Multi-scale cortical keypoints for realtime hand tracking and gesture recognition

M. Farrajota[1], M. Saleiro[1], K. Terzic[1], J.M.F. Rodrigues[1] and J.M.H. du Buf[1]

Abstract— **Human-robot interaction is an interdisciplinary research area which aims at integrating human factors, cognitive psychology and robot technology. The ultimate goal is the development of social robots. These robots are expected to work in human environments, and to understand behavior of persons through gestures and body movements. In this paper we present a biological and realtime framework for detecting and tracking hands. This framework is based on keypoints extracted from cortical V1 end-stopped cells. Detected keypoints and the cells' responses are used to classify the junction type. By combining annotated keypoints in a hierarchical, multi-scale tree structure, moving and deformable hands can be segregated, their movements can be obtained, and they can be tracked over time. By using hand templates with keypoints at only two scales, a hand's gestures can be recognized.**

I. INTRODUCTION

Automatic analysis of humans and their actions has received increasingly more attention in the last decade. One of the areas of interest is recognition of human gestures, as these are frequently used as an intuitive and convenient way of communication in our daily life. Recognition of hand gestures can be widely applied in human-computer interfaces and interaction, games, robot control, augmented reality, etc.

In computer vision there are numerous approaches for hand detection, tracking and gesture recognition, although to the best of our knowledge none is really biologically inspired. Kim et al. [7] presented a method for hand tracking and motion detection based on a sequence of stereo color frames. Bandera et al. [1] presented an approach to recognize gestures which are composed of tracked trajectories of different body parts, where each individual trajectory is described by a set of keypoints. Gestures are characterized through global properties of the trajectories which are involved. Suk et al. [17] explored a method for recognizing hand gestures in a continuous video stream based on a dynamic Bayesian network.

Holte et al. [5] presented an approach to invariant gesture recognition using 3D optical flow in a harmonic motion context. Employing a depth map as well as an intensity image of a scene, they used the latter to define a region of interest for the relevant 3D data. Their gesture recognition is based on finding a 3D version of optical flow which results in velocity-annotated point clouds. These are represented efficiently by introducing motion context. The motion context is transformed into a view-invariant representation by applying spherical harmonic basis functions, which yields a harmonic motion context representation. Finally, a probabilistic classifier is applied to identify which gesture best describes a string of primitives. Shen et al. [15] proposed a new visual representation for hand motions based on motion divergence fields, which can be normalized to gray-scale images. Salient regions detected by the MSER algorithm (Maximum Stable Extremal Regions) are then identified in the motion divergence maps. From each detected region, a local descriptor is extracted which captures the local motion pattern.

Our approach is similar to that of [10] in terms of simplicity, with hand tracking although we do not apply color segmentation. A recent development is the "Haar Cascade" for detecting e.g. eyes, mouths, noses and faces [18], [9], also for tracking hands [2]. Algorithms are already included in OpenCV and they are very fast because they employ Haar wavelets, but these wavelets only coarsely resemble Gabor wavelets which are used to model cortical simple cells in area V1.

Recently we presented cortical models based on multi-scale line/edge and keypoint representations, also with keypoint annotation [4], [12], [14]. These representations, all based on responses of simple, complex and end-stopped cells in cortical area V1, can be integrated for different processes: visual reconstruction or brightness perception, focus-of-attention (FoA), object segregation and categorization, and object and face recognition. The integration of FoA, region segregation and object categorization is important for developing fast gist vision, i.e., which types of objects are about where in a scene. We also developed an initial model for cortical optical flow based on keypoints [4]. Experiments have strengthened the idea that neurons in a specialized region of the cerebral cortex play a major role in flow analysis [21], that neuronal responses to flow are shaped by visual strategies for steering in 3D environments [20], and that flow processing has an important role in the detection and estimation of scene-relative object movements [19].

