Available online at www.sciencedirect.com
ScienceDirect
Computer Vision and Image Understanding 108 (2007) 196–203
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Robust real-time vision for a personal service robot

Gérard Medioni [a,*], Alexandre R.J. François [a], Matheen Siddiqui [a], Kwangsu Kim [a], Hosub Yoon [b]

[a] *Institute for Robotics and Intelligent Systems, Viterbi School of Engineering, University of Southern California, USA*
[b] *Electronics and Telecommunications Research Institute, Republic of Korea*

Received 17 September 2005; accepted 13 October 2006
Available online 23 January 2007
Communicated by Mathias Kolsch

Abstract

We address visual perception for personal service robotic systems in the home. We start by identifying the main functional modules and their relationships. This includes self-localization, long range people detection and tracking, and short range human interaction. We then discuss various vision based tasks within each of these modules, along with our implementations of these tasks. Typical results are shown. Finally, STEVI v.1, a demonstration computer vision subsystem that performs real-time people detection and tracking from stereo color video, illustrates our modular and scalable approach to integrating modules into working systems.

Keywords: Robust real-time vision; Software architecture; Integration; Personal service robot; Human computer interfaces; Social interaction; Computer vision; Robotics

1. Introduction

Personal service robots, whose purpose is to "educate, entertain, or assist, or protect in the home" (as defined by *Robotic Trends* magazine) are attracting significant interest in both the research and commercial communities. Such robots can be especially useful in addressing the needs of an aging population in industrial societies. To become accepted, these robots need to autonomously interact and communicate with ordinary people in a natural and social way. This defines a new research field coined "Social Robotics" [1–3]. In particular, a service robot must be aware of its environment and interact with it in a naturalistic manner [4]. This includes not only interaction with the environment, but with humans who might not be technologically savvy, and might even have some disabilities preventing them from using traditional control devices and modalities—this defines the subfield of assistive robotics [5].

The roles for such robots range from offering companionship, to performing safety monitoring, to providing physical augmentation to compensate for physical disability. In the home, typical personal service robot activities might involve the following scenarios: the robot actively and quietly monitors its environment until it recognizes that a potential user summons it or might be in need of assistance. Visual tasks associated with this behavior involve self-localization in the environment, and constant detection and tracking of non-static objects in the environment (principally people), some level of scene understanding and situation awareness. In response to some signal from a person, the robot might take some relevant actions such as moving closer to the potential user, and become engaged in a focused interaction with this potential user, whom it might identify as its "master" or privileged user. If the robot recognizes the user as somebody who may issue commands to it, it should be receptive to them, and respond accordingly. It should also communicate its understanding of the perceived commands.

* Corresponding author.
E-mail address: medioni@usc.edu (G. Medioni).

1077-3142/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2006.10.015

Our research is motivated by both the critical nature of visual perception for these service robots in the home as well as the integration of vision based software components into a working system. Visual sensing has inherent properties that makes it a very desirable modality, on its own or as part of a multimodal sensing complex. In particular, vision based sensing is passive and non-intrusive, and thus requires no modification of the environment, nor does it involve cumbersome apparatus to be donned by potential users interacting with the machine. As no single vision algorithm or technique provides a robust solution to all vision tasks, efficient and effective integration of disparate vision subsystems is key to success.

In the rest of this paper, we discuss the major functional modules for a vision based personal robot system. These include: self-localization, long-range awareness, short-range interaction. Finally, STEVI v.1, a demonstration computer vision subsystem that performs real-time people detection and tracking from stereo color video, illustrates our modular and scalable approach to integrating such modules into working systems.

2. Functional modules

The emphasis of our project is not to necessarily develop many new algorithmic building blocks, but rather to identify and integrate functional modules essential to a vision system for personal service robots. We believe that the complexity of the tasks supported by a vision system requires that models be formed at higher levels of abstraction than the visual signal. In particular, we design our vision systems so as to produce and make available rich 2.5D, 3D and symbolic level models of the environment. These models are not only necessary for higher level reasoning and planning, they also are a source of robustness in the processing through feedback to the lower semantic levels. For instance, we use two cameras as input, and process the two streams independently and in parallel. We perform stereo fusion at the symbolic level, rather than starting from a disparity map. This provides robustness to loss of calibration, loss of synchronization, or loss of signal from one camera.

