_i)$, where o_i are the values of the other $n-1$ output nodes ($i \neq a$). Being ξ_{min}, ξ_{max} the minimum and maximum training errors, respectively, τ a threshold, the value (true or false) of the novelty η is given by:

$$\eta = \xi > ((1-\tau)+(\xi_{min}+\xi_{max})/2)$$

Note that we still consider the activation threshold, which is adjusted empirically. This procedure was tested with other applications, and features, and proved empirically to be correct for getting new instances. Feed-forward activation (in training or matching) is calculated as:

$$o_i = \left(1 + e^{-\sum_{i=0}^{A} \varpi_{ij} x_i}\right)^{-1}$$

The following Equations are used for training the net:

$$\Delta \varpi_{ij}(t+1) = \varepsilon \delta_j o_i + \alpha \Delta \varpi_{ij}(t+1)$$

where o_i is as defined above and

$$\delta_j = \begin{cases} o_j(1-o_j)(y_j - o_j), & \forall j \in output \\ o_j(1-o_j)\sum_{k=1}^{B} \delta_k \varpi_{jk}, & \forall j \in others \end{cases}$$

6. Experimental results

In order to test both identification algorithms and to prove the usefulness of the feature set proposed, we have performed several experiments. Basically, several images of different animal instances preferably in a regularized positioning are selected, the images passed by all the processes mentioned above, and the features calculated. These features are used either as the model or as input to both tools. In the following, we detail the experimentations performed and their results.

6.1 Template matching results

For the first set of experiments (template matching), the table containing the feature set of selected animals is constructed in memory. Another image set, of dolphins already in memory, with poses varying a little or strongly in relation to the original one, is chosen from the database and presented to the system. By the results shown in Table 1, we can see that one of the animals is wrongly identified. That is, a feature set extracted from an image of a given animal present at the table was matched to another one. That is due to a certain degree of distortion of the animal pose in the presented image in relation to the original image pose, to image noise, to illumination bad conditions, or even to very similar features from a dolphin to another.

Minimum score	8,509124
Maximum score	9,914631
Positive identification	4
False identification	1

Table 2. Images of known dolphins are presented for template matching.

Next, in this experimentation set yet, another set of images of animals not previously in the table are presented. All of them have had wrong positive identification, of course, but, surprisingly, some of them even with high (good) scores (0.065% of total error). Table 3 shows the scores for this experimentation.

Minimum score	7,192648
Maximum score	9,341214
Positive identification	0
False identification	5

Table 3. Results for images of dolphins not present in the Table, using template matching.

6.2. Experiments with the backpropagation net

For experimentations using the BPNN, the net is first trained with a current set of features. In our work, it is initialized with no dolphins. For implementation purposes, the net is initialized with two output nodes, the first for all input feature values equals to 0 and the second for all input values equals to 1. So, in practice, there exist two virtual dolphins, only for the initialization of the memory. As all of the real images have not all the features in 0 nor in 1, exclusively, these virtual dolphins (outputs) will not influence. So, at the very first time, the network is trained in this configuration. The activation threshold is initially set to 0.90.

The first image is presented, its features extracted, and matched by the network. The memory presents some activation for the two existing nodes, under the threshold for both. Novelty given above is set to true meaning that a new instance of dolphin is discovered. The self-growing mechanism inserts the new set of features in the net, and retrains it, now containing a real dolphin features on it, besides the two initial nodes. Next, other set of features of a different dolphin is presented. The above procedure is repeated, and so on, until features of 8 different dolphins could be inserted in the net. The ninths has had activation over the threshold that was initially too low but was wrongly identified. So the threshold might be set upper, in order to be more selective or another way (function) to discover the new dolphins might has to be found. The first option is chosen, but we stop here this experiment in order to try with same pictures already in the memory, and to other cases of same dolphins with modified poses. So, as an initial result, in all 8 cases (not previously existing individuals) the activation is all under the threshold, so they could be inserted in the net as new instances. At the final configuration of the net (10 outputs), it took 500 epochs for training in less than 1 second in a Pentium 4 processor, 3.0 GHz.

Output	1	2	3	4	5	6	7	8
Activated	0.95	0.04	0.01	0.00	0.01	0.00	0.02	0.00
Desired	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Activated	0.03	0.92	0.04	0.00	0.01	0.00	0.00	0.01
Desired	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Activated	0.00	0.04	0.94	0.03	0.00	0.00	0.02	0.03
Desired	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Activated	0.00	0.00	0.02	0.97	0.00	0.01	0.00	0.02
Desired	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
Activated	0.01	0.01	0.00	0.00	0.97	0.01	0.00	0.01
Desired	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Activated	0.01	0.00	0.00	0.00	0.01	0.98	0.00	0.02
Desired	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Activated	0.03	0.00	0.03	0.00	0.01	0.00	0.97	0.00
Desired	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
Activated	0.00	0.04	0.03	0.02	0.01	0.03	0.00	0.96
Desired	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Table. 4. Output of the net (activated and desired).

