Computer Vision and Image Understanding 149 (2016) 209–228

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

An integrated artificial vision framework for assisting visually impaired users

Manuela Chessa[1,*], Nicoletta Noceti[1], Francesca Odone, Fabio Solari, Joan Sosa-García, Luca Zini

DIBRIS - Università degli Studi di Genova via Dodecaneso, 35, 16146 Genova, Italy

ARTICLE INFO

Article history:
Received 17 April 2015
Accepted 18 November 2015

Keywords:
Applications for the visually impaired
Scene understanding
Surface orientation
Time to collision
Face detection and recognition
Text detection and recognition
Object recognition
Bio-inspired computer vision

ABSTRACT

We present a conceptual framework inspired by biological vision which integrates low-level vision functionalities oriented to actions – typical of the so-called "where" dorsal pathway – with identification and recognition capabilities – common to the "what" ventral pathway. Although they proceed independently, these complementary vision models may provide a deeper scene understanding and a more efficient computational framework. In this work, we refer specifically to a set of video analysis modules, which include semantic annotations of the scene and 3D environment interpretation. We discuss and qualitatively evaluate possible connections and integrations between different functionalities, grounding our analysis on a set of specific use cases, depicting visually impaired users finding their way in unfamiliar environments.

1. Introduction

Visually impaired people find it difficult to operate in unfamiliar environments or unpredictable settings. These challenges limit their independence and widen the gap between them and the normally sighted population. They are likely to modify their life to adapt to this particular condition, limiting their actions to prevent failures and frustrations. Usually, after years of training they learn how to conduct a reasonably independent life, in known environments and under a repetitive schedule. Often, they need to be taken to places for many times until they learn by heart all the important details of a given path, using complementary senses in the case of severe visual deficiencies [1]. In less controlled circumstances, they may have problems in finding their way in unknown places or in avoiding physical barriers. Moreover, it may be difficult for them to choose clothes and to buy objects; it is challenging to find a shop of a specific type in a street they do not know well, or to enter a post office, as they do not know if the clerk is ready to serve them.

According to the European Blind Union[2], there are nearly 30 million blind and partially sighted people in the member countries of the European Blind Union. This figure is based on the premise that 1 in 30 people are blind or partially sighted, and takes into account the varying definitions of visual impairment. This number is expected to grow in the future, together with the increase of the citizens' average age. At the same time, we can assume the number of elderly people willing to use technology will also grow.

Therefore, in the last few years there has been a steady growth of research and development of methods and systems to assist visually impaired people. While there has been a considerable advancement at the level of research findings, still most issues have been addressed independently. This lead to the development of devices or dedicated apps that may, in principle, be considered interesting aids, but it is virtually impossible to imagine a user adopting them in parallel. Only recently, some products are starting to propose multiple functionalities (e.g. the ORCAM device, which appears to be largely based on an effective OCR).

In this paper, we explore the possibility of designing multifunctional aids, purely based on Computer Vision, to address general goals of scene understanding and wayfinding. We reason on the benefits of integrating different functionalities in the same framework, obtaining improved performances, a wider descriptive power, and a potential for developing more general purpose devices in the future.

In our work, we draw inspiration from biological visual processing. We refer to the neural modeling of the vision functionalities, that states how any living being that acts in the environment has the need of visually exploring the neighboring locations, in order to both navigating and recognizing objects. Such an ability has evolved in two

* Corresponding author.
E-mail address: manuela.chessa@unige.it (M. Chessa).

[1] Manuela Chessa and Nicoletta Noceti equally contributed to the paper.

[2] The European Blind Union (EBU) is a non-governmental and non-profit-making organisation founded in 1984, currently composed of 45 countries: http://www.euroblind.org/.

http://dx.doi.org/10.1016/j.cviu.2015.11.007
1077-3142/

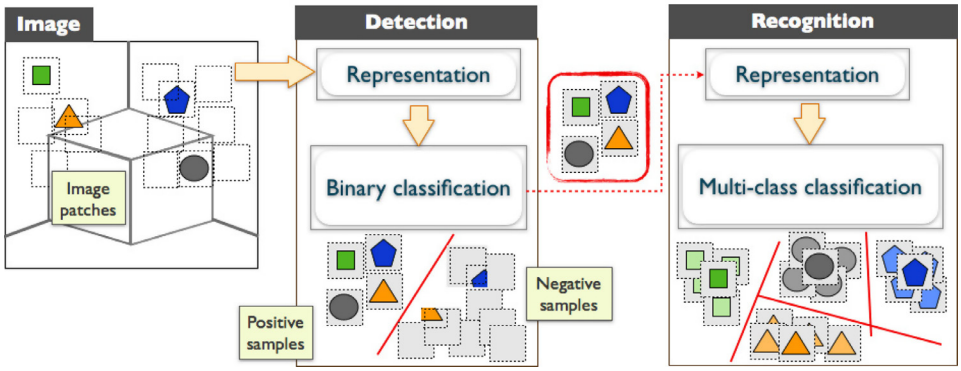

Fig. 1. A visual representation of detection and recognition pipeline. The actual detection relies on classifying image patches using a binary classifier that decides whether the patch contains or not an object of interest (e.g. shapes). Only positive samples undergo the recognition step, based on a multi-class classification, to associate a sample with an object class (e.g. square, circle,...).

different visual pathways in the primate cerebral cortex [2]. The ventral visual pathway (the *"What" stream*) [3] is devoted to build a detailed representation of the environment features required for cognitive operations, such as identification and recognition. The dorsal visual pathway (the *"Where" stream*) [4] provides a distributed representation of visual features, such as objects motion and distance, which enables the visual control of actions. In particular, such visual pathways are organized in hierarchical layers of neural architectures performing visual processing that becomes more informative and complex in higher layers.

The framework we propose has the aim of mimicking the first visual processing stages, which are present in the dorsal and ventral streams, in order to provide early visual capabilities to impaired people. Figs. 1 and 7 show the proposed instances of the "What" and "Where" visual pathways, respectively.

Although we do not consider a specific hardware, our reference technologies are wearable devices, such as sensorized glasses (usually mounting one camera). We simulate this configuration by processing video streams acquired by mobile phones.

We consider a few reference functionalities, which are summarized in the following. They are not intended to be an exhaustive list, but rather a rich starting point for future investigations.

Analysis of the environment. The physical constraints of an environment are 3D cues that normally sighted people perceive and understand very easily. For visually impaired people, instead, the same perceptual task becomes highly challenging, especially in unfamiliar places. In this respect, we provide an intuitive and simple way to obtain coarse information about the 3D structure of the scene (e.g. noticing the presence of walls and obstacles, and their temporal distances) in the user's field of view. The analysis of the environment is performed by a two-stage neural model, which first estimates low-level visual features, related to motion and depth, and then builds a sparse representation of the object surfaces and of the time to contact. Such information allows visually impaired people to be aware of the spatio-temporal structure of the scene, and, thus, to move in unknown environments.

Text reading. Natural scenes contain large amount of information directed to humans which is conveyed through text. The capability of "reading" textual information may favor visually impaired users moving in the environment for locating a specific shop, or allowing them to localize themselves by reading the streets names. We provide methods relying on a two-stage solution that first detects and then recognizes characters. Each character is processed independently. A reading functionality may be applied, in order to finally recognize words.

Object recognition. In everyday life, we often need to recognize and use large varieties of objects. In familiar settings, a visually impaired user may learn the position of specific objects by heart. When interacting with the external world, the same problem may be very challenging. Their limited visual ability triggers the refinement of alternative perception capabilities (e.g. tactile), and in some cases exploring the shape of a 3D object is sufficient to achieve the recognition. However, there are circumstances where the only shape is not informative enough. In such cases, the user may be able to roughly categorize the object (e.g. deciding that it is a box or a can) but cannot understand more specific properties (e.g. the content). As a use case for this wide scenario, we consider the problem of banknotes recognition. The same pipeline may be used to address a larger class of recognition problems, which aim at identifying objects observed by a close range.

Face detection and recognition. An important and quite general difficulty for blind users is in interacting with known and unknown people in unfamiliar environments. In the former case, they do not have clues on the presence of a known person in the environment, unless the other person sees them and sends them a vocal feedback; in the latter case, they may not know there is a person in their vicinity or they do not know whether this person is paying attention to them or not. This lack of visual contact greatly reduces the spontaneity of interpersonal contacts. We then consider a fast and effective face detection and recognition pipeline. The face detection module alerts the user on the presence of a person in the surroundings; in conjunction with time to contact, it may alert on the fact that the person is approaching. The face recognition module informs the user if a known person is present in the scene. It is possible to "introduce" a new person to the system for future occasions.

The reminder of the paper is organized as follows. In Section 2 we briefly review the literature of related works addressing assistive vision problems. Section 3 provides an overview of our methods for assigning semantic tags to a scene, with specific reference to two different scenarios of interest; Sections 4 and 5 are focused on the introduction of methods we used to extract low-level features, and on how we exploit them for understanding the coarse structure of the scene, respectively. Section 6 provides examples of integrated functionalities which are shown on a selection of use cases, while Section 7 is left to conclusion.

2. Related works

In the last decades, much attention has been devoted to providing technological aids to visually impaired users.

Many navigation devices have been devised to guide blind people, some of them based on non-visual sensors, such as laser emitters and

receivers or ultrasonic wide beam equipments (see [5] for a review). In addition, recently some authors have developed systems based on RGBD sensors (which combine images with infrared data, e.g. see [6]).

Since a general account of current research is out of the scope of this paper, here we focus on systems and tools based primarily on Computer Vision methods.

Optical Character Recognition (OCR) systems have been among the first Computer Vision products available on the market and particularly useful to visually impaired users. A classical OCR system translates the content of a scanned document into letters and words, successively available to be stored in files or to be read aloud by text-to-speech softwares. The combination of OCR systems with vocal synthesis clearly has the potential for being a great assistive technology by which visually impaired people can access printed information. The increasing availability of cheap mobile devices, equipped with cameras and good computing power, extended the scope of such applications. Here it is worth mentioning ABBYY technologies[3]. A commercial mobile OCR designed for visually impaired users is the KNFB Reader Mobile[4]. Another tool popular among visually impaired users is the Intel ®Reader[5], a commercial product consisting in a dedicated portable tablet device.

A more ambitious challenge is to detect and read text in unconstrained environment, without any prior on the position of text in the image [7,8]. This functionality also helps sign reading, object recognition, shops detection, and has a great impact on wayfinding. Only very recently this functionality appeared on commercial products. We mention the ORCAM system[6], which is attracting much attention of users' groups as it is promising for a wide scope of problems. Since a common challenge of OCR systems for visually impaired is the fact we cannot rely on the availability of good quality images, a large amount of research specifically concerning this aspect has been carried out in recent years (see for instance the thematic conferences ICDAR, and, for instance, [9–11]).

Specifically on sign reading it is worth citing the work described in [8], which introduced VIDI (Visual Integration and Dissemination of Information), a prototype system for detecting and recognizing signs, able to communicate their contents with a synthesized voice, and [12], in which the authors describe a set of algorithms for sign detection and recognition for a wearable system to be used by the blind, capable of recognizing a broad variety of signs. Besides sign reading, the general goal of wayfinding has been first addressed by proposing slight modifications of the environment, in order to produce a system of signs that could be easily read by the user (e.g. signs in Braille) or by ad hoc devices. Meaningful from the Computer Vision standpoint are the passive signs [13,14] and the reflective signs for infrared illuminators [15]. Navigation in traffic intersections has been addressed in [16,17]. Crosswatch [17] is a mobile phone-based system to help visually impaired pedestrians entering the crosswalk in the right direction. The system has been tested by blind users demonstrating the feasibility of the system. In [18], the authors deal with the problem of designing a Computer Vision algorithm to rapidly and reliably detect the walk light signal at a traffic light whenever it is present, in order to assist the users before starting to cross.

Sensory systems for mobility support are instead technologies aiming at detecting any feature in the environment that could be critical for a safe ambulation, such as obstacles, steps and points of access. The idea behind these applications is to replace or enhance the long cane, the widely used tool that ensures safe ambulation and that represents the baseline against which any Electronic Travel Aid (ETA) system should compete with. In general, these applications are based on depth estimation, that can be achieved for instance by stereo vision [19] or by active triangulation [20].

Another important element for visually impaired users is the ability to perform object recognition tasks, possibly with a reference to a specific set of objects of interest. On this respect, it is worth mentioning the Trinetra Project (CMU) based on barcode reading for object recognition [21]. In [22] the authors introduce ShelfScanner, a prototypical object detection system helping visually impaired users to autonomously find the needed groceries in a store. Another domain-specific object recognition system for impaired users is banknotes recognition (also known as *Paper Currency Recognition* (PCR)). Although banknotes counters, money changers and vending tickets are common on the market, their underlying technology relies on the assumption of having neither background information nor illumination changes. Such issues are instead typical of a Computer Vision application. The work in [23] describes a prototype camera-phone based system to recognize currency in real time from video. Taking into account the difficulty for an impaired individual to capture high quality images, the system exploits the video stream by processing each captured frame until a good view of the object is available. The same principle is exploited by LookTel Money Reader[7], a commercial iPhone application that recognizes currency in real time. A recent work [24] addresses the same problem with good performances.

In terms of improvements in the socialization quality, face recognition has been explored. The work of [25] describes the iCare Interaction Assistant, consisting of a wearable device for assisting visually impaired individuals during social interactions. The tool has been developed under the iCare project[8], providing a set of digital tools for blind people, and in particular proposing automatic face recognition to address prosopagnosia (also known as face blindness). Related to the problem of face recognition is also the work presented in [26]. Here the authors deal with the problem of person localization introducing the Social Interaction Assistant, a prototype wearable device that focuses on the localization of a person approaching the users while facing them. Face detection and recognition has been explored in [27], a feasibility study with the purpose of improving social inclusion and context awareness of visually impaired users.

3. Understanding the semantic content of a scene

In this section, we discuss the problem of providing semantic tags of the surrounding environment. From a technical standpoint, this functionality is often obtained by addressing *image classification* problems. We consider two distinct daily life scenarios. On the first one, the user is moving in the environment, hence there is the need of analyzing the surrounding world to understand the presence of objects of potential interest. In such a case, large portions of the acquired images may include clutter besides the object(s) of interest, calling for computational methods able to first detect and then recognize them. The second scenario refers to the user needing to recognize objects, typically hand-held, at a lower distance. In this context, we can assume that the majority of the image is occupied by the object of interest, possibly with some minor clutter, thus the detection step can be safely removed.

In the following, we review our approaches to the above mentioned problems.

