Robust localization and identification of African clawed frogs in digital images

Tek, F.B., Cannavo, F., Nunnari, G. and Kale, I.

NOTICE: this is the authors' version of a work that was accepted for publication in Ecological Informatics. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in Ecological Informatics, 23, pp. 3-12, 2014.

The final definitive version in Ecological Informatics is available online at:

https://dx.doi.org/10.1016/j.ecoinf.2013.09.005

Robust Localization and Identification of African Clawed Frogs in Digital Images

F. Boray Tek[a], Flavio Cannavo'[b],Giuseppe Nunnari[c], İzzet Kale[d]

[a]*Department of Computer Engineering, Işık University, Şile, İstanbul, Turkey, 34980, boray@isikun.edu.tr*

[b]*Istituto Nazionale di Geofisica e Vulcanologia, Piazza Roma 2, 95123 Catania, Italy, flavio.cannavo@ct.ingv.it*

[c]*Dipartimento di Ingegneria Elettrica, Elettronica e Informatica, Università di Catania, Viale A.Doria 6, 95125, Catania, Italy, gnunnari@dieei.unict.it*

[d]*Applied DSP and VLSI Research Group, Department of Electronic Network and Computer Engineering, Faculty of Science and Technology, University of Westminster, 115 New Cavendish Street, London, W1W 6UW, UK, kalei@westminster.ac.uk*

Corresponding author: F. Boray Tek, *Department of Computer Engineering, Işık University, Şile, İstanbul, Turkey, +902165287145, boray@isikun.edu.tr*

Abstract

We study automatic localization and identification of African clawed frogs (*Xenopus Laevis sp.*) in digital images taken in a laboratory environment. We propose a novel and stable frog body localization and skin pattern window extraction algorithm. We show that it compensates scale and rotation changes very well. Moreover, it is able to localize and extract highly overlapping regions (pattern windows) even in the cases of intense affine transformations, blurring, Gaussian noise, and intensity transformations. The frog skin pattern (i.e. texture) provides a unique feature for the identification of individual frogs. We investigate the suitability of five different feature descriptors (Gabor filters, area granulometry, HoG[1], dense SIFT[2], and raw pixel values) to represent frog skin patterns. We compare the robustness of the features based on their identification performance using a nearest neighbor classifier. Our experiments show that among five features that were tested, the best performing feature against rotation, scale, and blurring modifications was the raw pixel feature, whereas the SIFT feature was the best performing one against affine and intensity modifications.

Highlights

- A stable frog localization and skin pattern window extraction method is introduced.
- The method compensates for scale and rotation changes and blurring distortions.
- The pattern window is stable to allow the use of raw pixel values as a feature.

[1] Histogram of gradients
[2] Scale invariant feature transform

Keywords

Xenopus Laevis, automated frog identification, skin pattern recognition, area granulometry, SIFT, HoG.

1. Introduction

The African clawed frog (*Xenopus Laevis sp.*), is the largest of the genus Xenopus which are the only frogs with clawed toes. It is native to Sub-Saharan Africa. After the 1960s, it was farmed and distributed widely for pregnancy tests and the pet trade. More recently owing to its genetic simplicity, it has been used as a model in development biology and it has become one of the standard experimental amphibians (Kay and Peng 1991). However, releases from captivity and escapees have formed viable and invasive populations pan-globally (Matthews and Brand, 2004). Moreover, owing to its highly adaptable and carnivore feeding, it dominates and endangers local species. It is listed as a threat to local species for Chile, UK and preventively controlled in the US, California (Measey et al., 2012).

The field of image recognition has shown a rapid development in recent years. The capability to recognize entities is becoming crucial for many applications in different fields, such as surveillance, automation and control. Many recent studies are devoted to algorithms aiming at detection and recognition of objects, iris, or faces (Dalal et al., 2006; Lowe, 2004; Ma et al., 2003). However, the literature for automatic recognition or detection of insects, animals, or plants is quite rare. This study mainly concerns automatic recognition of African clawed frogs in a laboratory environment. However, the techniques and discussions reported here will be valuable for their detection, or localization, and identification in the wild. Moreover, it is as important for recognition of other skin- or fur-textured species since our models can be generalized.

A system which is able to locate, detect, and recognize the African clawed frog (*Xenopus Laevis sp.*) can be used for counting, tracking or monitoring of the individuals. This is a time-consuming and difficult task when performed manually. Though there are different types of methods which could be used for identification, a non-invasive procedure is possible based on the unique skin patterns of the individual frogs (Hubrecht and Kirkwood, 2010). The observed color of the skin can change through time; however the skin pattern is stable.

