# Efficient Alignment of Aerial Images Based on Virtual Forces

Claudius Stern, Christoph Rasche, Lisa Kleinjohann and Bernd Kleinjohann
Faculty of Computer Science, Electrical Engineering and Mathematics,
Department of Computer Science, C-LAB,
University of Paderborn, Germany
e-mail: {claudis, crasche, lisa, bernd}@c-lab.de

***Abstract*—Getting a contemporary aerial overview of a disaster area is a foremost task in search and rescue operations. We previously have introduced a novel method for registering a large amount of aerial images when camera parameters are almost unknown and no reference images are available. This paper provides two new methods, which improve our method for image registration based on virtual forces. The goal is to improve the performance of creating a contemporary overview map of a disaster area assembling several images taken by unmanned aerial vehicles (UAVs) equipped with cameras. In this paper, two new methods are introduced: a method for rotation estimation and a method for scale estimation. Both methods use fast heuristic approximation approaches and statistical methods to provide high robustness. We discuss the methods in detail and compare them to our previous approach regarding robustness and calculation speed. We can show that the new methods significantly increase the performance of the image registration process.**

***Keywords*-Virtual forces, image registration, UAV, map building**

## I. Introduction

For large-scale disasters—natural or human made—getting a contemporary overview of the disaster area is an important task in order to find victims and to plan the rescue actions. Often an area is affected, which is too large to efficiently get an overview at ground level. Classically, a manned aircraft or helicopter is used to gather aerial pictures of the whole scene. Nevertheless only in few cases the pictures are used to build a map. In the cases where a helicopter equipped with video cameras has been assigned an observation task, the video is seen only by the observer located in the helicopter. Typically, the video is not analyzed automatically and sometimes not even recorded. Slow changes of the environment like, for instance, the imminence of a flood may be recognized too late. Also typically, there is only one helicopter over the area due to the costs of such a mission. Hence, there is no possibility to get up-to-date images of more than one place at the same time. Furthermore, manned helicopters, which are not directly involved in rescue missions are often not allowed to fly over persons at low altitudes.

In recent years, Unmanned Aerial Vehicles (UAVs) have increasingly become a more viable choice for such situations. The research project SOGRO (German: ”Sofortrettung bei Großunfall mit Massenanfall von Verletzten”, English: ”Immediate rescue in a large-scale accident with mass casualties”) also makes use of UAVs for exploring a terrain where a large-scale accident or a natural disaster has caused many casualties.

In this paper, we present the improvement of an image registration approach developed in the course of the SOGRO project. In [1], we tackle the challenge to register sequentially arriving images to a consistent map of a disaster area. To achieve this, aerial images are delivered by a swarm of UAVs, which spread out over the area [2], [3]. This enables us to deliver up-to-date images of different parts of the disaster area at the same time.

Each UAV produces many images, which are partially overlapping. The images contain distinctive feature points, which can be extracted automatically. To improve the old-fashioned observer approach, we store the images in a database, analyze them, e.g., by extracting distinctive features, and using them to create a contemporary overview map of the disaster area. The extracted features are also stored in the database. They are used to compare different images to each other such that corresponding parts of images can be determined.

Corresponding feature points (key points) are connected with virtual forces. Our image registration approach introduced in [1] translates, rotates and scales the images such that the forces' lengths becomes minimal. We will describe that approach shortly in Section II.

Two new methods are presented in this paper, which increase the performance of our previous approach, supporting the ability using low-cost UAVs to automatically create a contemporary overview map of an area. We introduce a new fast heuristic rotation estimation method and a new fast heuristic scale estimation method. Both methods are compared to the former ones in terms of calculation speed and resulting scale factors.

The paper is organized as follows. In the next section, we will point out the problems of classical image registration and introduce the main idea of using virtual forces for image registration. In Section III, we will give an overview of key point extraction methods and compare our work with regard to other approaches. Section IV describes our new approach for rotation estimation in detail and Section V explains the new approach for scale estimation. In Section VI, we show some results of the evaluation regarding the use of the new methods compared to the old ones. In Section VII, we conclude the paper and give an outlook on future research.

