Texture Cue Based Tracking System Using Wavelet Transform and a Fuzzy Grammar

Manuel J. Ferreira[1], Cristina P. Santos[2], Joao Monteiro[3], *Members, IEEE*

***Abstract*— This paper addresses a system for fast object tracking based on texture cues, by using the wavelet transform and a fuzzy grammar classifier. The method is based on wavelet type features. The feature vector consists of 6 characteristics extracted from the wavelet detail images for each colour component. These texture characteristics automatically generate a fuzzy rule using a fuzzy inference classifier based on a fuzzy grammar. A learning phase is required for each texture but only uses one sample. This 2D tracking system of textured objects in image sequences is demonstrated on a robotic application using the platform developed by Sony – AIBO robot. The application ensures a real time tracking approach and can be parameterized in order to be flexible in face of different types of textures.**

I. Introduction

The most common approaches for object tracking are based mainly on the detection of one of the following cues: edges, colour and texture [1,2,3,4]. Edges techniques are generally based on the analysis of the gradients intensity. For applications with highly texture environments or objects in which light conditions are not stable or its interaction with the objects produces shadows, these techniques are not suitable. Texture segmentation techniques are recently been applied to object tracking, especially as a complement to a multi-cue tracker [1,5]. However, the most common texture segmentation techniques tend to be computationally intensive, and use classification methods that require a time expensive off-line learning phase. For these reasons, such approach isn't consistently used for tracking purposes.

In this paper, we propose a tracking system that uses a texture segmentation approach based on the wavelet transform and on a fuzzy grammar as classifier. This system reduces the expensive time consuming both in the processing and in the off-line learning phase. Specifically, features are extracted from detail images of wavelet transforms. This technique is applied to each R, G and B colour components of the image, performing a colour texture analysis.

This work was supported in part by the industrial protocol among University of Minho and Enermeter – Sistemas de Medição, Lda (www.enermeter.pt). Authors work at the Department of Industrial Electronics, University of Minho, Campus de Azurem, Guimarães, Portugal. (phone: +351-253-510190; fax: +351-253-510189; e-mails: mjf@dei.uminho.pt; cristina@dei.uminho.pt; joao.monteiro@dei.uminho.pt).

The developed approach is divided onto two phases: the *learning phase* and the *tracking phase*. In the *learning phase*, the texture for tracking is manually selected specifying a *Region of Interest* (ROI) for this image. A feature vector is extracted and a fuzzy rule, characterizing the texture, is generated by the fuzzy grammar. In the *tracking phase*, a feature vector is extracted from the overall input image. The fuzzy rule generated during the *learning phase* is evaluated, and a final classification is done. The size of the tracking window within each video frame is optimized in each iteration, thus reducing computation times.

As a case study, this tracking system was implemented and tested in an AIBO robot platform. This platform is being used as a companion robot, and one of its tasks is to maintain surveillance over dependent and aged people in a clutter environment. In the long run, the main purpose is to keep several moving persons under tracking. The developed system will identify a specific cloth for each individual, such that it is possible to independently track each texture present in the image. Herein, a much more simplified experiment is described, in which the AIBO dog successfully detects a texture moved to a different position.

This paper is structured as follows. Section II starts by briefly describing the tracking system architecture. Several experiments are described and results discussed in section III. Finally, the article ends with some conclusions and future works presentation.

II. The Tracking System Architecture

Fig. 1a shows the architecture of the tracking system, in which two paths were specified: one for the Learning phase (P1) and another for the Tracking phase (P2).

The first two modules are identical for both P1 and P2 and deal with object analysis.

The *Feature Extraction* module extracts the feature vector that best describes an object texture.

P1 has a *Fuzzy Grammar* module which generates the fuzzy rule that describes the texture using the extracted feature vector.

In the *Tracking phase*, the extracted feature vector is submitted to a *Parsing Procedure module* for the fuzzy grammar. In this module, the vector is submitted to the fuzzy rule specifically generated in the *Learning phase* for this texture. The output of the parsing procedure is a value in the interval [0,1] reflecting the grade of membership of the texture.

Finally, the *Classification* module uses the output of the parsing and verifies if the rule has a response higher than a pre-defined threshold. The result is a binary image where the blob corresponds to the presence of the texture under tracking.