3.1. Recognizing objects in the wild

To address the problems of detecting and recognizing classes of objects in the observed scene, we refer to a rather standard two-stage pipeline, summarized in Fig. 1. Indeed, it may be interpreted as a possible strategy to address perceptual tasks typical of the *What stream*.

[3] http://www.abbyy.com/textgrabber/
[4] http://www.knfbreader.com/products-mobile.php
[5] http://www.careinnovations.com/products/intel-reader-text-to-speech-technology
[6] http://www.orcam.com/
[7] www.looktel.com
[8] http://cubic.asu.edu/projects/index.php

Object detection can be formulated in terms of binary classification. Positive examples refer to the class of interest, negative examples are chosen among background elements or other classes. Usually, the actual detection phase is characterized by a large amount of computations and by a considerable unbalancing between positive elements (instances of the object class of interest) and negative ones. To cope with detection in real time, cascading or coarse-to-fine procedures are often adopted.

Object detection may be followed by *object recognition*, where we associate specific class instances with the detected objects. The recognition step, usually implemented by means of multi-class classification, is applied on detected elements only, thus it is less critical from the computational point of view. In what follows we consider two object classes, *human faces* and *text*, which are crucial for improving the users quality of life. Awareness on the presence of a person (possibly a known one) in the scene improves inclusion, the ability to locate and read text in cluttered environments improves way finding and stimulates independence. As a further motivation, these objects classes have been largely studied in Computer Vision and state of the art performances are to the standard level of market products.

An important user requirement we consider throughout the work is to provide real time feedback. Thus, all our methods are implemented with the aim of controlling the computational cost without degrading too much the recognition performances. Last, we take into account usability issues thus we do not pose any limit to the type of acquisition: the image we process are acquired with only a few constraints on the camera position.

3.1.1. Detecting and recognizing faces

In the application scenario we consider, frontal faces are particularly meaningful as they highlight the presence of people facing the user. Once a person has been identified, a further recognition layer may allow the user to identify the presence of known people.

A complete discussion on related works for face detection and recognition is out of the scope of this work, and we refer the interested reader to [28] for a deeper analysis. However, it is worth noting that face detection and recognition have been very often addressed by finding appropriate image descriptors – eigenfaces [29], fisherfaces [30], component-based methods [31], rectangular features [32], local approaches as [33] and, more recently, sparse coding [34], just to name a few. In large part of research, the representation problem has been addressed by coupling the design of appropriate image descriptors with feature selection methods, as Adaboost [35,36], genetic-based approaches [37], and LASSO methods [33].

In the following, we summarize our approach to face detection and recognition. The two actions are addressed by two independent modules, sharing a common philosophy but implementing different strategies. In both cases we start from the observation that human faces have a common low level structure but large intra-class variability. Also, they may be largely affected by environment changes or by variations due to time going by. Over the years, many effective methods based on the availability of large dictionaries of overcomplete features have been proposed (see e.g. [35,38]). Such dictionaries allow us to model local as well as global structures. At the same time, they carry a large amount of redundant information, therefore, to improve efficiency and computational performances, they are often coupled with a feature selection layer that introduces adaptivity to the data representation. This procedure also helps obtaining compact feature vectors which may be efficiently computed in a real time.

Given a training set $(\mathbf{x}_i, y_i), i = 1, \ldots, n$, with $\mathbf{x}_i \in X \subseteq R^m$ $y_i \in R$, we formulate the *feature selection problem* as a generalized linear relationship between input and output data as

$$\sum_{j=1}^{p} \phi_j(\mathbf{x}_i)\beta_j = y_i \tag{1}$$

where $\mathcal{D} = (\phi_j)_{j=1}^{p}$ is a dictionary (a collection of atoms or features). Problem (1) can be reformulated as $\Phi\beta = \mathbf{y}$, with Φ the feature matrix s.t. $\Phi_{ij} = \phi_j(\mathbf{x}_i)$ of size $n \times p$, $\mathbf{y} \in \{-1, +1\}^{n\times 1}$ the output vector, and $\beta \in \mathbb{R}^{p\times 1}$ the vector of weights. The goal of feature selection is to find a *sparse* $\boldsymbol{\beta}$ which approximates well the input-output relationship.

In image related problems, the dictionary matrix Φ is often large and rectangular ($p \gg n$), producing a linear system which is underdetermined. Under this assumption, algebraic solutions to the linear system are not feasible, thus one can rely on some form of regularization, as we will see later in the section.

Face detection. The dictionary we adopt for face detection is based on rectangle features [32], computed over different locations, sizes, and aspect ratios. For a 19×19 pixels image patch we obtain an overcomplete dictionary of about $64,000$ features. We perform feature selection by L_1 regularization, that is by solving the following problem:

$$\beta^* = \arg\min_{\beta} \left(\|\mathbf{y} - \Phi\beta\|_2^2 + \tau \|\beta\|_1 \right), \tag{2}$$

the so-called LASSO estimation [40,41]. Notice that in our case the feature matrix Φ would be of about 1 Gb (for a training set of $n = 4000$ elements and real numbers represented in single precision). To efficiently manage the matrix, we break the large dimensional system in many smaller problems obtained by considering subsets of features, uniformly sampled with replacement. Then, we perform S feature selection steps on feature subsets which may overlap. The final set of selected features $\tilde{\beta}$ will contain elements that have been selected each time they were included in the features subset considered. Details on this procedure may be found in [33], a visual sketch is shown in Fig. 2(a) (left).

At run time, object detection in images is usually performed by raster scanning the image (or part of it) at all possible positions and scales, applying a binary classifier to each patch (or window) obtained during the raster scan. A binary classifier is trained to decide if the image patch analyzed at a given position and scale belongs to the class of the object of interest or not. We design an efficient coarse-to-fine classifier, based on N classification layers, each one considering a small set of features. Target performance of each layer is chosen so that each classifier will not be likely to miss faces: we set the minimum hit rate to 99.5% and the maximum false positive rate to 50%. In this way, assuming a cascading of 10 classifiers, we would get $H = 0.995^{10} \sim 0.9$ and $F = 0.5^{10} \sim 3 \times 10^{-5}$.

The cascade of classifiers works as follows (see Fig. 2(a), right): a test image is given to the first classifier of the cascade (CL.1), which decides whether the input image is a face or not. If the answer is negative, no further computation is performed. Then, "easy" negatives are quickly discarded with a low computational burden. If the image is classified as a positive, in goes to the next classification layer (CL.2), and so on. A face is detected only if all N classifiers of the cascade answer positively. A final non maxima suppression concludes the detection procedure.

Once we detect a face in the scene, we try and recognize it against a set of models of known individuals. To this purpose, we represent each face image by means of an overcomplete dictionary of Local Binary Pattern (LBP) descriptors [35,42]. The representation scheme is exemplified in Fig. 2(b), left. We consider a grid of overlapping regions having different scale and aspect ratio (images are all rescaled to the size 40×40 pixels). We then extract uniform 8-bits LBPs [43] and quantize their values with 59-bins histograms, accordingly to the original work. The final over-complete description consists of 841 LBPs, thus each training image $\mathbf{x}_i$ is described by $841 \times 59 = 49,619$ features. Again, feature selection is advisable to control the computational cost at run time, but in this case we have to consider that our feature vector contains groups of features (the bins of a LBP

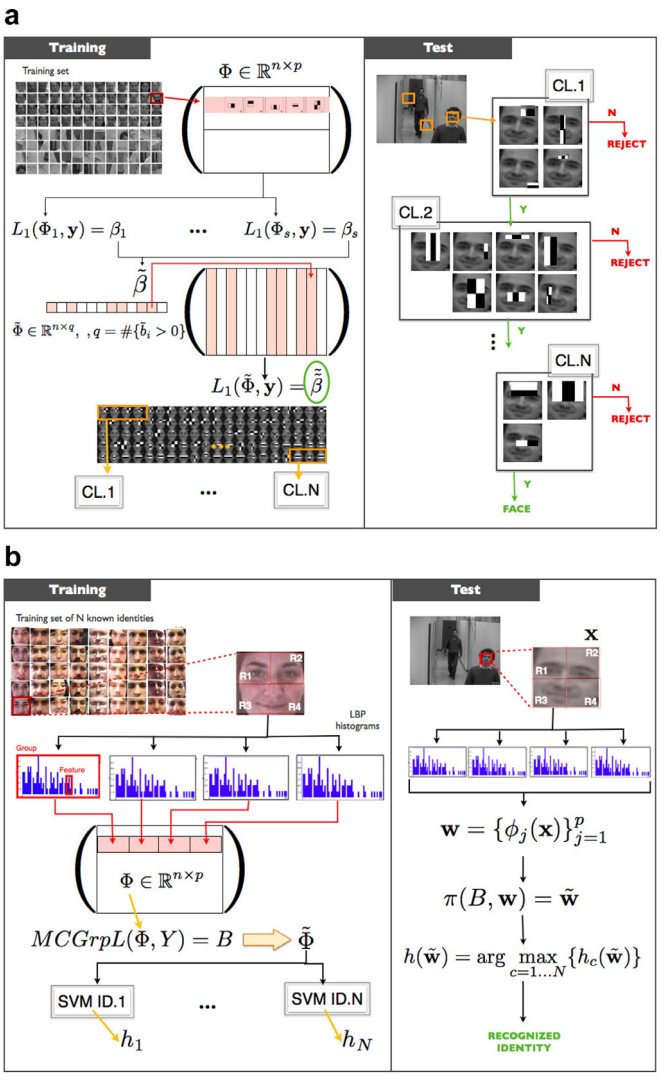

Fig. 2. Visual representations of face-based analysis. (a): Our pipeline for detecting faces on images is based on solving a binary classification problem using sparse face representations. Function L_1 implements Eq. (2). (b): In our method face recognition is addressed as a multi-class classification on data sparsified using structured feature selection. Function MCGrpL implements Multi-Class Group LASSO [39].

histogram) which need to be computed together (even if they have not been selected together). Thus, it is wise to select groups of correlated features rather than single features. The method we adopt to perform structured feature selection is Multi-Class Group-LASSO [39], which provides us with a set of features groups which are meaningful for all classes simultaneously, guaranteeing the benefit of computing a common description for all identities at run-time (instead of a diverse representation tailored for each subject).

Once we have found a sparse and meaningful representation for our multi-class problem, we train a multi-class classifier based on SVM and a *Winner-Takes-All* strategy [44]: the architecture is composed by N One-vs-All binary SVM classifiers where each image $\mathbf{x}_i$ is represented by mapping it on the sub-set of selected regions, by a projector $\pi(\mathbf{x}_i, B) = \{\phi_j(\mathbf{x}_i) | B_{j,\cdot} \neq 0\}$. Given the binary classifier of class c, the negative examples are randomly sampled from all the classes j, $j = \ldots N$, $j \neq c$. For a new datum $\mathbf{x}$, the resulting global classifier is then formulated as the argmax of the discriminant functions of the binary classifiers. The overall procedure may be enriched by a final temporal smoothing.

Experimental assessment. In this section we assess the modules with respect to the specific application setting we are considering. More general results and comparisons with the state of the art may be found in [33] and [39]. In Table 1, we report face detection results obtained on a dataset acquired in-house, where people appear in front of a camera moving freely. The test set includes 3000 images, the size of the training set is reported in the table. The table shows how we may achieve impressive performances with larger descriptors (of about 240 features), in particular if adopting a non linear kernel. If we want to reduce the information redundancy and speed up computation, we may obtain a sparser solution (of about 50 features) and still obtain comparable results. In this case, though, we have no benefit in choosing a non linear kernel. This is related to the fact that features are less correlated. Our method learns a representation from a relatively small dataset, if compared with classical choices, such as Adaboost schemes that may explore millions of negative examples [32]. Therefore we tested possible performance improvements for larger training sets. The results in the table show that in our case there is no benefit in enriching the training set size.

Table 1
Face detection performances – in-house dataset (see text).

Training Size (pos, neg)	# Sel. feat.	Kernel	EER	ER (FP = 5×10^{-3})
(2000,2000)	42	lin	0.0122	0.0621
(2000,2000)	240	lin	0.0119	0.04
(2000,2000)	42	poly (d = 2)	0.0120	0.0420
(2000,2000)	240	poly (d = 2)	0.0065	0.0078
(2000,20000)	53	lin	0.0134	0.0358

Table 2
Face detection performances under long time observations (5 hours acquisition).

# Sel. feat.	False positives	False negatives
42	4×10^{-7}	0.24
42 + 82	4×10^{-7}	0.18

Table 3
Summary of the performances of our method for face recognition on different datasets.

Dataset	# Identity	# Sel. feat.	Rec. Error
MOBO	24	26	0.031
		13	0.039
Choke Point	29	47	0.032
R309	12	45	0.204

We also performed a long term experiment by loosely annotating a 5 hours video. Table 2 shows the results obtained in this case. There is a degradation of the results due to the large data variability, but the performances are still good. Also we may notice an improvement if we add a post processing layer based on checking the presence of additional patterns (eyes and noses), obtained with an eye and nose detector equivalent to the face one.

The experiments on face recognition have been performed on three different datasets, two public benchmarks (appropriate for our case study)– *MOBO* [45] and *Choke Point* [46] – and a in-house dataset, *R309* [39]. The datasets present different variability in terms of image quality, lighting conditions, subjects appearance and pose. Table 3 reports a summary of the recognition rates of our method on the three datasets, showing very competitive performances. It is worth noting that such results are achieved with very sparse representations (from the order of 50k features to about 2.5k), favoring the computational efficiency. Notice moreover that the recognition rate for R309 dataset are significantly lower than for the other two datasets. This is due to the higher variability of the acquisitions, which have been made in an unconstrained environment, where the users were free to move, and considering a temporal span of some months. For these reasons,

Fig. 3. Sample frames from a video sequence where the subject speaks freely, moving closer and farther the video camera, and rotating her head.

such data are very reminiscent of the applicative scenario we have in mind in this work. Notice, however, in such a scenario we may count on the availability of image sequences, rather than single shots. Consequently, the analysis may be performed on more than one image, allowing for a statistics over time which is known to be beneficial for recognition purposes (as in [23,24]).

Fig. 3 shows examples of face detection and recognition, with the subject talking moving freely in the scene.

3.1.2. Detecting and reading characters

Natural scenes contain a large amount of information directed to humans which is conveyed through text. For these reasons in the last decades many "smart systems" tried to address the so-called OCR in unconstrained environments (see, for instance, [47–51]) in the attempt of bridging the gap between automatic document reading (which has largely been solved for printed latin characters) and a more generic automatic reading. In recent years, large improvements have been made in this direction. Important thematic conferences and challenges contributed enormously to promoting and stimulating research on relevant topics (see, for instance, the ICDAR series, which opened specific competitions on robust reading in both born-digital and real images).