## II. Virtual Forces for Image Registration

In [1], we have described the first steps towards using virtual forces for image registration. For better understanding, we motivate the use of virtual forces here again.

In order to build a contemporary map of a disaster area only using civil UAVs equipped with relatively inexpensive camera equipment, we encountered a challenge. Using classical approaches did not lead to satisfying results. We used SIFT [4] as key point extractor, later also SURF [5]. After a descriptor matching, RANSAC [6] was applied to filter the matches. This approach is well-known and broadly used, e.g., for satellite images. Assuming only rare information about flight attitude (describing, e.g., the position, the nick-angle, the roll-angle, and the yaw-angle) and the inavailability of reference imagery, only few images could be stitched due to accumulating perspective errors. This behavior already has been described by Brown and Lowe when successively concatenating homographies [7].

The challenge in our scenario is to be able to create a contemporary map of a disaster area without having flight attitude information or any reference imagery. In field tests, we also evaluated the reliability of off-the-shelf GPS receivers and decided not to rely on them. The GPS signal delivered by our receiver had a relatively high tolerance and even disappeared intermittently. So, we decided not to rely on any measured reference for the first steps. Our approach does not take any measured reference into account at this time. In future research, different measurements shall be included to increase the stability even more or to increase the achievable resolution.

Classical approaches typically use a static mapping function, local or global. In the global case, all images are needed in advance to calculate a consistent mapping function. In the local case, errors would propagate if the first image's parameters are erroneous. Using virtual forces for image registration addresses with both problems, mapping images in a flexible way and balancing the errors.

Images are regarded to be masses connected by forces that could be imagined to work like rubber bands or springs, which tend to have a length of zero. A virtual force has a start point, an end point, a length and a direction. The force's strength is equivalent to its length. Like in the classical image registration process, key points (distinctive image features) are extracted from the images. These key points are compared and matched against each other. Then each matching pair is fitted with a new virtual force, pulling the newest image to the previous ones, which it partially overlaps (otherwise no matchings would have been found). Here connections between the last image and its direct predecessor are established as well as connections to all other images, which it overlaps by a certain amount. Afterwards, an iterative process moves, rotates and scales the masses until an equilibrium is established, namely, the sum of all remaining forces is minimal. Assuming geometrically consistent matches, corresponding key points then will be next to each other, ideally with zero distance.

## III. Related Work

To find matching interest points (features) is essential for image registration. Several interest point detectors were introduced in the last decades. Such detectors are usually chosen by means of requirements of an application. There are simple feature detectors for edges and/or corners, e.g., like the Harris corner detector [8], which are easy to compute. As these detectors only indicate the presence of a feature, they are not directly usable for image registration. When using a group of features to match against another group of features, taking the geometrical structure of the groups into account, these kind of features are nevertheless usable for image registration. Especially when regarding video sequences, these features or, e.g., the features Shi and Tomasi introduced in [9] are trackable in subsequent video frames assuming some kinematic restrictions.

More general, for image registration purposes features are needed, which can be compared in terms of similarity. Two well-known feature detectors of this kind are the Scale-invariant feature transform (SIFT) [4] and Speeded Up Robust Features (SURF) [10], [5].

As described by Zitová and Flusser in [11], most image registration techniques use a kind of keypoints, which are compared and matched against each other. The matched points are then considered to represent the same location in the scene, which was captured. According to Zitová and Flusser, image registration is the process of overlaying two or more images of the same scene taken at different times, from different viewpoints, and/or by different sensors. A comprehensive survey of the image registration process and an overview of solution methods used by current approaches for different processing steps is given in [11]. They also provide a classification of different approaches and according to them the application of this work belongs to their classes *multiview analysis* and *multitemporal analysis*. The images used in our scenario can be delivered by several UAVs at different positions over the area, and also the same part of the area can be recorded repetitively as the intention is to provide a contemporary overview map.