In this paper we present a biologically-inspired method for tracking deformable objects based on keypoints extracted from cortical end-stopped cells. We focus on human hands and gestures which is necessary for joint human-robot manipulation of objects on top of a table: pointing and grasping etc. Our contributions are a realtime cortical hand detector, a new tracking and gesture recognition algorithm, and significantly faster keypoint annotation and tracking algorithms. The advantage of using annotated keypoints is that they provide more information than mere point clouds. The disadvantage is that the filtering involved is very expensive in terms of

[1] Vision Laboratory, LARSyS, University of the Algarve, 8005-139 Faro, Portugal {mafarrajota,masaleiro,kterzic,jrodrig, dubuf}@ualg.pt

CPU time, hence keypoint detection has been implemented on a GPU. The rest of this paper is organized as follows. In Section II we explain keypoint detection and annotation, and in Section III optical flow computation. Hand tracking is explained in Section IV, and we conclude with a discussion in Section V.

II. MULTI-SCALE KEYPOINT ANNOTATION

Keypoints are based on cortical end-stopped cells [12]. They provide important information because they code local image complexity. Moreover, since keypoints are caused by line and edge crossings, detected keypoints can be classified by the underlying vertex structure, such as K, L, T and + shaped junctions, and the angles can be employed. This is very useful for most if not all matching problems: object recognition, stereo disparity and optical flow. In this section we briefly describe the multi-scale keypoint detection and annotation processes.

Recently the original model [12] has been improved such that multi-scale keypoints can be detected in realtime. The improvements concern several important aspects: (1) a new approach to merging keypoints resulting from single- and double-stopped cell responses improves precision at coarse scales; (2) instead of applying many convolutions with filter kernels tuned to many scales and orientations, a Gaussian pyramid is used and all filters are applied in the frequency domain (FFT), which speeds up enormously keypoint extraction at coarse scales; (3) subpixel localization is used, which improves precision at fine scales and partially compensates the loss of precision at coarse scales caused by using the Gaussian pyramid; and (4) a scale selection mechanism is introduced, which significantly reduces the number of duplicated keypoints across scales. These improvements are detailed in a forthcoming paper. Below we briefly describe the algorithms.

The basic principle for multi-scale keypoint detection is based on Gabor quadrature filters which provide a model of cortical simple cells [12]. In the spatial domain (x, y) they consist of a real cosine and an imaginary sine, both with a Gaussian envelope. Responses of even and odd simple cells, which correspond to real and imaginary parts of a Gabor filter, are obtained by convolving the input image with the filter kernel, and are denoted by $R^E_{s,i}(x, y)$ and $R^O_{s,i}(x, y)$, s being the scale, i the orientation ($\theta_i = i\pi/N_\theta$) and N_θ the number of orientations (here 8) with $i = [0, N_\theta - 1]$. Responses of complex cells are modeled by the modulus $C_{s,i}(x, y)$. As mentioned before, there are two types of end-stopped cells, single and double. These are applied to $C_{s,i}$ and are combined with tangential and radial inhibition schemes in order to obtain precise keypoint maps $K_s(x, y)$. For a detailed explanation with illustrations see [12]. Below, the scale of analysis s will be given by λ, the wavelength of the Gabor filters, expressed in pixels, where $\lambda = 1$ corresponds to 1 pixel.

In order to classify any detected keypoint, the responses of simple cells $R^E_{s,i}$ and $R^O_{s,i}$ are analyzed, but now using $N_\phi = 2N_\theta$ orientations, with $\phi_k = k\pi/N_\theta$ and $k = [0, N_\phi - 1]$. This means that for each of the 8 simple-cell orientations on $[0, \pi]$ there are two opposite analysis orientations on $[0, 2\pi]$, e.g., $\theta_1 = \pi/N_\theta$ results in $\phi_1 = \pi/N_\theta$ and $\phi_9 = 9\pi/N_\theta$. This division into response-analysis orientations is acceptable according to [6], because a typical cell has a maximum response at some orientation and its response decreases on both sides, from 10 to 20 degrees, after which it declines steeply to zero; see also [3].

Classifying keypoints is not trivial, because responses of simple and complex cells, which code the underlying lines and edges at vertices, are unreliable due to response interference effects [3]. This implies that responses must be analyzed in a neighborhood around each keypoint, and the size of the neighborhood must be proportional to the scale of the cells. The validation of the line and edge orientations which contribute to the vertex structure is based on an analysis of the responses of complex cells $C_{s,i}(x, y)$. At a distance of λ, and for each direction ϕ_k, responses in that direction and in neighboring orientations ϕ_{k+l}, with $l = \{-2, -1, 0, 1, 2\}$, are summed with different weights equal to $1/2^{|l|}$. After this smoothing and detection of local maxima, each keypoint is then annotated by a descriptor of 16 bits which codes the detected orientations. In the case of keypoints caused by blobs with no underlying line and edge structures, all 16 bits are zero.

