UDC 004.932

C. Netramai,
H. Roth, Prof. Dr.-Ing.

REAL-TIME 3D MOTION ESTIMATION AND MAP BUILDING USING ENHANCED MULTI-CAMERA SYSTEM

Наведено вдосконалений метод побудови 3D фото реалістичної карти для систем з багатьма камерами. Основним досягненням є реалізація нового ймовірнісного підходу для побудови карт з використанням алгоритму FastSLAM 2.0. Запропонований алгоритм успішно реалізовано для систем з багатьма камерами, він здатний працювати в реальному часі. Експерименти у приміщенні виявили значне поліпшення роботи запропонованого алгоритму в порівнянні зі звичайним підходом.

Представлен усовершенствованный метод построения 3D фото реалистичной карты для систем с несколькими камерами. Основным достижением является реализация нового вероятностного подхода для построения карт с использованием алгоритма FastSLAM 2.0. Предложенный алгоритм успешно реализован для систем с несколькими камерами и способен работать в реальном масштабе времени. Эксперименты в помещении выявили значительное улучшение работы предложенного алгоритма по сравнению с обычным подходом.

This work presents an enhanced multi-camera system for a 3D photo-realistic map building task. The main contribution is the implementation of a new probabilistic approach for map building using FastSLAM 2.0 algorithm. The proposed algorithm is successfully implemented on the multi-camera system and is capable of real-time operation. Experiments within an indoor environment were conducted and the results show good improvement over the conventional approach.

Introduction. Map building is an active research topic among the machine vision community. In most cases the map is not available beforehand and the autonomous system has to incrementally build the map with zero knowledge. This brings forward a combination of the localization and mapping problem known as simultaneous localization and mapping or SLAM [1;4]. Although SLAM can be handled using a single stereo camera system [2;8] there is still a certain shortcoming. The main problem of the single camera implementation is the small field of view which leads to motion ambiguity error of the translation along the optical axis and small rotation about the axis perpendicular to the optical axis [5]. A solution to this problem is to use an optical lens with larger field of view. However, using optical lens with large field of view reduces the quality of the stereo calculation [3] which consequently results in the poor 3D map quality.

In previous work [6] a multi-camera system is introduced to overcome these problems. The multi-camera unit or so called MCU uses unique hardware arrangement to overcome motion ambiguity as well as maintaining the performance of the stereo algorithm since an optical lens with large field of view is not needed. More cameras also mean more data can be acquired at a time which is a big advantage for the map building task. The MCU is capable of estimating six degree of freedom (DoF) motion and producing a photo-realistic map of the observed environment in real-time. However the system faces difficulties when large measurement error and motion estimation error arise which lead to the corruption of the algorithm.

This paper presents a further enhancement to the existing MCU system. A stochastic approach for real-time SLAM, namely the FastSLAM 2.0 [4], is implemented in order to deal with the measurement and motion estimation error. FastSLAM 2.0 is an approach to the SLAM problem based on particle filtering where feature points within the map are estimated conditionally independent to the given camera trajectory. This approach eases the implementation for real-time system compared to EKF-based SLAM approach since the calculation complexity is lower. Section 3 describes FastSLAM 2.0 in more detail.

This paper is organized as follows-section 2 gives a brief review of the MCU system, section 3 describes the FastSLAM 2.0 implementation using MCU system, section 4 presents the experiment results and the last section contains the conclusion of this work

1. Multi-camera unit

The multi-camera unit (MCU) introduced in [6] is designed to improve sensitivity of the real-time 3D motion estimation as well as to increase the data acquisition rate for 3D map building task. By having three cameras pointing all perpendicular to each other in x-, y- and z-axis the correct six DoF motion can be efficiently detected since the motion ambiguity from one camera is always canceled out by the other two cameras. The raw output from the MCU includes six 2D images and three 3D images which are all calibrated and referred to the same reference frame. Fig.1 shows the current MCU hardware and its example output images.