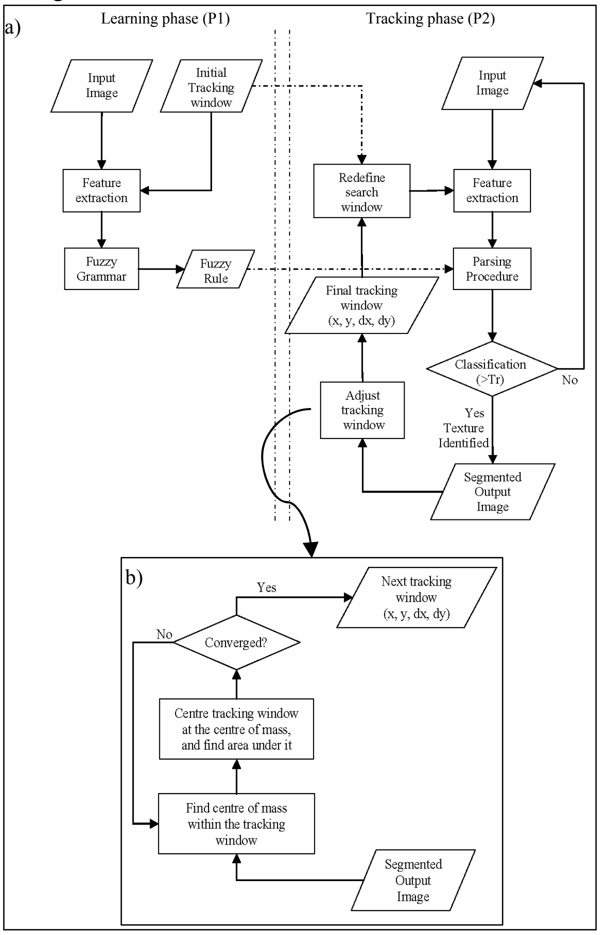

Fig. 1. Tracking system. a) Architecture of the processing system. b) Adjust tracking window module.

In the *tracking phase*, in order to reduce the processing time of each video frame while tracking the texture in the next video frame, the tracking window size is adjusted and an enlarged version of it is specified as the new ROI for the next image. This is ensured by the Adjust Tracking Window module (fig. 1b) which implements a procedure similar to the one employed in the CAMSHIFT algorithm [4], as follows. Firstly, calculate the blob's centre of mass. The tracking window is repositioned at this centre, and the area of the blob under the tracking window is quantified. This procedure is repeated until the area converges. The current size and position of the tracked texture are used slightly enlarged in the next video image, considering the tracked texture does not change faster than this enlargement.

A. Feature Extraction Module

Besides the classical approaches for texture segmentation, such as statistical, structural and spectral approaches, new ones, based on wavelet transform and Gabor filters [7,8,9], have recently deserved special attention. These methods process visual information in a multi-scale manner, similarly to the human vision system.

In [10], the author realises that the best compromise between processing time and classification rate is achieved with the wavelet transform among comparisons using Fractals, Feature Based Interaction Maps and Gabor filters for different types of texture (textile, cork, leather and paper). These aspects lead to the choice of the wavelet transform for texture segmentation in the present work. Further, these previous studies provide for a strong comparison of the wavelet transform against texture segmentation state-of-the-art, indicating that the approach taken has these advantages over many other possibilities.

To perform the wavelet transform, in the context of image processing, it is necessary to employ a two-dimensional discrete wavelet transform (DWT) [11]. The wavelet transform in this domain introduces the concept of variable time window with frequency. As illustrated in fig. 2a, an approximation coefficient, at level m+1 (V_{m+1}xV_{m+1}), is decomposed in four components: the approximation coefficient at level m (V_mxV_m) and the details at level m in three orientation coefficients: horizontal (V_mxW_m), vertical (W_mxV_m) and diagonal (W_mxW_m). These last three components are the detail images used to construct the feature vectors. Fig. 2b shows a textured image and its correspondent wavelet transform for the G component.

In this work, the applied wavelet functions are the two FIR filters presented in fig. 3. There are several types of wavelets functions that can be used in texture analysis. However, [8,10] supports that the type of wavelet function doesn't produce relevant changes in the analysis.