## IV. Fast Heuristic Rotation Estimation

In this section, we introduce the novel rotation estimation method and compare it to the former approach.

We assume two masses that are connected by some virtual forces. The former method presented in [1] was mainly inspired by physical application of forces to a mass. Besides an acceleration, a torque-like quantity was calculated and used for rotation. In the very first implementation, the center of gravity (CoG) of the mass itself was used as center of rotation. This had been enhanced so that the CoG of the forces' attraction points was used. Hence, the former method needed many iterations to stabilize.

The new method also uses the CoG of the key points in two images as center of rotation. Two center points are calculated: one for the start points in the first image and one for the end points in the second image. A CoG point is calculated for both masses, using the start points $CoG_{start}$ and the end points $CoG_{end}$ of the forces respectively.

$$CoG_{start} = \frac{\sum_{F_u} \text{start}(f_i)}{\|F_u\|} \qquad CoG_{end} = \frac{\sum_{F_u} \text{end}(f_i)}{\|F_u\|}$$

ISBN: 978-1-61208-187-8

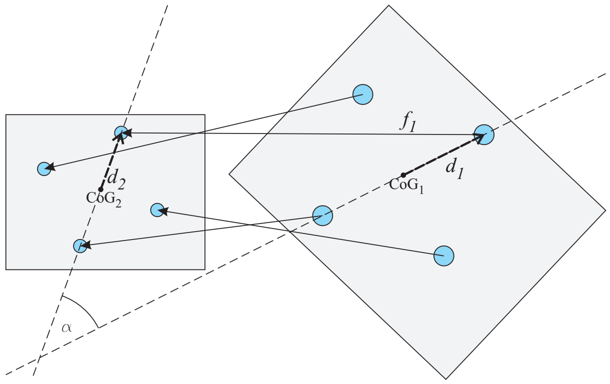

Fig. 1. Simple example for rotation calculation. The rotation angle $\alpha$ is derived by intersecting the lines going through the CoG points and a matching pair of key points. In this example, only one matching pair is used for demonstration.

Forces can be marked as ineffective, so $F_u \subset F$ denotes the forces, which are actually used from the set of totally available forces $F$, $f_i \in F$ denotes one force with a start point $\text{start}(f_i)$ and an end point $\text{end}(f_i)$. Both, start point and end point are given in local coordinates with respect to the according mass object.

For each force two lines are built: a first line through the CoG point of the start points and the start point of the force, and a second line through the CoG point of the end points and the end point of the force. These two lines are intersected and the angle between them is calculated. This angle represents the rotation angle, which is necessary to rotate the end-point-mass such that it is equally oriented like the start-point-mass. Fig. 1 shows a simple example for the rotation calculation described above. For demonstration only one force ($f_1$); is considered.

$$
\begin{aligned}
\overrightarrow{d_1} &= \overrightarrow{\text{start}(f_1)} - \overrightarrow{CoG_1} \\
\overrightarrow{d_2} &= \overrightarrow{\text{end}(f_1)} - \overrightarrow{CoG_2} \\
\alpha &= \arccos \frac{\overrightarrow{d_1} \times \overrightarrow{d_2}}{\|\overrightarrow{d_1}\| \|\overrightarrow{d_2}\|}
\end{aligned}
$$

With only one force considered, only two distance vectors are considered ($\overrightarrow{d_1}$, $\overrightarrow{d_2}$). Here, $\overrightarrow{d_2}$ denotes the distance vector from the CoG point of the start points to the actual start point of $f_1$, and $\overrightarrow{d_1}$ denotes the distance vector from the CoG point of the end points to the actual end point of $f_1$. The angle $\alpha$ can then be derived by calculating the arc cosine of the cross product of the two vectors divided by their length.