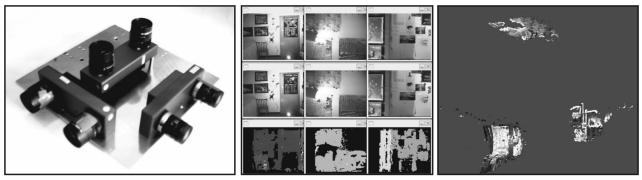


Fig. 1. The MCU hardware (a); the 2D images and disparity images from all three stereo cameras (b); the 3D points cloud composed of 3D images from all stereo cameras on the MCU) (c)

The overview of the MCU system is shown in Fig.2. The main processes include feature points extraction, motion estimation between two successive frames and map building. The result of these operations is a 3D photo-realistic map of the observed environment as well as the detailed 3D trajectory of the MCU during the map building process.

The detailed description of each process in the MCU system is given in the following subsections.

1.1. Feature extraction

Corner-type feature is used as feature point in this work. The corner response function (CRF) is used to extract corners from the 2D images. The goal of the CRF is to determine the response value of the target pixel which is defined by

$$R_{CRF} = \min((i_p - i_c)^2 + (i_{p'} - i_c)^2). \qquad (1)$$

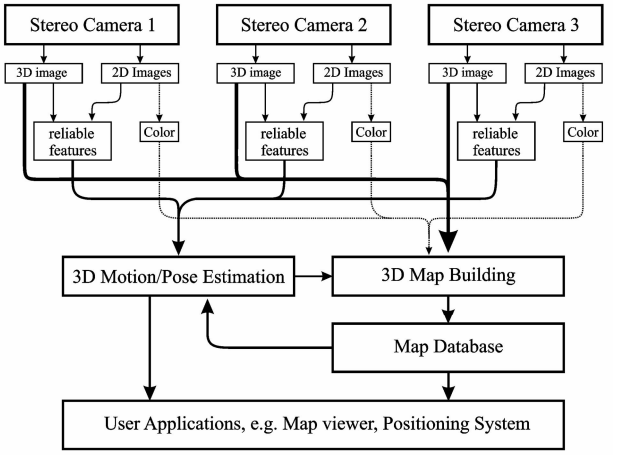


Fig. 2. System overview

Where i_p and $i_{p'}$ is the intensity of point p and p' whose locations defined by the specific shape of the mask and i_c is the intensity of the target pixel. More information about the *CRF* algorithm can be found in [9].

1.2. Feature matching

Feature matching relates the same feature point between two successive frames. The normalized cross correlation (NCC) is used for the 2D feature matching. The NCC coefficient between point f and t is defined by

$$\frac{\sum_{x,y}\left[f_{(x,y)} - \bar{f}\right]\left[t_{(x-u,y-v)} - \bar{t}\right]}{\left\{\sum_{x,y}\left[f_{(x,y)} - \bar{f}\right]^2 \sum_{x,y}\left[t_{(x-u,y-v)} - \bar{t}\right]^2\right\}^{0,5}}. \qquad (2)$$

Where $f_{(x,y)}$ and $t_{(x,y)}$ are the pixel intensity at corresponded locations within previous and current image frame and $\bar{f}$, and $\bar{t}$ are the mean intensity in the 9×9 pixels region under point f and t respectively.

1.3. Outlier detection

In practice it is possible that the NCC gives wrong matches. Two outlier detectors are implemented to eliminate these outliers. The first one uses color information or more precisely the hue value of the target pixels. The hue value is represented in degree and it runs from 0° to 360°. To simplify the process, the hue value is quantized using 60° interval (0°, 60°, 120°,…, 360°). If the hue values of the matched feature on both successive image frames are different then the match is considered an outlier. The second outlier detector uses motion information detected using 2D features. If the matched fea-

ture has different motion direction than the majority of the matches then it is considered an outlier. This motion detection using 2D features is described in the section 2.4.

1.4. 3D motion detection using 2D features

By using just the 2D images from the MCU, a very accurate 3D motion can be detected. This is done by simply calculating the 2D movement (up, down, left and right) of x-, y- and z-camera. This movement is derived from the displacement of the 2D feature points between two successive frames. By putting together the 2D movement information from all three cameras the direction of the 3D motion can be found. Fig. 3 illustrates the direction of the 2D movement from all three cameras and Table 1 gives some examples of the resultant motion according to this combination scheme.

