Teleoperated Visual Inspection and Surveillance with Unmanned Ground and Aerial Vehicles

Sebastian Blumenthal[1], Dirk Holz[1], Thorsten Linder[1], Peter Molitor[2],
Hartmut Surmann[2] and Viatcheslav Tretyakov[1]

[1] Department of Computer Science, University of Applied Sciences Bonn-Rhein-Sieg, St. Augustin, Germany

[2] Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS), St. Augustin, Germany

***Abstract*—This paper introduces our robotic system named *UGAV* (Unmanned Ground-Air Vehicle) consisting of two semi-autonomous robot platforms, an Unmanned Ground Vehicle (UGV) and an Unmanned Aerial Vehicles (UAV). The paper focus on three topics of the inspection with the combined UGV and UAV: (A) teleoperated control by means of cell or smart phones with a new concept of automatic configuration of the smart phone based on a RKI-XML description of the vehicle control capabilities, (B) the camera and vision system with the focus to real time feature extraction e.g. for the tracking of the UAV and (C) the architecture and hardware of the UAV.**

***Index Terms*—USAR, teleoperation, UGV, UAV, OCU, visual attention, computer vision**

I. Introduction

Mobile robotic systems for tele-exploration are gaining more and more importance, especially for industrial inspection tasks and rescue operations. In scenarios, like those that are addressed e.g. in *Urban Search And Rescue* (USAR), fully autonomous systems are not applicable because of safety or efficiency reasons. Here, human operators control semi-autonomous robot systems to gain information about environments where manually entering is considered harmful or that are not accessible at all. Basically, research efforts have been focussed on the following areas:

- Mobility and robustness of the robot platform,
- Development of reliable and accurate sensors for mobile robots and
- Human-Machine-Interfaces with high usability and acceptance for the human operator.

As every platform has its own advantages and disadvantages, they also have different applications and workspaces where they are particularly suitable. Hence, it is reasonable to use several platforms concurrently for the same task combining their individual strengths, i.e. so-called *multirobots* [1]. These cooperative systems are composed of several heterogeneous robots, e.g. a smaller mobile robot that is carried by a larger one and can be dropped off in cases where the larger robot can not further explore the environment because of its size. The larger robot, on the other hand, can carry different tools and batteries for the small-size robot. Recently, research groups have started to address the combination of ground and aerial vehicles. Whereas ground vehicles can enter e.g. collapsed building or mines, aerial vehicles can help to get an overview of the whole site [2].

One major problem is to deliver the required information about the surrounding of the robot to the operator. Cameras, mounted on Unmanned Ground Vehicles (UGVs) or Unmanned Aerial Vehicles (UAVs), have become the de facto standard sensor to provide this information. Our intention is to develop a platform to increase the level of reconnaissance during a USAR operation. To achieve this goal it is reasonable to combine a ground and an aerial vehicle. Here we introduce our approach named *UGAV* (Unmanned Ground-Air Vehicle) of a robotic system consisting of two semi-autonomous robot platforms, an UGV and an UAV (see Fig. 1). Both robots are equipped with camera systems for surveillance. The operator can directly control the UGV and the UAV (see Fig. 4). Furthermore the aerial vehicle can be commanded by the UGV for autonomous missions e.g. sending GPS coordinates which have to be observed and for autonomous landing. Especially for autonomous landing the ground vehicle has to detect and track the aerial vehicle. We present a real-time visual attention approach to track the UAV. The camera views from the vehicle are also presented to the operator. For teleoperating the UGV we use an off-the-shelf mobile device.

The paper is organized as follows: Chapter 2 gives a brief overview on the used *UGAV* hardware, whereas the teleoperation for the overall system is described in Chapter 3. Chapter 4 copes with acquisition of visual data and extraction of spatial information out of our new camera concept of a dodecahedron cube or a higher resolution omni camera. The architecture and control issues of the aerial vehicle are described in chapter 5.

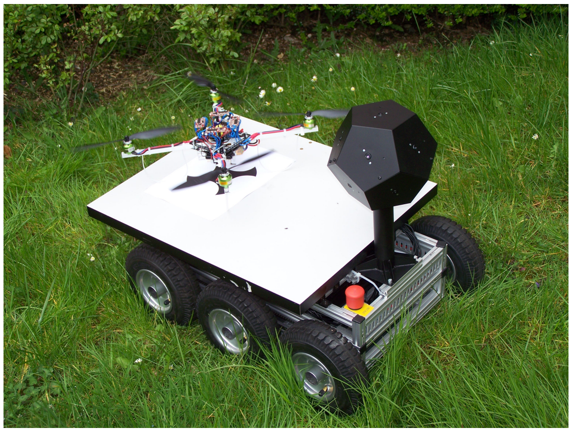


Fig. 1. First conceptual chassis of the *UGAV*. The ground platform is build of a VolksBot RT6 which is equipped with a panoramic vision system (sphere cube with 11 cameras) and a landing platform for the quadrotor.

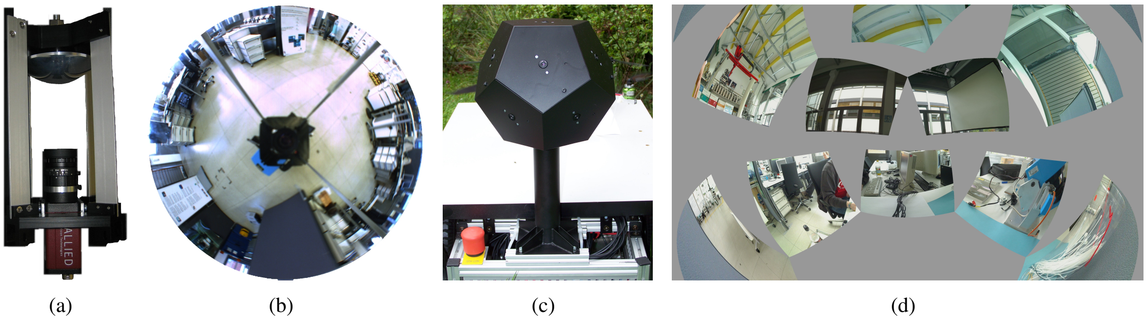


Fig. 2. Visions systems for the UGV. (a) shows the omnidirectional IAIS vision system. The camera aims towards a hyperbolic mirror. (b) is an image from the perspective of the omnidirectional camera. (c) The dodecahedron shaped camera system with eleven cameras. Each camera aims in another direction. (d) demonstrates an impression from panoramic images of the dodecahedron shaped camera.

II. Platform

Our robotic system (see Fig. 1) consists of an UGV and UAV which are briefly described in the following sections.

A. Unmanned Ground Vehicle

The UGV is based on a modular mobile platform called *VolksBot* [3], which has been designed specifically for rapid prototyping and applications in education, research and industry. The *VolksBot* system is developed, manufactured and sold by the Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS). It easily allows access to and replacement of several components such as motors, batteries and electronics as well as the extension of new hardware. For stability reasons in rough terrain we have chosen the six-wheeled version *VolksBot RT6* (see Fig. 3) out of the several variants of the *VolksBot* [4]. It has a size of $700 \times 480 \times 600$mm (L×W×H) and a weight of approx. 15 kg. As all six wheels are driven by the two 150W motors, the robot is even able to climb smaller stairs or steps. The robot has a maximum velocity of $1.1\,\mathrm{m/s}$ and a maximum payload of approx. 40 kg. For indoor applications front and rear wheels can be chosen to have no tread pattern to enhance rotation. Two Mac minis (CPU 2GHz, Memory 2GB) serve as computational units for processing sensor data and controling the UAV.