This application uses three levels for the wavelet transform, which results in a total of 9 detail images. The feature vector consists of features extracted from the detail images at each decomposition level and for each colour component, as follows: Mean (M), Standard Deviation (SD), Contrast Between adjacent - Next Neighbour - pixels in Vertical (CBNNV) and Horizontal (CBNNH) directions and Contrast Between alternated – Alternated Neighbour - pixels in Vertical (CBANV) and Horizontal (CBANH) directions ((1) to (6)).

Since the classifier is based on a fuzzy inference system, it implies that the magnitude of each element of the feature vector must be in the interval [0,1] and thus a normalization of each feature element is required (7).

The feature vector, FV, to be presented to the fuzzy grammar module consists of 6 features for each detail image of each colour component (6x9x3 = 162 features): $FV = [\mu_{MijR}, \mu_{DPijR}, \mu_{CVSVijR}, \mu_{CVSHijR}, \mu_{CVAVijR}, \mu_{CVAHijR}, \mu_{MijG},$

μ_{DPijG}, $\mu_{CVSVijG}$, $\mu_{CVSHijG}$, $\mu_{CVAVijG}$, $\mu_{CVAHijG}$, μ_{MijB}, μ_{DPijB}, $\mu_{CVSVijB}$, $\mu_{CVSHijB}$, $\mu_{CVAVijB}$, $\mu_{CVAHijB}$], with i= 0, 1, 2; j= 0, 1, 2.

$$M = \frac{1}{N}\sum_{i=0}^{N-1} I(i) \tag{1}$$

$$SD = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1}(I(i) - M)^2} \tag{2}$$

$$CBNNH = \frac{1}{N-1}\sum_{l=0}^{Nl-1}\sum_{c=0}^{Nc-1}\left|I(l \times Nc + c + 1) - I(l \times Nc + c)\right| \tag{3}$$

$$CBNNV = \frac{1}{N-1}\sum_{l=0}^{Nl-1}\sum_{c=0}^{Nc-1}\left|I((l+1) \times Nc + c) - I(l \times Nc + c)\right| \tag{4}$$

$$CBANH = \frac{1}{N-1}\sum_{l=0}^{Nl-1}\sum_{c=0}^{Nc-1}\left|I(l \times Nc + c + 2) - I(l \times Nc + c)\right| \tag{5}$$

$$CBANV = \frac{1}{N-1}\sum_{l=0}^{Nl-1}\sum_{c=0}^{Nc-1}\left|I((l+2) \times Nc + c) - I(l \times Nc + c)\right| \tag{6}$$

$$\mu_F = F / 255; \quad F \in \{M, SD, CBNNV, CBNNH, CBANV, CBANH\} \tag{7}$$

where I is the image, N is the number of pixels in the image, Nc and Nl are the number of columns and lines in the image, respectively.

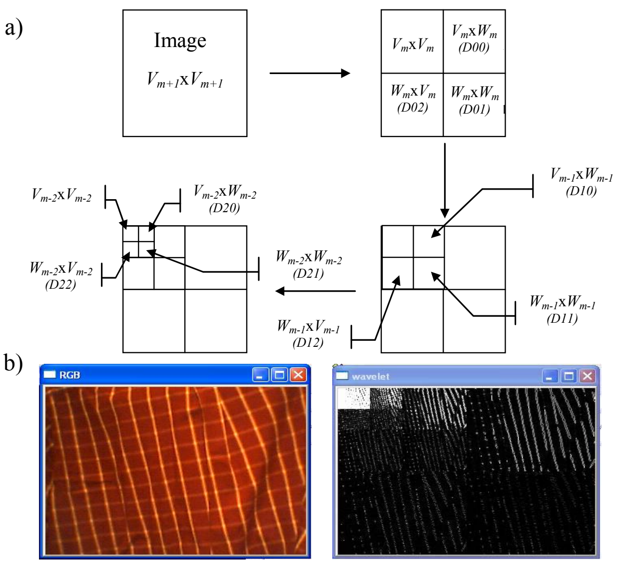

Fig. 2. Wavelet decomposition of an image. a) Final image obtained by the sub-spaces $V_{-i}xV_{-i}$, $V_{-i}xW_{-i}$, $W_{-i}xV_{-i}$, $W_{-i}xW_{-i}$ with i=1,2,3. b) Image of a textured object (left) and the corresponding wavelet transform (right) with three levels of decomposition.