On this respect, we developed a fast and general purpose text detection method to be applied on a wide set of (unconstrained) scenarios. The method is modular – in the sense it may accommodate more features and classifiers should they be needed – and portable on different architectures – since the implementation does not include any architecture-specific instruction.

Although the designed system is rather generic in its scope, we consider some specific goals, such as reading street names (which are often low contrast in many European cities), names on door bells (where illumination is poor), ingredients on food products (with some geometric deformation). Again, we also take in due account the users requirement of having a quick feedback from the system.

The method is composed by three main steps: image segmentation (integrated to features computation), text detection, and text recognition. In the following, we provide some details on the various steps.

Text features. We first discuss image segmentation and its efficient implementation. The latter includes the possibility of computing a set of features as segmentation takes place, in order to minimize the overall computational cost. Given the input image I we segment it to extract a set $C = \{c_i\}$ of connected components (or CoCos). In accordance with the state of the art findings [52], we adopt the Maximally Stable Extremal Regions (MSER) algorithm [53].

We adopt the linear implementation described in [54], which is based on a watershed procedure. The algorithm segments each image twice: first, we extract the connected components that can be obtained by thresholding the image as in $I \geq T$, with $0 \leq T \leq max_{val}$ (where max_{val} is the maximum gray level value), then we compute the connected components obtained by thresholding as $I < T$. The two procedures can be carried out in parallel. In order to minimize space and time computational cost, we do not store the list of coordinates of the connected component, but instead we compute directly features which will be useful for the following classification step. Thus, during image segmentation we compute and store:

1. *Pass index* (positive threshold i.e. $I \geq T$ or negative threshold i.e. $I < T$).
2. *Segmentation level*, that is the minimum (first segmentation pass) or the maximum (second segmentation pass) intensity value of a connected component
3. *CoCo seed*, coordinates of one contour pixel.
4. *Bounding box coordinates*, upper left corner and bottom right corner coordinates.
5. *Geometrical moments* up to the third order.

The first three elements are simply stored during the initialization of the connected component and are never updated during the segmentation. Both geometrical moments and bounding boxes can instead be computed efficiently in constant time as the segmentation proceeds.

Segmentation level, pass index and CoCo seed, allow us to navigate both the contour and the whole connected component later in the process, if needed. Indeed, to derive the elements of the connected component we can recursively visit all the pixels connected to the seed whose value is lower (higher) than the segmentation level. At each segmentation step we may have to add a pixel to a CoCo or to merge two CoCos. In the former case the bounding box coordinates are updated only if the coordinates of the new pixel are either smaller than the upper left corner or bigger than the lower right corner. The bounding box of the connected component can be used to compute in constant time the CoCo *aspect ratio* and, together with the number of pixels of the CoCo (moment M_{00}), the *compactness*.

The geometric moments are simple descriptors of a connected component shape. Geometric moments can be computed and updated efficiently, as when two CoCos are merged their geometric moments M^1 and M^2 can be summed up to compute the moments of the new component.

Text detection and recognition. To classify the connected components as text or non text elements, we adopt an architecture based on a cascade of linear and nonlinear SVM classifiers and filters (see Algorithm 1). Each level of the cascade filters the connected components using different feature vectors and classifiers. An element is classified as text only if it passes all the levels of the tests

The design of such a cascade is motivated by two reasons: first, it is computationally efficient, since it allows us to discard simple negatives samples with simple descriptions; second, it allows us to naturally combine different descriptions by placing them in different levels of the cascade:

- *Level I*: RBF SVM on a 2-dimensional input (aspect ratio and compactness). Both features are not rotation invariant, however if we assume that the characters are nearly horizontal, they are very effective for quickly discarding simple negatives.
- *Level II*: polynomial SVM (4th degree). It adopts normalized central moments up to the third order η_{ij}, computed from the geometrical moments. Normalized central moments can be computed efficiently and they are invariant to translation and scale. Therefore they are an effective choice to describe the CoCo shape for moderate rotation changes. To further reduce the computational cost of the SVM prediction we use an explicit mapping from the dual SVM formulation to the primal. This is computationally

Fig. 4. Text detection on outdoor scenes.

Algorithm 1 Character detection and recognition pipeline.

```
for all cc_i ∈ I do
    SVM({aspect ratio, compactness}_i,'rbf',σ_1) = f_1
    if f_1 > 0 then
        SVM({normalized central moments}_i,'poly','d = 4') = f_2
        if f_2 > 0 then
            SVM({LBP}_i,'linear') = f_3
            if f_3 > 0 then
                SVM({Zernike moments}_i,'linear') = f_4
                if f_4 > 0 then
                    SVM({asp.r., comp., NCM, LBP, Zernike}_i,'rbf', σ_2) =
                        f_5
                    if f_5 > 0 then
                        POSITIVE-CC-LIST ← cc_i
                    end if
                end if
            end if
        end if
    end if
    NMS(POSITIVE-CC-LIST )
    LINES-CC-LIST ← LINES(POSITIVE-CC-LIST )
    WORD-CC-LIST ← WORDS(LINES-CC-LIST )
end for
for all cs_i ∈ WORD-CC-LIST do
    MCSVM({asp. r., comp., NCM, LBP, Zernike}_i, 'rbf', σ_3) = class
end for
```

advantageous since the size of the input feature vector is small (7 features).

- *Level III*: linear SVM on the Local Binary Patterns (LBP) of the connected component. The connected component is a binary image, therefore LBPs simply encode different patterns on the connected component contour, while inner pixels do not carry useful information. Thus we scan contour elements starting from the seed and update a 255-dimensional LBP histogram, where each point of the contour is represented considering a 3×3 neighborhood. Within the same contour scan used to compute the LBP descriptor we also may compute the mean magnitude of the gradient to discard low contrast extremal regions.
- *Level IV*: linear SVM on the Zernike moments of the connected components. Zernike moments can be implemented efficiently by pre-computing their masks and multiplying them by the connected components binary matrix, resized to the same size of the mask, when needed. The mask size chosen in our work is 64×64. The overall size of the Zernike feature vector is 759.
- *Level V*: RBF SVM which uses all the feature vectors concatenated in a single description which is then normalized on a hypercube of side 2. The overall size of the descriptor is 1024 entries.

The detected characters are grouped in lines and words using considerations on their spatial vicinity. Then, a multi-class SVM using a standard one vs all scheme is used for character recognition. We train 72 different classifiers since we use different labels for upper and lower cases letters. Since we are not interested in distinguishing between upper and lower case characters, we removed from the negative set of a given uppercase (lowercase) character its corresponding lowercase (uppercase) samples. To optimize the computational performances, we employ linear classifiers on the whole feature vector obtained during the detection phase, thus we do not need to recompute features.

Table 4
Performances of the classification layers (cumulative performances combine the cascade performances up to the corresponding layer).

Layer	TPR	FPR	cumul. TPR	cumul. FPR
I	0.995	0.7608	0.995	0.7608
II	0.995	0.4673	0.990	0.3555
III	0.995	0.8344	0.985	0.2966
IV	0.995	0.5994	0.980	0.1778

Experimental assessment. In this section, we report an analysis of the method performances, considering its ability in detecting text. We consider three datasets of positive (text CoCos) examples: (i) *ICDAR2013*[9] contains characters with a ground truth provided with the ICDAR 2013 training dataset using both the natural and synthetic scenes; (ii) *CONTAINER*[10] is a private dataset of characters extracted from containers code images; (iii) *CHAR74K*[11] is a set of synthetic characters from the Char74k dataset. The negative examples (non text CoCos) have been extracted automatically from 192 full-hd images with no text. Since none of the listed dataset is an exhaustive representation of a general urban scenario, we also collected a set of 1737 images, which we used for a further evaluation of the algorithm. Overall, our quantitative analysis considers a training set of ~50K negatives and ~5.5K positive examples and a test set of ~74K negatives and ~3.4K positives sampled from the union of the datasets of above. For each layer of the classification cascade, with a KCV procedure, we select the parameter which achieves the lowest false positive rate fixed the true positive rate to 0.995. Table 4 reports the independent and cumulative classification performances of each of the first four classification layers. The equal error rate (EER) of the whole procedure is 0.961.

Fig. 4 shows sample results of character detection on a variety of outdoor settings. The images include night shots, occlusions, reflections, and a variety of fonts and sizes. All detection results are very accurate. Fig. 5 instead shows detection and automatic reading results on a variety of images shot indoors. The detection is confirmed to be very accurate, and automatic reading performs well, although there is room for improvement. It should be noticed though that we are simply performing character recognition, with no error correcting code based on a predefined dictionary.

[9] ICDAR 2013 datasets http://dag.cvc.uab.es/icdar2013competition.

[10] Dataset acquired for the project VIT - Vision for Innovative Transport - VII FP EU - SP4 Capacities Research for SMEs - n. 222199 http://www.vitproject.eu.

[11] http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

Fig. 5. Character detection and recognition on indoor environments.

3.2. Recognizing hand held objects

We now focus on a different setting, where the observed object is placed in the peripersonal space of the viewer. There are circumstances in which visually impaired people can understand the category of objects (e.g. a box), using alternative sensorial information (e.g. tactile), but cannot appreciate more specific visual properties (e.g. the fact that a box contains cereals of a given brand). Computer Vision systems may compensate by providing ways of discriminating between objects belonging to the same *broad* class. In this scenario, we consider the recognition of banknotes (also known as *Paper Currency Recognition* (PCR)) as a use case. We start by two observations that guided us in the design of this module. First, since the object is positioned in the peripersonal space of a subject, it is likely to occupy the majority of the image content. Second, we may assume the user through sensory substitution is certain that he is holding banknote, but cannot be sure of its value. Thus the aim of the system is to recognize this amount. This suggests us that the PCR task can be efficiently modeled as an image retrieval problem, skipping the object detection phase which we adopted in the previous sections and which is also used in the literature to address PCR. With this respect, it is worth citing methods available in the literature making use of neural networks (see, for instance, [55–57]), HMM [58], PCA [59] , or vector quantization [60].

3.2.1. Banknote recognition as an image retrieval problem

Today, a reference model for content-based image retrieval (CBIR) is Bag-of-Features (BoF) [61–63]. We adopt BoFs on SURF features [64], chosen for their computational efficiency. We consider an image gallery that includes the different types of Euro banknotes, lying on a plain flat background, on the two sides, at different angles. SURF features are extracted from each image and then used to estimate a vocabulary of visual words, or *visual dictionary*, via K-Means clustering. Each image is then associated with a global BoF descriptor – through *vector quantization* – which corresponds to the frequencies histogram of each visual word in the image.

Given a new *query image*, it is first represented according to the *visual dictionary*, then it is compared with all the images in the gallery. The obtained similarities are ranked, and a voting procedure is applied to the first k-positions of the ranking list. The estimated banknote value is identified as the one receiving the highest amount of votes. Fig. 6 shows the reference pipeline.

3.2.2. Experimental assessment

In this section, we report the experimental evaluation of our approach to PCR. We built a reference dataset of Euro banknotes – EUR 5 (first and second series), 10 (first and second series), 20 and 50 – composed of 96 images of 665 × 1182 pixels resolution. Images of both front and reverse sides of the banknotes have been acquired, also considering different viewpoint orientations (8 equally spaced angles between 0° and 315°). At the end, about 16 sample images have been available for each banknote class.

Table 5
Recognition rates (%) using different image descriptors. D is the size of the dictionary, d the length of the feature vector, $d = 64$ for SURF features.

Descriptors	Descriptor size	Average rec. rate
BoF [61]	D	0.97
MBoF [66]	$4 \times D$	0.96
VLAD [67]	$d \times D$	0.92
FISHER [68]	$2 \times d \times D$	0.94
SPM [69]	$21 \times D$	0.82

For performance evaluation, we also acquired a dataset of 340 query images, with about 36 to 90 images for each class of banknote. The images cover a wide variety of conditions, such as partial occlusion, blur, rotation, illumination and viewpoint changes. Differently from other datasets in the literature, typically collected by using scanners or under constrained conditions, our data thus represent a reasonable approximation of real world scenarios.

For building the BoF representations we adopted the Euclidean Distance; the number k of ranked positions considered in the voting strategy was set to 12. Before using them in the retrieval procedure, we normalize all the image descriptors by the so-called power-law normalization [65]. This normalization method processes the output image vector $v = (v_1, \ldots, v_D)$ as $v_i = \mid v_i \mid^{\beta} \times sign(v_i)$, with $0 \leq \beta < 1$ a fixed constant. The updated vector v is then L2-normalized. We fixed $\beta = 0.1$ in all experiments.

We evaluate the use of SURF features in combination with different image descriptors, reporting in Table 5 the retrieval performance on the query dataset.

The BoF method achieves the highest overall recognition rate. It is worth mentioning that, in general, neural network systems achieved recognition rate no larger than 95% (see e.g. [57,70]) with more complex and time consuming procedures.

4. Low level features for 3D scene (environment) interpretation

Distributed bioinspired architectures, which resort to populations of tuned neurons, can be used to estimate low level visual features, such as optic flow and disparity, from image sequences. Such features

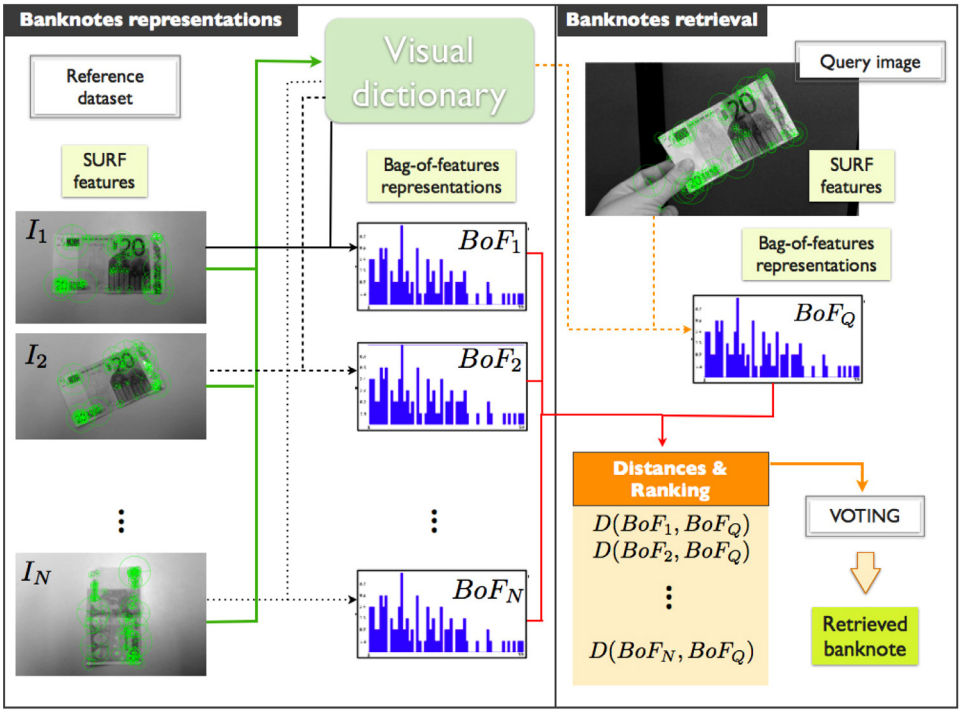

Fig. 6. Our pipeline for banknote recognition based on a CBIR approach. Reference images are represented using SURF features and a BoF vector making use of a pre-computed dictionary. During the test phase, a query image is represented in the same way, and it is compared with all the reference representations. A ranking on the comparisons combined with a voting strategy allows us to retrieve the most similar banknote.