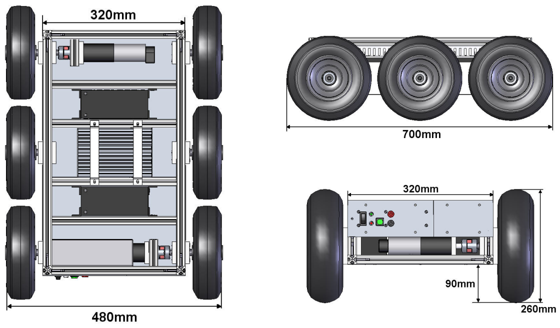


Fig. 3. Engineering drawing of the *VolksBot* RT6 chassis. RT6 is a six-wheeled robot platform with rough terrain capabilities. The left sketch shows top view, upper right shows lateral view and lower right shows front view.

B. Unmanned Aerial Vehicle

The UAV is a four-rotors aerial platform, a so-called *quadrotor* [5], that is capable of Vertical Take-Off and Landing (VTOL). Its flight control board is equipped with an inertial measurement unit consisting of 3-axes gyroscopes, 3-axes inertial sensors, 3-axes digital compass and a GPS module. For altitude control a pressure sensor is employed. Fusion of these sensors as well as the control of the four motors is done by means of an on-board 20 MHz-microcontroller (Atmel ATMEGA644P) and four brushless motor control boards. The on-board microcontroller communicates with the four brushless controllers via I^{2}C bus.

The quadrotor has a size of $650 \times 650 \times 220$mm (L×W×H) and a weight of 590 g. With an extra antenna the height increases to 550 mm and the weight increases to 620 g. With fully loaded batteries (2100 mAh) it can operate approx. 20 min. Its maximum payload is 350 g. The quadrotor is controlled either by the UGV or a human operator via WiFi, Bluetooth or an analog remote control unit. The architecture of the quadrotor is further explained in section V.

C. Vision sensors

For the USAR purpose the RT6 is equipped with one of the following vision systems. The first one consists of the IEEE1394 firewire camera ”AVT Marlin F-145-C” aiming towards a hyperbolic mirror. This camera can deliver up to 10 frames per second in high resolution color mode (1392 x 1038 pixel). The second vision system is build up from eleven off-the-shelf USB-webcams aiming in different directions. They are mounted in a dodecahedron shaped chassis with a size of $220 \times 220 \times 380$mm (L×W×H). Each camera delivers up to 15 frames per second at a resolution of 800×600 pixels. At a lower framerate, pictures with a resolution of 1600×1200 pixels can be acquired. All eleven cameras are connected to one up to four Mac minis via USB 2.0. Both platforms are suitable for teleoperated applications like USAR or visual surveillance scenarios.

III. Teleoperated Robot Control

A teleoperator is a physical device which is enabling an operator to move around, sense or mechanically manipulate objects by using a robot. These devices can be separated into two

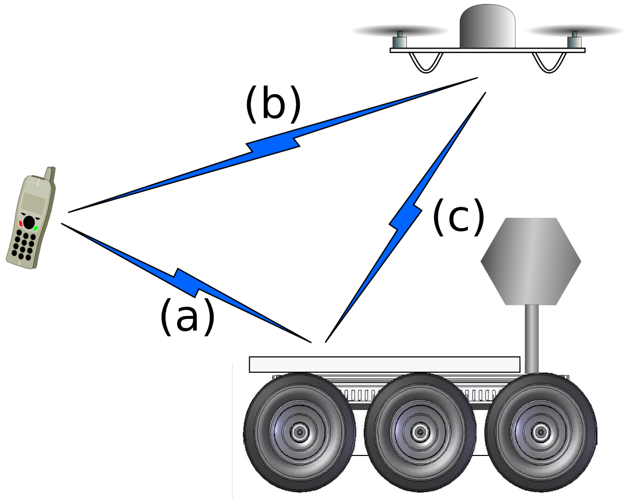


Fig. 4. Unmanned Ground-Air Vehicle (*UGAV*): The mobile device controls either the UGV (a) or the UAV (b). (c) UAV is commanded by the UGV.

device classes. Both have in common that the teleoperators allow physically separating the operator from the machine and the robot [6, 7].

The first class is named anthropomorphic which means that these teleoperators have a manlike physiognomy. Anthropomorphic teleoperators are mostly used in combination with a manipulator to allow a remote handling of objects (e.g. handling toxic or radioactive waste). The non-anthropomorphic devices build the second class. This class includes a lot of different device types like PCs, laptops, special hand controllers and vehicle cockpits [8]. If the teleoperator is a laptop or a similar device they are also called Operator Control Units (OCU).

OCUs are common for USAR robots and often built into waterproof boxes [7]. This mounting concept respects two essential requirements for the whole rescue equipment. The first one is given by the fact that every object which is used on a mission must be either able to be easily decontaminated or must be disposed to ensure that no biologic, toxic, chemic or radioactive contamination can effect the rescue teams or the population [9]. The second is given by the need of portability and robustness. The rescue team member must be able to carry the OCU and it can not be ensured that the OCU will not be dropped during a march to the operational area.

Each teleoperator needs a connection to its robot and widely varied techniques are in use to ensure optimal communication channels with or without respecting real time requirements [10]. There are different technologies available, either wired or wireless connections. As shown in [7, 9] the usability of wireless techniques is limited in the situation of exploring small voids from a collapsed building. Based on the nature of radio frequency transmission, the signal can be heavily disturbed or poor in such an environment. This might cause that the robot gets lost, as seen on the World Trade Center catastrophe [7].

The advantage of a wireless communication technique is the complete physical isolation between the robot and the operator. Data- and safety-tether might get caught and reduce the freedom of movement. Hence, the decision on which technique should be used heavily depends on situation and task.

A. Mobile device (PDA or Cell phone)

Instead of using an ordinary Laptop or a special remote device to control the robot, our approach uses mobile devices like (PDAs or cell phones). This concept is just starting and not quite common at the moment. However, it has multiple advantages out of the shelf. In fact there is a widespread research field evaluating the beneficing and usability of mobile devices in general for non-phone use (e.g. [11–13]):

- **Always available:** Cell phones are nowadays widespread and there are already trials to make them part of the normal rescue worker equipment [14]. Therefore the device is already up on the ground and can also be used for controlling the robot.
- **High social acceptance and limited teaching:** Handling of the OCU is one of the strongest barriers which detain rescuers to make use of the advantages or their robots. They have to be trained to use and interpret sensors of the system [7]. This training can be limited if they already now how the physical device works. This is given for mobile phones.
- **Man pack-able, light weight and size:** In an urban catastrophe scenario the equipment must be manportable. This means that rescue workers have to be able to transport their technical equipment to the ground of interest by themselves. Both, the OCU and the robot have to be carried. An OCU which is based on a mobile device like a cell phone does not have these requirements (they are still on the ground) and can be easily carried by the same person supplying the robot.
- **Long run time:** A long runtime is required for USAR for both the OCU and the robot as mentioned in [9]. Cell phones are able to operate for a sufficient time.
- **Robustness and substitutability:** Mobile devices are robust enough for the daily use. But they are not designed to be used on rough terrain or to be waterproof. These disadvantages can be managed by using special cases for the mobile devices to fit the requirements [15]. Additionally the cell phones are not as expensive as common OCUs and they are out of the shelf products. Following these facts the substitutability is to declare as high by using mobile devices as OCU.
- **General purpose utilizable concept:** Millions of mobile robots are actually drive in-house from small LEGO toys over vacuum cleaner up to service robots and their number will increase in the next decade. They will be in our houses as a service oriented gadget, in our cars as a driving assisted system or part of the technical equipment of rescuer workers. Therefore the need of interaction between such systems and human is growing. Cell phones are available to do these jobs and it is expected that they will be strongly influence our life [16].