In order to verify robustness of the training process, the same image set is then presented to the net. All of them are positively identified, as expected. The training and presented set are exactly the same, this is to test the net trustworthiness. Table. 4 shows the activation in the output layer nodes and their desirable activations (as same images are used). The maximum error is 0.076041 (ideally would be zero), minimum error is zero and the mean error is 0.015358 (ideally would be zero). This demonstrates that the training algorithm works on the desired way, converging to an adequate training.

Next, other 4 images of instances already present in the Table (1, 3, 5, 7), with little modification in pose are presented to the net. They are all positively identified, and, as can be seen in Table 4, the activation to its corresponding model is over 0.85 in all of them. The next experiment presents results for 3 images with strongly modified positioning in relation to the model. This means poses are rotated almost 30 degrees in relation to the models. There is one case with activation under 0.85, and with other nodes more activated than expected (see Table 6). Cases like this one could be considered as confusing, making the self-growing mechanism act, wrongly.)

Output	1	2	3	4	5	6	7	8
Activated	0.85	0.08	0.04	0.06	0.01	0.14	0.01	0.00
Desired	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Activated	0.05	0.14	0.90	0.10	0.01	0.00	0.06	0.05
Desired	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
Activated	0.00	0.04	0.00	0.00	0.92	0.04	0.02	0.02
Desired	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Activated	0.00	0.02	0.01	0.00	0.01	0.07	0.87	0.00
Desired	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00

Table. 5. Activation given by the last layer of the net for dolphins with little modification in pose.

Output	1	2	3	4	5	6	7	8
Activated	0.73	0.04	0.12	0.00	0.25	0.14	0.01	0.03
Desired	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Activated	0.11	0.04	0.02	0.20	0.12	0.80	0.04	0.12
Desired	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
Activated	0.23	0.15	0.03	0.00	0.21	0.10	0.85	0.01
Desired	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00

Table. 6. Activation given by last layer of net, dolphins with strongly modified poses up to 30 degrees rotated.

In relation to the training time, the graphic illustrated in Fig. 13 shows processing time in miliseconds versus number of dolphins (that is, number of nodes of last layer plus two) in the net. One can note that for 10 dolphins the net takes some 200 ms. The graphic illustrated at Fig. 14 shows number of epochs versus number of dolphins. We note that the input layer is composed of 16 nodes, a 16 degree dimensional feature vector space. With 28 dolphins, the hidden layer has 70 nodes. The graphics show an apparently exponential function. This is not a problem since the number of dolphins in a family should not be more than 50. Yet, in a practical situation, the system is trained only when a new animal is found. That can take some time without problems since the system is trained of-line. In this case, the weights can be saved to be used in the time it becomes necessary, not interfering in the system performance.

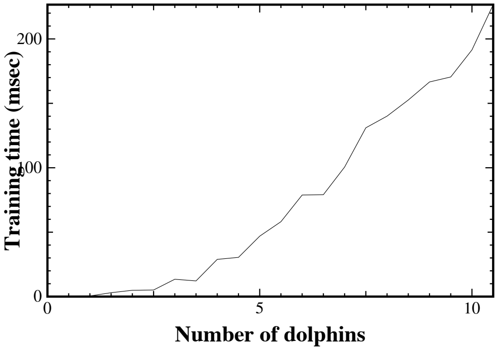


Fig. 13. Time versus number of dolphins in the net.

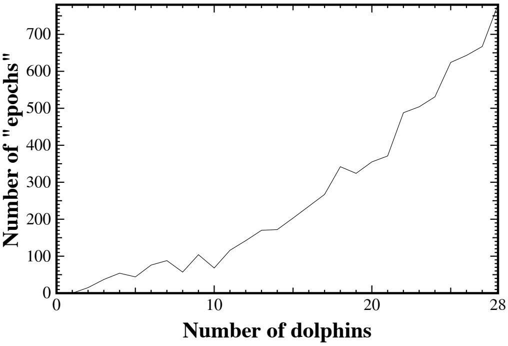


Fig. 14. Epochs versus number of dolphins.

For on-line situations or for situations in which the net should be trained fast, other approaches can be used. As for example, one solution is to increase the input space state that each node of the output can reach. One way to do that is to set each output for more than one instance, dividing the range of each output from zero to one in more than one value depending on the number of dolphins. For example, instead of one output for each dolphin, a binary notation can be used, addressing two dolphins. So, with 20 outputs, a number of 2^{20} of dolphins can be addressed by the net. For 50 dolphins, 6 nodes on the output layer are enough, instead of the 50 in the current implementation. Of course, training time may not decrease so fast as the number of nodes, but for sure it will be faster due to necessity of less computation for feed-forwarding the net and back propagating the error. By comparing the above results for both classifiers, the net and the straight-forward method using template matching, we could see that the net, besides slower, performed better producing activations with more reliable values. Yet, it is easier to deal with, inserting features for new instances in automatic way.