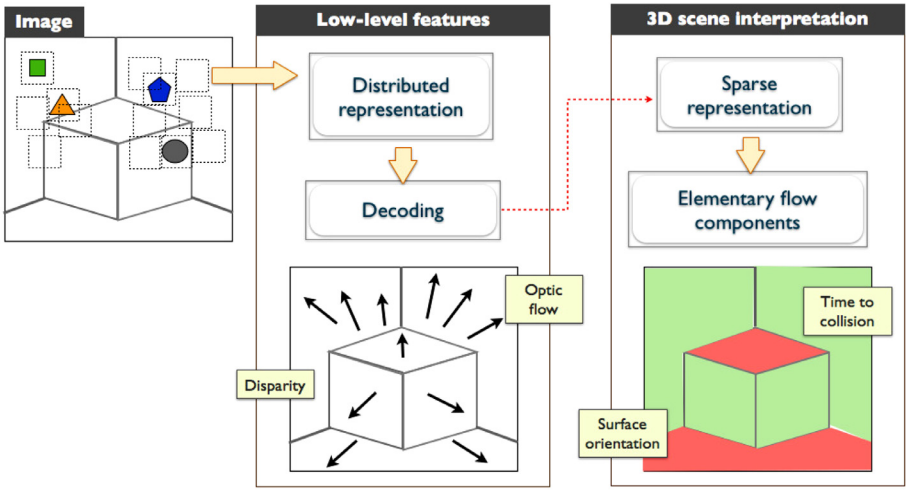

Fig. 7. A visual representation of low-level features and 3D scene interpretation. Low-level visual features are encoded by a network of neurons, such distributed representation can be decoded in order to compute the motion and depth information of the scene. Such visual features are afferent to higher layers, where a sparse representation (i.e. image locations of interest) allow us to estimate 3D descriptors of the scene.

are the basis of the 3D scene interpretation. Fig. 7 shows a sketch of the proposed neural architecture that mimics the dorsal visual pathway: the cortical areas V1 and MT (primary visual cortex and Middle Temporal area, respectively) provide the computation of low-level visual features, then higher visual areas, such as Medio Superior Temporal (MST) area, process complex visual patterns of optic flow (i.e. elementary flow components) in order to estimate properties of the environment, such as time to contact or time to collision (TTC) [4].

The literature includes several approaches to design networks of tuned cells and to properly combine their responses, in order to compute reliable features from the visual signal [71]. In distributed representations (i.e. population codes), the visual information is encoded by the activity pattern of hierarchical layers of neural units. The first layers are composed of simple and complex cells. The simple neural units can be approximated by a bank of band-pass filters [72] tuned to different features of the visual signal (e.g. spatial orientation and frequency). The complex units perform non-linear operations, such as squaring and maximum, over their afferents, i.e. the responses of the simple cells [73]. The selectivity to a specific feature (e.g. elemental vision attributes, such as direction of motion and binocular disparity [74]) emerges, in the next layer, from the combination of the responses of populations of cells from the previous layer.

The intra- and inter-layers processing of neural architectures are based on a set of *canonical neural computations* [75,76], which can be applied to solve different visual tasks. Such canonical computations,

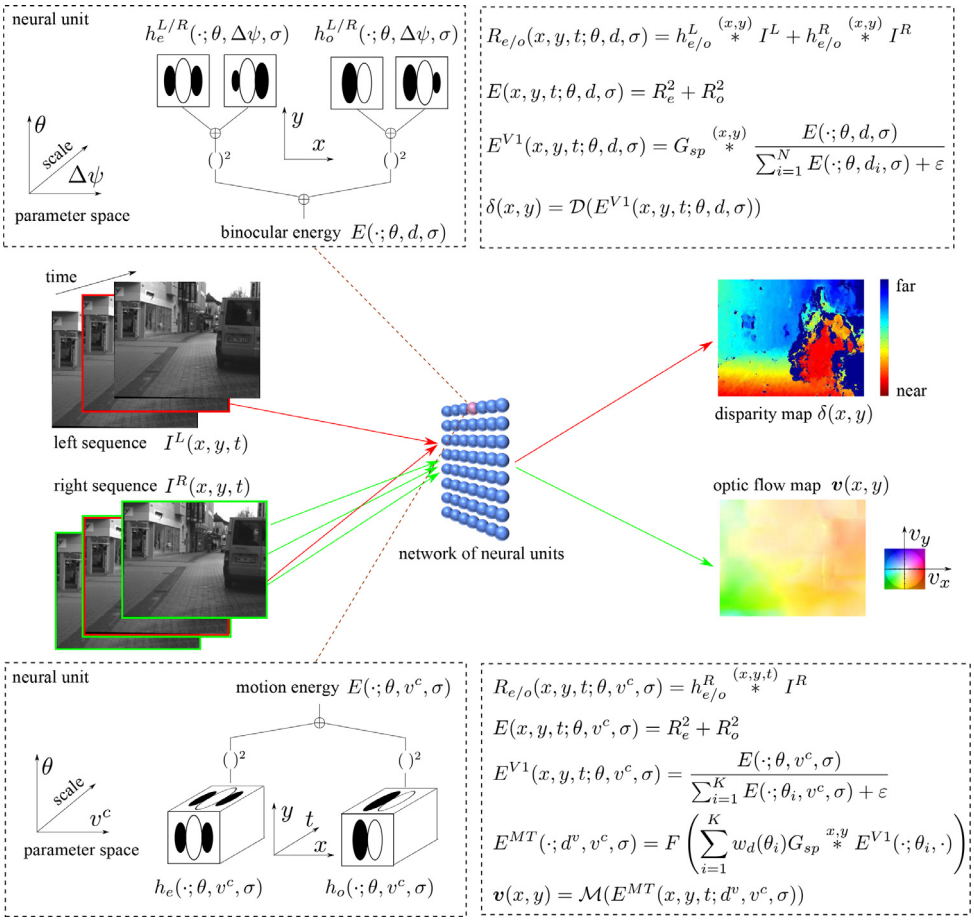

Fig. 8. The neural architecture for the computation of low level visual features. The input visual sequences are processed by a network of neural units that can estimate optic flow and disparity maps. (Top) On the left side, a neural circuit for the estimation of binocular energy and the related parameter space of the cells are sketched. The neural processing is described by the equations reported on the right side (see text for details). (Bottom) The motion energy, on which the optic flow computation is based, is shown on the left side, and the mathematical description of the hierarchical processing is on the right side.

which describe the functional models of the visual cortex, are summarized in the following:

- *Linear filtering.* The weighted sum performed by the receptive fields (RFs) of neural populations are approximated by the processing of filter banks. Such a processing is used to describe the neural responses in different areas of the visual system, e.g. the V1 [77] and the MT [78] areas.
- *Energy model.* It is a squaring operation applied on pairs of neurons characterized by the same spatial orientation tuning, but with different binocular phase tuning or with different spatial and temporal profiles. Such model describes the invariance that can be observed in the complex cells of *V*1 [73,79].
- *Divisive normalization.* The mechanism normalizes the response of a neural population by the summed activity of a pool of neurons characterized by a tuning to a specific parameter. Such a computation is used in area V1 [80], and in area MT [78], where it explains the non-linear properties of neurons. The divisive normalization is also used to remove noise from the responses of a neural population, thus improving the quality of the encoded information [81].
- *Soft-thresholding.* It allows obtaining narrow tuning curves in order to maintain sensor selectivity. Moreover, such a feed-forward mechanism can be considered as a valid alternative to the lateral inhibition mechanisms [82].
- *Pooling.* This mechanism performs a weighted sum of a group of neurons characterized by the same tuning parameter. In particular, such a mechanism allows us to mitigate uncertainty in the neural representation due to the ambiguous local representation of the visual signal of each neuron [83].

4.1. Implementation details

The canonical neural computations have been mainly used by considering simple experimental visual stimuli, in this paper our aim is to use them in real-world environments. To handle the frequency range of natural scene, a multi-scale approach has been adopted: in particular, we use a pyramidal decomposition [84] in order to diminish the computational load of the visual processing. We consider a coarse-to-fine refinement [85] for combining the estimates of the visual features among the pyramidal scales. Moreover, the spatio-temporal filters are decomposed into separable filters in space and time, in order to obtain a low computational load.

The choice of using bioinspired distributed architecture is due to the natural description of the corresponding algorithms in terms of parallel computations, thus exploiting the GPU architectures of nowadays mobile devices.

4.2. Motion (optic flow)

To estimate optic flow, we implement a feed-forward neural model, which is based on a three-layer hierarchical architecture that mimics the functional properties of V1 and MT visual cortical areas. The neural architecture for the computation of optic flow is shown in Fig. 8 (bottom).

Each neural unit is composed of two simple cells, which are described by even and odd RFs, $h_e(x,y,t;\theta,v^c,\sigma)$ and

$h_e(x, y, t; \theta, v^c, \sigma)$ respectively, characterized by a spatial orientation θ in the x-y domain, and by a preferred component velocity (speed) v^c in the x_θ-t domain, i.e. in the direction orthogonal to its preferred orientation [86]. The even and odd RFs are defined by using the following filters:

$$g(x, y; \theta, \sigma, \psi) = Be^{\left(\frac{-(x^2+y^2)}{2\sigma^2}\right)} e^{j2\pi(f_s cos(\theta)x + f_s sin(\theta)y + \psi)}, \quad (3)$$

$$p(t; f_t) = e^{\left(-\frac{t}{\tau}\right)} e^{j2\pi(f_t t)}, \quad (4)$$

where f_s and f_t denote the peak spatial and temporal frequencies of the RF (filter), σ and τ define the spatial and temporal support, respectively, and B is a normalization term. In particular, the even RF (similarly for the odd one, as described in [87]) is defined as $h_e(x, y, t; \theta, v^c, \sigma) = g_o(x, y; \theta, \sigma, \psi) p_e(t; f_t) + g_e(x, y; \theta, \sigma, \psi) p_o(t; f_t)$, by considering the real and imaginary components of the complex filters as g_o, g_e, p_e and p_o. The tuning of a cell for a specific speed is described by $v^c = f_t / f_s$ by setting $\psi = 0$.

The population response $E(x, y, t; \theta, v^c, \sigma)$ of complex cells is obtained according to the motion energy model [73]. The responses $E^{V1}(x, y, t; \theta, v^c, \sigma)$ of the V1 cells (obtained through a divisive normalization) are first pooled on the orientations (also on a spatial neighborhood by using a Gaussian G_{sp} function) through a set of MT linear weights $w_d(\theta)$, then a static non-linearity $F(\cdot)$ is applied. This gives rise to the responses $E^{MT}(x, y, t; d^v, v^c, \sigma)$ of the population of MT pattern cells [78,88], tuned to different speed v^c and direction d^v of the speed. The population of MT cells encodes a distributed representation of the velocity $\boldsymbol{v}$ of the stimulus. We adopt a linear approach $\mathcal{M}(E^{MT}(x, y, t; d^v, v^c, \sigma))$ to decode the MT population response [89,90]. The weights of such a linear decoding can be computed by using two possible approaches: to learn them from examples, or to analytically deduce them. Here, we consider the latter approach: in particular, first we decode the MT responses along each direction d^v to compute the speed, then we linearly weight such estimated velocities. The actual implementation of the neural architecture is described by the Algorithm 2.

Algorithm 2 Population code for optic flow computation.

```
for all scale do
    for all θ do                                   ▷ V1 Population coding
        for all v^c do
            LINEAR FILTERING
            MOTION ENERGY MODEL
            SPATIAL POOLING AND DIVISIVE NORMALIZATION
            SOFT-THRESHOLDING
        end for
    end for
    for all v^c do                                 ▷ MT Population coding
        POOLING ON θ
        STATIC NON-LINEARITY
    end for
    for all d do                                   ▷ MT Population decoding
        LINEAR COMBINATION OF v^c
    end for
    LINEAR COMBINATION OF d^v
    if scale > 1 then                              ▷ Multi-scale and coarse-to-fine
        WARPING AND MERGING OF THE ESTIMATES
    end if
end for
```

The proposed algorithm for optic flow computation has been described in [87], where the authors performed a quantitative evaluation with respect to the state of the art approaches for optic flow estimation. Here, we report the evaluation, in terms of Average Angular Error (AAE) and End Point Error (EPE), by using the standard Middlebury dataset [91] (see Table 6).

Table 6
Error measurements on Middlebury optic flow training set.

Sequence	AAE ± STD	EPE ± STD
Grove2	4.28 ± 10.25	0.29 ± 0.62
Grove3	9.72 ± 19.34	1.13 ± 1.85
Hydrangea	5.96 ± 11.17	0.62 ± 0.96
RubberWhale	10.20 ± 17.67	0.34 ± 0.54
Urban2	14.51 ± 21.02	1.46 ± 2.13
Urban3	15.11 ± 35.28	1.88 ± 3.27

Fig. 12 (third row) shows three sample optic flow maps computed for a real-world scene (the *Town* sequence), acquired by a moving camera mounted on a car moving in a pedestrian area. Both the camera and other objects in the scene are moving independently[12].

4.3. Stereo (disparity)

We consider a network of binocular neurons, tuned to specific values of spatial orientation θ, frequency f_s and phase difference $\Delta\psi$ between the left and right RFs of a cell, in order to compute disparity maps. The neurons' tuning to the binocular disparity is based on physiological and modeling studies [79,92], which showed that a difference in the phase of the left and right spatial RF profiles of a V1 binocular simple cell is able to encode the disparity information (*phase-shift-model*).