Given the plethora of specific vision algorithms and tasks, it is useful to identify the key components of a vision system for personal service robotics. Fig. 1 shows these functional modules and their inter-relationships.

The self-localization module uses the video stream from an omnidirectional camera to compute the 3D position and orientation of the robot in the world. This information can be used to issue motion commands to the robot, and to register the view centric 3D model of the observed world with the fixed environment.

The long-range awareness module detects and tracks people in the field of view of the robot, even if they are distant. The input consists of two synchronized video streams from two calibrated cameras. These streams are processed in parallel and fused at the symbolic level to produce 3D robot-centric descriptions of the people's locations. Integration with the 3D world pose of the robot in turn produces knowledge of their 3D position in the world, which allows the robot to approach the subject(s) in order to engage in a "conversation."

The short-range interaction module is activated when the robot is closer to people. Once again, the input consists of two synchronized video streams from a calibrated setup. Here, the resolution is sufficient to perform a finer analysis. We also process the two streams independently and fuse them only at the symbolic level. We detect the face, track it in 3D in real-time, and also detect and track the limbs. If needed, we classify the gestures in order to generate data for the planning and response actions.

Methods to address the functional modules are described in further detail in the following sections.

3. Self-localization with omni-directional camera

An assistive robot should always be able to provide answer to the question "Where am I" to move properly and to be prepared to respond to commands from its master. Thus, fast and accurate self-localization is an essential component of a personal service robotic system.

Omni-directional cameras are well suited for this purpose. In particular, they are capable of capturing a field of view encompassing large portions of the environment due to its extremely wide field of view. This results in efficient localization schemes that can be easily implemented.

We propose a simple and fast self-localization method using the folded catadioptric type of omni-directional camera [6], which can produce 30 frames per second at a resolution of 1600 by 1200 pixels. This method determines the position of the robot in a predefined map.

We convert the raw input image (1600×1200) into a panoramic image (1152×240) which compensates for misalignments in the imaging apparatus. Rectification allows us to easily find landmarks in the image. In particular we use three vertical lines as the landmarks to determine the physical location of the robot in a predefined model of the environment. Vertical lines are used as this system is intended for *in home* use and they are relatively easy to detect robustly in spite of partial occlusion as illustrated in Fig. 2.

To match these detected vertical edges with those from the predefined model of the environment, we use the magnitude and polarity of the edge, color values of the neighboring regions, taking into account the average image intensity for robustness to changes in illumination [7]. We also use the order constraint. To take into account occlusion, and for robustness purposes, we use a RANSAC based approach for matching: given three matching edges, we find the set of support for these matches, and repeat the process.

Once the vertical edges are matched, we can calculate the position of the robot relative to a predefined base posi-