7. Conclusion and future research

In this work, we have proposed a complete system for identification of dolphins. The system involves image acquisition, pre-processing, feature extraction, and identification. The novelties of the system are mainly in the proposed methodology for pre-processing the images based on KLT for extraction of the dorsal fin, the new feature set proposed for being used as input to the classifiers, and the self-growing mechanism itself. With that, the system does not need user intervention for training its memory, neither for identifying new subjects in the universe of animals. The system has proceeded as expected in all experiments.

Of course, other alternatives for the classifier can be tested in the following, for example, Bayesian nets, self organizing maps, or radial basis functions. Self organizing maps has the property of separating the input space in clusters. We do not know if this would be a good strategy, it has to be tested, perhaps for separation between families. It is important to remark that the feature set used can be enhanced. A good set must consider texture, besides shape features. In order to use texture, one must use an approach for avoiding water effects in the image illumination. There are some restrictions to be observed in the feature extraction. For example, in the case of substantial rotation, close to 90 degrees, the holes or cracks may not be visible. Holes may not be detected, prejudicing the system performance. Of course, a visual procedure in the considered picture would not produce any good result too, as in this case the fin may become visible from the front or from the back of the dolphin. So, besides the necessity of such improvements, we enhance the importance of the features proposed, which has produced good results.

The BP net used showed to be useful in the insertion of new individuals in the long term memory. Yet, some strategy can be developed in order for the system to learn what features are the best to segregate individuals from a given group in a more precise way. If the group changes, the fins characteristics may also change. In this way, a smaller feature set can be used, diminishing training time and growing in efficiency. Also, using weights in the features is another strategy to be tried. The idea is to determine the most relevant features, for specie specifics. A stochastic inference approach can be tried, as future work, in this track.

8. References

Araabi, B.; Kehtarnavaz, N.; McKinney, T.; Hillman, G.; and Wursig, B. A string matching computer-assisted system for dolphin photo-identification, *Journal of Annals of Biomedical Engineering*, vol.28, pp. 1269-1279, Oct 2000.

Araabi, B. N.; Kehtarnavaz, N.; Hillman, G.; and Wursig, Bernd. Evaluation of invariant models for dolphin photo-identification. *Proceedings of 14th IEEE Symposium on Computer-Based Medical Systems*, pp 203-209, Bethesda, MD, July 2001 (a).

Araabi, B. N. *Syntactic/Semantic Curve-Matching and Photo-Identication of Dolphins and Whales.*, PhD Thesis, Texas A&M University, 2001b.

Cesar Jr., R. and Costa, L. d. F. Application and assessment of multiscale bending energy for morphometric characterization of neural cells. *Reviews on Science. Instrumentation*, 68(5):2177--2186, 1997.

Cesar Jr., R. & Costa, L. F. Pattern recognition. *Pattern Recognition* , 29:1559, 1996.

Defran, R. H.; Shultz, G. M. ; and Weller, D. A technique for the photographic identification and cataloging of dorsal fins of the bottlenose dolphin. Technical report, *International Whaling Commission* (Report – Special issue 12), 1990.

Flores, P. Preliminary results of photoidentification study of marine tucuxi, Sotalia fluviatilis in southern brazil. *Marine Mammal Science*, 15(3):840--847, 1999.

Garcia, L. M.; Oliveira, A. F. ; and Grupen, R. Tracing Patterns and Attention: Humanoid Robot Cognition. *IEEE Intelligent Systems*, pp 70-77, July, 2000.

Gope, C; Kehtarnavaz, N. D.; Hillman, Gilbert R.; Würsig, Bernd. An affine invariant curve matching method for photo-identification of marine mammals. *Pattern Recognition* 38(1): 125-132, 2005.

Hillman, Gilbert R. Tagare, H. Elder, K. Drobyshevski, A. Weller, D. Wursig, B. Shape descriptors computed from photographs of dolphins dorsal fins for use as database indices. *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 20, 1998.

Jain, A. K. *Fundamentals of Digital Image Processing*. Prentice-Hall, October 1989.

Kreho, A.; Aarabi, B.; Hillman, G.; Würsig, B.; and Weller, D. Assisting Manual Dolphin Identification by Computer Extraction of Dorsal Ratio. *Annals of Biomedical Engineering*, Volume 27, Number 6, November 1999, pp 830-838, Springer Netherlands.

Link. L.d.O. *Ocorrência, uso do habitat e fidelidade ao local do boto cinza*. Master thesis, Marine Biology Department. Universidade Federal do Rio Grande do Norte, Brazil, 2000.

Mann, J. R. Connor, P. Tyack, and H. Whitehead. *Cetaceans societies - field studies of dolphins and whales*. The University of Chicago Press, Chicago, 2000.

Riedmiller, M.; and Braun, H. A direct adaptive method for faster backpropagation learning: The rprop algorithm. *Proc. of the International Conference on Neural Networks (ICNN'93)*, pages 123--134. IEEE Computer Society Press, 1993.

Rumelhart, D. E. ; Hinton, G. E. ; and Williams, R. J. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the microstructure of cognition*, volume 1: Foundations. The MIT Press, Cambridge, Massachusetts, 1986.

Werbos, P. Backpropagation: Past and future. *Proceedings of IEEE International Conference on Neural Networks*, pages 343--353, 1988.