In general, the mean angle $\bar{\alpha}$ between the matching pairs' directional vectors is used:

$$
\begin{aligned}
\overrightarrow{d_i^{start}} &= \overrightarrow{\text{start}(f_i)} - \overrightarrow{CoG_{start}} \\
\overrightarrow{d_i^{end}} &= \overrightarrow{\text{end}(f_i)} - \overrightarrow{CoG_{end}} \\
\alpha_i &= \arccos \frac{\overrightarrow{d_i^{start}} \times \overrightarrow{d_i^{end}}}{\|\overrightarrow{d_i^{start}}\| \|\overrightarrow{d_i^{end}}\|} \\
\bar{\alpha} &= \frac{\sum_{F_u} \alpha_i}{\|F_u\|}
\end{aligned}
$$

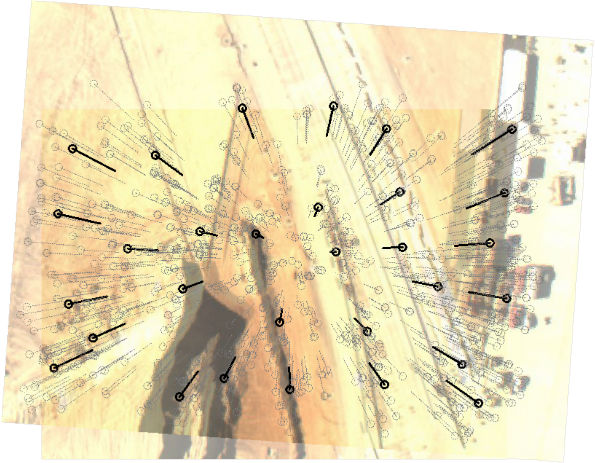

Fig. 2. Old scale method. Two previously aligned images with different scales are shown. The virtual forces, which connect the images, form a star shape. These forces act like gravitational forces and constrict the bigger image.

## V. Fast Heuristic Scale Estimation

In this section, we introduce a novel scale estimation method. The new method needs less computation time for scale changes and the algorithm itself is much more numerically stable than the former one. In non-controlled environments the UAV faces ascending or descending air currents and has to level them out. During that leveling, scale changes appear, as well as, when the territory rises or falls; whereas the altitude of the UAV remains stable. Hence, scale changes will occur and have to be tackled. Following is a brief description of the former method.

The former method needed previously adjusted images. Then the forces formed a star shape and the forces were applied like gravitational forces. Fig. 2 shows two images connected by forces, which were previously adjusted. The forces also have been filtered and only the solid black ones are effective. Nevertheless, that method required scaling iterations interleaved with adjustments to stabilize: The forces' vectors are treated as lines and are randomly selected for intersection. The mean of the intersection points is used as the center point for scaling. Due to the randomness of the intersection, the center point moves and after a few steps a readjustment of the mass is necessary. In particular, the calculation of the intersection points is numerically unstable due to the shrinking vector lengths when the matching points come close to each other. Nevertheless, that approach is robust due to the use of statistical methods to overcome this numerical instability. However, this procedure needed many calculation steps in iterations when it came to scale changes, e.g., due to an altitude change of the UAV. To cope with this behavior, a new method has been created.

The new method also uses a statistical approach to derive the center of the scale operation. In contrast to the former method, the CoG of the key points is used instead of the intersection

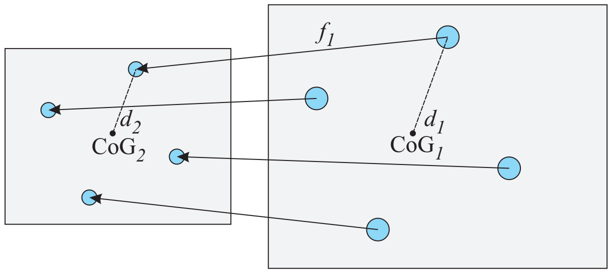

Fig. 3. Simple example for scale calculation. The scale factor for one iteration to scale the right mass to the left one is calculated as $1.001^{\|d_2\|-\|d_1\|}$. In this example, only one distance is used for demonstration.

points. This operation is independent of the actual position of the two masses. Like in the previous section, a CoG point is calculated for both masses. $CoG_{start}$ is calculated using the start points and $CoG_{end}$ is calculated using the end points of the forces.