The neural architecture for the computation of binocular disparity is shown in Fig. 8(top). The input stereo image sequences are processed by a large network of simple binocular cells by considering ψ^L and ψ^R for the left and right $h_e^{L/R}(x, y; \theta, \Delta\psi, \sigma)$ RFs, which are obtained from Eq. (3) ($\Delta\psi = \psi^L - \psi^R$, as described in [93]). Simple cells' responses $R_{e/o}(x, y, t; \theta, d, \sigma)$ are then combined into a binocular energy neuron $E(x, y, t; \theta, d, \sigma)$ [92]; such neural unit (complex cell) sums the squared responses of a quadrature pair of binocular simple neuron. The tuning to disparity is defined as $d = \Delta\psi / f_s$.

Given the distributed representation provided by the population of tuned neurons, the estimates of the disparity present in the visual signal can be decoded through $\mathcal{D}(\cdot)$ by considering the activity $E^{V1}(x, y, t; \theta, d, \sigma)$ of the population in a neighborhood of each location.

The actual implementation of the neural architecture is described by the Algorithm 3, where the interaction among the different layers and the canonical neural computations that are embedded in the population of neurons are highlighted.

Algorithm 3 Population code for disparity computation.

```
for all scale do
    for all θ do                                   ▷ Population coding
        for all Δψ do
            LINEAR FILTERING
            BINOCULAR ENERGY MODEL
            DIVISIVE NORMALIZATION
            SOFT-THRESHOLDING
        end for
    end for
    for all θ do                                   ▷ Population decoding
        CENTER OF MASS over Δψ
    end for
    LINEAR COMBINATION OF θ
    if scale > 1 then                              ▷ Multi-scale and coarse-to-fine
        WARPING AND MERGING OF THE ESTIMATES
    end if
end for
```

[12] The sequence has been recorded in the context of the EU DRIVSCO project, courtesy of HELLA, Hueck KG, Lippstadt.

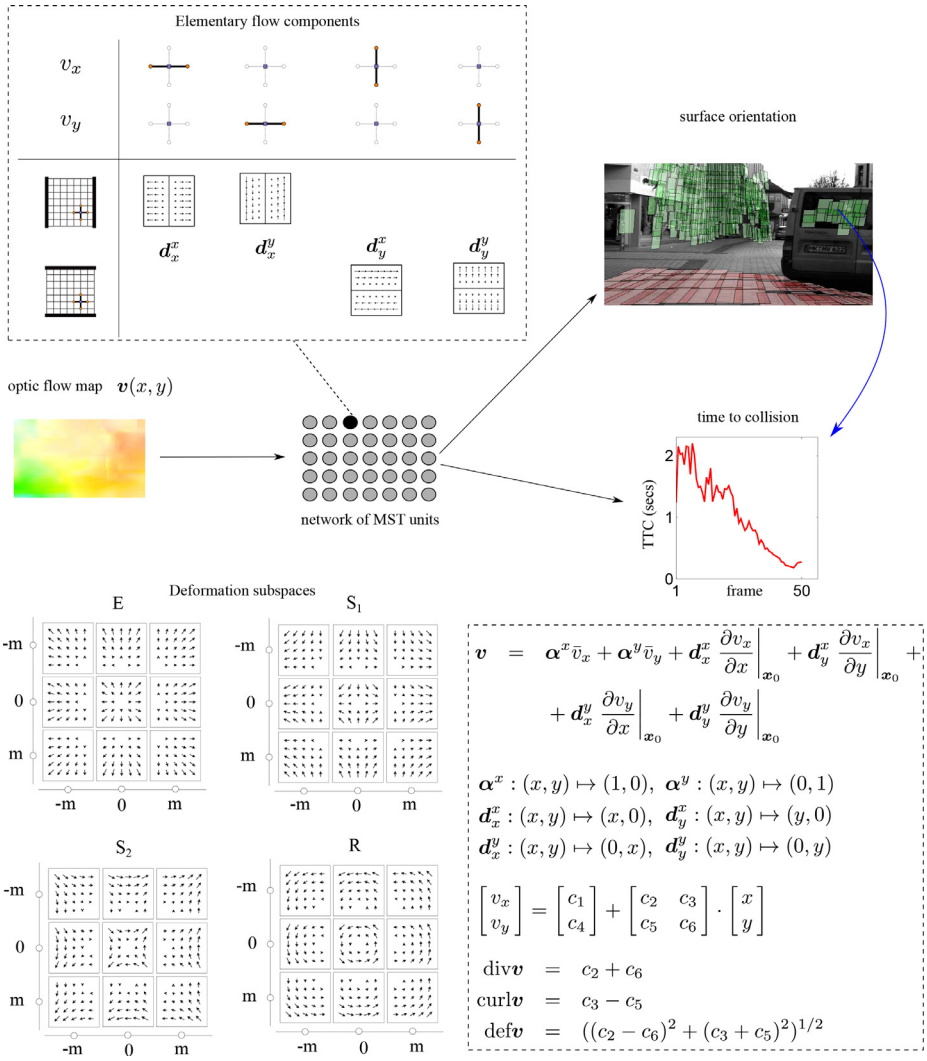

Fig. 9. The architecture for 3D scene interpretation. The afferent optic flow $\boldsymbol{v}(x,y)$ (i.e. the output of the previous layer, see Fig. 8) is processed by using elementary flow components (d_x^x, d_y^x, d_x^y, d_y^y) in order to estimate the affine coefficients (c_1, c_2, c_3, c_4, c_5, c_6) that are related to the differential invariants (div$\boldsymbol{v}$, curl$\boldsymbol{v}$, def$\boldsymbol{v}$). The surface orientation and the time to collision can be computed by using such differential invariants. The deformation subspaces can be obtained by combination of EFCs, which can be generated from basic lattice interaction schemes.

Table 7
Error measurements on Middlebury Stereo Evaluation (Version 2).

Image pair	All	Nonocc
Tsukuba	12.28	10.48
Venus	8.05	7.11
Teddy	30.54	17.58
Cones	24.30	14.19

The proposed algorithm for disparity computation has been described in [93]. Here, we report the evaluation of the algorithm, in term of percent of bad pixels (i.e. the percent of pixels whose error is greater than 1 pixel with respect the ground truth), by using the standard Middlebury dataset [94] (see Table 7).

By considering non occluded regions (Nonocc) only, the average percent of bad pixels is 12.34. Although the approach does not reach the top performances of the first algorithms in the website table[13], it is better than some state of the art methods (e.g. [95–97]).

Moreover, given the availability of the modern parallel computing architectures (e.g. GPUs), it is possible to obtain a fast and near real-time implementation of the proposed algorithm (see [98] for details about the GPU implementation of a preliminary version of this model).

5. Obstacle avoidance and 3D scene (environment) interpretation

The information about the 3D structure of the world and about the time to contact (TTC) has important consequences for the survival of living beings, and for their interaction with both the inanimate and animate objects in their environments. In the literature, many studies have shown that motion is a very powerful cue for 3D dynamic scene understanding. In particular, smooth variations in speed give rise to the perception of rigid 3D surfaces, while speed discontinuities to depth discontinuities [99]. As a consequence, it appears very probable that there are several neural mechanisms that have evolved to compute this information. For example, specific mechanisms for the estimation of TTC can be found in birds for the avoidance of rapidly approaching objects such as predators.

The collision avoidance is of vital importance for humans, too. Psychological research on collisions began with the seminal work

[13] http://vision.middlebury.edu/stereo/eval/.

by Gibson and Crooks [100], and progressed in several directions. Though experimental evidences show the importance of TTC evaluation also for humans, the problem of finding the neural mechanisms has not been addressed in higher mammals and especially in primates [101], yet. The perception of the orientation in depth of the surfaces has been addressed both in neuroscience [99] and in psychophysics [102]. Evidence for the selectivity for 3D structure (from motion) has been found in both MT and MST cells. Many cells in these cortical areas are sensitive to speed gradients, thus to the orientation of the surfaces. This supports the view that orientation of surfaces is represented in the dorsal stream of the visual pathway. Such physiological observations fit the psychophysical evidence [103], which suggests that the first-order representation, i.e. local orientation of surfaces, is one of the depth representations used by the human visual system.

Motion flow fields can be described in terms of their linear decompositions, based on their first-order properties. From this perspective, local spatial features of flow field at a given location, can be of two types: (1) the average flow velocity at that location, and (2) the structure of the local variation in a neighborhood of same location. The former relates to the smoothness constraint, i.e. to structural uniformity; the latter is related to linearity constraint, i.e. to structural gradients [104]. Velocity gradients provide important cues about the 3D structure of the environment. Formally, they can be described by a first-order Taylor decomposition, around the image point $x_0 = (x_0, y_0)$: $v = \bar{v} + \bar{\mathbf{T}}x$, where $\bar{v} = v|_{x_0} - \bar{\mathbf{T}}x_0$ and

$$\bar{\mathbf{T}} = \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} \end{bmatrix}_{x_0} . \tag{5}$$

By breaking down the tensor in its dyadic components, the motion field can be locally described through two-dimensional maps ($\boldsymbol{g}$: $R^2 \mapsto R^2$) representing elementary flow components (EFCs) (see Fig. 9).

The maps $\boldsymbol{\alpha}^x$ and $\boldsymbol{\alpha}^y$ are pure translations, and the maps d^x_x, d^x_y, d^y_x, d^y_y represent cardinal deformations, basis of a linear deformation space. In this way, we have four classes of deformation gradients (see Fig. 9): one stretching (v^i_i) and one shearing (v^i_j) for each cardinal direction, e.g. $v^x_x = a_1\alpha^x + a_2 d^x_x = m_1$, see [105] for details. The EFCs can be combined to obtain deformation subspaces representing elementary deformations such as expansion (E), shear (S_1 and S_2) and rotation (R), see Fig. 9. Such a choice gives to the model maximum flexibility: every gradient deformation within a single class will be built through the same recurrent network, just by changing its driving inputs on the basis of direct local measures on the input optic flow. The description of the optic flow through the EFCs can be interpreted in terms of an affine model. The affine coefficients (c_1, c_2, c_3, c_4, c_5, c_6, see Fig. 9) can be computed by using several approaches (e.g. [106,107]). Here, we use an adaptive pattern matching that is based on the EFCs [105].

5.1. Surface orientation from optic flow

The orientation of a surface with respect to the line of sight is usually represented by two angles called *slant* σ, and *tilt* τ [108]. The slant of a surface σ is defined as the angle between the normal to the surface and the line of sight. The tilt τ specifies the direction in which the surface is slanted and is defined as the angle between the x-axis of the image plane and the projection into the image plane of the normal to the surface. From these angles, and by considering a planar approximation of the surfaces in a small area around a given point, it is possible to derive the normal $\mathbf{n} = (n_1, n_2, n_3)$ to such a planar surface, where $n_1 = \sin\sigma\cos\tau$, $n_2 = \sin\sigma\sin\tau$ and $n_3 = \cos\sigma$. As described in [105], the descriptors of the surface orientation can be computed

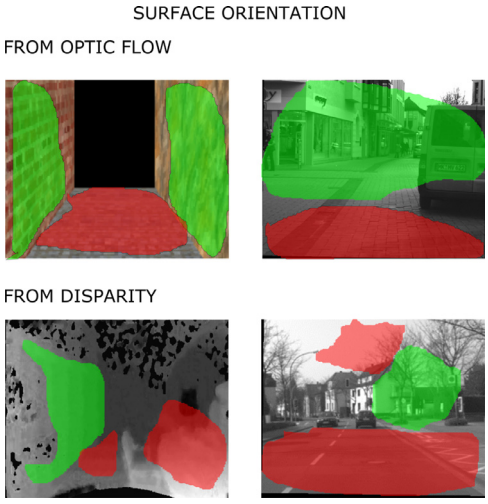

Fig. 10. Surface orientation estimation from optic flow (top) and disparity (bottom). Top left: a frame from a virtual scene. Top right: a frame from the automotive *Town* sequence. Bottom left: a frame from a virtual scene. Bottom right: a frame from the automotive *City* sequence. Horizontal (red) and vertical (green) surfaces estimated from the affine description of optic flow and disparity are overlapped to each frame. The *Town* and *City* sequences have been recorded in the context of the EU DRIVSCO project (courtesy of HELLA Hueck KG, Lippstadt). They are recorded by a camera mounted inside a car moving in urban and suburban contexts. The spatial resolution of each frame is 640 × 512 pixels. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

from the optic flow affine coefficients. In particular, the relative motion between an observer and the environment is described as a rigid-body motion through the translational velocity $\mathbf{T} = (T_X, T_Y, T_Z)^T$, and the angular velocity $\Omega = (\Omega_X, \Omega_Y, \Omega_Z)^T$. The relationship between the differential invariants (div$\boldsymbol{v}$, curl$\boldsymbol{v}$, def$\boldsymbol{v}$), which can be described in terms of the affine coefficients, and the behaviorally quantities for scene interpretation can be defined through the viewer translation $\mathbf{A}$ and the surface orientation $\mathbf{F}$, see [105] for details. Thus, we can summarize the complete procedure to estimate surface orientation by following Algorithm 4.

Algorithm 4 Computation of surface orientation.

for all *scale* **do**
  SUBSAMPLE THE OPTIC FLOW
  for all points of interest in the image **do**
    CALCULATE THE DIFFERENTIAL INVARIANTS
    $\text{div}\mathbf{v} = \frac{2T_Z}{D} + \mathbf{F}\cdot\mathbf{A}$
    $\text{curl}\mathbf{v} = -2\Omega_Z + |\mathbf{F}\cdot\mathbf{A}|$
    $\text{def}\mathbf{v} = |\mathbf{F}||\mathbf{A}|$
    RELATE F TO σ AND τ
    $|\mathbf{F}| = \tan\sigma$
    $\angle\mathbf{F} = \tau$
    COMPUTE $\mathbf{n} = (n_1, n_2, n_3)$
  end for
end for

Fig. 10 (top row) and Fig. 12 (third row, right) show an estimation of the surface orientation by using the affine description of the optic flow. In this paper, we are not interested in providing an accurate 3D reconstruction of the surfaces, as it shown in Fig. 9, since a main user requirement is to maintain a low computational cost. A further requirement is to control the amount of information to be delivered to users. Thus we divide the scene surfaces into two classes: horizontal (e.g. ground plane, represented in red in the figure) and vertical (represented in green) surfaces.