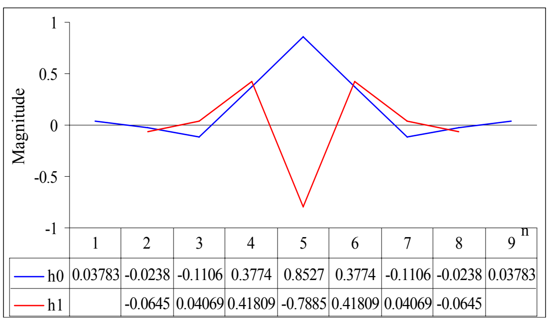

Fig. 3. Wavelets filters $h_0[n]$ and $h_1[n]$.

B. Fuzzy Grammar Module

This module only exists in the *learning phase* (P1). After the extraction of the feature vector that characterizes a texture, it is necessary to classify it according to its attributes.

Regarding the classifiers and recognizers, there are different approaches. The most common solutions use recognizers based on the calculus of metrics like Euclidean, Minkowsky e Mahalanobis distance measures. However, these recognizers, as well as the ones based on neural, fuzzy logic and neurofuzzy networks, demand a great amount of samples from the population to perform learning [12,13]. This application has to deal with a high diversity of texture objects. To fulfil this constraint, the *learning phase* must be done with a unique sample of each type of texture.

In this work, a fuzzy system modelling approach was developed in which a fuzzy inference system identifies the fuzzy rules representing relationships among the features extracted from the wavelet detail images. There are several approaches that generate these fuzzy rules. The most often applied are based on statistics, neural networks and genetic algorithms [12,14,15]. However, these methods poorly satisfy the needs of the present application, specifically the possibility to learn using only a characteristic vector. Therefore, a fuzzy grammar approach [15,16] was applied. Fuzzy grammar is a pattern classification syntactic model used to represent the structural relations of patterns and describes the syntax of the fuzzy languages that generate the fuzzy rules. This inference system is capable of generating a fuzzy rule using only one sample of a pattern.

Herein, a brief review of some basic concepts of fuzzy grammar is presented. Fuzzy grammar GF is a quintuple $GF=(V_N, V_T, P, S_0, \mu)$, in which V_N and V_T are finite disjoint sets of non-terminal and terminal vocabulary respectively, such that $V=V_N \cup V_T$ is the total vocabulary of the grammar. P is a finite set of production rules of the type $\alpha \rightarrow \beta$, with $\alpha \in V_N$ and β is a member of the set V^* of all strings (including the null string ε). $S_0 \in V_N$ is the starting symbol. μ is the mapping of $P \rightarrow [0,1]$, such that $\mu(p)$ denotes the possibility of the current language sentence $p \in P$.

The syntax of the developed language $L(GF)$ is depicted in Fig. 4 and includes four different steps:

1) The codification of the features to primitives (Table I).

TABLE I

Codification of features to primitives, with i=0,1,2; j=0,1,2; f=R,G,B

Feature	Primitive
μ_{Mijf}	FWDijMFf
μ_{SDijf}	FWDijSDFf
$\mu_{CBNNVijf}$	FWDijCBNNVFf
$\mu_{CBNNHijf}$	FWDijCBNNHFf
$\mu_{CBANVijf}$	FWDijCBANVFf
$\mu_{CBANHijf}$	FWDijCBANHFf

2) The definition of linguistic terms *HistVar:c*, as follows:

$$HistVar:c \ = \Pi(x, 0.2, c \times 0.1) \qquad c = 0 \ldots 10. \qquad (8)$$

The parameter *c* is chosen such that the eleven membership functions cover the all universe of discourse, *X*, and have disjointed maximums.