Fig. 3 shows a simple example for scale calculation. Two masses are shown, both with key points and connected by forces. On both masses the CoG points are marked. For demonstration, only one distance pair is depicted. Typically all distance pairs are taken into account and the arithmetic mean is used as scale factor. Listing 1 shows the basic algorithm to calculate the scale of two masses. Each force connected to both images has a start point and an end point. The start points are located on one image, the end points on the other. For both point sets a center of gravity is calculated. Then the mean distance of each start point to the start point center is set in relation to the mean distance of each end point to the end point center. The resulting ratio is used as scale factor.

$$\begin{aligned}
\textit{start diff}_i &= \|\text{start}(f_i) - CoG_{start}\| \\
\textit{end diff}_i &= \|\text{end}(f_i) - CoG_{end}\| \\
\textit{mean diff} &= \frac{\sum_{F_u} \textit{end diff}_i - \textit{start diff}_i}{\|F_u\|} \\
\textit{mean scale} &= 1.001^{\textit{mean diff}}
\end{aligned}$$

A more sophisticated variant has been implemented to support multiple masses as potential force sources. There, at first, the forces are grouped according to their source mass. For each group of forces the basic algorithm is used to calculate a scale factor. Then a mean scale factor is calculated from the individual scale factors of each group.

As mentioned before, the scale center is of great importance, too. If two masses were aligned before, the scale operation should not move the mass. In the former approach, the scale center was a bit volatile, so the pure scale operations had to be interleaved by adjusting operations. The new approach improves this behavior. The scale center is calculated as the center of gravity of all start points and end points respectively. Assuming aligned masses and geometrically consistent matches (i.e., the matching points on the images represent the same area in reality), the two centers for start points and end points would lie above each other. Scaling around this point assures that the scale operation does not move the center of gravity in terms of global coordinates.

```
scaleWithCoG()
  $CoG_{start}$ = calculateCenter(startPoints($F_u$))
  $CoG_{end}$ = calculateCenter(endPoints($F_u$))
  foreach $f_i \in F_u$ do:
    start_diff = $\|\text{start}(f_i) - CoG_{start}\|$
    end_diff = $\|\text{end}(f_i) - CoG_{end}\|$
    mean_diff = mean_diff + (end_diff - start_diff)
  mean_diff = mean_diff / $\|F_u\|$
  mean_scale = pow(1.001, mean_diff)
```

Listing 1. Basic algorithm to scale the last mass according to the virtual forces. The mean scale is an exponential function of the mean difference of the distances from the start points to the start point center and the distances from the end points to the end point center.

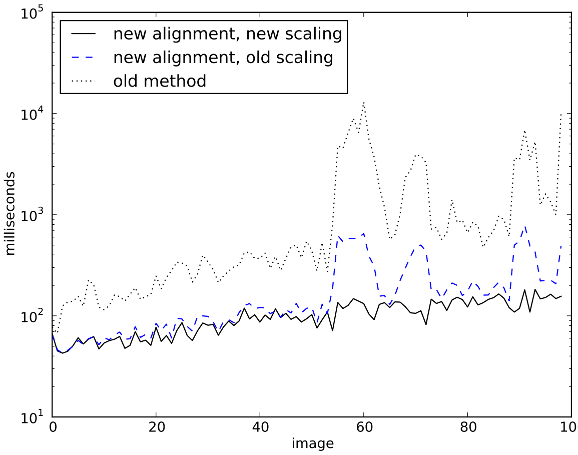

Fig. 4. Comparison of the computation times between old and new method. The total computation times including alignment and scaling are shown. The values are the mean of ten independent runs on identical image sets. Note the logarithmic scale for computation time.

