Evaluation of Pose Hypotheses by Image Feature Extraction for Vehicle Localization

Kristin Schönherr[1], Björn Giesler[1] and Alois Knoll[2]

[1] Audi Electronics Venture GmbH, 85080 Gaimersheim, Germany,
[2] Institute of Computer Science VI, University of Technology Munich,
85748 Garching b. München, Germany

Abstract. For the realization of driving assistance and safety systems vehicles are being increasingly equipped with sensors. As these sensors contribute a lot to the cost of the whole package and at the same time consume some space, car manufacturers try to integrate applications that make use of already integrated sensors. This leads to the fact that each sensor has to fulfil several functions at once and to deliver information to different applications. When estimating very precise positioning information of a vehicle existing sensors have to be combined in an appropriate way to avoid the integration of additional sensors into the vehicle.
The GPS receiver, which is coupled with the navigation assistant of the vehicle, delivers a rough positioning information, which has to be improved using already available information from other built in sensors. The approach discussed in this paper uses a model-based method to compare building models obtained from maps with video image information. We will examine, if the explorative coupling of sensors can deliver an appropriate evaluation criteria for positioning hypotheses.

Key words: Localization, probability density function, pose hypotheses, quality factor

1 Introduction

Precise vehicle localization is turning into one of the most important challenges for driving assistance and security systems. Sensors are already available on the market, that provide exact vehicle localization, accurate enough to fulfil the demands. But the cost of these systems is still prohibitive for production vehicles. Additionally, integrating extra sensors into the vehicle is a major challenge to automotive design. The mentioned available high precise sensors do not fulfil these requirements at all. Therefore the preferred solution is to use information from already-integrated automotive sensors for multiple applications. The camera as sensor for image information delivery is already integrated in series-production vehicles for lane detection and parking assistance. It seems that the camera may be a suitable application-independent candidate for vehicle localization, see [7], [6]. In combination with additional sensor information like the rough position data of a GPS[3] receiver and precise maps (GIS[4]), we are doing research on how to obtain the exact vehicle position.

Creating appropriate evaluation criteria for the position hypotheses is a challenge, especially when utilizing combined sensor information. A probability density function to evaluate a position hypothesis upon the given sensor data, should reflect the precise vehicle position in form of a distinctive maximum. Model-based object pose estimation algorithms known from computer vision, such as RAPiD[5] [3], minimize distance information between projected model edges and detected object edges within a video image for estimating object positions. These algorithms work well in the laboratory; it is worthwhile though to evaluate, if they can work in automotive practice as well.

We build a 3D-building lattice model (compare [2], [4] and [5]) from precise outline information originating from highly accurate map material. This 3D lattice model is overlayed over the video image using a hypothesis for the car pose. It is assumed that every building has the same height and that building outlines are often occluded by spurious objects (objects that are not part of the model, and cannot contribute to the positioning process, such as parking or moving cars). Therefore the focus is set on vertical building edges, which are the most probable to be at least partially unoccluded. The model generation as well as the necessary image processing, has to be adapted to the generation of vertical model and object edges.

The sequence schema shown in Fig.1 illustrates determination and evaluation the quality factor of different vehicle position hypotheses step by step.

Different position hypotheses are generated based on precise ground truth position, determined by a highly accurate reference sensor system. The distance values between vertical model and object edges are used to generate a quality factor, though evaluating position hypotheses. The transferred quality function should show a maximum near the ground truth position.

2 Extraction of Vertical Lattice Lines

For evaluating a pose hypothesis, we use accurate map material to extract hypothetical building outlines. Thus it would be visible, where the car at the postulated hypothetical position is located.

At first the preparation of the precise map material, in which building outlines are stored as polygon lines, is explained . The point of view which is represented by the position information, decides whether a model line is visible or not (compare [11]).