3) The definition of fuzzy modifiers (*FM*): "More than", "Less than" and "Between". The *FM* "More than" *LT* is defined by

$$\mu_{MT}\langle LT\rangle = \begin{cases} 1 & x \geq L \\ S(x, L-lb, L-lb/2, L & x < L \end{cases} \qquad (9)$$

where *L* is a threshold value and *lb* is the bandwidth value of the *S* membership function [12]. The *FM* "Less than" *LT* is given by

$$\mu_{LT}\langle LT\rangle = \begin{cases} 1 & x \leq L \\ 1 - S(x, L, L+lb/2, L+lb & x > L \end{cases} \qquad (10)$$

The *FM* "Between" LT_1 and LT_2, is given by

$$\mu_B < TL_1 >< TL_2 > \ = \begin{cases} 1 - S(x, w_1, w_1 + lb/2, w_1 + lb) & x > w_1 \\ 1 & w_2 \leq x \leq w_1 \\ S(x, w_2 - lb, w_2 - lb/2, w_2) & x < w_2 \end{cases} \qquad (11)$$

where w_1 and w_2 are threshold values [12].

4) The definition of fuzzy operators (*FO*) which define the relations between the linguistic terms and primitives. The following *FO* were defined:

a) &, representing the AND of two primitives. It is given by the Yager intersection.

b) >, representing "More than" *LT* and is given by μ_{MT}<*LT*>.

c) <, means "Less than" *LT* and is given by the function μ_{LT}<*LT*>.

d) ||, describes "Between two" *LT* and is given by μ_B<LT_1><LT_2>.

e) #, means a "Separator between a" *primitive* and a *LT*.

f) ( ), imposes a hierarchy in the rule.

Consider texture depicted in Fig. 2b. Fig. 5 illustrates the values of the eleven membership function ∏ for the primitive FWD00MFR (Fig. 5a) and primitive FWD22CBNNHFR (Fig. 5b). Primitive FWD00MFR has non-zero degrees of membership for *LT* HistVar:0, *LT* HistVar:1 and *LT* HistVar:2. The highest fuzzy value is obtained using *LT* HistVar:0. Thus, HistVar:0#FWD00MFR is part of the fuzzy rule which characterizes this texture. Similarly, HistVar:1#FWD22CBNNHFR is part of the fuzzy rule which characterizes this texture.

The final rule will characterize the texture but herein we present part of this rule for detail image D00 and component R, created by the fuzzy grammar:

HistVar:0#FWD00MFR&HistVar:0#FWD00SDFR&HistVar:0#FWD00CBNNHFR&HistVar:0#FWD00CBNNVFR&HistVar:0#FWD00CBANHFR&HistVar:0#FWD00CBANVFR.

Language -> $L(G_F) = \{x, \mu(x) | x \in V^*_T, S \Rightarrow x\}$

$G_F = (V_N, V_T, P, S_0, \{\mu\})$

V_N={S_0, Name, ElementSet ,Primitive, TermSet, Element, Term}

V_T={FWD00MFR, …, HistVar:0,..HistVar:10,+,..,#}

$S_0 \rightarrow$ 'Rule' RuleName' 'ElementSet

ElementSet	→	ElementSet '&' ElementSet '('ElementSet {'|' ElementSet }')' '('ElementSet { '+' ElementSet } ')' Element λ
Element	→	TermSet '#' Primitive Primitive
TermSet	→	'>' Term '<' Term '('Term '| |' Term')'
RuleName	→	Obj1 other
Primitive	→	FWD00MFR, ... other
Term	→	'HistVar:0' ... 'HistVar:10'

Fig. 4. Syntax of the developed fuzzy language L(GF).

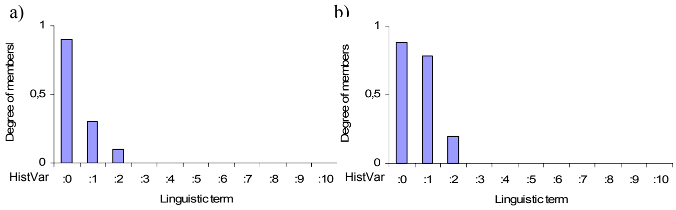

Fig. 5. Membership degree of Linguistic Terms. a) Primitive FWD00MFR. b) Primitive FWD22CBNNHFR.

If more than one linguistic term gives fuzzy values superior to 0.75; fuzzy modifiers like "More than", "Less than" and "Between", are applied to combine the obtained results.