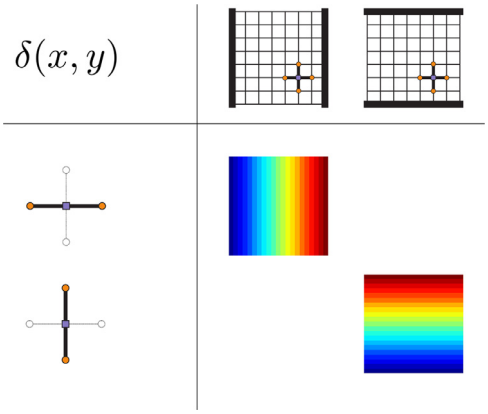

Fig. 11. Elementary disparity components. The basic lattice interconnection schemes, which generate the disparity gradients along the cardinal axes (the color code is the same used for the disparity maps), are shown.

5.2. Surface orientation from disparity

In a similar way we have proposed for the optic flow, we can estimate the orientation of the surfaces by using a single stereo pair. As we have previously described (see Section 4.3), disparity map $\delta(x, y)$ is a function of two variables and it can be linearly approximated as $\delta(x, y) = \delta_0 + \nabla\delta \cdot \bar{x}$, where $\nabla\delta = (\partial\delta/\partial x, \partial\delta/\partial y)$. By considering the affine description, the linear approximation of the disparity map is a plane $\delta(x, y) = c_1 + c_2 x + c_3 y$, where $c_2 = \partial\delta/\partial x$ and $c_3 = \partial\delta/\partial y$ are the basis (elementary components) of the linear representation (see Fig. 11). Fig. 12 (second row, right) shows an estimation of the surface orientation obtained by using disparity. Such information can be used to validate the surface orientation from optic flow, and to add the distances of the surfaces with respect the observer by using a calibrated stereo camera.

5.3. Time to collision from optic flow

The TTC corresponds to the map of the temporal distance between the observer and *any* point in the scene. The idea of employing TTC from first-order derivatives of the optical flows goes back to the early 90s [109–111]. In general, such methods share the drawback of being sensitive to errors in the estimates of optical flow. As an alternative, families of simple fixed flow divergence templates have been proposed in an attempt to overcome the problems associated with the computation of image velocity derivatives. Along this line, Meyer [112] proposes a technique for applying the theoretical analysis of [113] in realistic situations.

In this work, we propose an alternative to the method presented in [112], which works on the image velocity field, while avoiding an explicit differentiation of the optic flow: the affine coefficients are obtained from the matching with the *adaptive templates* defined in Section 5, working on the optic flow obtained in Section 4.2.

A relative motion between an observer and an object induces a corresponding image motion field, which is divergent in nature when the object moves towards the observer. From it, we can derive an estimate of the TTC with the object in the field of view of the moving observer. More generally, computing the TTC is a complex problem that implies, in principle, solving in advance the structure from motion problem and especially separating the translational and rotational components of relative rigid motion. Though, it has been observed that under proper simplification assumptions, at least bounding values of the TTC can be directly related to the first-order differential invariants of the image flow by simple algebraic relationships: $\frac{2}{\mathrm{div}\mathbf{v}+\mathrm{def}\mathbf{v}} \leq t_c \leq \frac{2}{\mathrm{div}\mathbf{v}-\mathrm{def}\mathbf{v}}$.

The actual implementation of the approach to compute TTC is described by the Algorithm 5.

Algorithm 5 Computation of TTC.

for all *scale* **do**
  SUBSAMPLE THE OPTIC FLOW
  for all points of interest in the image **do**
    CALCULATE THE DIFFERENTIAL INVARIANTS
    CALCULATE div**v** AND def**v**
    if def**v** ~ 0 **then**
      $t_c = \frac{2}{\mathrm{div}\mathbf{v}}$
    end if
  end for
end for

Fig. 12 (first row, right) shows the time to collision for a pedestrian crossing a street.

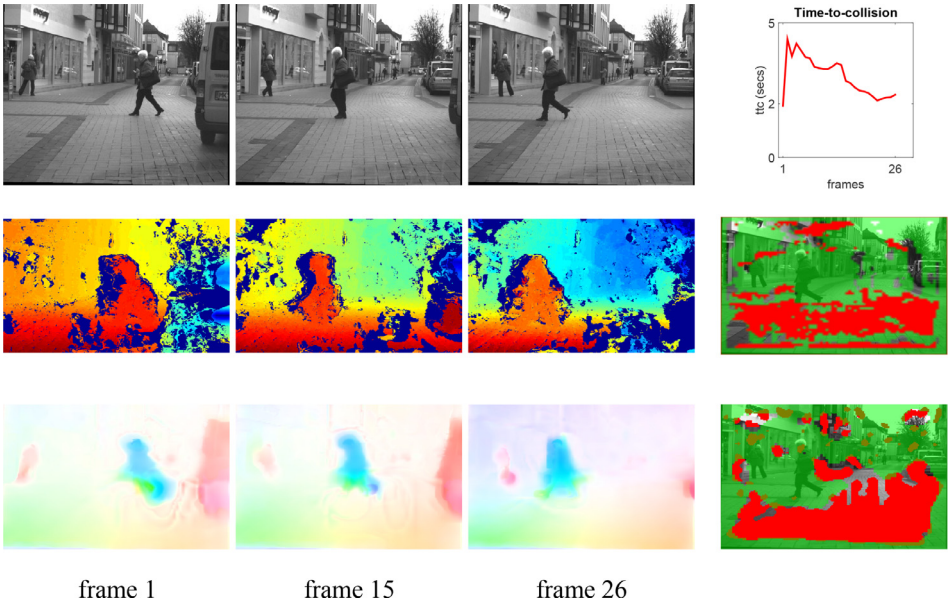

Fig. 12. Low-level features and 3D scene interpretation for the *Town* sequence. First row: three frames of the selected sequence and the time to collision for the pedestrian. Second row: the disparity maps, which show the distances in the scene, and the surface orientation (red and green code for horizontal and vertical surface, respectively) computed by using disparity information. Third row: the optic flow and the related estimation of the surface orientation.

Fig. 13. Some snapshots of the video acquired for the use case "Indoor navigation". The spatial resolution of the images is 1280 × 720 pixels. Frame 1 and 60 show that there are no constraints on the position of the acquisition device, indeed the horizontal lines of the scene are not parallel with respect to the image frame. Frame 150, 190 and 385 show the structure of the considered scene.

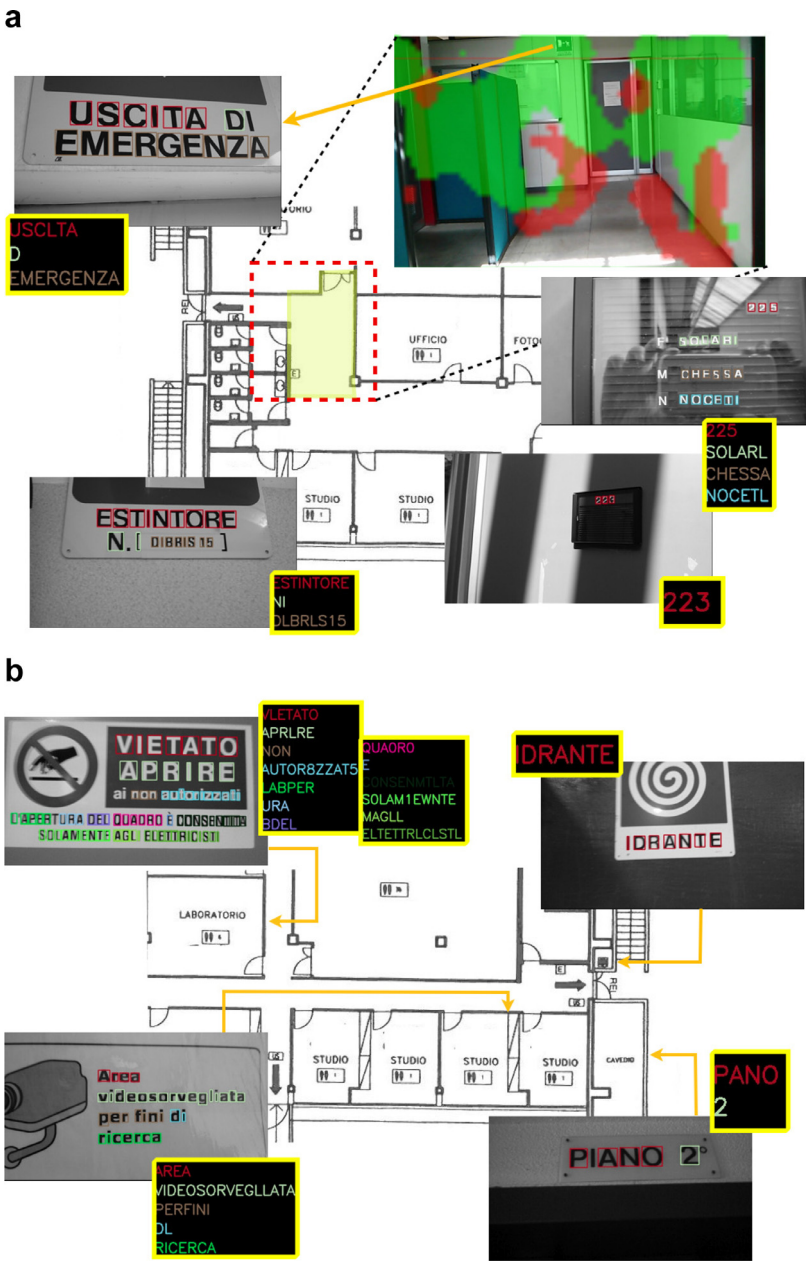

Fig. 14. Examples of integrated functionalities combining the estimation of the surface orientations and the detection and recognition of text. (a): The presence of main vertical structures (in green) suggests regions of interest where performing text detection. (b): Further examples of detected text and recognized characters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

6. Towards an integrated assistive visual system

In this section, we consider the main concepts behind a complex artificial vision framework for assisting visually impaired users. Our main goal is to discuss the benefits of integrating different functionalities, focusing in particular on the ones described in the previous sections. The general aim is to guide the user during the navigation in a real-world scene, building a coarse representation of the 3D structure and associating semantic tags with it, which may indicate obstacles or suggest the presence of objects of potential interest.

In the following, we first provide some general considerations for the design of such a system, then we show qualitative results of particular use cases we have chosen as proof of concept of the integrated functionalities.

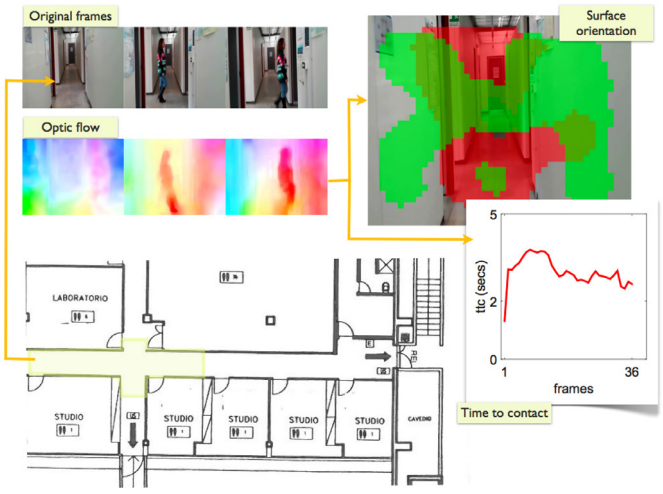

Fig. 15. Example of use of the optical flow estimate to compute a coarse representation of the scene structure and the time to contact with respect to a subject moving in front of the observer.

6.1. Assistive vision: some considerations

The application scenarios we have considered throughout the paper call for complementary aspects, which should be considered when designing a prototype system.

Technical specifications. In the following we briefly summarize the main issues:

- *Hardware and sensors*. The acquisition setup should be based on the use of portable devices that a user can easily bring with himself during daylife activities. With this respect, issues related to the design of such devices (e.g. the ergonomicity) should be taken into account. From the software standpoint, we have to consider that the computational power provided by portable devices is still limited, while the battery consumption is still an open issue.
- *Image quality*. Partially correlated to the previous point, a main issue is the fact that portable devices usually have poor optics (while the image resolution may be adequate) and limited capabilities in terms of computational power. Moreover, we need to consider that for a visually impaired user is in fact impossible to precisely direct and focus the camera towards the object of interest. Even when the camera is pointed in the correct direction there might be a problem with the distance of observation, and with the presence of egomotion effects on the acquired images. Depending on the task, a possible solution, still a challenge of research nowadays, is to equip the user with wearable devices coupled with non-visual feedback helping him/her to adjust the viewpoint and so favor the image analysis.
- *The need for real-time non-visual feedback*. Since the considered methods are expected to assist users during daily activities, the presence of real-time feedback from the system becomes of fundamental importance to guarantee the achievement of the desired functionalities. Also, such feedback should be devised so to be delivered through non-visual channels, as using audio stimuli. This implies that the outcomes of the modules should be easily summarized by sensorial substitutions, which in turns suggests not to recover detailed dense features that should be difficult to be represented through non-visual signals.
- *System usability and feedback from the users*. An important aspect that should guide the design of an assistive system is the feedback on the usability provided by final users.

What has been done. We now analyze our specific choices, with a reference to the previous list.

- Our work focused so far on the design and the development of software modules, thus we did not make any specific hardware choice. At the same time, to mimic the quality of data in a perspective prototype system, we process videos acquired by mobile phones, hand held by a user moving freely in the reference environment. Therefore, our data are characterized by a high variability with respect to background clutter, lightening conditions and amount of egomotion.
- As pointed out earlier, our experiments have been performed on video streams acquired by a user moving freely in the scene. In this way, we have been able to appreciate the challenges of motion blur, camera jitter, partial objects framing. Most of the times, temporal continuity attenuates the effects of these sources of noise. Indeed, in the “What” stream, voting methods can help us obtaining a smooth feedback out of multiple temporally adjacent observations. The “Where” stream benefits of the temporal

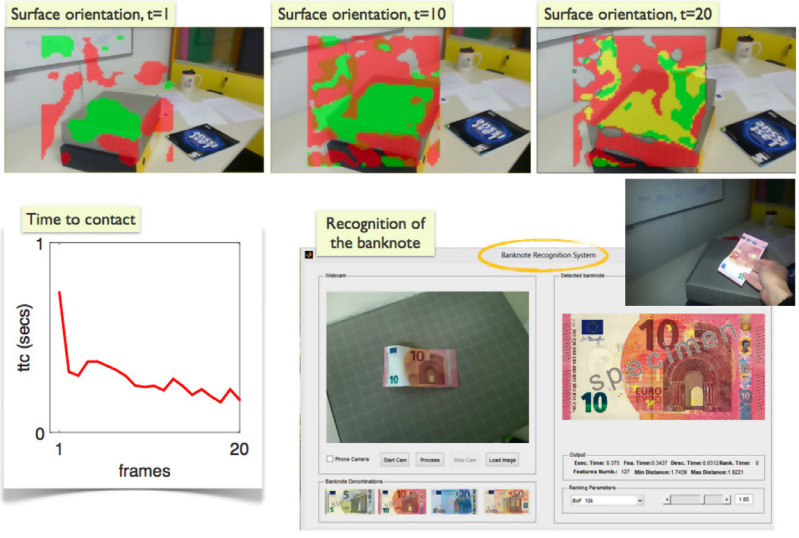

Fig. 16. An example of laboratory situation that mimics a set of functionalities a user may require at a counter desk. The correct distance and orientation of the surface, on which the user can place the banknote, are obtained analyzing the surface normals and the time to contact. Once that the banknote has been placed on the correct surface the recognition is achieved. The spatial resolution of the images is 1920 × 1080 pixels.