2.1 Backface Culling

In our map, building outline information is stored as polygons, whose edges are arranged in a clockwise orientation. For every line the normal vector is derived

[3] Global Positioning System

[4] Geographic Information System

[5] Real-time Attitude and Position Determination

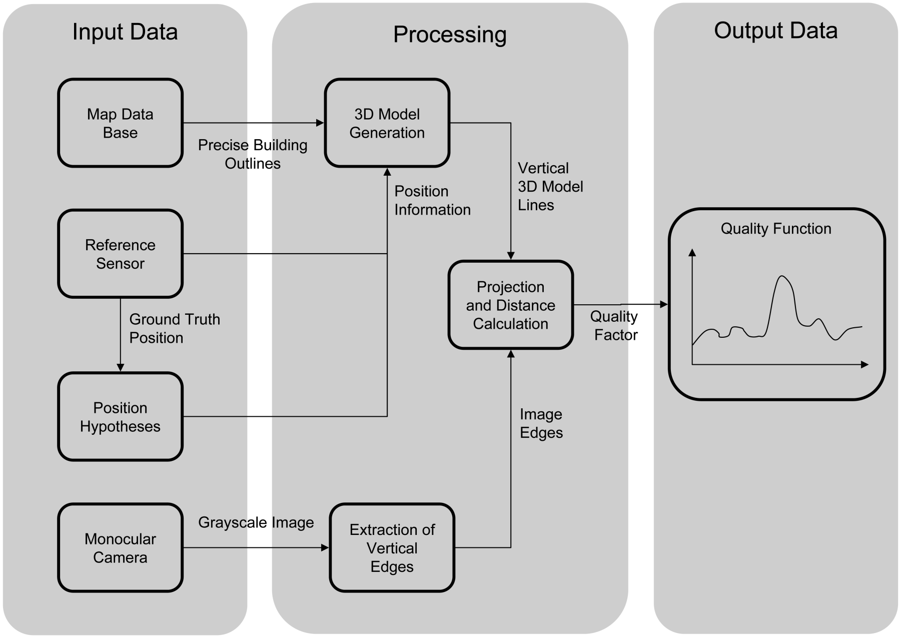


Fig. 1. Overview of processing and data information for evaluation of position hypotheses

and compared with the viewing direction vector, to distinguish between visible and invisible edges. If the normal vector is directed along the viewing direction vector, the polygon line is backfaced and thus not visible. This line is not considered for futher calculations. If the normal vector is directed contrary to the viewing direction, this polygon line is considered to be visible, which is then taken for succeeding calculation. In mathematical terms, the calculation can be derived using the scalar product of the normal vector and the viewing direction vector. The following equation (1) shows the case discrimination.

$$\begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \cdot \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \qquad \begin{cases} > 0 : \text{unvisible} \\ \leq 0 : \text{visible} \end{cases} \tag{1}$$

$$n_i = \text{normal vector of polygon line;}$$
$$v_i = \text{view direction}$$

For this step there is no z-coordinate (the up vector in our coordinate system), so the backface culling [1] method is reduced to a 2-dimensional problem.

2.2 Ground Plane Based Angle Separation

After the rough differentiation between visible and non visible borders, we take into account inter-object occlusions. From a certain viewing direction, some buildings can occlude others. The aim is to identify the outline edges, which are actually visible. Therefore, we sort the remaining edges of the backface culling operation by their distance to the viewer. We select the line closest to the viewer and determine the angular range it occupies. The angle range of the next lines is then compared to the already known occupied range. In the case of complete occupation the line is skipped. In case of partially occupation the line is segmented. With this procedure all the visible outlines are determined.

2.3 Visible Vertical 3D Model Lines

Starting from the visible outlines of the buildings, we add an assumption of height, see Fig. 2.