Observing the skin patterns (Figure 1) the first question that arises is whether the popular texture analysis descriptors will be successful in differentiating them. There are many texture analysis and classification studies in the literature (Chen et al., 1998; Mirmehdi et al., 2009; Wang and Yong, 2008). Moreover, a variety of techniques have been developed for measuring texture similarity (Huang and Dai, 2003; Manjunath and Ma, 1996). Most of these techniques rely on comparing values of what are known as second-order statistics such as the degree of contrast, directionality and regularity (Tamura et al., 1976); or periodicity, directionality and randomness. Alternative methods of texture analysis for image retrieval include the use of

Gabor filters and fractals (Kaplan, 1999). Gabor filter is widely adopted to extract texture features from the images for image retrieval (Manjunath and Ma, 1996), and has been shown to outperform other wavelet transform based feature descriptors.

However, recent advances on image representation have provided some new powerful feature descriptors which are applied to a wide range of recognition and classification problems. Popular ones include SIFT (Lowe, 2004), HoG (Dalal et al., 2006), and others (Mikolajczyk and Schmid, 2005). It is an interesting question whether these modern feature descriptors could be successful in X. Laevis skin pattern representation.

While our main objective is to detect and recognize a frog's identity, this study focuses mostly on the suitability of the different feature descriptors for the task. It extends our previous study (Cannavo' et al., 2012) in major ways: it tests and shows the stability of our skin pattern window extractor; it compares popular feature descriptors: Histogram of Gradients and Scale Invariant Feature Transform (HoG and dense SIFT) with Gabor filters, and morphological area granulometry; finally, given the stability of the pattern window extractor it questions whether it is possible to use a specifically located window of raw pixels as a feature descriptor.

Our main approach can be summarized in four steps: localization of the frog body, skin pattern window extraction, feature extraction, and identification. The first step is necessary to localize and segment the body of the frog in the laboratory container; the second step is crucial to locate the central part (pattern window) of the frog's skin; the third step consists of calculating the features of the pattern window; finally, the last step is the classification of the feature vector to find the closest match from a database of pre-recorded frogs.

This paper is organized as follows. Our methodology comprised of the pre-processing of the images, localization of the frog body, the skin pattern region extraction algorithm, and the calculation of different feature descriptors are discussed in Section 2. The description of data and experimental results are given in Section 3. Discussions and conclusions are presented in Section 4 and 5, respectively.

2. Methodology

In our dataset, the frogs are pictured in random positions and orientations in a white plastic bowl container which is filled with water (Figure 1). The background is mostly comprised of the white container (bowl). The scale and the illumination conditions are uncontrolled and differ. The first part of the proposed algorithm is to analyze the frog position and its orientation. First, the image is thresholded to identify background (bowl) and foreground (frog) regions. Then the component with the largest area is chosen to be an estimate of the frog location. Then the localized body is further processed to extract a central square region which we name as the *pattern window*. This window is transformed (normalized) to be invariant to position, scale, and orientation changes.

2.1 Frog localization

In this stage, the grey level frog image is thresholded to obtain a binary image by using Otsu's adaptive thresholding method (Otsu, 1979). Otsu's method searches exhaustively for the threshold that minimizes the intra-class variance (the variance within the class), defined as a weighted sum of variances of the two classes:

$$\sigma_P^2(T) = P_1(T)\sigma_1^2(T) + P_2(T)\sigma_2^2(T) \tag{1}$$

, where the weights P_i are the probabilities of the two classes separated by a threshold T, computed from the histogram as T, and σ_i^2 variances of these classes. This method is robust to isolate the frog's body from the background (in the described pictures), which has the white container as the background. In the obtained black and white image (b&w) the noisy small components are removed using a morphological area opening operation (Soille, 1999). Then, the holes in the binary image are filled by morphological reconstruction (Soille, 1999). After this, an erosion/opening operation is applied to smooth the borders. At the end of this process, the frog's body is the component with the largest area in the b&w image. The morphological opening operation with a disk shaped structuring element that has a radius equal to the frog's pseudo-radius provides a symmetrical body by removing the arms and legs. The pseudo-radius value is estimated from the area of the largest component. Then in order to calculate the orientation of the body the second order geometrical moments are calculated on binary image $f(x,y)$ (Hu, 1962).

$$m_{p,q} = \sum_x \sum_y x^p y^q f(x,y) \tag{2}$$

The first order moments ($p + q = 1$) give the coordinates of barycenter (i.e. center of mass), whereas the second order moment provides the orientation of the body. The line that crosses the barycenter of the body which has the angle obtained from frog's orientation is considered as the body line of the frog. Figure 2 (a-e) demonstrates the steps from gray-scale conversion to the localization of the barycenter.