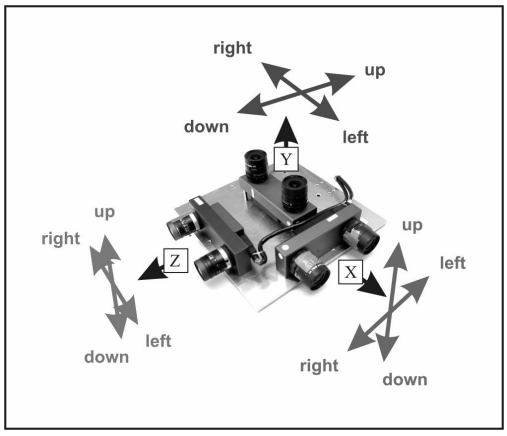


Fig. 3. Direction of the 2D movement of all three stereo cameras on the MCU

1.5. Six DoF motion estimation

Now that the correct matched feature points are found, the six DoF motion (*Tx, Ty, Tz, roll, pitch, yaw*) can be estimated. This is done by applying the RANSAC algorithm to find initial value of the transformation parameters between two successive frames. These initial values are then used by the iterative closest point (ICP) algorithm to iteratively determine the final motion estimation.

Applying the ICP algorithm is straightforward since the correspondences between feature points are known. The ICP algorithm determines the optimal rotation R and translation t by iteratively minimizing the Euclidean distance between the two set of matched feature points

$$\min \left\| p - p'(R,t) \right\| . \quad (3)$$

Where p and p' are the feature points from the previous time step and current time step respectively. More detail about motion estimation can be found in our previous work [9] where the similar approach is used.

Table 1. MCU motion using 2D movement information

Cam X	Cam Y	Cam Z	MCU motion
up	-	up	up
down	-	down	down
-	left	left	forward
-	right	right	backward
up	right	-	ccw, z-axis
down	left	-	cw, z-axis
left		left	ccw, y-axis
right		right	cw, y-axis

(note: cw=clockwise, ccw=counter-clockwise)

2. 3D map building using FastSLAM 2.0

In order to obtain a real-time performance the use of full resolution 3D images is avoided. It is sufficient and more efficient to use just the feature points since the amount of data points to process is much lower. This can be safely done since the feature points are actually sharing the same coordinate with the full resolution 3D images and the resultant transformation derived using feature points can be directly applied to the full resolution images to produce photo-realistic 3D map.

Let the map consisting of N feature points be denoted as $\Theta = \theta_1, \ldots, \theta_N$ and the trajectory of the MCU is denoted as $s^t = s_1, \ldots, s_t$ where t is the time index and s_t is the pose of the MCU at time t. The goal of the SLAM algorithm is to estimate the posterior distribution

$$p(\Theta, s^t \mid z^t, u^t, n^t) . \quad (4)$$

Where $z^t = z_1, \ldots, z_t$ is the sequence of measurements of the feature point locations and $u^t = u_1, \ldots, u_t$ is the sequence of the MCU motion estimation. In [4], FastSLAM 2.0 factorizes this posterior distribution as

$$p(\Theta, s^t \mid z^t, u^t, n^t) = p(s^t \mid z^t, u^t, n^t) \prod_{n=1}^{N} p(\theta_n \mid s^t, z^t, u^t, n^t) . \quad (5)$$

The trajectory of the MCU is sampled using particle filter with *m* particles and the feature point locations are estimated independently of each other using *N* EKFs.

At each time step, the new poses of the MCU within each particle are sampled based on the latest trajectory information and the measurement of the feature point locations

$$s_t^{[m]} \sim p(s_t \mid s^{t-1,[m]}, u^t, z^t, n^t). \quad (6)$$

The feature points are updated according to the following posterior

$$p(\theta_{n_t} \mid s^{t,[m]}, n^t, z^t) = \\ = \eta p(z_t \mid \theta_{n_t}, s_t^{[m]}, n_t)\, p(\theta_{n_t} \mid s^{t-1,[m]}, z^{t-1}, n^{t-1}). \quad (7)$$

Where η is a normalize $\eta^{[m]} = p(z_t \mid s^{t-1,[m]}, u^t, z^{t-1}, n^t)^{-1}$. In the next step the importance weight for each particle is calculated

$$w_t^{[m]} = \frac{p(s^{t,[m]} \mid z^t, u^t, n^t)}{p(s^{t-1,[m]} \mid z^{t-1}, u^{t-1}, n^{t-1})\, p(s_t^{[m]} \mid s^{t-1,[m]}, z^t, u^t, n^t)}. \quad (8)$$

Finally the resample of the particle is done based on the calculated importance weight

$$p(s^{t,[m]}) \in S_t \propto w_t^{[m]}. \quad (9)$$

At each iteration, the MCU position and orientation as well as the feature point locations are updated using the state variable of the particle with highest importance weight. The full resolution 3D images are then merged to the existing map according to the most up-to-date position and orientation of the MCU.