Disadvantages of using mobile devices for teleoperation are the limited computing power and small screen sizes. This reduces the usability of mobile devices to simple teleoperation tasks which is not always desirable. Furthermore, mobile devices are limited by the numbers of supported interfaces and are hardly extendable. Some of these disadvantages can be compensated by the capabilities of the robot. For example the limited computing power can be compensated partly by acquiring pre-computed data from the robots.

B. Android

As a representative for the upcoming smart phone generation a simulation software development kit called *Android* is used. Android is a new platform for mobile devices like cell phones. The Open Handheld Alliance is developing Android as an open software Operating System (OS). The kernel of Android is a Linux in version 2.6. The developers have setup a framework on top of the Kernel to guarantee that the whole OS can be used via JAVA applications. Therefore the main programming language (currently the only application language which can be used) is JAVA [17]. Other cell phone operating systems which are programmable in JAVA use mostly the JAVA mobile version J2ME. However, J2ME is functionally reduced and not as powerful as the normal JAVA (J2SE) [18]. The new framework makes use of the Dalvik Virtual Machine. This allows for using nearly the full standard JAVA libraries on mobile devices. A special advantage of Android is the open source concept which will allow extending the OS for the needs of USAR.

The Android project includes support for GSM and UMTS telephone networks. Furthermore, it supports WiFi, Bluetooth and USB. Therefore the platform will allow several communication options. This fact allows us to use a widespread communication background (e.g. if a near location network like WiFi is not available the system can use long-distance networks like the GSM network).

The expected CPU-power, size, weight and runtime can be approximated by the current available smart phones. The computing power of smart phones is up to about 600 MHz. This is far less than the current power of PCs or Laptops. These facts challenge our project and it is to be evaluated if the CPU-power can reach our goals. Nevertheless, first results indicate a positive outcome on that question. An additional positive effect by using cell phones in USAR is given by the growing popularity of the Global Position System (GPS) for these mobile devices. For sure the most Android phones will have a built in GPS antenna which allows for determining the position of the Operator [19]. This information can be used to interact with the robot and provides basic means for homing and extended path planning routines.