2.2 Pattern region extraction

Once the barycenter is located and the orientation of the body is estimated, the algorithm proceeds to seek a stable region in the frog's back to prepare for the identification process. Despite the fact that the skin is patterned from head to the tip of the legs, we have considered the central part of the back which is relatively stable (compared to the legs and arms) and large (compared to the head). These considerations are also useful in locating the head of the frog: due to the stronger legs, the rear part of the body is bigger than the front part. Hence, we determine the head-side of the frog by calculating the projection of the b&w image along the body direction. In the projection, the lighter side and the tip are assigned to be the head/head-nose region, whereas the heavier side and the tip are assigned as the rear region and the tail points, respectively. The head-tail localization is refined further with an optimization step, where we devised a sequential search algorithm that is inspired from the simulated annealing process which is a random-search technique for a minimum in a generalized system (Kirkpatrick et al., 1983). In our case, given the initial body orientation from the barycenter, it starts searching around with relatively high angle variations and then decreasing it to locate the optimal (orientation) angle of the body line which maximizes the head-tail distance (Figure 2(f)). A further improvement consists of using the simulated annealing optimization algorithm to adjust the head and tail coordinates to search for the best, in terms of symmetry, isosceles triangle for the head (Figure 2(g)). In fact, given a pair of points, head and tail, it is possible to pick out a triangle with vertices in the head-nose and frog's side flanks. Flank points are initialized by the crossing points between the peripheral of the body and a line which passes from the center of the head-tail segment and that is perpendicular to the head-tail line. We measure the triangle symmetry by the difference between the two side edges formed by the head-flank points. We use the simulated annealing algorithm to find the most symmetric triangle:

$$\min_{H,T}|HF1 - HF2| \tag{3}$$

, where H and T stands for Head and Tail points, and $HF1$ and $HF2$ are the lengths of the segments between the head and the first flank found in the clockwise direction and the opposite one respectively.

To avoid flipping or rotation problems, the orientation coherence of flank points is assured by the sign of the cross-product of the found vectors $\overline{HF1}$ and $\overline{HF2}$. In case of a negative magnitude the flank points are swapped.

The last step of the algorithm extracts a square window (i.e. pattern region). This is done by locating the two flanks of the body and the other two corners on the body line as the corners of the square (Figure 2(h)).

Then the square region obtained is rotated to reach the standard position of 90 degrees (the frog in vertical position facing to the top of the image frame and tail to the bottom). It is then resized to a standard size of 200x200 pixels by using a bilinear interpolation.

2.3 Stability of the skin pattern region extractor

An ideal pattern region extractor must find the same pattern region window for the same frog under different conditions. We modeled these different conditions with some transformations and distortions: rotation, scale change, affine distortion, blurring, Gaussian noise, and pixel intensity change. The set of rotations included 10 different angles in [0,$\pi/2$]; the scale set included scaling with different factors (0.25, 0.5, 0.75, 1.25); the affine set included transformation with different affine coefficients (-0.25, -0.135, -0.02, 0.02, 0.135, 0.25), the blur modification iterated from (1 to 5) with an increasing size box mean filter; and Gaussian noise of mean 0 and increasing variance values (0.0005, 0.0010, 0.0015, 0.0020, 0.0025) were added to introduce distortions. The pixel intensity modification was simply performed by a contrast stretching operation: compute the minimum and maximum values of the image; and remap all pixel values such that the new minimum and the new maximum will be 0 and 255, respectively. Our choice for these modifications is mainly motivated by the fact that the frog's body is moving, it is flexible and it was pictured underwater. Furthermore, the frog skin can change colors and the identification can take place under different illumination conditions at different times, which will cause pixel intensity changes. Figure 3 shows examples of pattern window after some of these modifications were applied to the original image of one example frog. While the blurring and scaling effects are obvious, one must carefully inspect to see slight differences in the squares extracted from the rotation and affine transformed images. In this example, rotation causes apparent differences in the bottom regions, whereas the affine transform clearly deforms the shapes. In addition, a group of bubbles seen in the left top corner display a translation with respect to the original. Nevertheless, a careful observer can still state that these patterns belong to the same individual frog.

We have examined all of the patterns that were extracted under different modifications and intensities. We observed that the algorithm was successful in the sense that the pattern windows of the original and modified images overlapped significantly in almost all of the cases. However, in only few cases (4 in 1800), the affine transform distortion caused the pattern region extractor to locate badly positioned pattern windows.

To quantifiably measure the robustness of the pattern window extractor, we have used a simple error metric Mean Square Error (MSE) which is widely used for measuring the difference of two images from each other (Wang et al., 2004):

$$MSE(I_0, I_1) = \frac{1}{wh}\sum_{k=1}^{wh}\big(I_0(k) - I_1(k)\big)^2 \quad (4)$$

, which calculates the MSE between two monochromatic images (I_0, I_1) that are of size $[wxh]$. The MSE was calculated individually for each frog and then averaged to obtain a single number for each modification at each intensity (or parameter). A lower MSE value for a modification suggests that the newly found pattern window from the modified image is close to the one of the original. A lower average MSE indicates that this is the case for the majority of the frogs in our database. However, to ease interpretation of the MSE values we have added a control value which shows the MSE value obtained with a transformed image of an already extracted window. This control value shows the difference created by the application of the modification; that is a value we would expect to observe if the pattern window extractor were not adjusting to the modification. Figure 4 shows the MSE plots for the different modifications: (a) with respect to the rotation angle, (b) with respect to the different affine transformation coefficients, (c) with respect to the different scale coefficients, (d) with respect to the increasing blur iterations, Gaussian noise variance, and contrast stretching operations. The latter does not have any control due to the fact that there is no blurring, noise, or illumination change compensation in the pattern window extractor algorithm. However, these modifications also distort the input images and therefore they will affect the representation performance of the features.