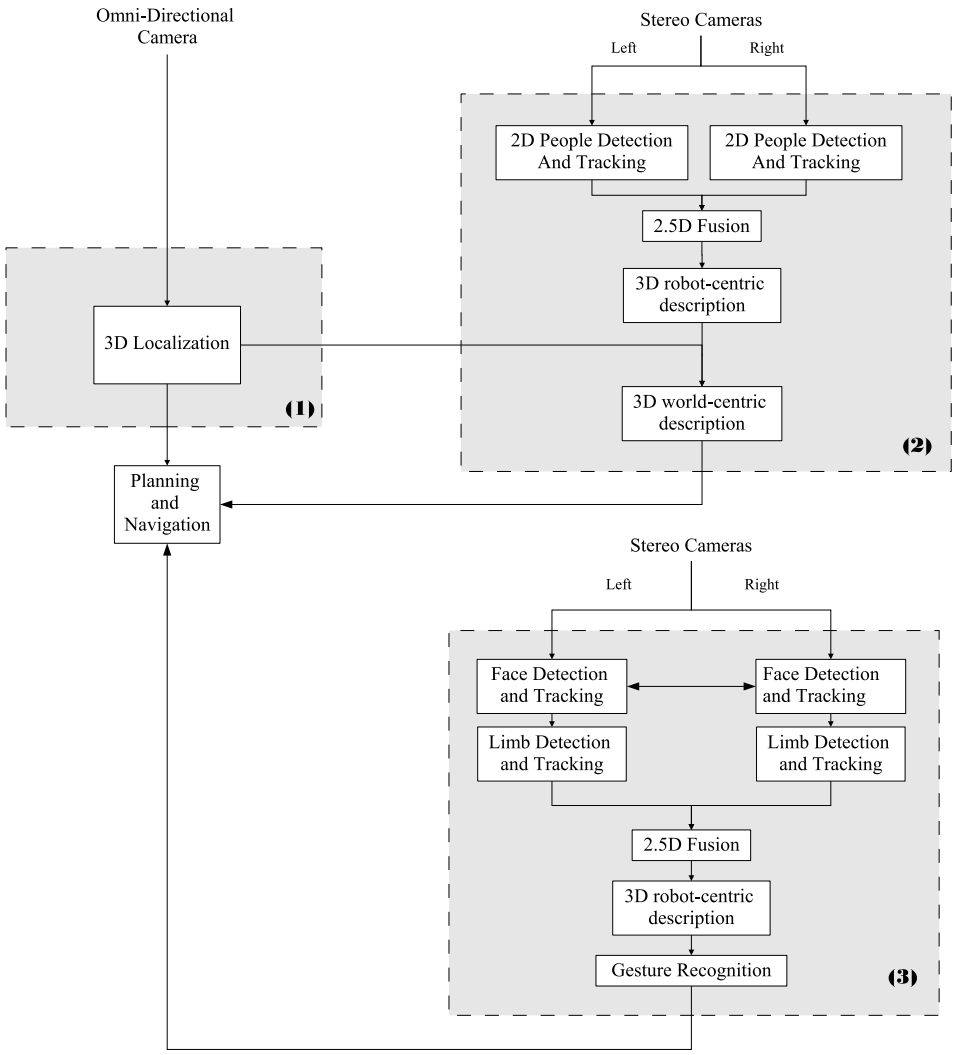

Fig. 1. Functional modules: (1) self-localization with omnidirectional camera; (2) people detection and tracking at long range; (3) interaction with humans at short range.

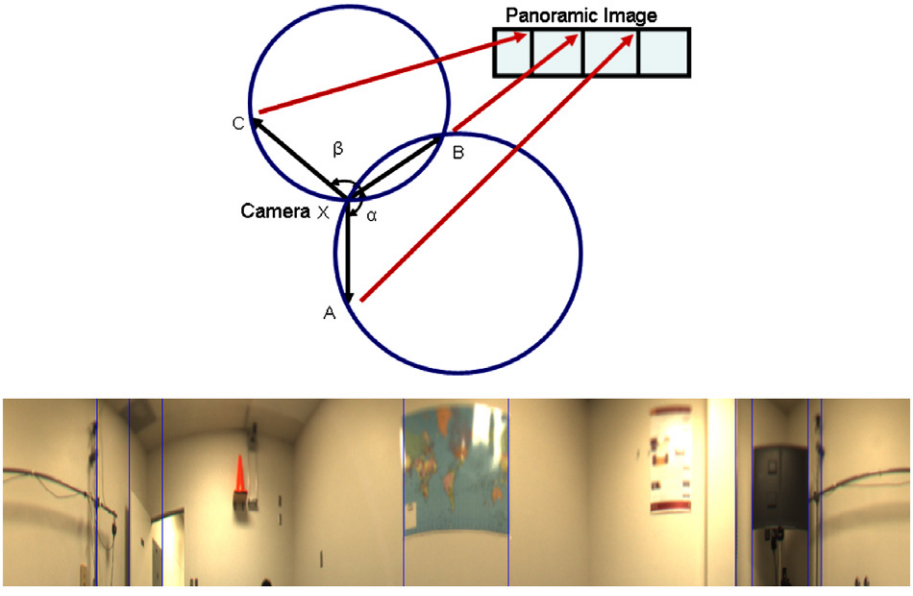

Fig. 2. Getting position from angles (top), detected edges (bottom).

tion using the pixel displacements of corresponding edges and the geometry as illustrated in Fig. 2.