This method is an improvement of the previous method [4]. It provides a more detailed descriptor of the underlying line and edge structures, with a significant increase in performance and with a negligible loss of precision. The first five images in Fig. 1 illustrate keypoint detection and annotation at the given scales. For more illustrations see [12].

III. OPTICAL FLOW

Keypoint detection may occur in cortical areas V1 and V2, whereas keypoint annotation requires bigger receptive fields and could occur in V4. Optical flow is then processed in areas V5/MT and MST, which are related to object and ego motion for controlling eye and head movements.

Optical flow is determined by matching annotated keypoints in successive camera frames, but only by matching keypoints which may belong to a same object. To this purpose we use regions defined by saliency maps. Such maps are created by summing detected keypoints over all scales s, such that keypoints which are stable over scale intervals yield high peaks. In order to connect the individual peaks and yield larger regions, relaxation areas proportional to the filter scales are applied [12]. Here we simplify the computation of saliency maps by simply summing the responses of end-stopped cells at all scales, which is much faster and yields similar results. Figure 1 (bottom-right) illustrates a saliency map.

We apply a multi-scale tree structure in which at a very coarse scale a root keypoint defines a single object, and at progressively finer scales more keypoints are found which convey the object's details. Below we use five scales: $\lambda = [4, 12]$ with $\Delta\lambda = 2$. All keypoints at $\lambda = 12$ are supposed to represent individual objects, although we know that it is

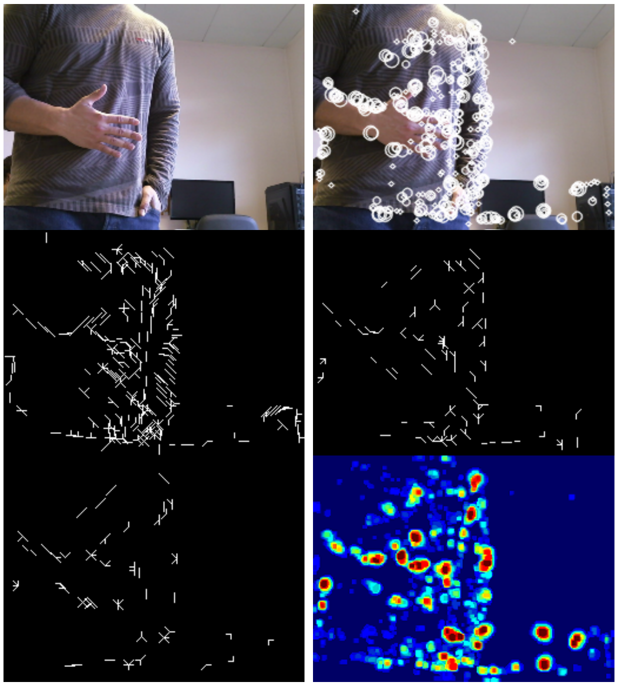

Fig. 1. Left to right and top to bottom: input frame, keypoints detected at all 5 scales, keypoint annotation at scales $\lambda = 4$, 8 and 12, and the frame's saliency map where red indicates higher and blue lower saliency.

possible that several of those keypoints may belong to a same object. Each keypoint at a coarse scale is related to one or more keypoints at one finer scale, which can be slightly displaced. This relation is modeled by down-projection using grouping cells with a circular axonic field, the size of which (λ) defines the region of influence; see [4].

As mentioned above, at a very coarse scale each keypoint – or central keypoint CKP – should correspond to an individual object [12]. However, at the coarsest scale applied, $\lambda = 12$, this may not be the case and an object may cause several keypoints. In order to determine which keypoints could belong to the same object we combine saliency maps with the multi-scale tree structure.

At this point we have, for each frame, the tree structure which links the keypoints over scales, from coarse to fine, with associated regions of influence at the finest scale. We also have the saliency map obtained by summing responses of end-stopped cells over all scales. The latter, after thresholding, yields segregated regions which are intersected with the regions of influence of the tree. Therefore, the intersected regions link keypoints at the finest scale to the segregated regions which are supposed to represent individual objects.