The FastSLAM 2.0 and all other software modules within this work are implemented using C++ language. The operational speed of the MCU (including the image grabbing from three stereo cameras, stereo image calculation, feature extraction, feature matching, motion estimation and FastSLAM with 100 particles) is approximately 2 Hz on a PC system with Intel Pentium Core 2 Quad 2.4 GHz CPU and 4 GB RAM running 32-bits Linux operating system.

3. Results

Some significant results are presented here to compare the performance of the MCU against a single camera system. A single stereo camera system is emulated by using only one camera on the MCU to supply input images to the same software algorithm. Several tests including motion estimation using 15 degree rotation around y-axis (illustrated in Fig.4,a) and motion estimation using 50 mm translation along y-axis (illustrated in Fig.4,d) are conducted and the results are shown in Fig.4.

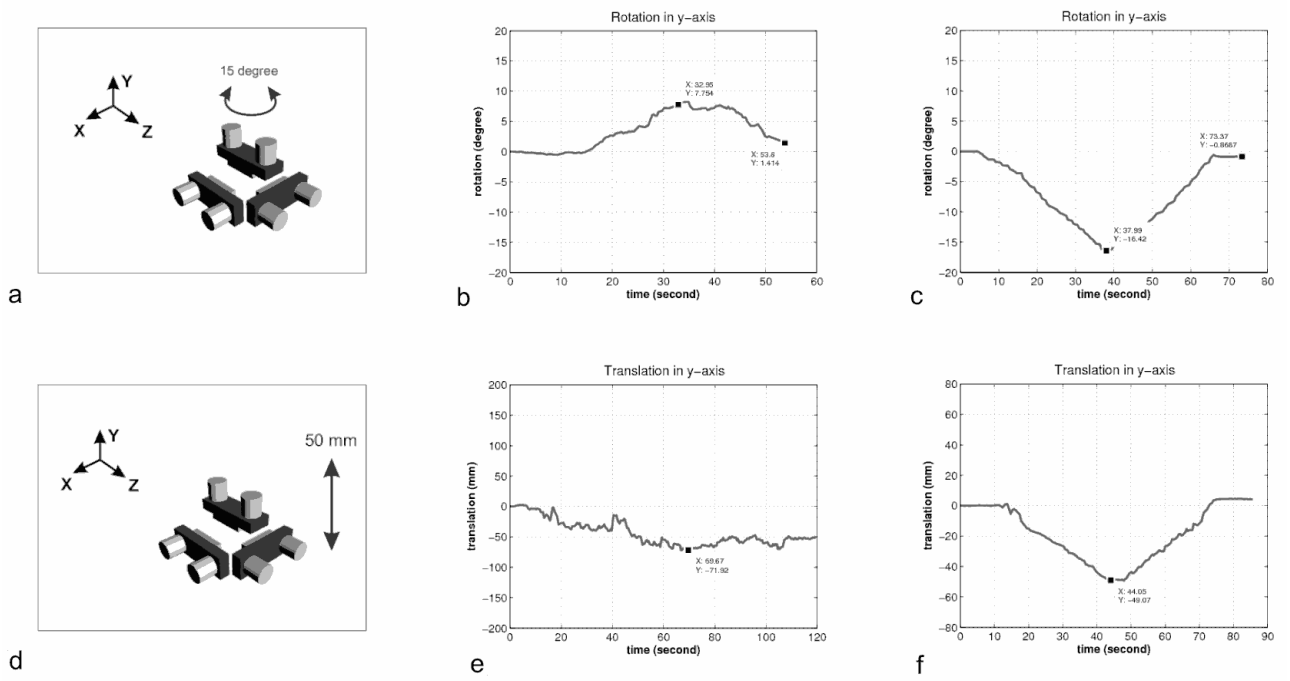


Fig. 4. Motion estimation results using single camera and multi-camera system: a – motion estimation using 15 deg rotation around y-axis; b – rotation estimation using only the x-axis camera; c – rotation estimation using three cameras; d – motion estimation using 50 mm translation along y-axis; e – translation estimation using only the y-axis camera; f – translation estimation using three cameras

By using only one camera, namely the x-axis camera, the small rotation during each movement is mistaken as translation due to the motion ambiguity. This motion ambiguity occurs since the rotation takes place in the axis that is perpendicular to the optical axis and therefore the rotation estimation result is poor (Fig.4,b) compared to the result obtained using three cameras setup (Fig.4,c) where the estimated rotation (-16.42 degree) is close to the real value (-15.00 degree).