In our experiments, we captured images at predefined positions within a single room and compared the inferred robot position with that of ground truth data. The error was between 0.11 and 3.64 inches as the robot moved in a range of 54 inches in the room of 160×120 inches. The average error distance was 1.2 inches for 13 positions. This level of accuracy is sufficient for our vision related tasks.

We intend to provide added robustness by combining this vision-based localization with odometry readings. And we also intend to combine the Simultaneous Localization and Mapping(SLAM) process with our method for fully autonomous navigation [8,9].

4. Long-range awareness: people detection and tracking

In order for a personal service robot to function, it must have awareness of its environment. In particular, it is critical for the robot to know whether, when and where people are present.

For this task face detection is very useful. The literature contains many approaches to face detection ([10,11] offer recent surveys). Appearance-based and learning-based techniques have produced very good results and show further promise [12]. Among these approaches, learned the cascades of week detectors are the most compatible with the real-time requirements [13–15].

At a distance detecting heads, rather than faces, may be more appropriate. In [16,17] methods are proposed to track elliptical head regions.

In our system we detect heads heuristically use image intensity, skin colored pixel detection, and circle detection from edges pixels. Color is very useful to infer the presence of skin when the face is near frontal. Image intensity is useful when captured heads are darker than the background. This heuristic occurs frequently enough to be useful. Edges are useful to delineate an elliptical head shape.

For each feature type we produce a set of candidate ellipses corresponding to head location hypotheses. Ellipse from different the different feature sets are merged to generate potential heads. This detection algorithm, implemented on a standard PC platform, runs at 6–8 frames per second. An example result is shown in Fig. 3.

Fig. 3. Example of head detection from a video stream.

5. Short-range interaction

Once the robot is closer to the people in the room, image resolution is sufficient to allow a more detailed analysis of users. This allows one to estimate a more refined human models which, as shown in the following sections, could included the head and arms. Such information can be used for a variety of human centered tasks, such gesture recognition and analysis.

5.1. Face detection and tracking from two image streams

Stereo face detection combines the results of monocular face detection from algorithms such as [13] applied to the left and right images to infer the presence of new faces in the scene. Cross validation in left and right views increases the system's robustness to false positives in the monocular face detection process. Furthermore the disparity between the detected faces determines the size of the corresponding bounding box. Each detected face area instantiates a target candidate to which is attached a color model (histogram) initialized from the face area in each image of the stereo pair. A mean shift-based algorithm [18] tracks the targets (or rather their color models) that have sufficient combined support in both images. Fig. 6 shows composite images of tracking information overlayed on the input left and right images. The apparent size of the face area is estimated using stereo information. Color-based tracking is robust to head orientation (as long as enough skin color remains visible). The target labels are consistent in left and right views (and across time, not visible in the figure).

5.2. 3D face tracking

Precise tracking of the 3D position, orientation and shape of the face is helpful for identification and facial gesture recognition [19]. A number of techniques and approaches can be found in the literature (see e.g. [12] for an overview).

We have a cooperative agreement in place with the Ecole Polytechnique Fédérale de Lausanne (EPFL), that gives us access to a state-of-the-art tracker developed by Fua et al. [20]. This module combines a recursive and key-frame-based approach to minimize tracking drift and jitter, and reduce the number of key frames required for stable tracking. Fig. 4 shows tracking results obtained with this method.

We have ported this submodule on our system, but have not integrated it at this time. It will enable us to analyze changes in face expressions and possibly assist in speech understanding.

Fig. 4. 3D face tracking.

5.3. Limb detection and tracking

For limb detection, we make use of a 2D articulated upper body model, and initialize this model with the head found by the method in Section 5.1. This constrains the possible locations of the arms (i.e. they need to be close to the head) as well as provides information about skin color, an important visual cue used for both detection and tracking. In addition to skin color, we make use of edge and boundary information.