III. Results

During the *learning phase*, for initial tracking window size specification, it is necessary to consider the type of texture, specifically periodical or random aspects. Therefore, the following was settled for the *learning phase*: 1) initially the user chooses the initial tracking window; 2) this tracking area is divided in non-overlapping windows (NOW), whose size is set by the operator (Fig. 6a); 3) for each NOW the wavelet transform is applied, and the 6x9 features for each colour component are extracted. Each element of the final feature vector is the mean value of each feature for each NOW; 4) a fuzzy rule is created with this feature vector. In the *tracking phase*, the search window (that initially corresponds to the all image) is also divided in NOW with the same size as the ones of the *learning phase*, but now overlapped (Fig. 6b) by (*dx*,*dy*), where *dx, dy* are the displacements relatively to the previous position. This procedure ensures different grades of performance.

Firstly, the feasibility and efficiency of the texture segmentation approach have been studied by performing a set of experiments using 30 different types of textures (8 of them are presented in fig. 7a). Each image has a resolution of 640x480 pixels and was acquired using a PAL compatible CCD Camera and a Matrox acquisition board. The approach was applied with a NOW of 50x50 pixels.

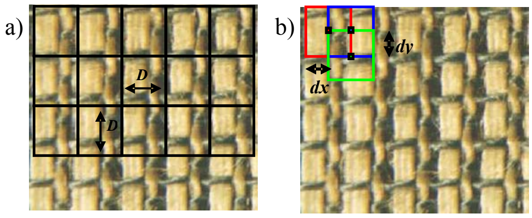

Fig. 6. Decomposition process of the tracking window for the application of the DWT. a) Learning phase. b) Tracking phase: Red window *dx*=0,*dy*=0; Blue window *dx*=D/2,*dy*=0; Green window *dx*=D/2,*dy*=D/2.

Fig. 7b shows the response of each texture rule (gray bars) as well as the overall response of the rule that characterizes the other textures (red bars).

A specific advantage of the developed approach is that, when a texture is presented to the inference system it gives a response with high value (higher than 0.85) for the rule that describes this texture. The rules corresponding to the other textures give low value responses (less than 0.3). This means that the system creates disjoints rules and assures a good classification. Another advantage is that only one sample can be used during the *learning phase*. The above results show that the developed approach can be applied both to different types of textures and when the environment is cluttered with several types of textures.

In a second set of experiments, the developed approach was integrated with the AIBO platform. This platform uses wi-fi wireless connectivity and a vision system with an image size of 412x320 pixels and acquisition step, through wi-fi, of 29ms. Wavelet decomposition was applied with 3 levels and a NOW of 45x34 pixels. These specifications yield a processing time of 40ms. The tracking system was developed in C++, and a DLL was created to encapsulate the parsing procedure which was developed with the compilers yacc and lex [6].

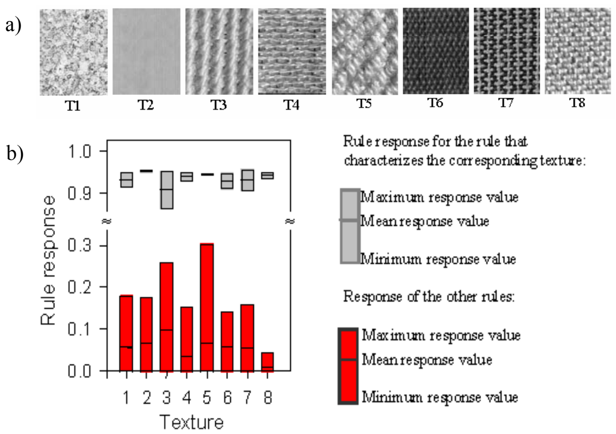

Fig. 7. a) Some examples of the textures used to texture segmentation procedure evaluation [10]. b) Rule response for the images of fig. 7a.

The image of fig. 8a was acquired by the camera of the AIBO robot, and image 8b results from the application of the wavelet transform for the G component. The first step in the *learning phase*, is to select the initial tracking window, which is done by selecting the texture to track (blue box in fig. 8a). The application evaluates the feature vector for the wavelet decomposition under that window and creates the rule for this particular texture. During the *tracking phase*, the developed approach searches for this texture in the image. Firstly, it searches over the all image and, in the next video images, an optimized search window is found to search for this texture. Fig. 8c shows a video frame when the texture was moved to a different position. The segmentation result is depicted in Fig. 8d.