It can be seen that the pattern window locater successfully compensates for rotation and scale changes in all the tested angles and scaling factors respectively (Figure 4(a, c)). It is also clear from the MSE plots that our toughest tests for the features are against affine and the intensity modifications. Particularly, the affine transformation modifications using larger coefficients caused the pattern window to produce larger MSE errors than the control (Figure 4(b)). This is mainly introduced by the procedure in the algorithm which geometrically estimates the body size and orientation to normalize the angle and size.

2.4 Feature extraction

The next step in our method is to extract features of the normalized pattern window. We studied the suitability of Gabor filters (Lianping et al., 2004), Histogram of Gradients (Dalal et al., 2006), SIFT and dense SIFT (Liu et al., 2011), area granulometry (Meijster and Wilkinson, 2001), and the pattern window pixel values as feature descriptors.

2.5 Gabor filter

Feature description in this approach is done by filtering the image with a bank of filters, each filter having a specific frequency and orientation. The filtering process consists of a convolution between the image and the filter. Then, the feature values are extracted from these filtered output images. A Gabor filter can be described by the following two-dimensional function (Feichtinger and Strohmer, 1998):

$$g(\bar{x},\bar{y}) = e^{-\frac{\bar{x}^2+\bar{y}^2}{2\sigma^2}} co\,s(2\pi f\bar{x}) \tag{5}$$

$$\begin{aligned} \bar{x} &= xcos(\theta) + ysin(\theta) \\ \bar{y} &= -xsin(\theta) + ycos(\theta) \end{aligned} \tag{6}$$

, where $\bar{x}$, $\bar{y}$ are the rotated coordinates, and f is the frequency. As suggested in the previous works (Lianping et al., 2004), the variance σ can be chosen to be equal to the period $1/f$. Convolutions with the filters of different size and orientations produce equally numbered output images:

$$G(X,Y) = \left|\sum_x \sum_y I(X-x, Y-y)g(x,y)\right| \tag{7}$$

The features extracted from these output images are the averages and the standard deviation on the non-overlapping cells of an nxn grid of the image (Lianping et al., 2004; Manjunath and Ma, 1996). As it can be expected, the Gabor filter responses are sensitive to changing orientation and scale. The frequencies used in the Gabor filters have been chosen by commonly used octave progression pattern as 1/4, 1/16, and 1/32. For the orientation, we used four equally spaced angles in the range (0° and 90°). We have tested several variations for the number of frequencies and orientations to observe the effects on the performance, as we report in Section 3.

2.6 Granulometry

Feature description in this approach is done by using the granulometry spectrum of the grayscale image. Granulometry was introduced by Matheron (1975) as a tool to extract size distribution from binary images. By performing a series of morphological openings with a family of Structuring Elements (SE) of increasing size, we can obtain the granulometry function which maps each SE to the amount of image volume, i.e. sum of gray levels removed after the opening operation with the corresponding SE. However, the choice of SE is important and the computational time required for the morphological opening with each SE is high. On the other hand, the computation of area granulometry is quite efficient and was employed in some recent studies as a feature descriptor (Rao and Dempster, 2001; Tek et al., 2009 and 2010). Area granulometry using morphological area openings (Soille, 1999) is sensitive to the area of the connected components in the consecutive threshold levels, where it extracts the distribution of the areas of the components. The area granulometry (f^{ag}) of an image can be defined as the difference between the sums of the pixel values of the output images after consecutive area opening (γ_T^a) operations with area threshold T. The consecutive area thresholds are of an increasing order $T_k > T_{k-1}$:

$$f^{ag}(k) = \sum_I \gamma_{T_{k-1}}^a(I) - \sum_I \gamma_{T_k}^a(I) \tag{8}$$

Due to the properties of morphological area opening (Soille, 1999), area granulometry is invariant to translation and rotation. However, it is sensitive to scale and affine variations. The area granulometry extracts the size distribution of the bright components, whereas *anti*-granulometry extracts the distribution of dark components by simply calculating the same function (8) on the negative of the grey level image (Meijster and Wilkinson 2001). We calculate area granulometry for N linearly spaced area thresholds in the range $(1 \leq T_k \leq [wxh/4])$ where w and h are the width and height of the local window, respectively. The values were then normalized by dividing them with $wxh/4$. Together with the anti-granulometry values, the granulometric feature descriptor forms a $2N$ element vector. We comment on the effects of the parameter N in Section 3.