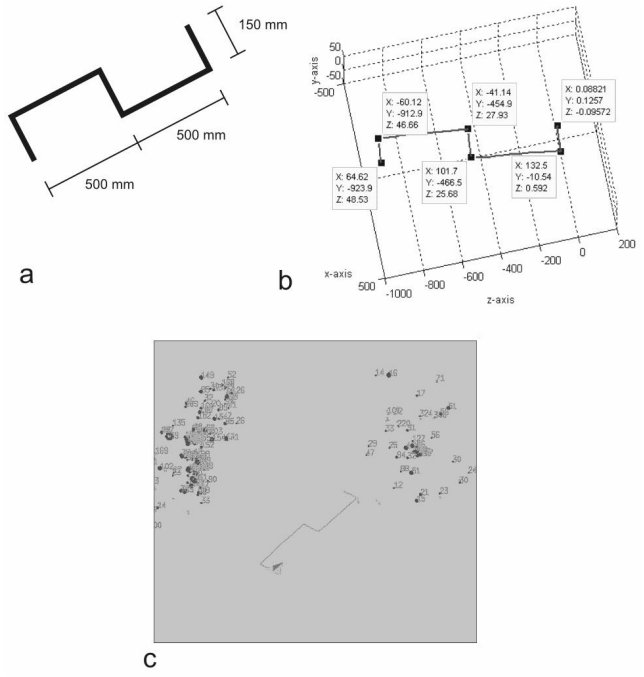


Fig. 5. Motion estimation along predefined trajectory (a) actual trajectory (b) estimated trajectory and (c) real-time plot of the trajectory and the feature points being observed during the test

For pure translation, the weakness of a single camera setup can be observed when the direction of translation is parallel to the optical axis of the camera because the error from stereo depth estimation corrupts the motion estimation process. By using only the y-axis camera to observe the translation along y-axis an inaccurate test result is obtained (Figure 4e) while the result obtained using the setup with three cameras (Fig.4,f) shows a good estimated value (-49.07 mm) compared to the real value (-50 mm).

In the next test the MCU is moved along a predefined trajectory which consists of five straight line segments (Fig.5, a). The MCU system produces the resultant trajectory which is close to the real trajectory. This result is shown in Fig. 5.

For the map building task, the MCU system captures multiple 3D images from all stereo cameras and incrementally merges them to the existing map in order to create a photo-realistic map. Fig.6 shows an example of a 3D photo-realistic model obtained from the MCU system.

The quality of the obtained map can be observed in Fig.7 where the close-up view of the finished 3D model is rendered at high resolution.