C. XML Robot Instructions

Since most of the remote control devices are specially made for the robots there is no need for universal remote control information. Cell phones on the other hand have different inhomogeneous operating systems. To use them for control a vehicle, we store the control parameters in a XML file on the robot. Each cell phone can request a *RKI* (Robot Known Instruction) file to configure the control program. Furthermore a simple JAVA control client can also be requested from the vehicle. Figure 5 shows an example of an *RKI* file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Descriptor File for the RKI(Robot Known
Instructions) -->
<RKI>
<instruction ID="1001" name="ahead">
<labels>
<label>Drive forward</label><!-- Complete caption-->
<label>Ahead</label><!--Caption -->
<label>Go</label><!--Short caption -->
</labels>
<explanations>
<eplanation>This instruction is used to drive the
robot ahead. The driving speed is not settable by
this command, therefore the preset robot speed is
used.
</eplanation><!--Complete eplanation -->
<eplanation>Drive the robot ahead with out speed
settings.</eplanation><!--Short eplanation-->
</explanations>
<values>
<value>
<boolean default="true">true</boolean>
</value>
</values>
</instruction>
...
<instruction ID="1004" name="back_set">
<labels>
<label>Drive backwards</label>
<label>backward</label>
<label>back</label>
</labels>
<explanations>
<eplanation>This instruction is used to drive the
robot backwards. The driving speed is set by this
command. The speed can be set in up to 20 centimeter
per second.</eplanation>
<eplanation>Drive the robot backwards with speed
settings.</eplanation>
</explanations>
<values>
<value>
<integer bytes="1" default="0x00" min="" max="">
<SI entity="cm/s"/><!--Physical meaning of the value-->
<!-- Min, average and max possible values-->
<min> <constant value="0"/> </min>
<average> <constant value="10"/> </average>
<max> <constant value="20"/> </max>
</value>
</values>
</instruction>
</RKI>
```

Fig. 5. Example of a *RKI* (Robot Known Instruction) file. The file is stored at the robot and is request by the control client.

D. First Results

We use the Android simulator (version m5-rc15) to evaluate the usability of the platform for an OCU. The simulator is able to mimic the behavior of a mobile phone running Android [20]. The OCU software was written in JAVA and matches following use-cases:

- Open up a bidirectional wireless connection between a ground based robot and the OCU.
- Display the robot camera stream (unidirectional).
- Navigate the robot via basic command sets and a graphical user interface.

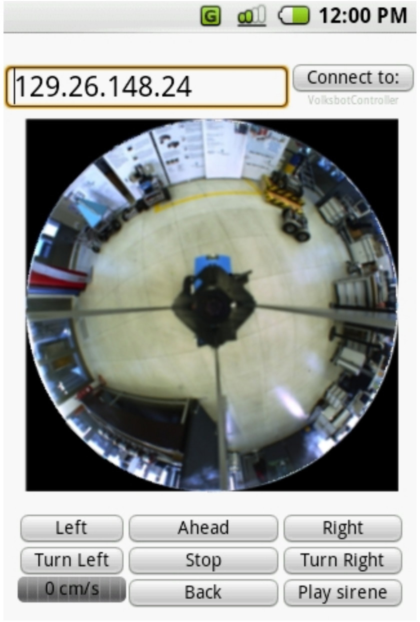
The outcome of the first test rides are public available as videos on [21] and [4][1]. As you can see there the Android simulator was running on a conventional notebook (Intel Mobile 1.73 GHz, 1 GB RAM) and used the IEEE 802.11 b/g (WiFi) standard to setup a communication channel. A three-wheeled robot (*VolksBot* RT3) was used during the testing period with

[1] http://www.volksbot.de/videos/Android-2.divx.avi

same VolksBot Motor Controller (VMC) as on the RT6. The IAIS-Vision system on top allows a panoramic view (see Fig. 2(a)). The user interface is setup by respecting that also non-specialists must be able to control the robot. The whole interface can be controlled through buttons and the cell phones touch pad, if available. This makes the handling intuitive and respects the request to limit the need of training (see Fig. 6).

The prototype OCU software is a single screen which is separated into three parts. On top of the screen the operator inserts the address of the robot and can start the connection via a button. In this prototype version of the OCU the address is based on IP4 and a local network domain name. The communication is similar to that of standard client-server architectures (robot: server; OCU: client).

Visual information is a fundamental for USAR. It is needed e.g. for navigation and finding victims [7, 22–24]. As a result the camera view is displayed in the middle of the cell phone screen. This explicit region shows the image of the on board robot camera. If the robot is equipped with an omnidirectional camera system like the IAIS-Vision system the operator obtains a panoramic view over the surroundings. The bottom of the user interface is reserved for motion control. The OCU software allows for commanding translational and rotational velocities as well as some predefined motions like for instance turning on the spot (if available).



129.26.148.24

Fig. 6. User interface of teleoperator software in the Android environment.

For safety reasons the robot is setup with a watchdog functionality. This means it decelerates if there is no new command within the last two seconds and stops immediately if the signal to the OCU gets lost. In our testing environment this safety functionality works well, but as seen in [7] this behavior is critical. An outcome of this safety behavior can be that the robot gets lost during a mission and is maybe not recoverable. Therefore this behavior is to be extended by the functionality of searching for an alternative communication channel and by autonomous homing skills.

IV. Vision systems

A fundamental problem in the field of vision for mobile robots is the **online** perception of the environment. The vision systems deliver an overview of the scene which supports the operators impression of the whole scenery and provides visual information for controlling and steering the semi-autonomous robot. Another crucial task comprises finding victims and inspection of buildings or cluttered terrain for revealing structural damages or hazardous areas [23]. Therefore the UGV is equipped with one of the following panoramic camera systems: (A) One camera aiming towards a hyperbolic mirror which enables a hemispherical but distorted view (see Fig. 2(a)) and (B) a dodecahedron cube consisting of eleven cameras aiming in different directions. They are mounted in a penta-dodecahedron shaped polyhedron to achieve a near full spherical view (see Fig. 2(c)). The grabbed frames from the cameras are undistorted but the image processing results in higher computational efforts.

To improve the operator's *scenery awareness* we tested different feature extraction mechanisms, according especially to their real-time applicability. Image features can be used i.e. to calculated depth information (for mapping) if they occur in several different images or to help loop closing (for SLAM). Our approach is based on features in the image data. Figure 7 shows a schematic overview of the system.

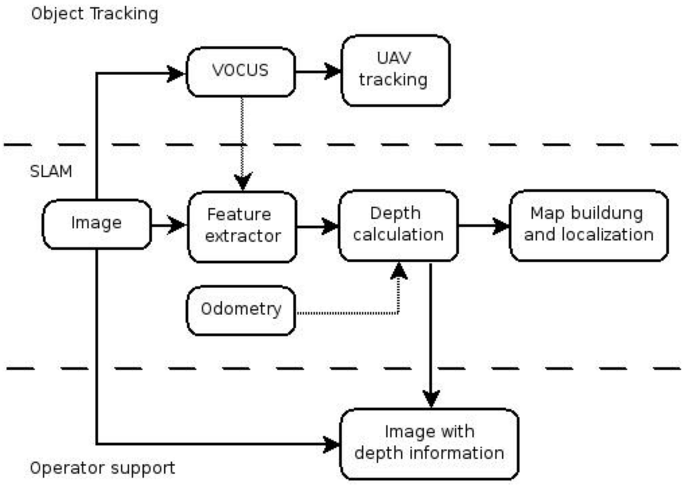


Fig. 7. Schematic overview of the vision systems.

Features need to be robust against changes in scale, rotation and occlusion. Two widely used algorithms for the feature extraction are the SIFT algorithm from [25] and SURF algorithm from [26]. Both algorithms are robust against changes in scale, rotation and occlusion. Sim et. al. [27] and also Karlson et. al. [28] demonstrates vision-based robot navigation systems using SIFT features, but only offline. The SURF algorithm provides comparable good results and needs less computational power because of utilizing integral images and approximated digital filters (Haar wavelet) but is also not real-time capable [29]. To recognize a feature in different images a unique descriptor is necessary. While the SIFT descriptor consists of a 128 dimensional vector SURF needs only 64 dimensions by default. Hence the shorter descriptor yields advantages in the nearest neighbor matching algorithm. The matched feature

TABLE I