2.7 Histogram of gradients (HoG)

Histogram of the local gradients' feature is based on the idea that the local distribution of the gradient orientations can represent the local appearance (Dalal et al., 2006). It is simply a histogram of the local gradient orientations on a dense grid of positions that was determined by the size of the local cells (i.e. regions). As Dalal et al. (2006) point out that HoG was sensitive to gradient computation kernels; however, for their problem (human detection) the simplest kernel (e.g. [-1 1] and $[-1\ 1]^T$) seemed to work best. HoG is proposed to be invariant to translations and rotations which are small with respect to the local cell size. We have used Matlab vl_feat toolbox for the (Vedaldi and Fulkerson, 2008) implementation of the HoG method with default number of orientation bins which divide the 0-π interval to nine linearly spaced bins. We report our findings on the effects of cell size parameter in Section 3.

2.8 SIFT and dense SIFT

Scale Invariant Feature Transform (SIFT) has become quite a popular descriptor quick after it was introduced by Lowe (2004). Similar to its late variant HoG, the SIFT descriptor calculates the local orientation histograms in a local window; however instead of calculating a single 1-D histogram for a fixed sized cell, it calculates 16 histograms (from sub-regions) of cells of varying size and orientation. In addition, the center of the cell where the descriptor will be extracted is not fixed; instead it is determined by the local extrema of the Gaussian scale-space. This approach creates a varying number of feature descriptors per image where the descriptors are localized only on some salient points (i.e. keypoints). Lowe (2004) proposes that the SIFT descriptors are invariant to scaling, rotation, translation, and partially invariant to affine transformation and illumination changes.

On the other hand, the *dense* SIFT (Ce et al., 2011) approach overrides the keypoint localization procedure and forces the descriptor to be extracted from a regular grid of cells in fixed orientations. Thus, similar to the HoG, the cell size parameter determines the number of feature descriptors extracted from the image. The reason that we focused on dense SIFT besides the original (Lowe's keypoint localizer SIFT) is due to the fact that the pattern region is already refined to a degree where all pixel locations are informative. We have used Matlab vl_feat toolbox (Vedaldi and Fulkerson, 2008) implementation of dense SIFT with default bin size of (4x4); default number of orientations, which divide the 0-π interval into eight linearly spaced bins. We report on the effects of varying cell sizes in Section 3 and use the SIFT with default parameters (Vedaldi and Fulkerson, 2008).

2.9 Raw pixel values

Human observers can differentiate frog skin patterns easily by the naked eye. So one important question in our study is whether we really need an abstract and second order features or is it possible to use the set of raw pixel values of the pattern window as a feature descriptor. The feasibility of this depends on the stability of the pattern window extractor, which can be regarded as the feature extractor in this case. The pattern window extractor can be regarded as stable if it locates the same square region for the same frog in every run and independent of the conditions whatever results in a different appearance.

To study the performance of the raw pixel feature in different scales, we introduced a scale reduction factor n which reduces the dimensions of the pattern window image by a factor of 2^n, which allows us to control the dimensionality of the feature descriptor.

3. Results

The original dataset included 60 images of varying sizes ([1592x1194] or [1280x960]) of African clawed frogs. All of the images were taken under similar conditions where the frog is alive; lies in a white box; and in water (Figure 1). While the training set (database) contained all of the original pictures of 60 frogs, six different test sets included 1860 modified test pictures in total: rotation with 10 different angles (60*10), affine transform with 6 different coefficients (60*6), scale with 4 different factors (60*4), blur with 5 different iterations (60*5) and Gaussian noise with 5 different variance values (60*5). In order to evaluate the performance of the features we have utilized a nearest neighbor classifier based on L1-norm distance (Dasarathy, 1991). For each modification test set and for each feature, we count and compare the percentage of frogs that were correctly recognized (i.e. identified with the original). Thus, a feature calculated from the pattern window of a modified input picture is identified as the one which has the shortest L1 distance among all in the database.

In brief, the results have shown that the raw pixel window feature was the best, compared to the other four competitors. However, it is not as good as SIFT in affine transform and performed the worst in intensity modification tests, which prevents us from stating that it is a winner. The averaged recall/precision rates among all tested pictures in different tests were as follows: rotations-raw (98.6/98.5%), scaling-raw (95.4/96.3%), affine-SIFT (93.3/94.7%), blurring-raw (97.7/96.9%), noise-SIFT (99/99.2%), and intensity-HoG (98.3/97.5%). Table 1 shows the best precision/recall trade-offs for different features for different tested parameters. The parameter tests and details are explained further below.