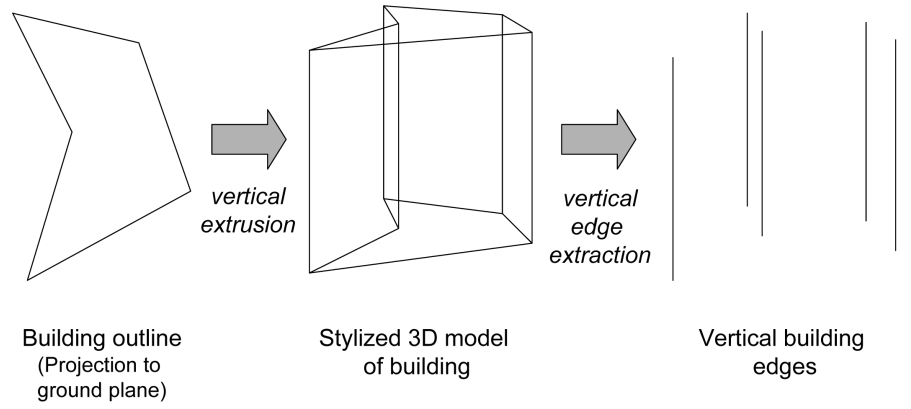


Fig. 2. Generation of vertical building edges by using the complete outline

This results in a 3D building model, that contains all lattice edges of these buildings which should be visible from the viewer's current position.

It has to be noticed, that in comparison with the video image, bottom edges of the generated model are often occluded by unmapped objects. Additionally, the top edge of the building can not be used for calculation, because the actual height is not known and our assumption of it is certainly incorrect in most cases.

Therefore we reduce the 3D building-environment-model to vertical lattice lines, which represent the vertical building edges within in the image. To correlate between the model and the object edge within the image, the next preprocessing step is necessary, which especially extracts the vertical edges from the image.

3 Model and Image Combination

The vertical lattice lines, which are projected into the image, when compared to the real object edges, should present the difference between ground truth pose and the pose hypothesis.

Taking a closer look at the pose of a vehicle, direction and x / y position are particularly interesting. To test our algorithm, we take the ground truth position as a starting point and distort it randomly, what represents the different hypotheses. The variation of the pose information results from the error range of a GPS receiver, allowing us to simulate GPS inaccuracies while still knowing the ground truth position.

Based on tracking methods like RAPiD algorithm [3], the deviation between object and model edges is determined by distance calculation. To do this, we divide the projected model lines into equal line segments. Based on these control points, orthogonal search vectors in the image domain are created, that are used to look for corresponding edges, see Fig. 3. The vertical object edges are generated by image preprocessing with a Sobel mask in x direction.

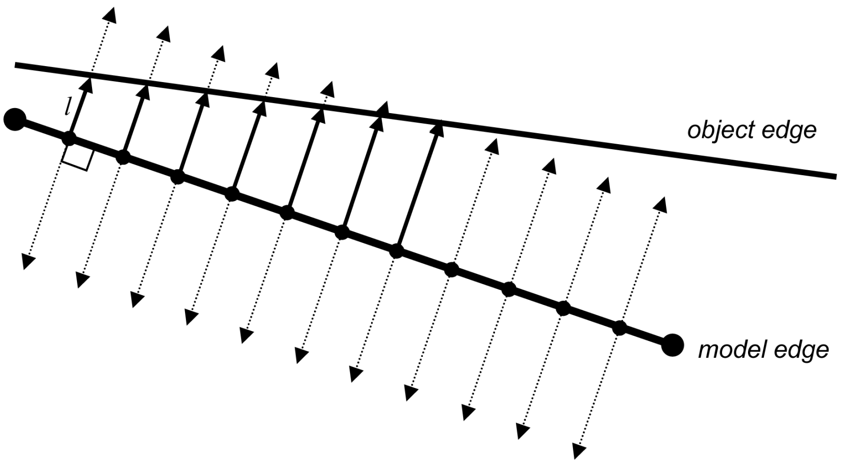


Fig. 3. Determination of distances in orthogonal direction relating to model edge

Fig. 4 shows the camera images with the projected model lines, from which the distance values are calculated. One of the problems with this approach is the large number of misdetections (i.e. image edges caused by unmapped environment features or image noise) and non-detections (i.e. a model edge that does not find a match in the image domain because the image edge is too weak).

We use a RANSAC[6]-type algorithm [10] to filter all extracted object edges to remove outliers, but this does not remove all outliers reliably and it still leaves the problem of non-detections.

Our goal is to obtain a function, that compares the hypothetical edge model with the edges found in the image and delivers a clear maximum, where the position hypothesis and ground truth are close. Therefore a good weight for the interpolated points with and without distance information has to be found.