Summarized, the new method has a faster convergence rate, is independent from position and rotation of the mass to be scaled and is numerically stable.

## VI. Results

Using the new method for rotation estimation, we have achieved a significant speed-up of the registering performance. In Fig. 4, the computation times of the old method [1] and the new method are compared to each other. The total computation times including alignment and scaling are shown. The values are given in milliseconds and are calculated as the geometric mean of ten independent runs on identical image sets of 100 images. As it can be seen, the new method outperforms the old one and provides a much more stable runtime behavior. Also the impact of the new scaling method can be seen here. The middle curve depicts the runtime of the new rotation estimation method combined with the old scaling method.

For this evaluation, images taken the bottom-view camera of a “Parrot AR.Drone” [12] are used. The camera lacks a stabilization so it is a good test to the robustness of our

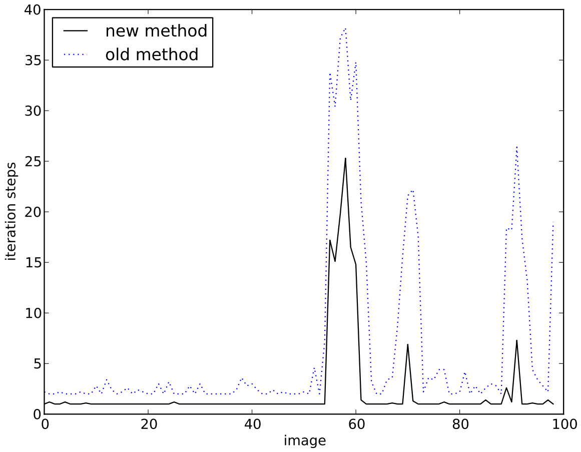

Fig. 5. Comparison of the iteration steps needed to stabilize between old and new method. The values are the mean of ten independent runs on identical image sets. A scale change starts at image 55. Here, the iteration steps of the old method increase significantly. The new method better adapts to scale changes due to its faster convergence.

approach against perspective distortions. Nevertheless, some problems are to be expected at increased camera tilt levels, when the perspective distortion becomes too large. Here a false scale change is to be expected.

Fig. 5 shows the performance of the new scale estimation method compared to the former one in terms of iteration steps. In the first 54 images, the performance of both approaches is nearly equal. A scale change starts at image 55 and immediately the number of iteration steps of the old method increases significantly over the number of iteration steps of the new method. Most of the time, the new method only needs one iteration step and while the scale changes, it significantly outperforms the old method.

In Fig. 6, a related comparison of the computation times and the scale factor is shown. The underlying set of images was taken with the aforementioned ”Parrot AR.Drone”'s bottom-view camera. The image set was taken at approximately the same height above ground for all images. Here the effect of large perspective distortion occurs, which causes a false reaction of the heuristic scale estimation. Nevertheless, it can be seen that the new method of scale estimation is much less sensitive to such distortions. The resulting scale factor is as expected: flat lines in phases of plain flight and a limited scale change during phases with increased camera angles. These occur as the “Parrot AR.Drone” is a Quadrotor, and it has to change its flight attitude in order to fly in a given direction.

At last we evaluated different implementations of the new scaling approach. We changed the scale estimation function to a function, which should be able to calculate the scale difference between two images in one single step. For this purpose we used the mean quotient of the distances of the start points to their center and the distances of the end points to their center. In the aforementioned example showed in Fig. 3,