Now, each annotated keypoint of frame i can be compared with all annotated keypoints in frame $i-1$. This is done at all scales, but the comparison is restricted to an area with radius 2λ instead of λ at each scale in order to allow for larger translations and rotations. In addition, (1) at fine scales many keypoints outside the area can be skipped since they are not likely to match over large distances, and (2) at coarse scales there are less keypoints, λ is bigger and therefore larger distances (motions) are represented there. The tree structure is built top-down, but the matching process is bottom-up: it starts at the finest scale because there the accuracy of the keypoint annotation is better. Keypoints are matched by combining three similarity criteria with different weight factors:

(a) The distance D serves to emphasize keypoints which are closer to the center of the matching area. For having $D = 1$ at the center and $D = 0$ at radius 2λ, we use $D = (2\lambda - d)/2\lambda$ with d the Euclidean distance (this can be replaced by dynamic feature routing [4], [13]).

(b) The orientation error O measures the correlation of the attributed orientations, but with an angular relaxation interval of $\pm 2\pi/N_\theta$ applied to all orientations such that also a rotation of the vertex structure is allowed. Similar to D, the summed differences are combined such that $O = 1$ indicates good correspondence and $O = 0$ a lack of correspondence. Obviously, keypoints marked "blob" do not have orientations and are treated separately.

(c) The tree correspondence C measures the number of matched keypoints at finer scales, i.e., at any scale coarser than the finest one. The keypoint candidates to be matched in frame i and in the area with radius 2λ are linked in the tree to localized sets of keypoints at all finer scales. The number of linked keypoints which have been matched is divided by the total number of linked keypoints. This is achieved by sets of grouping cells at all but the finest scale which sum the number of linked keypoints in the tree, both matched and all; for more details see [4].

The three parameters are combined by grouping cells which can establish a link between keypoints in frame $i-1$ and i. Mathematically we use the similarity measure $S = \alpha O + \beta C + \gamma D$, with $\alpha = 0.4$ and $\beta = \gamma = 0.3$. These values were determined empirically. The candidate keypoint with the highest value of S in the area (radius 2λ) is selected and the vector between the keypoint in frame $i-1$ and the matched one in frame i is computed. Remaining candidates in the area can be matched to other keypoints in frame i, provided they are in their local areas. Keypoints which cannot be matched are discarded. Figure 2 (top two rows) illustrates a sequence of 10 frames with a moving hand with detected optical flow vectors.

IV. HAND TRACKING AND MOTION RECOGNITION

Moving objects are segregated and detected by analyzing the optical flow vectors of their multi-scale tree structures. Only trees with keypoints with sufficiently large vectors (displacements of more than 2 pixels between frames) are considered. Deformable objects can be distinguished from rigid ones because some or even all their multi-scale trees possess different motion vectors, i.e., different directions and/or velocities.

Hands performing gestures are a particular class of deformable objects. Hand and gesture recognition is obtained by using a simple and fast algorithm. This algorithm relates