COMPARISON BETWEEN SIFT- AND SURF-BASED FEATURE DETECTION ALGORITHMS. THIS IS DONE ON FULL RESOLUTION IMAGE AND ON A ROI IN THE IMAGE.

Algorithm	Time per picture	Number of features
SIFT	4300 ms	2777
SURF	1180 ms	1475
SIFT (ROI)	3100 ms	147
SURF (ROI)	391 ms	116

pairs, and the changes in their bearing due to the robots movement are consulted for calculating depth information (λ). This is done by a triangulation-like linear algebra approach where the $\vec{x}$ represents the current position in global coordinates and $\vec{a}$, $\vec{b}$ are unit vectors describing the directions to the feature. $\vec{x}_1(\lambda) = \vec{x}_{01} + \lambda\vec{a}$, $\vec{x}_2(\kappa) = \vec{x}_{02} + \kappa\vec{b}$ and $\vec{c} = \vec{x}_{02} - \vec{x}_{01}$ leads to the linear system of equations:

$$\left(\begin{array}{cc|c} a_x & -b_x & c_x \\ a_y & -b_y & c_y \end{array}\right)$$

Table I shows our results testing the two algorithms on full resolution images with a special Region Of Interest (ROI) in the image. Both algorithms are not suitable for online data processing. The main drawback is that the processing time of the algorithm depends on the number of pixels. Therefore the number of pixels has to be reduced e.g. by selecting a small amount of interesting sections from the image. Instead of the random selection of Regions Of Interest (ROI) as proposed by Davision et al. [30], our approach is inspired by the biological process of searching for an object in a visual scene from humans [31, 32].

A. *Human visual attention*

Human attention is caught by regions with object-specific features such as color or orientations. The implemented visual attention system consists of a bottom-up part computing data-driven saliency and a top-down part which enables goal-directed search [32]. The most salient regions are detected with respect to color, intensity and orientation. Bottom-up saliency results from uniqueness of features, e.g., a black sheep among white ones, whereas top-down saliency uses features that belong to a specified target, e.g., red when searching for a red ball. The bottom up part, is based on the well-known model of visual attention by Koch & Ullman [33] used by many computational attention systems [34, 35]. It computes saliencies according to the features intensity, orientation, and color and combines them in a saliency map. The most salient region in this map yields the focus of attention. The top-down part uses predefined feature weights to excite target-specific features and inhibit others e.g. for searching interesting red regions. On one hand the feature weights can be learned offline – as we presented in a previous paper with balls [36] – and on the other hand feature weights can be selected from a planning module to initiate a goal directed search. The important difference to our previous work is that we reimplementated the software and reduced the computation time *from* $10\,\mathrm{s}$ for a VGA picture to *less than* $50\,\mathrm{ms}$ [37]. That allows us to process all camera data online. Figure 8 shows four images of a teleoperated quadkopter flight marked with the bottom up saliency regions.

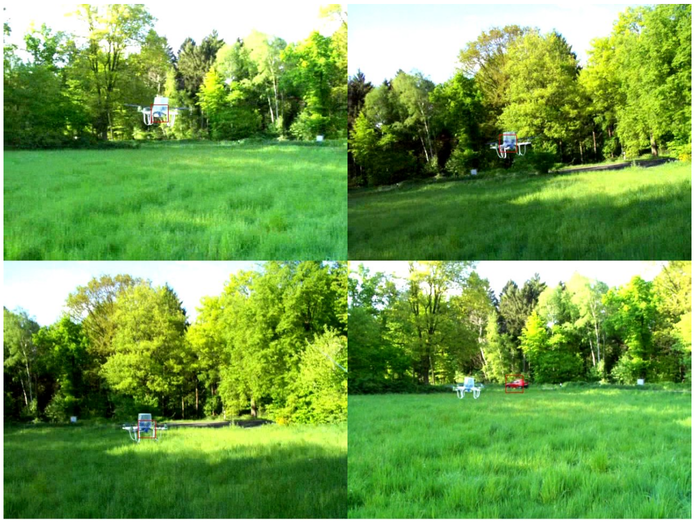


Fig. 8. Most salient regions in four frames of the quadkopter flight. In more than 80% of the images is the quadkopter the most attentive object. When a red car drives through the scene the attention goes to it (bottom right) A video with all images can be found under http://www.volksbot.de/videos/vocus_copter-2008.avi.

In the following we give a brief introduction to the visual attention system VOCUS (Visual Object detection with a CompUtational attention System) which detects salient regions in images.

A.1 **Bottom-up saliency**

The first step for computing bottom-up saliency is to generate image pyramids for each feature to enable computations on different scales. Three features are considered: Intensity, orientation, and color. For the feature intensity, we convert the input image into gray-scale and generate a Gaussian pyramid with 5 scales s_0 to s_4 by successively low-pass filtering and subsampling the input image, i.e., scale $(i+1)$ has half the width and height of scale i.

The intensity maps are created by center-surround mechanisms, which compute the intensity differences between image regions and their surroundings. Two kinds of maps are computed, the on-center maps I''_{on} for bright regions on dark background, and the off-center maps I''_{off}: Each pixel in these maps is computed by the difference between a center c and a surround σ (I''_{on}) or vice versa (I''_{off}). Here, c is a pixel in one of the scales s_2 to s_4, σ is the average of the surrounding pixels for two different radii. This yields 12 intensity scale maps $I''_{i,s,\sigma}$ with $i \,\epsilon\, \{\text{on}, \text{off}\}, s \,\epsilon\, \{s_2\text{-}s_4\}$, and $\sigma \,\epsilon\, \{3, 7\}$.

The maps for each i are summed up by *inter-scale addition* $\bigoplus$, i.e., all maps are resized to scale 2 and then added up pixel by pixel yielding the intensity feature maps $I'_i = \bigoplus_{s,\sigma} I''_{i,s,\sigma}$.

To obtain the orientation maps, four oriented Gabor pyramids are created, detecting bar-like features of the orientations $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$. The maps 2 to 4 of each pyramid are summed up by inter-scale addition yielding 4 orientation feature maps O'_θ.

To compute the color feature maps, the color image is first converted into the uniform CIE LAB color space [38]. It represents colors similar to human perception. The three parameters in the model represent the luminance of the color (L), its position between red and green (A) and its position between yellow and blue (B). From the LAB image, a color image pyramid P_{LAB} is generated, from which four color pyramids P_R, P_G, P_B, and P_Y are computed for the colors red, green, blue, and yellow. The maps of these pyramids show to which degree a color is represented in an image, i.e., the maps in P_R show the brightest values at red regions and the darkest values at green regions. Luminance is already considered in the intensity maps, so we ignore this channel here. The pixel value $P_{R,s}(x,y)$ in map s of pyramid P_R is obtained by the distance between the corresponding pixel $P_{\text{LAB}}(x,y)$ and the prototype for red $r = (r_a, r_b) = (255, 127)$. Since $P_{\text{LAB}}(x,y)$ is of the form (p_a, p_b), this yields: $P_{R,s}(x,y) = ||(p_a,p_b),(r_a,r_b)|| = \sqrt{(p_a - r_a)^2 + (p_b - r_b)^2}$.

On these pyramids, the color contrast is computed by on-center-off-surround differences yielding 24 color scale maps $C''_{\gamma,s,\sigma}$ with $\gamma \,\epsilon\, \{\text{red, green, blue, yellow}\}$, $s \,\epsilon\, \{s_2\text{-}s_4\}$, and $\sigma \,\epsilon\, \{3, 7\}$. The maps of each color are inter-scale added into 4 color feature maps $C'_\gamma = \bigoplus_{s,\sigma} \hat{C}_{\gamma,s,\sigma}$.

A.2 **Fusing Saliencies**

All feature maps of one feature are combined into a conspicuity map yielding one map for each feature: $I = \sum_i \mathcal{W}(I'_i)$, $O = \sum_\theta \mathcal{W}(O'_\theta)$, $C = \sum_\gamma \mathcal{W}(C'_\gamma)$. The bottom-up saliency map S_{bu} is finally determined by fusing the conspicuity maps: $S_{bu} = \mathcal{W}(I) + \mathcal{W}(O) + \mathcal{W}(C)$ The exclusivity weighting $\mathcal{W}$ is a very important strategy since it enables the increase of the impact of relevant maps. Otherwise, a region peaking out in a single feature would be lost in the bulk of maps and no pop-out would be possible. In our context, important maps are those that have few highly salient peaks. For weighting maps according to the number of peaks, each map M is divided by the square root of the number of local maxima m that exceed a threshold t: $\mathcal{W}(M) = M/\sqrt{m} \quad \forall m : m > t$. Furthermore, the maps are normalized after summation relative to the largest value within the summed maps. This yields advantages over the normalization relative to a fixed value (details in [39]).