We were able to calculate the average precision/recall values for the different features with different parameters; however, for clearness, we report here only the ones which produced a trade-off: resolution factor for the raw pixel value window; number of bins for the granulometry; cell size for HoG; cell size, frequencies and scales for Gabor features; cell size for dense SIFT; and SIFT was used with default parameters of the VLFEAT toolbox (Vedaldi and Fulkerson, 2008).

Figure 5-10 show the average precision-recall points calculated for the different tests, for the features of different parameter settings. The individual points are indexed according to their parameter settings and markers differ with respect to the different features. The increasing index value of the markers denotes different parameters for the different features: the window resolution decrease for raw features ((200x200, 100x100, 50x50, 25x25, 12x12, 6x6, 3x3); for Hog and dense SIFT the cell size decreases (100,50,20,10); for the granulometry feature the number of bins decrease (30,20,12,7,6,5); and for the Gabor filters the cell size parameter (20,50,100) increase. Note that the individual precision/recall points that are plotted in these figures must not be interpreted as samples of a single precision-recall curve. Rather, each point probably lies on a different curve, which we could not reconstruct completely due to the fact that the NN classification does not generate a posterior Bayesian probability to be used for a decision threshold.

Looking at Figures 5 to 10, the first and the most important observation was that the raw pixel feature performed better compared to all other features in rotation, scaling and blurring tests. We can also observe that the coarser resolutions (reduced size windows) provided better performing features against the finer. An exception to this was the unacceptably poor performance in the intensity modification test set (Figure 11). However, a similar relation (cell size versus performance) was also observable in the winner of this test (HoG) and the granulometry features.

In terms of the performance, the raw window feature was closely followed by the SIFT feature with two wins in very important tests (Figure 7 and 9), despite whilst running with the default parameters out of the VLFEAT toolbox (Vedaldi and Fulkerson, 2008).

HoG was the winner of the intensity modification test (Figure 10) with larger cell sizes, despite not performing as well for the noise and affine modification tests (Figure 7 and 9).

Except for the intensity modification test (Figure 10), the granulometry feature did not performed well. In this test, increasing the number of bins had an inverse effect on its performance.

The Gabor filter feature performed the best against Gaussian noise test; however, it was not much effective for the rest. We have experimented on different number of scales, orientations, and cell sizes (to calculate the mean and standard deviation of the sub-regions). We observed that increasing the number of scales beyond three did not affect the results, whereas doubling orientations from four to eight did result in a slight improvement. In most cases, increasing the cell size had an inverse effect on its performance. Similar to Gabor, the dense SIFT under performed. Smaller cell sizes were slightly better than the larger ones.

SIFT outperformed all other results against affine modification (Figure 7). Note that SIFT uses keypoint localization and produces a variable number of feature vectors (i.e. histograms) for each image. Hence, only for the SIFT features, we had to utilize the so called SIFT feature matching approach, rather than the nearest neighbor classifier. The SIFT feature matching rejects some matching features by checking their relative value with the respect to the best match. This approach was proposed by Lowe (2004) and is implemented in (Vedaldi and Fulkerson, 2008).

4. Discussions

Overall, the results show that there was no clear winner which can perform best for all modification tests. However, the raw pixel feature could be the best candidate if it did not perform very poorly for the intensity modifications. Considering that a flexible (and moving) frog is to be pictured underwater, in an uncontrolled environment, perhaps the most important robustness tests were the affine and intensity modifications.

SIFT performed best against affine modifications, which is most probably due to the feature point (keypoint) selection strategy, which prefers affine invariant feature points. This robustness against affine modifications is most valuable when we consider that two pictures of a live frog may not be taken from the exact same view angle.

As we saw from the stability experiments, the scale and rotation modifications are compensated reasonably well by the pattern region extractor. This is also confirmed by the rotation and scale sensitive features (e.g. raw pixels, Gabor, HoG) performing well against these modifications.

To improve the raw pixel feature performance under intensity modification, it may be possible to store and use for matching more than one appearance model per frog. In addition, it is also possible to use it jointly with a different feature(s), for which HoG, or SIFT seem to be the appropriate candidates.

Finally, the average time for calculation of the different features were in the following order: raw pixels, HoG, area granulometry (<100ms), dense SIFT, followed by Gabor filters (<1s). This suggests that it is quite feasible to think of a joint feature scheme for the classification.

5. Conclusions

In this paper, we studied the problem of recognizing African clawed frogs' identities from their skin patterns. We proposed a novel, and as far as our literature search is accurate, the only frog body localization and skin pattern region extraction algorithm reported in the open literature. We tested and reported its accuracy and stability, which showed that it is quite successful in compensating scale and rotation changes. Moreover, our qualitative observations showed that it extracts majorly overlapping windows even in the case of intensive modifications.