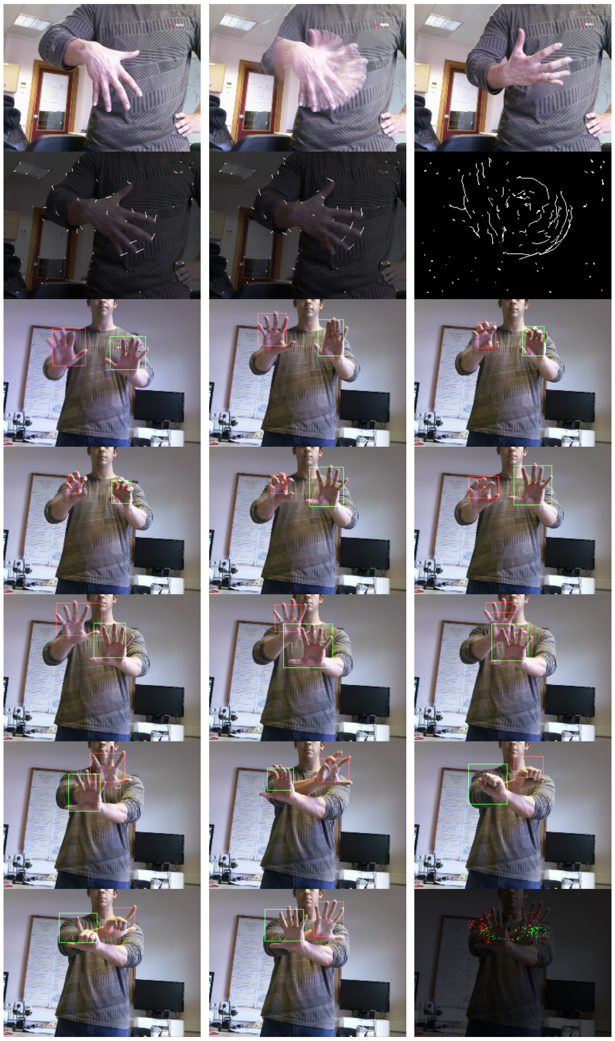

Fig. 2. Top two rows, left to right: initial, combined and final frames of a moving hand sequence; optical flow of two frames; and combined optical flow of the sequence. Bottom five rows: another sequence with tracked hands marked by their bounding boxes. Bottom-right: the combined centers of the boxes.

keypoint positions in previously prepared templates with those detected in acquired image frames. The templates are prepared by simply capuring images of a person with specific hand gestures, after which the hand regions are selected and the keypoint information is stored in small lists; see below. The matching algorithm exploits keypoints at scales not too fine, $\lambda = 8$ and 12, because the number of keypoints is not too big and we are not interested in tiny details. At each scale, and for each template, the angle and the Euclidean distance from each keypoint to all other keypoints are computed. Let us call these primary and secondary keypoints. This yields many but relatively small lists, one for each primary keypoint. Since angles and distances to secondary keypoints are relative to a primary keypoint, all lists are translation and rotation invariant. Typically, a template counts 10 keypoints at scale $\lambda = 8$, such that there are 10 lists each with 9 elements. At scale $\lambda = 12$ there are less. At the moment we only use five templates; see Fig. 3.

Let us first assume that no prior information of a new image frame is available: no known hand position and gesture, and no tracking information. All already available keypoints (because of optical flow) at the two scales and in the entire frame are processed sequentially. In the matching process, the primary keypoint of one template list is positioned at a frame's keypoint, and its secondary keypoints are matched with those in the frame: at positions according to the angles and distances. In order to introduce some flexibility in the matching, for the number of hand-gesture templates cannot be too large, we apply a position tolerance: about 1/5th the size of the template, for example 20×25 pixels in the case of a template of 100×125 pixels. The lists are also mirrored about the major dimension of the template (for the palm and back side of the hand) and rotated by applying only 16 angles because of the position tolerance. Hence, each list involves checking 32 keypoint configurations, or typically $2 \times 10 \times 32$ lists per template, but the matching is fast because a discrete lookup table is used and both the lists and the lookup table are in the CPU's cache memory. When at least 50% of all keypoints in a template list match those in the neighborhood of a frame's keypoint, at one of the two scales, the matching template determines the hand's gesture, its position is known as is its bounding box.

Translation and rotation invariance are obtained by considering (rotated and mirrored) relative angles and distances between keypoints. In order to also achieve scale (size) invariance in the future, each gesture must be represented by several templates captured with different hand sizes (hand-camera distances). A larger number of sizes results in more reliable detection, but costs more CPU time. However, the additional cost is rather low because it only involves matching of many but very small keypoint lists.

By combining optical flow with the hand-gesture detector, hands can be tracked and recognition becomes more robust and faster. Tracking is achieved by combining the last valid hand template, its position in the last frame, and the actual optical flow. This reduces false positives and speeds up the tracking process. At the beginning of the process, camera frames are processed until at least one hand has been detected. Then, the processing of the following frames is simplified by searching the area(s) around the position(s) of the last detected hand(s). Nevertheless, the remaining part of the frames must be analyzed because a hand can be temporarily occluded or a new one can enter the frame. However, this can be done once or twice per second.