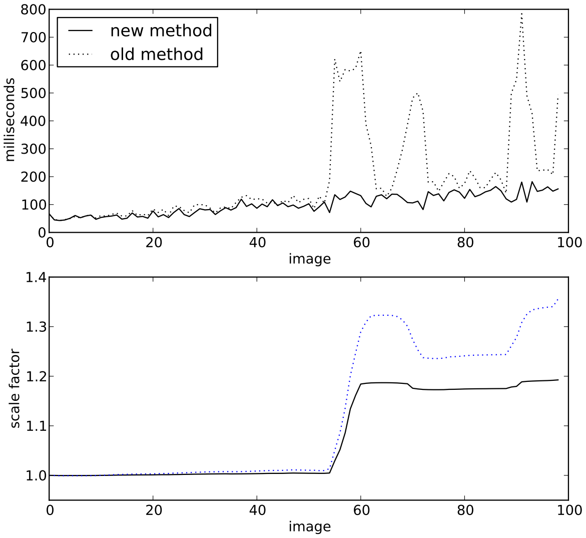

Fig. 6. Comparison of the computation times between old and new method related to the scale factor. The values are the mean of ten independent runs on identical image sets. A scale change starts at image 55. The scale changes are false positives due to perspective distortion. The new method's reaction is much less than that of the old method.

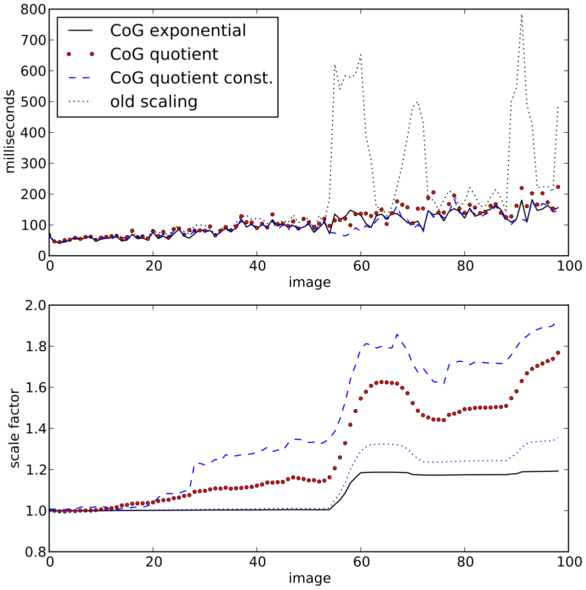

Fig. 7. Comparison of the computation times and the resulting scale factor of the old method with different versions of the new scale estimation method.

the scale factor would be calculated as $d_1/d_2$ to scale the right image to the left one. Fig. 7 shows the calculation times and the resulting scale factors of four different scale estimation implementations. All implementations, which use a variant of the new scale estimation, are using the CoG points of the start points and the end points respectively. The chart lines

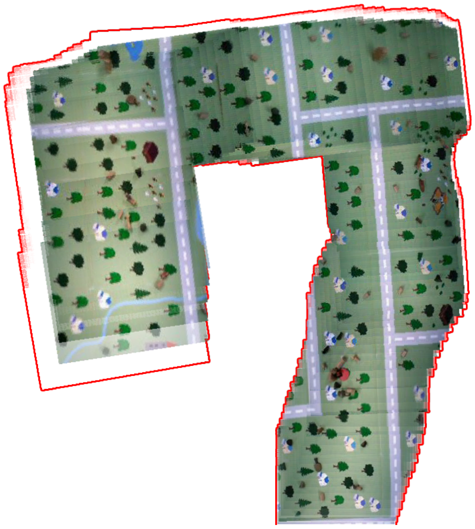

Fig. 8. Comparison between new and old scaling method. Shown are the registered images using the new scaling method. The underlying contour marks the extent of the registered images, if the old scaling method would have been used.

are named correspondingly. The scaling method using the direct quotient method (CoG quotient const.) needs constant computation time but is very sensitive to any distortions and reacts with false scale changes. Executing that method iteratively (CoG quotient) increases its scaling performance but yet the quotient method is worse than the old scaling method. Using the exponential function for scale estimation in combination with the CoG points (CoG exponential) reaches our goal and outperforms the old scale method in terms of calculation time as well as in the resulting scale factor.

Fig. 8 shows the registered images using the new methods. Underneath the images a contour is drawn. It marks the extent of the images, if the old scaling method would have been used. The contour, starting very close to the images, soon deviates in scale. That deviation starts to grow, when the "Parrot AR.Drone" moves sidewards due to the assymetric horizontal and vertical flare angles.