The application was tested with different illumination conditions and the results have shown that a drift in the illumination doesn't affect the efficiency of the tracking procedure.

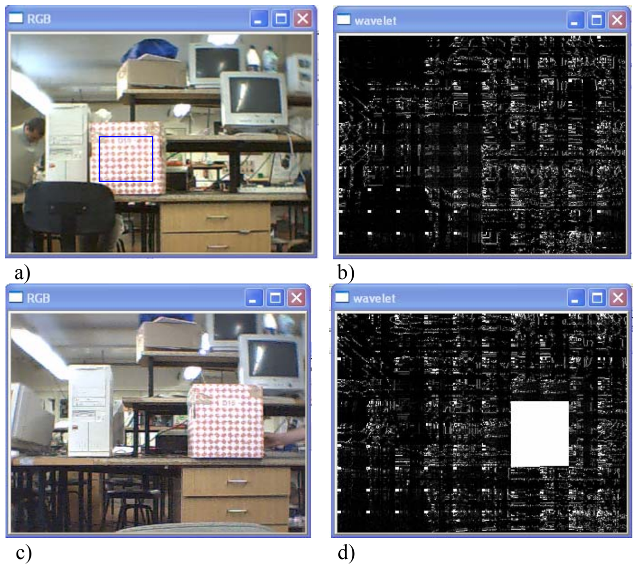

Fig. 8. Experimental results with the AIBO platform. Learning phase: a) Image of the object to track. b) Wavelet transform. Tracking phase: c) Image of the object to track. d) Segmentation result.

Consider Fig. 9a where an original image and its respective darker (a luminance decrease of 20%) and brighter (a luminance increase of 15%) versions are depicted. Fig. 9b presents the degree of membership for 5 linguistics terms of the primitive FWD00MFG for images of Fig. 9a. Even for the most sensible primitive (FWD00MFG), the fuzzy rule remains unchanged for a drift in illumination.

IV. CONCLUSION

In this paper, we have described the use of a combination of wavelet-based texture cues and fuzzy logic to create a fast texture tracking system. Specifically, 6 features are extracted from detail images of wavelet transforms. This technique is applied to each R, G and B colour components of the image, performing, in such a way, a colour texture analysis. A fuzzy grammar, specifically developed for this application, uses this feature vector to generate a fuzzy rule that characterizes the texture. The developed approach is divided in two phases: a *learning phase* and a *tracking phase*. During the *learning phase* the fuzzy rule is created. In the *tracking phase*, a tracking for the desired texture is performed, and a final classification is done. Further, a procedure similar to the CAMSHIFT algorithm was implemented in order to reduce the processing time of each video frame while tracking the texture.

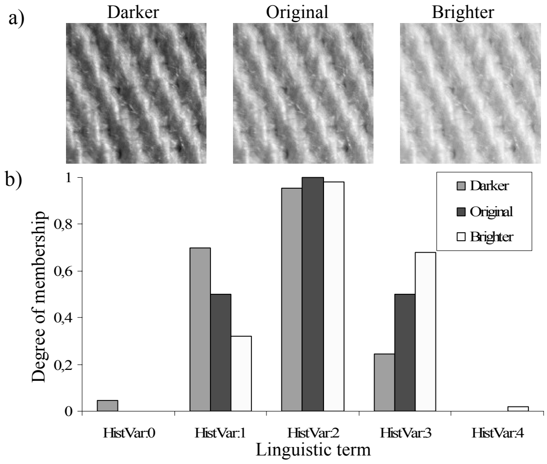

Fig. 9. Testing results with illumination drift. a) Original image and darker and brighter versions. b) Linguistic terms.

A crucial factor for tracking textured objects is the need to reduce expensive time consuming both in the processing and in the off-line *learning phases*. The combination of wavelet transform and fuzzy grammar revealed to be a suitable approach to achieve fast texture tracking. Another important factor for tracking textured objects is to be able to deal with clutter environments with diversified textures, especially if light conditions are not stable. Because the developed approach creates disjoint rules thus assuring good classifications, the system is capable to work in clutter environments even when different type of textures were simultaneously present in the same image. An advantage of this approach, when compared to other approaches, is that the *learning phase* is done with a unique sample of each type of texture.