For final gesture recognition we assume that the person keeps his hand stable at about the same position, such that the optical flow of the hand is zero or very small. This can be a few camera frames, i.e., a fraction of a second, but it depends on the application: in human-robot interaction while manipulating objects on a table, only occasional final gestures like pointing and grasping are important, but in a game a continuous stream of positions and gestures may be required. The bottom five rows in Fig. 2 show a sequence

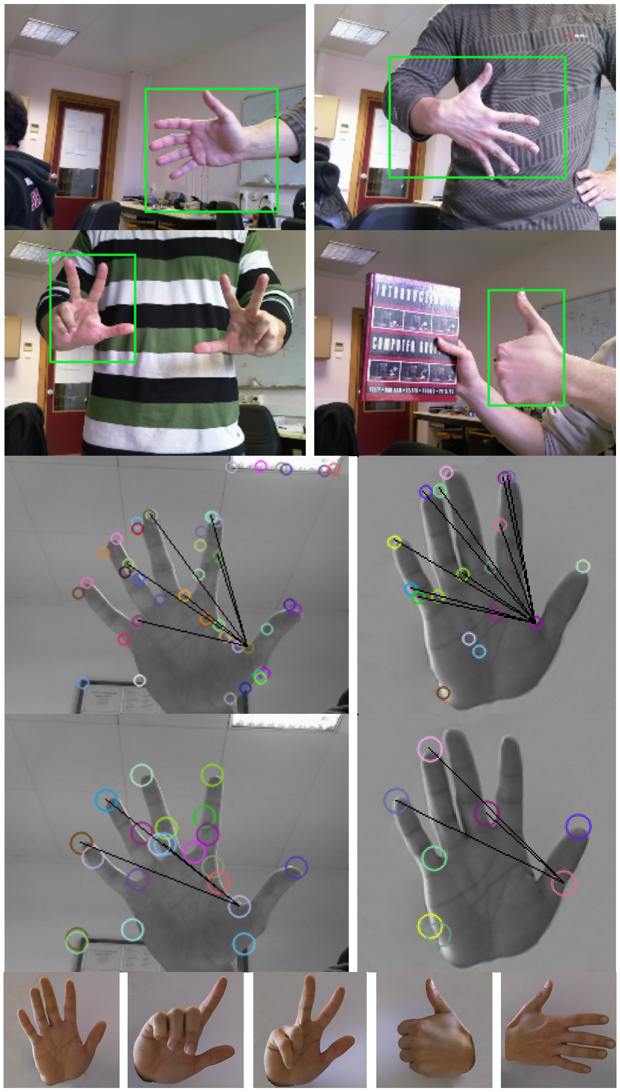

Fig. 3. Top two rows: four examples of recognized gestures. Middle two rows: template matching at $\lambda = 8$ (3rd row) and $\lambda = 12$ (4th row). Bottom: the five hand templates used.

with two tracked hands. The bottom-right image combines the centers of the bounding boxes. Even if the hands are very close or partly overlapping, the tracking process can separate them. Figure 3 shows recognized gestures (top), the matching process at two scales (middle), and the five templates (bottom). Our method can also be applied to track other deformable objects, for example human bodies; see Fig. 4. This figure shows a sequence of frames while a person straightens after picking up a bottle, and then brings his arm with the bottle close to the body while also straightening his head. In contrast to hand gestures, templates of body gestures remain to be developed and applied.

V. DISCUSSION

In this paper we presented a biologically inspired method for hand detection, with tracking and gesture recognition. After