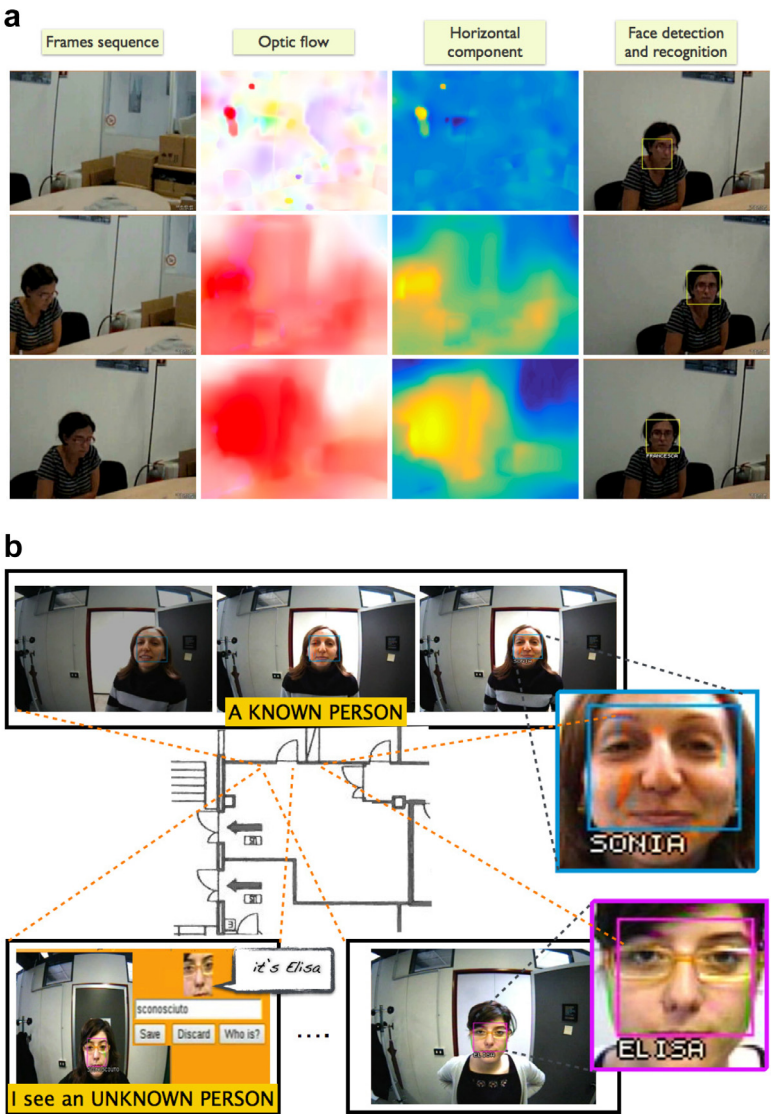

Fig. 17. Examples of use cases exploiting face detection and recognition functionalities. (a): Obstacle detection guides face detection and recognition. (b): A simple simulation of a system for controlling the access on a given environment. The spatial resolution of the input images for the described modules is 320 × 240 pixels.

continuity, since optic flow estimation is performed through a temporal filtering, and surface orientation estimation and time to contact are continuously updated from previous estimates.

- Although we did not consider the important issue of providing appropriate feedback to the user, our main concern throughout the work has been to design computationally efficient modules.

6.2. Use cases

We now restrict our attention on specific use cases to discuss the validity of our conceptual framework. We consider an indoor navigation scenario, with a user freely moving in a few areas of the building of our department. For each use case we start by describing the reference scenario.

Use case 1 - Indoor navigation. *The user is walking along a corridor of an unfamiliar building. She is looking for room number 223, which is somewhere in the corridor. The video acquired by her mobile is summarized in Fig. 13. The corridor is narrow, and few textures are present on the walls and on the floor. The illumination of the scene is quite common in indoor office buildings: a mixed presence of natural illumination (including shades of light) and artificial one.*

The optical flow is computed to extract coarse information on the main horizontal and vertical structures in the scene. Such cues guide the detection of objects of interest, only on regions of the images potentially including such objects (the room number will be written on vertical structures, colored in green in the figure). In Fig. 14 we depict instances of such functionalities. For clarity, the output of the software modules are overlapped with the map of the considered environment. The estimation of the main surface orientation suggests the presence of walls (vertical surfaces highlighted in green) over which text detection can be performed. Examples of detected and recognized characters are reported. Once text is detected, a screen shot is presented to the users with the detected text highlighted with colored rectangles. Characters belonging to the same word are associated with the same color.

While the user is walking down the corridor, she is approaching a person moving towards the right (see Fig. 15) . Optical flow, combined with the surface estimation, also allows us to compute the time to contact with respect to obstacles – in this case, the person crossing the corridor.

Use case 2 - At the counter. *A user is approaching a counter desk in a supermarket, envisaging she requires to pay and thus to recognize*

the amount of a banknote. She extracts a banknote from her wallet and places it on the flat surface in front of her.

To simulate the situation in a laboratory, we consider a video sequence depicting a simple scene where a box (the counter desk) has been placed on a table. Fig. 16 shows a visual sketch of the use case, including some sample frames and a screenshot of the recognition software where the correct retrieval result can be appreciated. In this case, the estimation of the surface normals allows the system to notice the presence of flat surfaces (in green). Also, the system guides the correct point of view to favor the subsequent recognition (in yellow at time $t = 20$), and the correct distance (thanks to the time to contact estimation). The user now places the banknote on the flat surface and banknote recognition takes place.

Use case 3 - Social interaction. *A user is about to walk in a room where she is not aware of the presence of other people. She feels uncomfortable and aware of her disability. Then she switches on the face detection module and enters the room. The face detection module informs the user of the presence of another person in the scene. Immediately after, she is also informed that the person is known to the user.*

Fig. 17(a) shows an example with a video sequence (some sample frames are reported on the first column) acquired by a user from her point of view while having a look at a desk. The analysis of the optical flow suggests the presence of a potential obstacle (the regions associated with the highest intensity in the second column, corresponding to the images on the first one). A more accurate segmentation can be achieved by considering the horizontal component only, which, as an add-on, provides coarse properties on the camera motion (in this example mostly a translation). After that a region of interest has been identified, face detection is performed only on it, thus obtaining detections which are more accurate as the observed subject modifies her head pose. Finally, face recognition is achieved.

The same technology may be used to implement a smart video doorphone, and to control the access to the user home. In the lab, we simulate such a situation by placing a camera in the vicinity of a room entrance. Fig. 17(b) shows a visual representation of this setting. Face detection highlights the presence of a person in the observed scene. Face recognition provides further info on the identity of the subject. In case the person is correctly recognized, the system provides an output of his identity. If the subject has never been seen before by the system, the user is able to enroll the new identity on the system, so that future recognition can be possible.

7. Conclusion

In this paper, we described a framework composed of different Computer Vision modules meant to assist visually impaired users. We proposed a selection of functionalities, which cover wide scope issues, such as semantic understanding and the analysis of a 3D environment. Such functionalities have first been developed independently, and their effectiveness for the specific application scenario has been discussed on benchmark data. Then, we analyzed the benefits of integrating complementary functionalities in the same framework, by considering appropriate use cases that exemplify the actions of a visually impaired user visiting an unfamiliar building or socially engaging with others. Integration has the clear benefit of centralizing different functions on the same system, but it also (and more importantly) produces enhanced information and improved performances.

The devised framework paves the way to a system, which may follow constantly the visually impaired user in the daily activities. In principle, this would also be a challenging test bed for life long learning, where the computational models are expected to be stable over time and to adapt coherently to environment changes and to novel scenarios.

Future works will also be devoted to enrich the computational model as well as to the development of a working prototype. With a reference to the technical specifications listed in Section 6 on the *hardware and sensors*, we will consider a system composed of sensorized glasses and a mobile device. Some modules will be engineered to run directly on the mobile phone (the text recognition module is already running on Android phones) or on remote servers on the cloud. The use of stereo camera will be considered also in relation to the benefits for the visual processing and to the possible issues in terms of computational power and battery consumption.

As for the *real-time feedback* to users, work will be devoted in designing simple and usable interfaces, which trigger different functionalities for instance via verbal commands. Concerning this specific aspect, but more in general all the aspects related to the usability of the prototype, the advice of user groups will be crucial. Taking inspiration from the analysis reported in [27], for the specific case of face recognition, we plan to provide a thorough investigation on the users feedback, collecting the point of view of both visually impaired people and their relatives. This will guide us in possibly adapting the current implementation with respect to the users requirements, providing the correct type of human-machine interface, but it may also suggest the presence of preferred tasks or new missing functionalities.

References

[1] E. Hill, J. Rieser, M. Hill, J. Halpin, R. Halpin, How persons with visual impairments explore novel spaces: strategies of good and poor performers, J. Vis. Impair. Blind. 87 (8) (1993) 295–301.
[2] M.A. Goodale, D.A. Westwood, An evolving view of duplex vision: separate but interacting cortical pathways for perception and action, Curr. Opin. Neurobiol. 14 (2) (2004) 203–211.
[3] T. Serre, A. Oliva, T. Poggio, A feedforward architecture accounts for rapid categorization, Proc. Nat. Acad. Sci. 104 (15) (2007) 6424–6429.
[4] P.J. Mineault, F.A. Khawaja, D.A. Butts, C.C. Pack, Hierarchical processing of complex motion along the primate dorsal visual pathway, Proc. Nat. Acad. Sci. 109 (16) (2012) 972–980.
[5] M. Bousbia-Salah, M. Bettayeb, A. Larbi, A navigation aid for blind people, J. Intel. Robot. Syst. 64 (3–4) (2011) 387–400.
[6] M. Salerno, M. Re, A. Cristini, G. Susi, M. Bertola, E. Daddario, F. Capobianco, Audinect: an aid for the autonomous navigation of visually impaired people, based on virtual interface, Int. J. Human Comput. Interact. 4 (1) (2013) 25–33.
[7] X. Chen, A.L. Yuille, Detecting and reading text in natural scenes, in: Proceedings of IEEE CVPR, vol. 2, 2004, pp. 366–373.
[8] P. Silapachote, J. Weinman, A. Hanson, R. Weiss, M.A. Mattar, Automatic sign detection and recognition in natural scenes, in: Proceedings of IEEE Work on Computer Vision Application for the Visually Impaired, 2005, p. 27.
[9] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. Wu, A. Ng, Text detection and character recognition in scene images with unsupervised feature learning, in: Proceedings of IEEE ICDAR, 2011, pp. 440–445.
[10] K. Wang, B. Babenko, S. Belongie, End-to-end scene text recognition, in: Proceedings of IEEE ICCV, 2011, pp. 1457–1464.
[11] P. Sanketi, H. Shen, J. Coughla, Localizing blurry and low-resolution text in natural images, in: Proceedings of IEEE WACV, 2011, pp. 503–510.
[12] M. Mattar, A. Hanson, E. Learned-Miller, Sign classification using local and meta-features, in: Proceedings of IEEE CVPRW, 2005, p. 26.
[13] J. Coughlan, R. Manduchi, Color targets: fiducials to help visually impaired people find their way by camera phone, J. Image Video Process. (2007) 10–23. Special Issue on Image and Video Processing for Disability.
[14] H. Bagherinia, R. Manduchi, Robust real-time detection of multi-color markers on a cell phone, J. Real-Time Image Process. 8 (2) (2011) 1–17.
[15] B. Tjan, P. Beckmann, R. Roy, N. Giudice, G. Legge, Digital sign system for indoor wayfinding for the visually impaired, in: IEEE CVPRW, 2005.
[16] M.S. Uddin, T. Shioyama, Bipolarity and projective invariant-based zebra-crossing detection for the visually impaired, in: Proceedings of IEEE CVPRW, 2005, p. 22.
[17] J. Coughlan, H. Shen, A fast algorithm for finding crosswalks using figure-ground segmentation, in: Proceedings of ECCVW, 5, 2006.
[18] V. Ivanchenko, J. Coughlan, H. Shen, Real-time walk light detection with a mobile phone, in: Computers Helping People with Special Needs, in: *LNCS*, vol. 6180, 2010, pp. 229–234.
[19] V. Pradeep, G. Medioni, J. Weiland, Robot vision for the visually impaired, in: Proceedings of IEEE CVPRW, 2010, pp. 15–22.
[20] M.A. Hersh, R. Farcy, R. Leroux, A. Jucha, Electronic travel aids and electronic orientation aids for blind people: technical, rehabilitation and everyday life points of view, in: Proceedings of Conference on Assistive Technology for Vision and Hearing Impairment, 2006, p. 12.
[21] P. Narasimhan, Trinetra: assistive technologies for grocery shopping for the blind, in: Proceedings of IEEE Symposium on Research in Assistive Technologies, 2007, pp. 147–148.