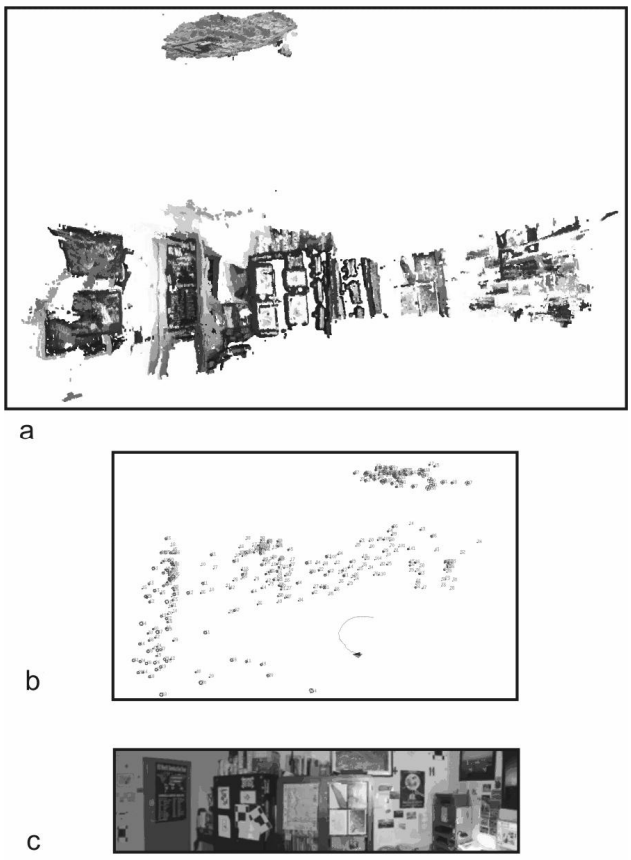


Fig. 6. An example of 3D photo-realistic model obtained from the MCU system (a), the 3D model of the test environment captured after a half circle trajectory (b), feature points and MCU trajectory maintained by FastSLAM algorithm (c) the panoramic image of the test environment

Fig. 7. Close up of the 3D photo-realistic model

Conclusion

This work presents an enhanced multi-camera system for real-time 3D motion estimation and map building task. The main improvement is the full implementation of the Fast-SLAM 2.0 approach which improves the robustness of the system against the feature point measurement and motion estimation error. The system has been tested within an indoor environment for motion estimation and map building task where accurate six DoF motion is correctly estimated in real-time. Lastly a 3D photo-realistic map of the test environment with good texture quality is also obtained.

References

1. Dissanayake G., Newman P., Clark S., Durrant-Whyte F., Csorba M. A solution to the Simultaneous Localization and Map Building (SLAM) Problem / G.Dissanayake, P.Newman, S.Clark, F.Durrant-Whyte, M.Csorba // IEEE Transaction on Robotics and Automation, – Vol. 17. – No. 3, 2001.

2. Garcia M. A., Solanas A. 3D Simultaneous Localization and Modeling from Stereo Vision, International Conference on Robotics & Automation, New Orleans, LA, April 2004.

3. Knöppel C. Stereobasierte und Spurgenaue Erkennung von Straßenfahrzeugen im Rückraum eines Straßenfahrzeuges, Dissertation, University of Magdeburg, 2001, Page 26-29.

4. Montemerlo M., Thrun S., Koller D., Wegbreit B. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges, Proc. IJCAI, Acapulco, Mexico, August 2003.

5. Negahdaripour S, Yu C. H. Robust Recovery of Motion: Effects of Surface Orientation and Field of View, Proc. CVPR, University of Michigan, Ann Harbor, June, 1988.

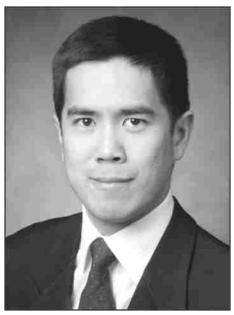
6. Netramai C., Roth H., Real-time Photo-realistic 3D Map building using Mobile Multiple Stereo Cameras Setup, 3DTV Conference 2007, Kos Island, Greece, May 2007.

7. Netramai C., Melnychuk O., Joochim C., Roth H. Combining PMD and Stereo Camera for Motion Estimation of a Mobile Robot, The 17th IFAC World Congress, Seoul, Korea, July 2008.

8. Nistér D., Naroditsky O., Bergen J., Visual Odometry, Proc. CVPR 2004, Washington, DC, June 2004.

9. Trajković M., Hedley M. Fast Corner Detection, Image and Vision Computing, Volume 16, Issue 2, 1998, Pages 75-87.

Received 07.02.2011



M.Sc. Chayakorn Netramai
Doctoral student at the Center for Sensor System (ZESS), University of Siegen, Germany
E-Mail: netramai@ipp.zess.uni-siegen.de

Postal Adresse: Zentrum für Sensorsysteme Universität Siegen, Paul-Bonatz-Strasse 9-11, 57076 Siegen, Germany

Telephone: +49 271 740-3323
Fax: +49 271 740-2336

Prof. Dr.-Ing. Hubert Roth
Head of the Institute of Automatic Control Engineering (RST), University of Siegen, Germany
E-Mail: hubert.roth@uni-siegen.de

Postal Adresse: Elektrotechnik und Informatik Lehrstuhl für Regelungs- und Steuerungstechnik (RST), Hölderlinstr. 3, 57076 Siegen, Germany

Telephone: +49 271 740-4439
Fax: +49 271 740-4382