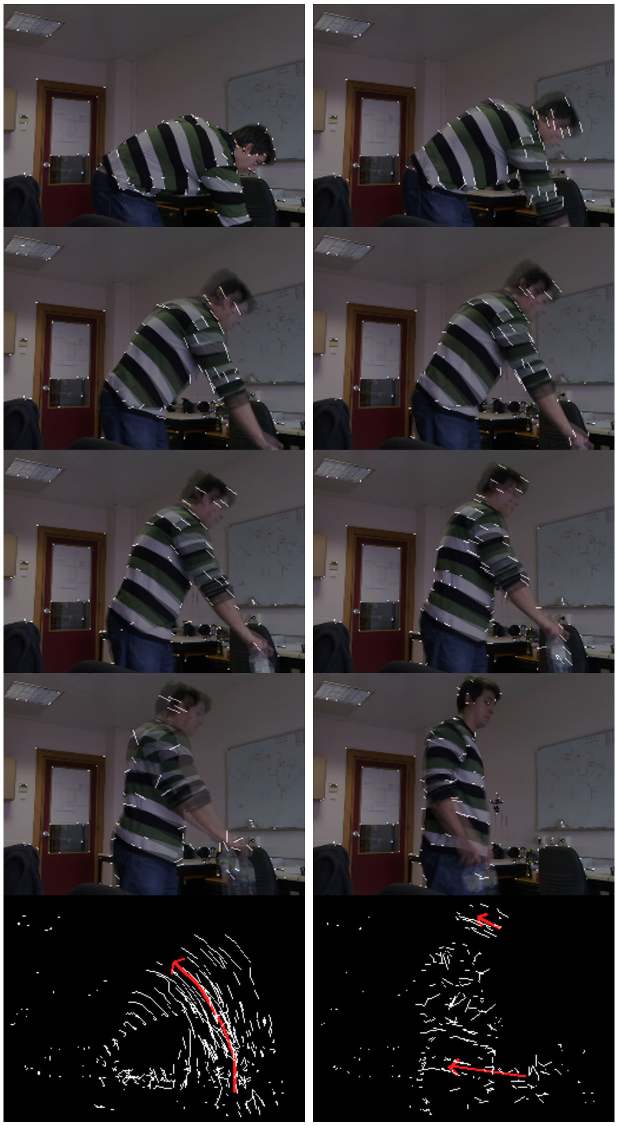

Fig. 4. The optical flow model applied to a person after fetching a bottle from the floor. The sequence shows vectors between successive frames. The two bottom images show the combined vectors while straightening (left), followed by bringing the arm and bottle close to the body and moving the head (right). Significant motions attributed to tracked, segregated regions are indicated by the red arrows.

optimizing the keypoint-detection algorithm and by limiting the number of scales, the method works in realtime when using a webcam, and it yields good results despite the fact that color information has not yet been used. The method was expected to work well because of our previous experience with cortical models: the keypoint scale-space provides very useful information for constructing saliency maps for Focus-of-Attention (FoA), and faces can be detected by grouping facial landmarks defined by keypoints at eyes, nose and mouth [12]. In [14] we have shown that the line/edge scale-space provides very useful information for face and object recognition. Obviously, object detection and recognition are related processes, with a seamless integration in the cortical so-called where and what pathways, i.e., the dorsal pathway (where is it?) and ventral one (what is it?). However, there may be no clear dichotomy in the sense that keypoints are only used in the where pathway and lines and edges only in the what pathway.

Since local clusters of keypoints are mostly related to individual moving objects, object segregation can be achieved and objects can be tracked. Cortical areas MT and MST are involved in optical flow and in egomotion, but recent results obtained with fMRI showed no clear neural activity in their ventral (what) and dorsal (where) sub-areas. Instead, there is elevated activity in between the sub-areas [16]. This might indicate that optical flow at MT level is processed separately or involves both pathways. The fact that the use of only keypoints can lead to very good results in optical flow and object (hand) segregation and tracking may indicate some "preference" of the dorsal (where) pathway for keypoints. This idea is strengthened by the fact that area MT also plays a role in the motion-aftereffect illusion [8], which is tightly related to motion adaptation and prediction.

Being a biologically inspired model, keypoint detection involves filtering input frames with many kernels (complex Gabor functions). We apply eight orientations but only a few scales in order to achieve realtime processing when using a normal webcam: five scales for optical flow and region segregation, of which only two scales are used for hand and gesture detection. The main limitation is the Gabor filtering with keypoint detection. The improved algorithm has already been implemented on a GPU, allowing to process at least 10 frames/s with a maximum resolution of 600×400 pixels and using at least 6 scales if they are not too fine. The GPU's memory of 1 GByte is the bottleneck for using larger images and fine scales because of the Gaussian pyramid.

Ongoing research focuses on motion prediction, a process which occurs in cortical area MST. In addition, instead of only extrapolating hand positions, also the gestures can be tracked and extrapolated, such that the number of templates to be matched can be reduced. Nevertheless, although currently a few distinct gestures are being used, extrapolation may involve more "intermediate" gestures and therefore templates. The ultimate goal is to apply a 3D hand model in the entire process. This can be done by employing cheap and off-the-shelf solutions like a Kinect [11] or two webcams with a biological disparity model. The same applies to human bodies: the tracking and prediction of body joints by exploiting all spatio-temporal information.