## VII. Conclusion & Outlook

We presented new methods for rotation estimation as well as for scale estimation. The fast heuristic rotation estimation significantly increases the overall performance of the map building process and provides a smoother runtime behavior than its predecessor.

Different implementations of the new approach for the fast heuristic scale estimation were compared to the old scale estimation method and against each other in terms of numerical stability, calculation time and resulting scale factor calculation. The best one—based on an exponential function of the mean difference of two vector lengths—provides a scale estimation method independent from position and rotation of the object to be scaled. It outperforms the old method in all tests: the calculation is numerically stable, the calculation is faster and is less dependent on scale changes, and the calculated scale factor is less sensitive to perspective distortions.

The next step will be the integration of perspective projection into the mapping as well as the usage of sporadic reference information. On the one hand, this will allow the building of geo-referenced maps and, on the other hand, we can make use of the sporadic reference information, e.g., to increase the resolution of the map. Another future work will be the evaluation of different kinds of features in order to increase the mapping speed even more.

## Acknowledgments

This contribution was developed in the course of the SOGRO project funded by the BMBF (German ministry of education and research) under grant number 13N10164.

## References

[1] C. Stern, C. Rasche, L. Kleinjohann, and B. Kleinjohann, "Towards using virtual forces for image registration," in *The 5th International Conference on Automation, Robotics and Applications (ICARA 2011)*, Wellington, New Zealand, Dec. 2011.

[2] C. Rasche, C. Stern, W. Richert, L. Kleinjohann, and B. Kleinjohann, "Combining autonomous exploration, goal-oriented coordination and task allocation in multi-uav scenarios," in *Autonomic and Autonomous Systems (ICAS), 2010 Sixth International Conference on*, Mar. 2010, pp. 52 –57, accessed 25-January-2012. [Online]. Available: http://dx.doi.org/10.1109/ICAS.2010.16

[3] C. Rasche, C. Stern, L. Kleinjohann, and B. Kleinjohann, "Coordinated exploration and goal-oriented path planning using multiple uavs," in *International Journal on Advances in Software*, vol. 3, no. 3&4. IARA, 2010, pp. 351–370.

[4] D. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1999, pp. 1150 –1157 vol.2, accessed 25-January-2012. [Online]. Available: http://dx.doi.org/10.1109/ICCV.1999.790410

[5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008, similarity Matching in Computer Vision and Multimedia. [Online]. Available: http://www.sciencedirect.com/science/article/B6WCX-4RC2S4T-2/2/c2c03b6165996e30312e5b7c7b681155

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, June 1981, DOI: 10.1145/358669.358692.

[7] M. Brown and D. Lowe, "Recognising panoramas," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, Oct. 2003, pp. 1218 –1225 vol.2, accessed 25-January-2012. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2003.1238630

[8] C. Harris and M. Stephens, "A combined corner and edge detector," in *Fourth Alvey Vision Conference*, 1988, pp. pp. 147–151, accessed 25-January-2012. [Online]. Available: http://www.assembla.com/spaces/robotics/documents/abzMnAOEer3zB7ab7jnrAJ/download/harris88.pdf

[9] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, jun 1994, pp. 593 –600, accessed 25-January-2012. [Online]. Available: http://dx.doi.org/10.1109/CVPR.1994.323794

[10] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *ECCV 2006*, ser. Lecture Notes in Computer Science, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin / Heidelberg, 2006, vol. 3951, pp. 404–417, accessed 25-January-2012. [Online]. Available: http://dx.doi.org/10.1007/11744023_32

[11] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977 – 1000, 2003, accessed 25-January-2012. [Online]. Available: http://dx.doi.org/10.1016/S0262-8856(03)00137-9

[12] "Parrot AR.Drone," 2012, accessed 25-January-2012. [Online]. Available: http://ardrone.parrot.com