Like face detection, human body pose estimation and tracking from images and video has been studied extensively in computer vision. Of relevance to this approach are those methods that make use of articulated (or near articulated) models [21–23]. These methods align an articulated model of the human body using various image cues such as appearance, edges, color, and motion. A major component of such systems is devoted to the machinery used to perform this alignment. Statistical approaches can be found in [24], where a learned appearance based detector is used to aid the sampler based tracking system. In [21] Lee uses data driven MCMC sampling to obtain likely poses. Other approaches offer a more bottom up framework [25,26], for example, by assembling low level features such as parallel pairs of edge segments.

Out strategy for limb detection is based on the work of [22]. Here human models consist of a collection of 2D part models representing the limbs, head and torso. The individual parts are organized in a tree structure, with the head at the root. The human model is aligned with an image by first searching for each individual part over translations, rotations. Following this, the optimal pose is found by combining the results of each individual part detection result efficiently.

Our limb detector is tuned for users wearing short sleeve shirts with their forearms exposed. This assumption is fairly valid in warmer environments, allows us to avoid explicitly modeling clothing and subsequently enables fast detection of limbs. The detected limbs can subsequently be efficiently tracked using a gradient based scheme that seeks to optimize the overlap of rectangular forearm box with the underlying skin and boundary pixels. Re-detections are triggered if a track score falls below a threshold. An example of this combined detection/tracking method which runs 10 frames per second is shown in Fig. 5.

6. Integration

The integration of heterogeneous software components into a viable real-time and low latency system, poses significant software design and implementation challenges. To facilitate this task, we leverage the Software Architecture for Immersipresence (SAI) framework [27]. We describe an early version of STEVI [28], an integrated demonstration computer vision subsystem that performs real-time people detection and tracking from stereo color video, as shown in Fig. 7. STEVI v.1's tracking core is based on the algorithm discussed in 5.1.

SAI specifies a formal architectural style [29] whose underlying extensible data model and hybrid (shared memory and message-passing) distributed asynchronous parallel processing model facilitates the specification of

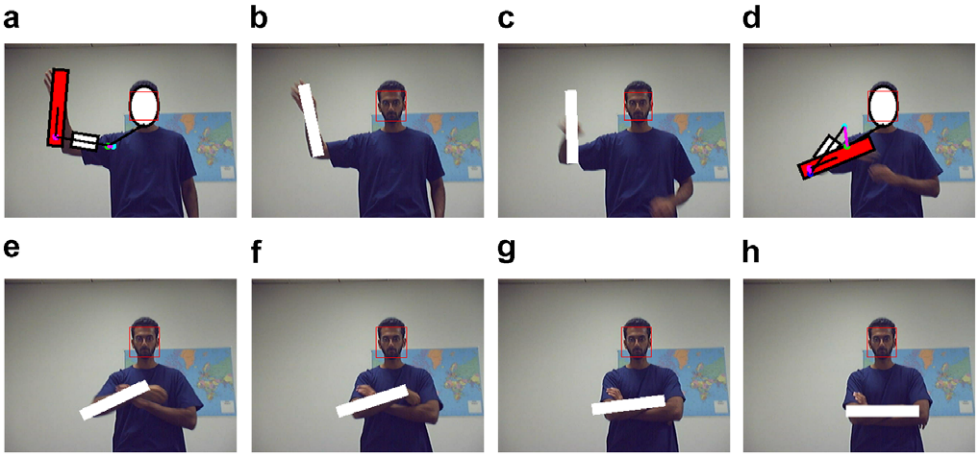

Fig. 5. This figure shows the results of the limb detector and tracker. In (a) the initial pose is detected and successfully tracked until frame (c). Here the tracker lost tracker and needed to be re-initialized.

Fig. 6. Composite images of tracking data overlayed on left and right input color images. The apparent size of the face area is estimated using stereo information. Stereo color-based mean shift tracking is robust to head orientation. Target labels are consistent in the left and right views.

asynchronous data stream processing software systems. A graph-based notation for architectural designs allows intuitive system representation at the conceptual and logical levels, while at the same time mapping closely to the physical level.