A.3 **The Focus of Attention (FOA)**

To determine the most salient location in S_{bu}, the point of maximal activation is located. Starting from this point, region growing recursively finds all neighbors with similar values within a threshold and the FOA is directed to this region. Finally, the salient region is inhibited in the saliency map by zeroing, enabling the computation of the next FOA.

A.4 **Search mode**

In search mode, firstly the bottom-up saliency map is computed. Additionally, we can determine a top-down saliency map that competes with the bottom-up map for saliency. The top-down map is composed of an excitation and an inhibition map. The excitation map E is the weighted sum of all feature maps that are important for a goal directed search, namely the features with weights greater than 1. The inhibition map I contains the feature maps that are not present in the goal directed search, namely the features with weights smaller than 1:

$$
\begin{aligned}
E &= \frac{\sum_i (w_i \cdot \text{Map}_i)}{\sum_j (w_j)} && \forall i : w_i > 1, \\
I &= \frac{\sum_i ((1/w_i) \cdot \text{Map}_i)}{\sum_j (w_j)} && \forall i : w_i < 1.
\end{aligned}
$$

The top-down saliency map $S_{(td)}$ is obtained by: $S_{(td)} = E - I$. The final saliency map S is composed as a combination of bottom-up and top-down influences. When fusing the maps, it is possible to determine the degree to which each map contributes by weighting the maps with a top-down factor $t \in [0..1]$: $S = (1-t) \cdot S_{(bu)} + t \cdot S_{(td)}$.

B. *First Results*

First results are shown in Table I. The well known feature extraction algorithms SIFT and SURF are **not** online capable in the current form. Furthermore, Table II indicates the precision of the feature extraction using the SURF algorithm. The robot is moved to several predefined positions and features are calculated in a specific ROI. Afterwards these features are matched to features in the initial image (at position 0). Based on the features the distances are calculated and compared to the real distances. The average error is around 3%.

TABLE II
GROUND TRUTH COMPARISON BETWEEN MEASURED DISTANCES AND CALCULATED DISTANCES WITH THE SURF ALGORITHM. IMAGES ARE TAKEN FROM PREDEFINED POSITIONS ALONG A LINE AND MATCHED TO THE IMAGE POSITION 0.

Position $[cm]$	Number of matched feature points	Average deviation $[\%]$
10	6	3.28
20	6	1.92
30	6	2.47
50	5	1.09
70	6	2.31
100	4	1.89
150	5	2.1

A second result is shown in figure 8 and the corresponding video which show that VOCUS is excellent appropriate for a selection of interesting regions especially in real-time (less then 40 ms for VGA images). The ROIs are used to guide the attention of the teleoperator in USAR missions and to track the quadrotor which is necessary for an autonomous landing at the ground vehicle. Our experimental results show an accuracy of approx. 80% while tracking the quadrotor. The omnidirectional view, the depth information in the images, the gained map and the ability of autonomous recovery of connection failures assists the operator in USAR scenarios.

V. QUADROTOR

This section focuses on the architecture of the Quadrotor. The hardware of the UAV was already described in section II. The UAV is an important component of the *UAGV* system which allows the operator to get a better overview of the environment.

The main goal of our project is to develop a multipurpose teleoperated VTOL platform with such autonomous features as automatic take-off and landing, position control, localization with return back function and obstacle avoidance, suitable for both indoor and outdoor USAR applications. Another crucial feature in our setup is the collaboration with an UGV. Having these autonomous features the copter will not require specially trained personnel to accomplish tasks and will return back or land safely in case of loosing a control signal.

Because of mechanical simplicity compared to conventional helicopters, a four-rotors configuration of our UAV has been chosen. The quadrotor has higher maneuverability than *Lighter-Than-Air* vehicles and *Fixed-Wings* vehicles. However, blimps for instance are easy to control if there are no disturbances like wind, whereas Fixed-Wings vehicles have higher operation ranges. Although there are approaches to increase the maneuverability of Fixed-Wing Micro Aerial Vehicles e.g. to allow hovering [40], the vehicles maximum payload gets reduced.

The quadrotor has two pairs of counter-rotating, fixedpitch blades located at the four corners of the vehicle. It is capable of vertical take-off and landing and it doesn't require complex mechanical control linkages for rotor actuation. Instead, it relies on fixed pitch control. Furthermore, it is capable of changing the moving direction by varying only the motor speeds [41].

The quadrotor is a highly non-linear and unstable platform and requires stability controllers to cope with its fast dynamics. There are many articles with dynamics description and some aspects of computer simulation addressed to these stabilization issues [42–47]. Stabilizing the platform is still a challenging problem. Some expensive VTOL platforms are commercially available[2] but closed for extensions, therefore not usable for research but attempts are being made to introduce such platforms [42].

Most of the researchers are using commercially available toys as HMX-4 or RCtoys' Draganflyer [43]. Our choice fell to non-commercial open-source project *MikroKopter* [5] with available pre-assembled flight and brushless control boards and open-source software. The flight control board contains a 3D accelerometer unit to calculate and align with the gravity component of the earth. In order to provide automatic leveling of the copter, a complementary filter has been implemented that processes the integrated angular velocity of three gyroscopes and the calculated Euler angles from the accelerometers. The output of the filter is used in a Proportional-Integral-Derivative (PID) controller.

Figure 9 shows the functional diagram of the quadrotor. The main component is the flight controller in the middle of the diagram. The flight controller board with the pressure sensor, gyros and a accelerator, controls the 4 brushless motors via four brushless controllers which are connected to the flight controller over an I^2C bus. Each of these units needs 5V power just as the Hokuyo laser scanner and the GUMSTIX boards. The Bluetooth device as well as the 3D compass need only 3V power supply which is generated by a voltage converter. Signals are then translated via a signal level translator to satisfy flight-control microprocessor's requirements. The total quadrotor device is controlled by a $40\,\mathrm{MHz}$ analog radio link.

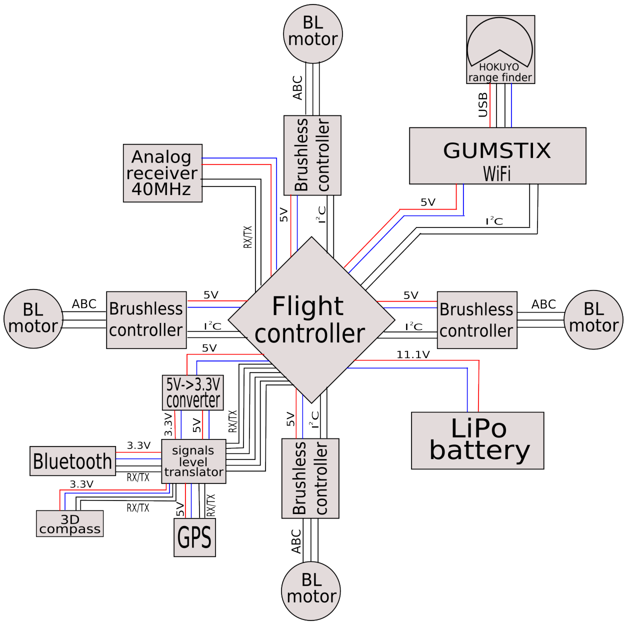


Fig. 9. The quadrotor's functional diagram with distributed laser range-finder data processing.