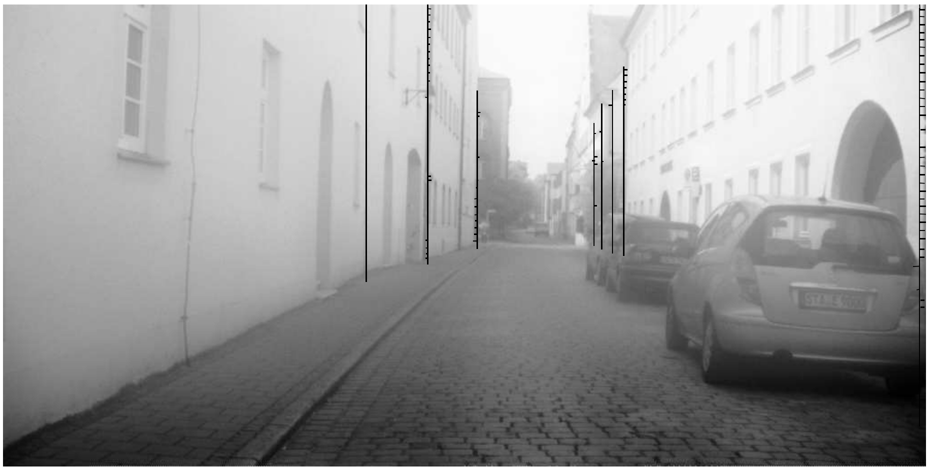


Fig. 4. Camera image with projected model lines and distance values

4 Determination of a Quality Factor

Starting from the exakt position and direction of precise reference sensors, hypotheses in the known GPS deviation range are scattered. Pose hypotheses are only generated outside of building footprints. The Even-Odd-Rule-Algorithm [9] is used to sort out pose hypotheses, which lie inside a footprint. Based on the pose hypothesis, the intersection points in x or y direction with building outline are counted. The pose hypothesis lies outside of a footprint with an even number of intersection points, otherwise the hypothesis is placed inside the footprint, see Fig. 5. With this method concave building outlines can also be handled.

All conditions for generating and evaluating a quality factor are now fulfilled. Different mathematical methods to determine the quality factor are considered according to performance and functional characteristics. The generated quality function using the distance values is being examined for its maxima. It is our goal, that the maximum of the quality function is at close range to the ground

[6] **Ran**dom **Sa**mple **C**onsensus

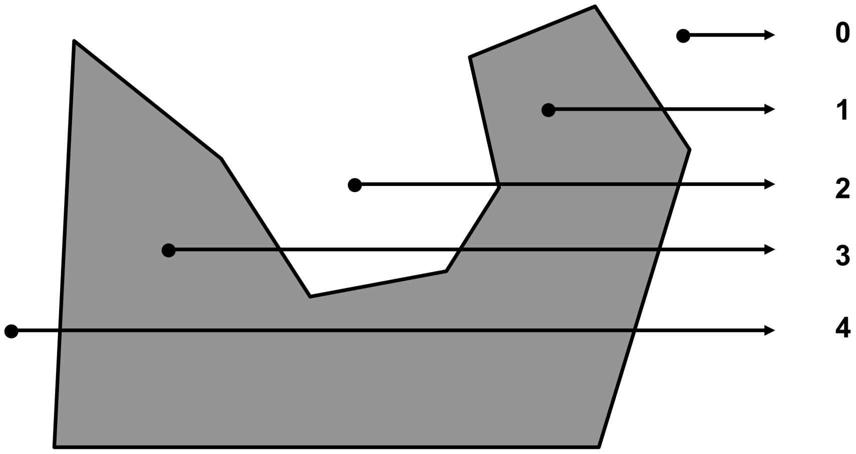


Fig. 5. Even-Odd-Rule checks number of intersection points along x direction

truth position. The analysis was started using the weighting W_1, where the quality is being calculated for every hypotheses $s_{i,t}$:

$$W_1(s_{i,t}) = \frac{n_{\text{gef}}}{\sum_{j=0}^{n_{\text{gef}}} l_j} \tag{2}$$

Here the amount of control points n_{gef}, where a corresponding edge to an image model edge was found, is divided by the sum of the pixel lengths l_i of the normal vectors of all found control points. It is assumed, that the current projection of the model edge for the actual position and direction gets the better the more corresponding control points are found and the shorter the lengths of the normals are. Tests carried out, using the this weighting on an idealized model, at first confirmed this approach, because the weight of model edges and the extracted edges from the video image are lying nearly one upon the other.