SAI offers intuitive abstractions, emphasizing the flow of data in the system. The modularity of the model allows distributed development and testing of particular elements, and easy maintenance and evolution of existing systems. SAI also naturally supports distributed and parallel processing. Where sequential and synchronous approaches impose arbitrary constraints on the performance of the underlying system, SAI's asynchronous concurrent model allows designing for optimal (theoretical) system latency and throughput.

Fig. 7 shows the conceptual specification graph for STEVI v.1 in the SAI notation. In the SAI model, *volatile data* flows down streams (notation: thin lines) defined by connections between processing centers called *cells* (notation: squares), in a message passing fashion. Repositories called *sources* (notation: circles) hold persistent data to which connected cells (connection notation: thick lines) have shared concurrent access. The directionality of stream connection between cells expresses dependency relationships. Dashed boxes in the figure identify three main groups of components: (1) the *Stereo input and pre-processing* subsystem regroups the multiple camera video input and subsequent pre-processing of the video frames flowing on the stream; (2) the *Stereo multi-target detection and tracking* subsystem is based on the algorithm discussed in 5.1; (3) the *Visualization* subsystem provides a visual insight into the tracker's state for debugging and demonstration purposes. The design is inherently parallel, and the modularity of the SAI style facilitates design evolution.

Note that the graph presents an intuitive view of the system. Because the visual language in which it is expressed has well defined semantics, the graph is also a formal specification (in this case at the conceptual level). SAI abstractions allow a continuum of intermediate-level representations from conceptual to physical specifications. SAI promotes the encoding of system logic in the structural organization of simple computing components rather than in the complexity of the computations carried by individual components. SAI designs exhibit a rich variety of structural and functional *architectural patterns*, suitable for systematic study and reuse.

To implement the system as specified, we leverage the Modular Flow Scheduling Middleware (MFSM;