ACKNOWLEDGEMENTS

This work was partially supported by the Portuguese Foundation for Science and Technology (FCT) project PEst-OE/EEI/LA0009/2011, EU project NeuralDynamics FP7-ICT-2009-6 PN: 270247, FCT project Blavigator RIPD/ADA/109690/2009 and by FCT PhD grants to MF (SFRH/BD/79812/2011) and MS (SFRH/BD/71831/2010).

REFERENCES

[1] J.P. Bandera, R. Marfil, A. Bandera, J.A. Rodríguez, L. Molina-Tanco, and F. Sandoval. Fast gesture recognition based on a two-level representation. *Pattern Recogn. Lett.*, 30(13):1181–1189, 2009.

[2] A.L.C. Barczak and F. Dadgostar. Real-time hand tracking using a set of cooperative classifiers based on haar-like features. In *Research Letters in the Information and Mathematical Sciences*, pages 29–42, 2005.

[3] J.M.H. du Buf. Responses of simple cells: events, interferences, and ambiguities. *Biol. Cybern.*, 68:321–333, 1993.

[4] M. Farrajota, J.M.F. Rodrigues, and J.M.F. du Buf. Optical flow by multi-scale annotated keypoints: A biological approach. *Proc. Int. Conf. on Bio-inspired Systems and Signal Processing (BIOSIGNALS 2011), Rome, Italy, 26-29 January*, pages 307–315, 2011.

[5] M.B. Holte, T.B. Moeslund, and P. Fihl. View-invariant gesture recognition using 3D optical flow and harmonic motion context. *Comput. Vis. Image Underst.*, 114(12):1353–1361, 2010.

[6] D.H. Hubel. *Eye, Brain and Vision*. Scientific American Library, 1995.

[7] H. Kim, G. Kurillo, and R. Bajcsy. Hand tracking and motion detection from the sequence of stereo color image frames. *Proc. IEEE Int. Conf. on Industrial Technology*, pages 1–6, 2008.

[8] A. Kohn and J.A. Movshon. Neuronal adaptation to visual motion in area MT of the macaque. *Neuron*, 39:681–691, 2003.

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

[10] C. Manresa, J. Varona, R. Mas, and F. Perales. Hand tracking and gesture recognition for human-computer interaction. *Electronic Letters on Computer Vision and Image Analysis*, 5(3):96–104, 2005.

[11] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3D tracking of hand articulations using kinect. *Proc. BMVC*, pages 101.1–101.11, 2011.

[12] J. Rodrigues and J.M.H. du Buf. Multi-scale keypoints in V1 and beyond: object segregation, scale selection, saliency maps and face detection. *BioSystems*, 2:75–90, 2006.

[13] J. Rodrigues and J.M.H. du Buf. A cortical framework for invariant object categorization and recognition. *Cognitive Processing*, 10(3):243–261, 2009.

[14] J. Rodrigues and J.M.H. du Buf. Multi-scale lines and edges in V1 and beyond: brightness, object categorization and recognition, and consciousness. *BioSystems*, 95:206–226, 2009.

[15] X. Shen, G. Hua, L. Williams, and Y. Wu. Dynamic hand gesture recognition: An exemplar-based approach from motion divergence fields. *Image and Vision Computing (In Press)*, 2012.

[16] A. Smith, M. Wall, A. Williams, and K. Singh. Sensitivity to optic flow in human cortical areas MT and MST. *European J. of Neuroscience*, 23(2):561–569, 2006.

[17] H. Suk, B. Sin, and S. Lee. Hand gesture recognition based on dynamic Bayesian network framework. *Pattern Recogn.*, 43(9):3059–3072, 2010.

[18] P. Viola and M.J. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. IEEE CVPR*, 1:511–518, 2001.

[19] P.A. Warren and S.K. Rushton. Optic flow processing for the assessment of object movement during ego movement. *Current Biology*, 19(19):1555–1560, 2009.

[20] K.P. William and J.D. Charles. Cortical neuronal responses to optic flow are shaped by visual strategies for steering. *Cerebral Cortex*, 18(4):727–739, 2008.

[21] R.H. Wurtz. Optic flow: A brain region devoted to optic flow analysis? *Current Biology*, 8(16):R554–R556, 1998.