However the method is very vulnerable to inhomogeneous areas or occlusions in the video image where the method determines values nearly as high as for the appropriate model. As the weighting $W_1(s_{i,t})$ is rating most of the hypotheses with the same quality, the hypotheses are never converging to one specific point when performing tests with a high amount of iterations.

The weighting $W_1(s_{i,t})$ delivers the same value to an independent amount of control points n_{gef} with constant normal lengths l. But an urban scenario is normally characterized by high amounts of adjacent buildings and many image edges.

Therefore, in addition to the found control points n_{gef}, we also consider the demanded control points n_{ges}, even if they do not yield a match. This approach appears in weighting $W_2(s_{i,t})$. It is assumed, that the result improves by the relation $\frac{n_{\text{gef}}}{n_{\text{ges}}}$ between found control points and demanded control points. Therefore the outcome should be the more precise the more corresponding control points were found within the video image. As a result, the amount of found control points n_{gef} is squared to give more weight to those edges, that result from a high amount of found pixels. Additionally we square the relation $\frac{n_{\text{gef}}}{n_{\text{ges}}}$ to give it more weight.

This results in an extended weighting $W_2(s_{i,t})$:

$$W_2(s_{i,t}) = \frac{\left(\frac{n_{\text{gef}}^2}{n_{\text{ges}}}\right)^2}{\sum_{j=0}^{n_{\text{gef}}} l_j} \tag{3}$$

We evaluate more than these two weightings described here in detail, to generate a convenient propability density function. Especially the ratio between control points and distance values is additionaly varied.

5 Practical Results for Quality Factor

The variation of the mathematical methods is analysed by the different quality functions. Fig. 6 shows the quality functions of the different methods. A smoothed result of quality factor for each position hypothesis is represented in these diagrams, that means the average of all hypotheses in an area of 10cm is calculated for eliminating single outliers. Single hypotheses with a high quality factor which have nearly identical distances to ground truth, but different position values are downgraded. The comparison of the different equations shows, that the used values are basically correct, only the ratio between control points and distance values has to be balanced. The diagram 6 a) shows different peaks, whereas the other mathematical equations deliver the assumed maximum at ground truth position.