[22] T. Winlock, E. Christiansen, S. Belongie, Toward real-time grocery detection for the visually impaired, in: Proceedings of IEEE CVPRW, 2010, pp. 49–56.
[23] X. Liu, A camera phone based currency reader for the visually impaired, in: Proceedings of ACM Conference on Computers and accessibility, 2008, pp. 305–306.
[24] Z. Solymár, A. Stubendek, M. Radványi, K. Karacs, Banknote recognition for visually impaired, in: Proceedings of European Conference on Circuit Theory and Design, 2011, pp. 841–844.
[25] S. Krishna, G. Little, J. Black, S. Panchanathan, A wearable face recognition system for individuals with visual impairments, in: Proceedings of ACM Conference on Computers and accessibility, 2005, pp. 106–113.
[26] L. Gade, S. Krishna, S. Panchanathan, Person localization using a wearable camera towards enhancing social interactions for individuals with visual impairment, in: Proceedings of ACM International Workshop on Media Studies and Implementations That Help Improving Access to Disabled Users, 2009, pp. 53–62.
[27] L. Balduzzi, G. Fusco, F. Odone, S. Dini, M. Mesiti, A. Destrero, A. Lovato, Low-cost face biometry for visually impaired users, in: Proceedings of IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications, 2010, pp. 45–52.
[28] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: a literature survey, ACM Comput. Surv. 35 (4) (2003) 399–458.
[29] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cognit. Neurosci. 3 (1) (1991) 71–86.
[30] P.N. Belhumeur, J.P. Hespanha, D.J. Kriengman, Eigenfaces vs. sherfaces: recognition using class specific linear projection, IEEE PAMI 19 (7) (1997) 711–720.
[31] A. Mohan, C. Papageorgiou, T. Poggio, Example-based object detection in images by components, IEEE PAMI 23 (4) (2001) 349–361.
[32] P. Viola, M.J. Jones, Robust real-time face detection, IJCV 57 (2) (2004) 137–154.
[33] A. Destrero, C. De Mol, F. Odone, A. Verri, A sparsity enforcing method for learning face features, IEEE Trans. Image Process. 18 (2009) 188–201.
[34] M. Yang, D. Zhang, Y. Jian, D. Zhang, Robust sparse coding for face recognition, in: Proceedings of IEEE CVPR, 2011, pp. 625–632.
[35] G. Zhang, X. Huang, S. Li, Y. Wang, X. Wu, Boosting local binary pattern (LBP)-based face recognition, in: *LNCS*, 3338, 2005, pp. 179–186.
[36] H. Xiangsheng, S. Li, Y. Wang, Jensen–Shannon boosting learning for object recognition, in: Proceedings of IEEE CVPR, vol. 2, 2005, pp. 144–149.
[37] J. Shelton, G. Dozier, K. Bryant, J. Adams, K. Popplewell, T. Abegaz, K. Purrington, D. Woodard, K. Ricanek, Genetic based LBP feature extraction and selection for facial recognition, in: Proceedings of ACM Southeast Regional Conference, 2011, pp. 197–200.
[38] B.A. Olshausen, D.J. Fieldt, Sparse coding with an overcomplete basis set: a strategy employed by v1, Vis. Res. 37 (1997) 3311–3325.
[39] L. Zini, N. Noceti, G. Fusco, F. Odone, Structured multi-class feature selection with an application to face recognition, Pattern Recognit. Lett. 55 (2015) 35–41.
[40] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc., Ser. B 67 (2005) 301–320.
[41] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. B 58 (1) (1996) 267–288.
[42] T. Ahonen, A. Hadid, M. Pietikäinen, Face recognition with local binary patterns, in: Proceedings of ECCV, vol. 3021, 2004, pp. 469–481.
[43] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, PAMI 24 (7) (2002) 971–987.
[44] B.J. Jain, F. Wysotzki, A competitive winner-takes-all architecture for classification and pattern recognition of structures, in: Proceedings of International Conference on Graph based Representations in Pattern Recognition, 2003, pp. 259–270.
[45] R. Gross, J. Shi, The CMU Motion of Body Database, Technical Report, Robotics Institute, Carnegie Mellon University, 2001.
[46] Y. Wong, S. Chen, S. Mau, C. Sanderson, B.C. Lovell, Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition, in: Proceedings of IEEE CVPRW, 2011, pp. 74–81.
[47] N. Ezaki, M. Bulacu, L. Schomaker, Text detection from natural scene images: towards a system for visually impaired persons, in: Proceedings of IEEE ICPR, vol. 2, 2004, pp. 683–686.
[48] L. Neumann, J. Matas, Real-time scene text localization and recognition, in: Proceedings of IEEE CVPR, 2012, pp. 3538–3545.
[49] L. Zini, A. Destrero, F. Odone, A classification architecture based on connected components for text detection in unconstrained environments, in: Proceedings of IEEE AVSS, 2009, pp. 176–181.
[50] Y. Shao, C. Wang, B. Xiao, Y. Zhang, L. Zhang, L. Ma, Text detection in natural images based on character classification, in: Advances in Multimedia Information Processing, 2011, pp. 736–746.
[51] P. Shivakumara, R.P. Sreedhar, T.Q. Phan, S. Lu, C.L. Tan, Multioriented video scene text detection through Bayesian classification and boundary growing, IEEE Trans. Circuits Syst. Video Technol. 22 (8) (2012) 1227–1235.
[52] A. Shahab, F. Shafait, A. Dengel, ICDAR 2011 robust reading competition challenge 2: reading text in scene images, in: Proceedings of IEEE ICDAR, 2011, pp. 1491–1496.
[53] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image Vis. Comput. 22 (10) (2002) 761–767.
[54] D. Nistér, H. Stewénius, Linear time maximally stable extremal regions, in: Proceedings of IEEE ECCV, 2008, pp. 183–196.
[55] N. Jahangir, A.R. Chowdhury, Bangladeshi banknote recognition by neural network with axis symmetrical masks, in: Proceedings of IEEE International Conference on Computer and Information Technology, 2007, pp. 1–5.
[56] J.-K. Lee, S.-G. Jeon, I.-H. Kim, Distinctive point extraction and recognition algorithm for various kinds of euro banknotes, Int. J. Control Autom. Syst. 2 (2004) 201–206.
[57] F. Takeda, L. Sakoobunthu, H. Satou, Thai banknote recognition using neural network and continues learning by DSP unit, in: Proceedings of Knowledge-Based Intelligent Information and Engineering Systems, 2003, pp. 1169–1177.
[58] H. Hassanpour, P.M. Farahabadi, Using hidden Markov models for paper currency recognition, Expert Syst. Appl. 36 (6) (2009) 10105–10111.
[59] A. Ahmadi, S. Omatu, T. Fujinaka, T. Kosaka, A reliable method for classification of bank notes using artificial neural networks, Artif. Life Robot. 8 (2) (2004) 133–139.
[60] T. Kosaka, S. Omatu, T. Fujinaka, Bill classification by using the LVQ method, in: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, vol. 3, 2001, pp. 1430–1435.
[61] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Proceedings of IEEE ECCV, vol. 1, 2004, p. 22.
[62] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Proceedings of IEEE CVPR, vol. 2, 2006, pp. 2161–2168.
[63] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: Proceedings of IEEE CVPR, 2007, pp. 1–8.
[64] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, in: Proceedings of IEEE ECCV, 2006, pp. 404–417.
[65] F. Perronnin, Y. Liu, J. Sánchez, H. Poirier, Large-scale image retrieval with compressed fisher vectors, in: Proceedings of IEEE CVPR, 2010, pp. 3384–3391.
[66] J. Sosa-Garcia, F. Odone, Mean BoF per quadrant - simple and effective way to embed spatial information in bag of features, in: Proceedings of VISAPP, 2015, pp. 297–304.
[67] H. Jegou, M. Douze, C. Schmid, P. Perez, Aggregating local descriptors into a compact image representation, in: Proceedings of IEEE CVPR, 2010, pp. 3304–3311.
[68] J. Sanches, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the fisher vector: theory and practice, IJCV 105 (3) (2013) 222–245.
[69] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of IEEE CVPR, 2009, pp. 1794–1801.
[70] F. Takeda, S. Omatu, High speed paper currency recognition by neural networks, IEEE Trans. Neural Netw. 6 (1) (1995) 73–77.
[71] A. Pouget, P. Dayan, R.S. Zemel, Inference and computation with population codes, Annu. Rev. Neurosci. 26 (2003) 381–410.
[72] J. Daugman, Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, J. Opt. Soc. Amer. A/2 (1985) 1160–1169.
[73] E. Adelson, J. Bergen, Spatiotemporal energy models for the perception of motion, J. Opt. Soc. Amer. 2 (1985) 284–321.
[74] E. Adelson, J. Bergen, The plenoptic and the elements of early vision, in: M. Landy, J. Movshon (Eds.), Computational Models of Visual Processing, MIT Press, 1991, pp. 3–20.
[75] M. Kouh, T. Poggio, A canonical neural circuit for cortical nonlinear operations, Neural Comput. 20 (6) (2008) 1427–1451.
[76] M. Carandini, D.J. Heeger, Normalization as a canonical neural computation, Nat. Rev. Neurosci. 13 (1) (2012) 51–62.
[77] J.A. Movshon, I.D. Thompson, D.J. Tolhurst, Spatial summation in the receptive fields of simple cells in the cat's striate cortex, J. Physiol. 283 (1978) 53–77.
[78] E.P. Simoncelli, D.J. Heeger, A model of neuronal responses in visual area MT, Vis. Res. 38 (5) (1998) 743–761.
[79] D. Fleet, H. Wagner, D. Heeger, Neural encoding of binocular disparity: energy models, position shifts and phase shifts, Vis. Res. 36 (12) (1996) 1839–1857.
[80] D.J. Heeger, Normalization of cell responses in cat striate cortex, Vis. Neurosci. 9 (2) (1992) 181–197.
[81] S. Deneve, A. Pouget, P. Latham, Divisive normalization, line attractor networks and ideal observers, in: Advances in Neural Processing Systems, The MIT Press, 1999, pp. 104–110.
[82] N.J. Priebe, D. Ferster, Inhibition, spike threshold, and stimulus selectivity in primary visual cortex, Neuron 57 (4) (2008) 482–497.
[83] B.S. Webb, T. Ledgeway, F. Rocchi, Neural computations governing spatiotemporal pooling of visual motion signals in humans, J. Neurosci. 31 (13) (2011) 4917–4925.
[84] E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, J.M. Ogden, Pyramid methods in image processing, RCA Eng. 29 (6) (1984) 33–41.
[85] E.P. Simoncelli, Course-to-fine estimation of visual motion, in: Proceedings of IEEE Workshop on Image and Multidimensional Signal Processing, 1993, pp. 128–129.
[86] E.H. Adelson, J.A. Movshon, Phenomenal coherence of moving visual patterns, Nature 300 (5892) (1982) 523–525.
[87] F. Solari, M. Chessa, N. Medathati, P. Kornprobst, What can we expect from a v1-mt feedforward architecture for optical flow estimation? Signal Process.: Image Commun. (2015).
[88] J.H. Maunsell, D.C. Van Essen, Functional properties of neurons in middle temporal visual area of the macaque monkey. I. Selectivity for stimulus direction, speed, and orientation, J. Neurophysiol. 49 (5) (1983) 1127–1147.
[89] A. Pouget, K. Zhang, S. Deneve, P.E. Latham, Statistically efficient estimation using population coding, Neural Comput. 10 (2) (1998) 373–401.
[90] K.R. Rad, L. Paninski, Information rates and optimal decoding in large neural populations, in: Proceedings of NIPS, 2011, pp. 846–854.

[91] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, R. Szeliski, A database and evaluation methodology for optical flow, IJCV 92 (1) (2011) 1–31.
[92] I. Ohzawa, G. DeAngelis, R. Freeman, Stereoscopic depth discrimination in the visual cortex: neurons ideally suited as disparity detectors, Science 249 (1990) 1037–1041.
[93] M. Chessa, V. Bianchi, M. Zampetti, S.P. Sabatini, F. Solari, Real-time simulation of large-scale neural architectures for visual features computation based on GPU, Netw.: Comput. Neural Syst. 23 (4) (2012) 272–291.
[94] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, IJCV 47 (1/2/3) (2002) 7–42.
[95] S. El-Etriby, A.K. Al-Hamadi, B. Michaelis, Dense depth map reconstruction by phase difference-based algorithm under influence of perspective distortion, Mach. Graph. Vision Int. J. 15 (3) (2006) 349–361.
[96] L. Nalpantidis, A. Gasteratos, Stereo vision for robotic applications in the presence of non-ideal lighting conditions, Image. Vis. Comput. 28 (6) (2010) 940–951.
[97] N. Manap, J. Soraghan, Disparity refinement based on depth image layers separation for stereo matching algorithms, J Telecom. Elec. Comp. Eng. 4 (1) (2012) 51–64.
[98] M. Chessa, G. Pasquale, Graphics processing unit-accelerated techniques for bio-inspired computation in the primary visual cortex, Concurr. Comput.: Pract. Exp. 26 (10) (2014) 1799–1818.
[99] D.K. Xiao, V.L. Marcar, S.E. Raiguel, G.A. Orban, Selectivity of macaque MT/V5 neurons for surface orientation in depth specified by motion, Eur. J. Neurosci. 9 (5) (1997) 956–964.
[100] J.J. Gibson, L.E. Crooks, A theoretical field-analysis of automobile-driving, Am. J. Psychol. 51 (1938) 453–471.
[101] M. Lappe, Building blocks for time-to-contact estimation by the brain, in: Time-to-contact, in: Advances in Psychology, vol. 135, 2004, pp. 39–52.
[102] J.A. Saunders, B.T. Backus, Perception of surface slant from oriented textures, J. Vis. 6 (9) (2006) 882–897.
[103] J. Koenderink, A. van Doorn, A. Kappers, Surface perception in pictures, Percept. Psychophys. 52 (1992) 487–496.
[104] J. Koenderink, Optic flow, Vis. Res. 26 (1) (1986) 161–179.
[105] M. Chessa, F. Solari, S.P. Sabatini, Adjustable linear models for optic flow based obstacle avoidance, CVIU 117 (6) (2013) 603–619.
[106] Y. Yacoob, M. Black, Parameterized modeling and recognition of activities, CVIU 73 (2) (1999) 232–247.
[107] T. Nir, A.M. Bruckstein, R. Kimmel, Over-parameterized variational optical flow, IJCV 76 (2) (2008) 205–216.
[108] A. Witkin, Recovering surface shape and orientation from texture, Artif. Intell. 17 (1981) 17–45.
[109] M. Subbarao, Bounds on time-to-collision and rotational component from first-order derivatives of image flow, Comput. Vis. Graph. Image Process. 50 (3) (1990) 329–341.
[110] M. Tistarelli, G. Sandini, On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow, Trans. PAMI 15 (4) (1993) 401–410.
[111] R. Nelson, J. Aloimonos, Obstacle avoidance using flow field divergence, Trans. on PAMI 11 (10) (1989) 1102–1106.
[112] F. Meyer, Time-to-collision from first-order models of the motion field, IEEE Trans. Robot. Autom. 10 (6) (1994) 792–798.
[113] M. Subbarao, A. Waxman, Closed form solutions to image flow equation for planar surfaces in motion, Comput. Vis. Graph. Image Process. 36 (2/3) (1986) 208–228.