Natural drift of the gyros, accelerometers and constant air movement (wind, convention currents) makes it difficult to achieve a stable hovering for a long period of time. The position drifting problem is managed for outdoor environments using GPS [48–50]. However, for indoor use GPS becomes inapplicable due to low signal strengths. There are also difficulties because of low payload of UAVs and integrated hardware which limits the platform to be extended with additional sensors. Roberts et al. [51] use a platform that is similar to ours except for being slightly modified to increase circuit integration. The platform was equipped with sonar for altitude control and four infrared range finders for hovering control. Matsue et al. [52] employed three infrared range sensors to measure the height above the ground and the distances to two perpendicular walls. Roberts' platform showed good hovering results in empty rooms and was able to avoid large obstacles while Matsue's could follow walls. Kim [53] used 6 degrees of freedom inertial unit for conventional MicroKopter hovering stabilization. Other approaches to the drifting problem employ external sensing for position stabilization. Castillo et. al [44] used Polhemus sensor for position tracking, Mori [54] proceeded on-board camera data on external PC and Gurdan [55] performed experiments in a laboratory equipped with an indoor motion tracking system by VICON that can measure the position vector of specific points on the body of the robot.

Although significant results were achieved, existing approaches lack of flexibility. Using external localization systems limits the copter's workspace to the area visible by that system. For avoiding collisions with obstacles and navigating in office buildings still no sufficient results have been shown.

[2] http://www.airrobot.de/, http://www.microdrones.com/

A. First results

At the current state of the project we achieved significant altitude and attitude stability of the platform with 3-axes gyroscopes and accelerometers and an air pressure sensor. The copter is remotely controlled and able to fly indoor and outdoor with just little adjustment of the flight trajectory by the operator. Altitude control is performed automatically according to the set point defined by the operator. Also the orientation is automatically controlled according to the starting position given during take-off by the 3D electronic compass. The quadrotor hovering near a window on our campus at Castle Birlinghoven in Sankt Augustin is shown in Figure 10.

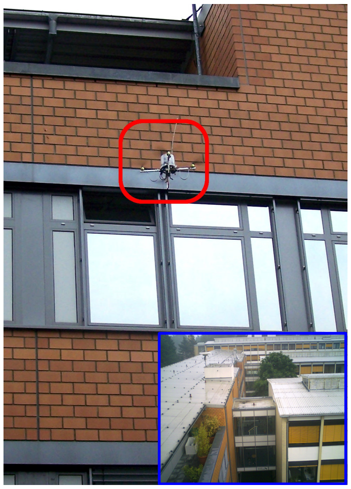


Fig. 10. Quadrotor (red square) flying near the building. It can carry a camera which allows us to acquire an interior overview through windows. Lower right (blue square) shows the camera view of the quadrotor flying above roofs.

Nevertheless it still requires operator presence in visible distance. To extend outdoor exploration options the quadrotor is equipped with GPS for position control. For remote software debugging, parameters transfer and future remote control using mobile devices the Free2move Bluetooth module [56] is used. The quadrotor is a valuable supplement in an USAR scenario especially if it cooperates with an UGV system.

VI. Conclusions and Future Work

This paper presented our robotic system *UGAV* consisting of two semi-autonomous robot platforms, an UGV (*VolksBot*) and an UAV (*Quadrotor*). Furthermore, we described three main topics of combined UGV and UAV. (1) The teleoperated control with cell or smart phones with the new concept of automatic configuration of the smart phone based on a RKI-XML description of the vehicle control capabilities. (2) The camera and vision system with the focus to real time feature extraction e.g. for the tracking of the UAV and (3) the architecture and hardware of the UAV.

Needless to say a that lot of work remains to be done:

1. A future version, the OCU will be able to use Bluetooth or telephone numbers and supports the searching for the best communication channel.
2. With the vision system we will build maps online. Since the spatial information of features is known, they will be transformed into landmarks. These landmarks are inserted into a global map representing the environment. With the help of the map the robot is able to localize itself by comparing the current feature bearings with the stored landmarks. This will be done by triangulation and a linear algebra approach. The obtained map is useful if the remote connection to the robot is lost. In future work the robot should be able to backtrack its path autonomously until it reaches a position where the connection can be reestablished. In case of a complete breakdown of communication the return to the initial position is feasible.
3. The quadrotor will be equipped with a HOKUYO laser range-finder [57] pursues the goal of multiple obstacles avoidance along with solving the hovering drift problem and flying indoor. With more and more increasing density of integrated circuits and increasing computational power it becomes possible to overcome computational expenses for micro aerial vehicles. Gumstix [58] 400MHz embedded computer running under Linux OpenEmbedded enables to efficiently process the laser data, to correct trajectory fluctuations and to compute obstacles avoidance commands. Total weight of 200 g for the navigation system is within the normal payload of our quadrotor. Furthermore, it provides interfaces e.g. 3 x RS232 ports, USB host, I^2C bus, WiFi and microSD card for information storage.
4. With GPS we are planning to improve hovering stability, resistance against wind and implement semi-autonomous navigation which will include returning to the UGV or approaching a requested position as well as following GPS tracks.

Acknowledgments

The authors would like to thank our Fraunhofer colleagues for their support namely Reiner Frings, Jochen Winzer, Stefan May, Paul Plöger and Kai Pervölz. This work was partly supported by the DFG grant CH 74/9-1 (SPP 1125).

References

[1] R.R. Murphy, “Marsupial and shape-shifting robots for urban search and rescue”, *Intelligent Systems and Their Applications, IEEE [see also IEEE Intelligent Systems]*, vol. 15, no. 2, pp. 14–19, Mar/Apr 2000.

[2] G. S. Sukhatme, J. F. Montgomery, and R. T Vaughan, *Robot Teams: From Diversity to Polymorphism*, chapter Experiments with Cooperative Aerial-Ground robots, pp. 345–368, A.K. Peters, Ltd., 2002.

[3] Thomas Wisspeintner, Walter Nowak, and Ansgar Bredenfeld, *RoboCup 2005: Robot Soccer World Cup IX*, vol. 4020, chapter VolksBot - A Flexible Component-Based Mobile Robot System, pp. 716–723, Springer Berlin Heidelberg, 2006.

[4] Fraunhofer Institut Intelligent Analyse und Informationssysteme (IAIS), “Volksbot”, http://www.volksbot.de/, 2008.

[5] Holger Buss and Ingo Busker, “Mikrokopter”, http://www.mikrokopter.de/, May 2008.

[6] S. Tachi, “Real-time remote robotics-toward networked telexistence”, *Computer Graphics and Applications, IEEE*, vol. 18, no. 6, pp. 6–9, Nov/Dec 1998.

[7] R.R. Murphy, “Trial by fire [rescue robots]”, *Robotics & Automation Magazine, IEEE*, vol. 11, no. 3, pp. 50–61, Sept. 2004.

[8] T. B. Sheridan, “Teleoperation, telerobotics and telepresence: A progress report”, *ControlEng. Practice.*, vol. 3, no. 2, pp. 205–214, February 1995.

[9] J. Casper and R.R. Murphy, “Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center”, *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, vol. 33, no. 3, pp. 367–385, June 2003.

[10] P. Fiorini and R. Oboe, “Internet-based telerobotics: problems and approaches”, *Advanced Robotics, 1997. ICAR ’97. Proceedings., 8th International Conference on*, pp. 765–770, July 1997.

[11] P.A. Roche, M. Sun, and R.J. Sclabassi, “Using a cell phone for biotelemetry”, *Proceedings of the IEEE 31st Annual Northeast Bioengineering Conference*, pp. 65 – 66, April 2005.

[12] A. Sekmen, A.B. Koku, and S. Zein-Sabatto, “Human robot interaction via cellular phones”, *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 4, pp. 3937–3942, Oct. 2003, ISBN: 0-7803-7952-7.

[13] B.A. Myers, J. Nichols, J.O. Wobbrock, and R.C. Miller, “Taking handheld devices to the next level”, *Computer*, vol. 37, pp. 36 – 43, December 2004.