The developed 2-D tracking system of textured objects in image sequences was integrated on a robotic application using the platform developed by Sony – AIBO robot. A fast texture tracking system was achieved with processing times in the order of the 40 ms in the AIBO platform.

A long term goal of this work is to track individual humans. This experiment is being currently performed in a simplistic form. In future work, a more stronger comparison against the texture tracking state-of-the-art systems will be done, in order to provide for an evaluation of disadvantages/advantages of the developed approach.

An improvement in the global tracking performance in which concerns colour, may be achieved with the integration of specific colour features based on the hue component histogram. The use of this fuzzy grammar classifier allows this integration in a simplified and clear manner. Other possible improvements include the creation of a database during the *learning phase* which would enable to automatically detect a texture that had already been presented to the system during the *learning phase*.

REFERENCES

[1] Pressigout, M., E. Marchand (2005), Real-time planar structure tracking for visual servoing: a contour and texture approach. In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'05, August, 2005.
[2] Taylor, G., L. Kleeman (2003), "Fusion of multimodal visual cues for model-based object tracking", 2003 Australasian Conference on Robotics and Automation, Brisbane, 1-3 December 2003.
[3] Shahrokni, A., T. Drummond, and P. Fua (2004), Proc. Eur. Conf. on COmputer Vision, "Texture Boundary Detection for Real-Time Tracking", Prague, Czech Republic, May 2004.
[4] Bradski, G. R. (1998), Computer vision face tracking as a component of a perceptual user interface. In Workshop on Applications of Computer Vision, pages 214–219, Princeton, NJ, Oct. 1998.
[5] Giebel, J., D. M. Gavrila and C. Schnörr (2004), A Bayesian Framework for Multi-Cue 3D Object Tracking, Proc. of the European Conference on Computer Vision, Prague, Czech Republic, 2004.
[6] Bumble-Bee (1999), Parser Generator Manual [online], Bumble-Bee Software, Available: http://www.bumblebeesoftware.com.
[7] Teuner, A., Olaf Pichler and Bedrich J. Hosticka (1995), "Unsupervised Texture segmentation of images using tuned matched gabor filters", IEEE transactions on image processing, vol. 4, n. 6, pp. 863-870.
[8] Livens, S. (1998), "Image analysis for material characterization", PhD Thesis, vol. 1, Universitaire Instelling Antwerpen, Antwerpen, Belgium.
[9] Wouwer, G. V. (1998), "Wavelets for multiscale texture analysis", PhD Thesis, Universitaire Instelling Antwerpen, Antwerpen, Belgium.
[10] Ferreira, M.J. (2004), Desenvolvimento de um protótipo para a identificação, classificação e quantificação de defeitos, aplicável em ambiente industrial, Unpublished Thesis (PhD), Universidade do Minho, Braga.
[11] Benedetto, J J., Michael W. Frazier (1994), Wavelets mathematics and applications: CRC Press, 1994.
[12] Bezdek, J.C. & Pal, S.K. (ed) (1992), Fuzzy Models for pattern recognition, IEEE Press. New York.
[13] Micheli-Tzanakou, E (2000), Supervised and Unsupervised Pattern recognition, feature extraction and computational, Industrial Electronics Series, J. David Irwin, Ed., CRC Press, Florida.
[14] Looney, C.G. (2002), "Pattern Recognition", in Hyngsuck Cho, (Ed.), Opto-Mechatronic Systems Handbook: Technical Applications - Handbook Series for Mechanical Enginnering, Vol 10, CRC Press.
[15] Ivancic, F. & Malaviya, A. (1998), "An Automatic Rule Base Generation Method for Fuzzy Pattern Recognition with Multi-phased Clustering", in IEEE Conference of Knowledge Engineering System, Proceedings of KES'98, IEEE Press, Adelaide, pp 66-75.
[16] C. P. Santos, Manuel João Ferreira, Computer Vision and Fuzzy Rules applied to an Industrial Desktop Robot, Assembly and Automation Journal, 2006