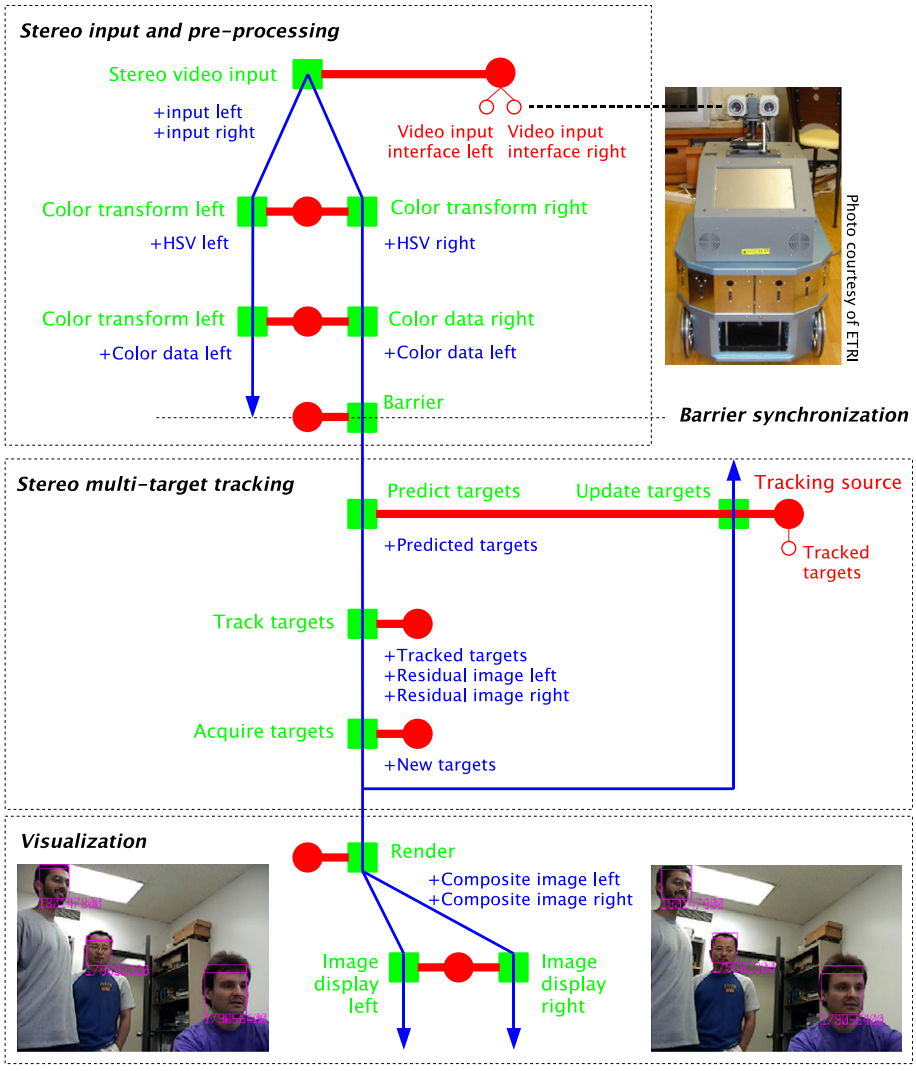

Fig. 7. STEVI v.1 conceptual level system architecture in SAI notation. Composite images of tracking data overlayed on left and right input color images illustrate system output.

mfsm.sourceforge.net), an open source architectural middleware that provides code support for SAI's architectural abstractions in a multi-threaded environment. The MFSM project comprises a number of open source functional modules that allow to focus coding effort on project specific components of the system. For example, modules for video input, image display, and other general purpose modules were directly reused. The basic computer vision algorithms underlying the system are established in the literature. STEVI v.1's implementation encapsulates face detection and color-based tracking functionalities provided by the Intel Open Source Computer Vision Library (OpenCV) [30]. The initial STEVI v.1 system was developed on the Windows platform. The same system was ported to Linux with minimal effort, thanks to the cross-platform MFSM code and module base, and the availability of Intel's OpenCV for both platforms. Furthermore, as MFSM is inherently multi-threaded, the resulting systems can directly and transparently take advantage of multiple CPUs, hyper-threading and multi-core processor architectures.

The running systems reliably detect and track people at approximately 15 frames-per-second on contemporary hardware. Video input (images 320×240 pixels) is acquired by two color cameras (USB or Firewire) mounted on a calibrated stereo rig. The most computationally expensive task is face detection, already performed on reduced-resolution images for efficiency purposes. The visualization subsystem, useful only for demonstration purposes, also consumes a non-negligible part of computation power. STEVI has been integrated and tested on our Weber-N robot platform.

The use of a well-defined, modular design approach, suitable for interactive systems, not only facilitates the development of a working prototype, but also ensures that the development of subsequent evolutions of STEVI will directly benefit from the work done on v.1.

7. Conclusion

In this work, we have addressed visual perception for service robots in the home. We have described and identi-

fied the main functional modules that would comprise such a vision-based system: self-localization, long range people detection and tracking, and short range human interaction. We have identified sub-problems and solution methods within each of these modules. Finally, STEVI v.1, a demonstration computer vision subsystem that performs real-time people detection and tracking from stereo color video, illustrated our modular and scalable approach to integrating such modules into working systems.

Acknowledgments

This work was conducted in the context of a collaborative project with the Electronics and Telecommunications Research Institute (ETRI), a Korean non-profit, government-funded research organization.