The pattern of the frog skin is a unique identifier which is used for manual identification by the experts. Our aim was to find the best descriptor to represent the skin patterns. In addition we tried to seek an answer the question whether we needed an abstract representation at all. Hence, we have compared five candidate features (Gabor filters, area granulometry, HoG, dense SIFT, and raw pixel values). The detailed experiments using a nearest neighbor classifier showed that the raw pixel values of the extracted pattern window (in coarse resolutions) was the most effective feature, thanks to the stability of the pattern window extractor.

However, it performed the worst against the intensity modification test. In addition, it is not as robust as the SIFT against affine modifications, which can certainly result in poorer recognition performance during a real use scenario.

It may be possible to create a joint feature using the raw pixel values and one of the other features, for which the HoG, and SIFT features seem to be good candidates. An important future work is to test our algorithm and features on a large database of African clawed frogs which include pictures taken at different times with varying conditions.

The focus of this study is on the automatic recognition of African clawed frogs in a laboratory environment. However, we believe that this work can be extended to the natural habitats by an additional pre- frog detection step which must locate frogs in uncontrolled environments. The generalized visual object detectors can be used for this purpose. Subsequently, our technique for locating the skin pattern region window can be used. Since it works for frog bodies which are very flexible, it can be adapted to other skin-textured or fur-textured species if adapted to the respective body shapes and geometries. Moreover, the comparison of various features presented here must be relevant to the different skin-fur textures.

Acknowledgments

The authors would like to gratefully thank Dr. Kathleen E Mathers, Mrs. Clare Brazill-Adams, and Mrs. Ola Puchalska-Osorio, from the Biological Services, MRC-National Institute for Medical Research, London, UK for useful discussions and the set of frog images used in this study.

References

Cannavo, F., Giuseppe, N., Kale, I., Tek, F.B., 2012. Texture recognition for frog identification. In Proceedings of the 1st ACM international workshop on multimedia analysis for ecological data (MAED '12). ACM, New York, NY, USA, pp. 25-30.

Chen, C.H., Pau, L.F., Wang, P.S.P., 1998. The hand-book of pattern recognition and computer vision, (2nd Edition), World Scientific Publishing Co, pp. 207-248.

Dalal, N., Triggs, B., Schmid, C., 2006. Human detection using oriented histograms of flow and appearance, In European Conference on Computer Vision, Lecture Notes in Computer Science, pp. 428-441.

Dasarathy, B.V., 1991. Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer Society ed.

Feichtinger, H. G. 1998. Gabor analysis and algorithms: Theory and applications. Birkhauser Boston.

Gonzales, R.C., Woods, R.E., 1992. Digital image processing, 3rd ed., Addison-Wesley.

Hu, M. K., 1962. Visual pattern recognition by moment invariants, IRE Trans. Info. Theory, 8, pp.179-187.

Huang, P.W., Dai, S.K., 2003. Image retrieval by texture similarity, Pattern Recognition, 36, pp. 665-679.

Hubrecht, R., Kirkwood, J., E.B., 2010. The UFAW handbook on the care and management of laboratory and other research animals, John Wiley & Sons.

Kaplan, L.M., 1999. Extended fractal analysis for texture classification and segmentation, Image Processing, IEEE Transactions on, 8, pp. 1572-1585.

Kay, K.B., Peng, H.B.,E.B., 1991. Xenopus Laevis: Practical uses in Cell and Molecular Biology. Methods in Cell Biology, 36, Academic Press, California.

Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P., 1983. Optimization by simulated annealing. Science, 220, pp. 671–680.

Lianping, C., Guojun, L., Dengsheng, Z., 2004. Effects of different Gabor filter parameters on image retrieval by texture, Proc. of the 10th International Multimedia Modeling Conference (MMM'04), pp. 273-278.

Liu, C., Yuen, J., Torralba, A., 2011. SIFT flow: dense Correspondence across scenes and its applications, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 33, pp. 978-994.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision, 60, pp. 91-110.

Ma, L., Tan, T., Wang, Y., Zhang, D., 2003. Personal identification based on iris texture analysis, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 25, pp. 1519-1533.

Manjunath, B.S., Ma, W.Y., 1996. Texture features for browsing and retrieval of image data, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 18, pp. 837-842.

Matheron, G., 1975. Random sets and integral geometry, Wiley, New York.

Matthews, S.; Brand, K., 2004. Africa invaded: the growing danger of invasive alien species. The Global Invasive Species Programme (GISP) Secretariat, Cape Town, South Africa.

Measey, G.J., Rödder, D., Green, S.L., Kobayashi, R., Lillo, F., Lobos, G., Rebello R, Thirion, J.M., 2012. Ongoing invasions of the African clawed frog, Xenopus laevis: a global review. Biological Invasions, 14, pp. 2255-2270.

Arnold, M., Wilkinson, M.H.F., 2001. Fast computation of morphological area pattern spectra, In Proc. Int. Conf. Image, 3, pp. 668-671.

Mikolajczyk, K., Schmid, C., 2005. A performance evaluation of local descriptors, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 27, pp. 1615-1630.