[14] Bradley J. Betts, Robert W. Mah, Richard Papasin, Rommel Del Mundo, Dawn M. McIntosh, and Charles Jorgensen, “Improving situational awareness for first responders via mobile computing”, Technical memorandum, NASA Ames Research Center, Moffett Field, CA 94035-1000, March 2005.

[15] Andres Industries, ”, http://www.andres-industries.de/, May 2008.

[16] W. Webb, “From ”cellphone” to ”remote control on life”: how wireless communications will change the way we live over the next 20 years”, *2002 IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, pp. 7 – 11, June 2002.

[17] “Android - an open handset alliance project : What is android”, http://code.google.com/android/what-is-android.html, May 2008.

[18] Kim Topley, *J2ME in a Nutshell*, O’Reilly, March 2002, ISBN: 0-596-00253-X.

[19] “Android - an open handset alliance project: Location-based service apis”, http://code.google.com/android/toolbox/apis/lbs.html, May 2008.

[20] “Android - an open handset alliance project: Android emulator”, http://code.google.com/android/reference/emulator.html, March 2008.

[21] “Volksbot videos”, http://www.youtube.com/user/Volksbot, May 2008.

[22] J. Craighead, B. Day, and R. Murphy, “Evaluation of canestas range sensor technology for urban search and rescue and robot navigation”, http://www.crasar.org/, 2006.

[23] R. Murphy, J. Casper, J. Hyams, M. Micire, and B. Minten, “Mobility and sensing demands in usar”, *Industrial Electronics Society, 2000. IECON 2000. 26th Annual Confjerence of the IEEE*, vol. 1, pp. 138–142 vol.1, 2000.

[24] Robin R. Murphy, “Rescue robotics for homeland security”, *Communications of the ACM, Special Issue on Emerging technologies for homeland security*, vol. 47, pp. 66–68, March 2004.

[25] David G. Lowe, “Distinctive image features from scale-invariant keypoints”, in *International Journal of Computer Vision*, November 2004, vol. 60, pp. 91–110.

[26] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, “Surf: Speeded up robust features”, in *Computer Vision ECCV 2006*. 2006, vol. 3951/2006, pp. 404–417, Springer Berlin / Heidelberg.

[27] R. Sim, P. Elinas, M. Griffin, and J. J. Little, “Vision-based SLAM using the Rao-Blackwellised particle filter”, in *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR)*, Edinburgh, Scotland, 2005, pp. 9–16.

[28] N Karlsson, “The vslam algorithm for robust localization and mapping”, in *Int. Conf. on Robotics and Automation*, 2005.

[29] Hauke Strasdat, Cyrill Stachniss, Maren Bennewitz, and Wolfram Burgard, “Visual bearing-only simultaneous localization and mapping with improved feature matching”, in *Fachgesprche Autonome Mobile Systeme (AMS)*, 2007.

[30] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse, “Monoslam: Real-time single camera slam”, in *TRANSAC TIONS ON PATTERN ANALYSI S AND MACHINE INTELLIGENCE*, June 2007, vol. 29.

[31] U. Neisser, *Cognitive Psychology*, Appleton-Century-Crofts, N.Y., 1967.

[32] Simone Frintrop, *VOCUS: A Visual Attention System for Object Detection and Goal-Directed Search*, PhD thesis, University of Bonn, July 2005.

[33] C. Koch and S. Ullman, “Shifts in selective visual attention: towards the underlying neural circuitry”, *Human Neurobiology*, pp. 219–227, 1985.

[34] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis”, *IEEE Trans. on Pattern Analysis & Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[35] G. Backer, B. Mertsching, and M. Bollmann, “Data- and model-driven gaze control for an active-vision system”, *IEEE Trans. on Pattern Analysis & Machine Intelligence*, vol. 23(12), pp. 1415–1429, 2001.

[36] Sara Mitri, Simone Frintrop, Kai Pervölz, Hartmut Surmann, and Andreas Nüchter, “Robust Object Detection at Regions of Interest with an Application in Ball Recognition”, in *Proceedings IEEE 2005 International Conference Robotics and Automation (ICRA ’05)*, Barcelona, Spain, April 2005, pp. 126–131, Conf-A.

[37] Stefan May, Maria Klodt, Erich Rome, and Ralph Breithaupt, “Gpu-accelerated affordance cueing based on visual attention”, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, USA*, 2007.

[38] R. E. Burger, *Colormanagement. Konzepte, Begriffe Systeme*, Springer, 1997.

[39] S. Frintrop, E. Rome, A. Nüchter, and H. Surmann, “A bimodal laser-based attention system”, *J. of Computer Vision and Image Understanding (CVIU), Special Issue on Attention and Performance*, 2005.

[40] W. E. Green and P. Y. Oh, “A fixed-wing aircraft for hovering in caves, tunnels, and buildings”, *American Control Conference, 2006*, p. 6, 2006.

[41] G.M. Hoffmann, H. Huang, S.L. Waslander, and C.J. Tomlin, “Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment”, *Proc. AIAA Guidance, Navigation, and Control Conf., Hilton Head, SC, August*, 2007.

[42] P. Pounds, R. Mahony, and P. Corke, “Modelling and Control of a Quad-Rotor Robot”, *Proceedings of the Australasian Conference on Robotics and Automation*, 2006.

[43] P. McKerrow, “Modelling the Draganflyer four-rotor helicopter”, *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*, vol. 4, pp. 3596–3601, 2004.

[44] P. Castillo, A. Dzul, and R. Lozano, “Real-time stabilization and tracking of a four-rotor mini rotorcraft”, *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 4, pp. 510–516, 2004.

[45] L. Beji and A. Abichou, “Streamlined Rotors Mini Rotorcraft: Trajectory Generation and Tracking”, *International Journal of Control, Automation, and Systems*, vol. 3, no. 1, pp. 87–99, 2005.

[46] S. Bouabdallah, P. Murrieri, and R. Siegwart, “Design and control of an indoor micro quadrotor”, *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*, vol. 5, pp. 4393–4398, 2004.

[47] A. Mokhtari and A. Benallegue, “Dynamic feedback controller of Euler angles and wind parameters estimation for a quadrotor unmanned aerial vehicle”, *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*, vol. 3, pp. 2359–2366, 2004.

[48] “Microdrones gmbh”, http://www.microdrones.com, May 2008.

[49] O. Meister, R. Mönikes, J. Wendel, N. Frietsch, C. Schlaile, and G.F. Trommer, “Development of a GPS/INS/MAG navigation system and waypoint navigator for a VTOL UAV”, *Proceedings of SPIE*, vol. 6561, pp. 65611D, 2007.

[50] E. Courses and T. Surveys, “Enhancement of GPS Signals for Automatic Control of a UAV Helicopter System”, *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*, pp. 1185–1189, 2007.

[51] J.F. Roberts, T. Stirling, J.C. Zufferey, and D. Floreano, “Quadrotor Using Minimal Sensing For Autonomous Indoor Flight”, *European Air Vehicle Conference and Flight Competition EMAV2007*, 2007.

[52] A. Matsue, W. Hirosue, H. Tokutake, S. Sunada, and A. Ohkura, “Navigation of Small and Lightweight Helicopter”, *Trans. Jpn. Soc. Aeronaut. Space Sci*, vol. 48, pp. 177–479, 2005.

[53] B. KIM, Y. CHANG, and M.H. LEE, “System Identification and 6-DOF Hovering Controller Design of Unmanned Model Helicopter”, *JSME International Journal Series C*, vol. 49, no. 4, pp. 1048–1057, 2006.

[54] R. Mori, T. Kubo, and T. Kinoshita, “Vision-Based Hovering Control of a Small-Scale Unmanned Helicopter”, *SICE-ICASE, 2006. International Joint Conference*, pp. 1274–1278, 2006.

[55] D. Gurdan, J. Stumpf, M. Achtelik, K.M. Doth, G. Hirzinger, and D. Rus, “Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz”, *Robotics and Automation, 2007 IEEE International Conference on*, pp. 361–366, 2007.

[56] “Technical description of free2move bluetooth module f2m03gx / gxa”, http://www.free2move.se/pdf/F2M03GX_GXA.pdf, May 2008.

[57] “Hokuyo automatic co. ltd.”, http://www.hokuyo-aut.jp/02sensor/07scanner/urg.html, May 2008.

[58] “gumstix inc.”, http://www.gumstix.com/, May 2008.