References

[1] T. Fong, I. Nourbakhsh, K. Dautenhahn, A survey of socially interactive robots, Robotics and Autonomous Systems, Special issue on Socially Interactive Robots 42 (3–4) (2003) 143–166.
[2] C. Breazeal, Socially intelligent robots, ACM Interactions 12 (2) (2005) 19–22.
[3] D.J. Feil-Seifer, M.J. Matarić, Socially assistive robotics, in: Proceedings of the International Conference on Rehabilitation Robotics, Chicago, Il, 2005.
[4] K. Kawamura, R.T. Pack, M. Iskarous, Design philosophy for service robots, in: IEEE International Conference on Systems, Man and Cybernetics: Intelligent Systems for the 21st Century, vol. 4, Vancouver, BC, Canada, 1995, pp. 3736–3741.
[5] D.P. Miller, Assistive robotics: an overview, in: Assistive Technology and Artificial Intelligence, Applications in Robotics, User Interfaces and Natural Language Processing, Springer-Verlag, London, UK, 1998, pp. 126–136.
[6] J. Gaspar, N. Winters, E. Grossmann, J. Santos-Victor, Toward robot perception using omnidirectional vision, in: S. Patnaik, L. Jain, G. Tzafestas, V. Bannore (Eds.), Innovations in Machine Intelligence and Robot Perception, Springer-Verlag, 2004.
[7] L. Tang, S. Yuta, Navigation of mobile robots in corridor network environment based on memorized robot's motion and omni-directional images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2002.
[8] J. Castellanos, J. Montiel, J. Neira, J. Tardos, The spmap: a probabilistic framework for simultaneous localization and map building, in: Proceedings of the IEEE Transactions on Robotics and Automation, 1999.
[9] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, M. Csobra, A solution th the simultaneous localization and map building(slam) problem, in: Proceedings of the IEEE Transactions on Robotics and Automation, 2001.
[10] M. Yang, D. Kriegman, N. Ahuja, Detecting faces in images: a survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (1) (2002) 34–58.
[11] E. Hjelmas, B. Low, Face detection: a survey, Computer Vision and Image Understanding 83 (3) (2001) 236–274.
[12] S.Z. Li, J. Lu, Face detection, alignment, and recognition, in: G. Medioni, S. Kang (Eds.), Emerging Topics in Computer Vision, Prentice Hall, Upper Saddle River, NJ, 2004, pp. 385–455.
[13] P. Viola, M.J. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2) (2004) 137–154.
[14] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, H. Shum, Statistical learning of multi-view face detection, in: Proceedings of the European Conference on Computer Vision, 2002, p. IV:67 ff.
[15] R. Lienhart, A. Kuranov, V. Pisarevsky, Empirical analysis of detection cascades of boosted classifiers for rapid object detection, in: B. Michaelis, G. Krell (Eds.), Pattern Recognition—Proceedings of the 25th DAGM Symposium, Springer Verlag, Magdeburg, Germany, 2003, pp. 297–304.
[16] S.T. Birchfield, An elliptical head tracker, in: Conference Record of the Thirty-First As global Conference on Signals, Systems and Computers, 1997, pp. 2:1710–1714.
[17] S.T. Birchfield, Elliptical head tracking using intensity gradients and color histograms, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 1998, pp. 232–237.
[18] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619.
[19] M. Turk, M. Kölsch, Perceptual interfaces, in: G. Medioni, S. Kang (Eds.), Emerging Topics in Computer Vision, Prentice Hall, 2004.
[20] L. Vacchetti, V. Lepetit, P. Fua, Stable real-time 3d tracking using online and offline information, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (10) (2004) 1385–1391.
[21] M.W. Lee, I. Cohen, Human upper body pose estimation in static images, in: Proceedings of the European Conference on Computer Vision, 2004, pp. II:126–138.
[22] P.F. Felzenszwalb, D.P. Huttenlocher, Pictorial structures for object recognition, International Journal of Computer Vision 61 (1) (2005) 55–79.
[23] D.D. Morris, J.M. Rehg, Singularity analysis for articulated object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998.
[24] L. Sigal, S. Bhatia, S. Roth, M.J. Black, M. Isard, Tracking loose-limbed people, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2004, pp. I: 421–428.
[25] D. Ramanan, D.A. Forsyth, Finding and tracking people from the bottom up, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2003, pp. II: 467–474.
[26] S. Ioffe, D.A. Forsyth, Probabilistic methods for finding people, International Journal of Computer Vision 43 (1) (2001) 45–68.
[27] A.R. François, Software Architecture for Computer Vision, in: G. Medioni, S. Kang (Eds.), Emerging Topics in Computer Vision, Prentice Hall, Upper Saddle River, NJ, 2004, pp. 585–654.
[28] A.R. François, G.G. Medioni, A vision system for personal service robots: resilient detection and tracking of people, Tech. rep. 06-877, Computer Science Department, University of Southern California.
[29] A.R. François, A hybrid architectural style for distributed parallel processing of generic data streams, in: Proceedings of the International Conference on Software Engineering, Edinburgh, Scotland, UK, 2004, pp. 367–376.
[30] G.R. Bradski, The OpenCV Library, Dr. Dobb's Software Tools for the Professional Programmer.