Mirmehdi, M., Xie, X., Suri, J.S., 2009. Handbook of texture analysis, Imperial College Press.

Rao, K.N.R.M., Dempster, A.G., 2001. Area-granulometry: an improved estimator of size distribution of image objects, Electronics Letters , 37, pp. 950-951.

Otsu, N., 1979. A threshold selection method from gray-level histograms, Systems, Man and Cybernetics, IEEE Transactions. on, 9, pp. 62-66.

Soille, P., 1999. Morphological image analysis: principles and applications, Springer-Verlag, pp. 173-174.

Tamura, H., Mori, S., Yamawaki, T., 1976. Texture features corresponding to visual perception, Systems, Man and Cybernetics, IEEE Transactions. on, 6, pp. 460-473.

Tek, F.B., Dempster, A.G., Kale, I., 2010. Parasite detection and identification for automated thin blood film malaria diagnosis. Computer Vision and Image Understanding 114, pp. 21-32.

Tek, F.B., Dempster, A.G., and Kale, I. ,2009. Computer vision for microscopy diagnosis of malaria. Malaria Journal,8, pp. 1-14.

Vedaldi A. and B. Fulkerson, 2008. VLFeat: An open and portable library of computer vision algorithms. URL = http://www.vlfeat.org/

Wang, Z.-Z., Yong, J.H., 2008. Texture analysis and classification with linear regression model based on wavelet transform, Image Processing, IEEE Transactions on, 17, pp. 1421-1430.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli E.P. , 2004. Image quality assessment: from error visibility to structural similarity, Image Processing, IEEE Transactions on, 13, pp. 600-612.

FIGURE & TABLE CAPTIONS

Figure 1 Example images from the X. Laevis image set.

Figure 2 Pattern window (back-square) localization process: (a)-(b), the original RGB image is converted to grayscale; (b)-(c), the grayscale image is thresholded; (c)-(d), the morphological opening operation with a disk-shaped structuring element which has a radius equal to the frog's pseudo-radius (calculated from the area) provides a symmetrical body by removing the arms and legs; (d)-(e), the barycenter and the orientation of the body are calculated, the line crossing the barycenter of the body which has the angle obtained from frog's orientation is considered as the body line of the frog; (e)-(f), simulated annealing process to improve the head localization by maximizing the head-tail distance; (f)-(g), adjust the head and tail coordinates by searching for the best isosceles triangle for the head; (g)-(h), extract a square from the back by locating the two flanks of the body and the other two corners on the body line as corners of the square; (h)-(i), resize and normalize extracted square to be the pattern window.

Figure 3 Extracted pattern windows after different modifications applied to the original image: (a) Original, (b) rotation angle of 65^{o}, (c) scaling by a factor of 0.25, (d) affine transform with coefficient (0.25), (e) contrast stretch, (f) blurring iteration 5.

Figure 4 MSE error of the pattern window extractor against different modifications: (a) with respect to the different rotation angles, (b) with respect to the different affine transformation coefficients, (c) with respect to the different scale coefficients, (d) with respect to the increasing blur iterations, Gaussian noise variances and illumination change by contrast stretching.

Figure 5. The rotation test performance of each feature with different parameters. Precision and recall rates were averaged for 60 frogs in the training test. The individual points are indexed according to their parameter settings; and markers differ with respect to the different features. By the increasing index value, the window resolution decrease for raw features ((200x200, 100x100, 50x50, 25x25, 12x12, 6x6, 3x3); for Hog and dense SIFT the cell size decrease (100,50,20,10); for granulometry the number of bins decrease (30,20,12,7,6,5); and for Gabor filters the cell size parameter (20,50,100) increase.

Figure 6. The scale test performance of each feature with different parameters. Precision and recall rates were averaged for 60 frogs in the training test. Refer to Figure 5 caption for the explanation of the parameters corresponding to the different marker indexes.

Figure 7. The affine distortion test performance of each feature with different parameters. Precision and recall rates were averaged for 60 frogs in the training test. Refer to Figure 5 caption for the explanation of the parameters corresponding to the different marker indexes.

Figure 8. The blurring test performance of each feature with different parameters. Precision and recall rates were averaged for 60 frogs in the training test. Refer to Figure 5 caption for the explanation of the parameters corresponding to the different marker indexes.

Figure 9. The noise test performance of each feature with different parameters. Precision and recall rates were averaged for 60 frogs in the training test. Refer to Figure 5 caption for the explanation of the parameters corresponding to the different marker indexes.

Figure 10. The intensity modification test performance of each feature with different parameters. Precision and recall rates were averaged for 60 frogs in the training test. Refer to Figure 5 caption for the explanation of the parameters corresponding to the different marker indexes.

Table 1 The best average precision/recall rates for each feature/modification pair. The best (green) and worst (yellow) performing features were marked for every modification. Note that the parameters of the best performing feature can differ among the rows.