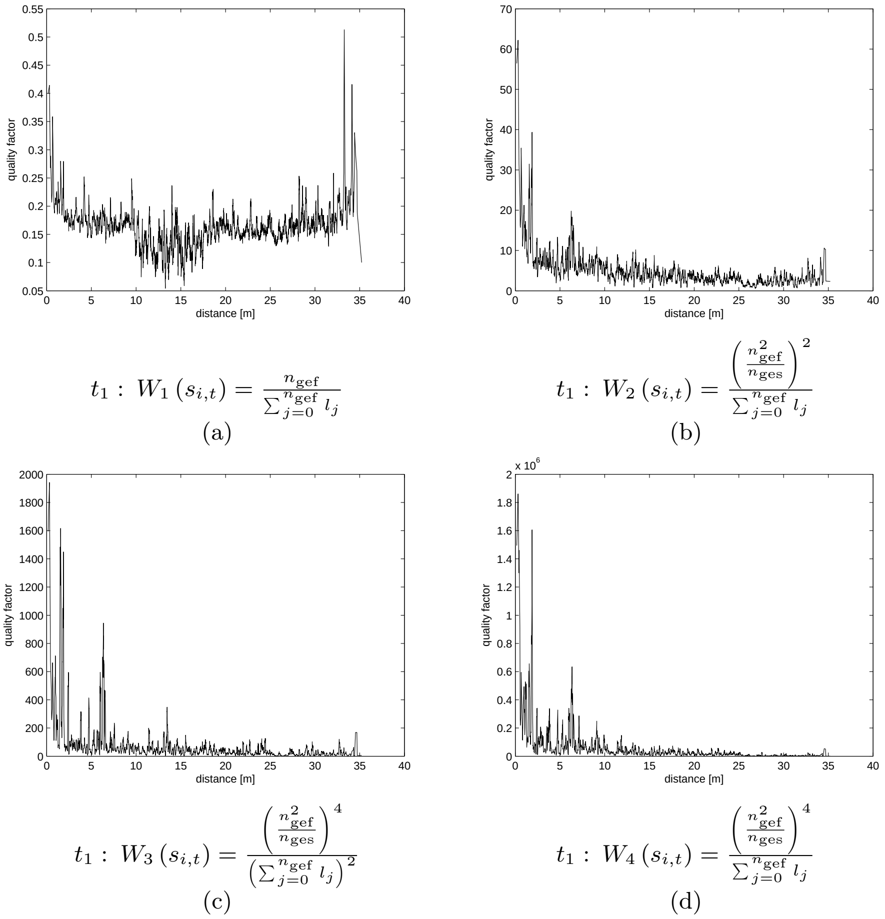
The equation relating to Fig. 7 is extracted for further consideration, because of the clear maximum at ground truth position, compared to the other lower peaks. The diagrams 7 a) and b) show the result at time t_1 and after a time step by t_2. The quality function of the second diagram is more distorted with different high potential peaks. The Fig. 7 c) illustrates the multiplication of the two functions above and a clear maximum at ground truth position is shown. So time analysis eliminates the disruptive peaks, what justifies the utilization of a Particle filter (compare [8]) for pose estimation. Therefore the determined weighting for each position hypothesis is transformed into a probability density function. The zoomed in figure supports our expectancy to reach an accuracy of less than 1m by the combined sensor approach.

6 Conclusion

With the combined use of map material and image data, we are able to generate a probability density function, which shows a clear maximum near ground truth position. It does not always find one single maximum only, but false / additional maxima are shifting around, caused by movement of the car. This is not the case with the main maximum, what makes the probability density function a good candidate for further filtering. We hope to reduce the deviation of the GPS sensor (currently more than 20m) to less than 1m with this approach. So transferring the preferred weighting to a Particle filter for pose estimation turned out be a promising method.

References

1. A. Nischwitz, M. Fischer, P. Haberäcker: *Computergrafik und Bildverarbeitung,* Friedr. Vieweg u. Sohn Verlag, GWV Fachbuchverlag, Wiesbaden, 2007, pp. 67-68
2. D.G. Lowe: *Fitting Parameterized Three-Dimensional Models to Images,* IEEE Trans. on Pattern Analysis and Machine Intelligence, 1991, pp. 441-450
3. C. Harris, C. Stennet: *RAPID - A Video Rate Object Tracker,* Proceedings of the British Machine Vision Conference, 1990, pp. 73-77
4. M. Amstrong, A. Zissermann: *Robust object tracking,* Proceedings of the Asian Conference on Computer Vision, 1995, pp. 58-62
5. V. Lepetit, P. Fua: *Monocular Model-Based 3D Tracking of Rigid Objects: A Survey,* Foundations and Trends in Computer Graphics and Vision, 2005
6. A. J. Davison: *Real-time simultaneous localisation and mapping with a single camera,* Proceedings of the International Conference on Computer Vision, 2003
7. A. J. Davison, D. W. Murray: *Mobile Robot Localisation Using Active Vision,* Proceedings of Fifth European Conference on Computer Vision, 1998, pp. 809-825
8. F. Dellart, D. Fox, W.Burgard, S. Thrun: *Monte Carlo Localisation for Mobile Robots,* IEEE International Conference on Robotics and Automation, 1999
9. A. Mathias, U. Kanther, R. Heidger: *Insideness and collision detection algorithm,* Proc. Tyrrhenian International Workshop on Digital Communications - Enhanced Surveillance of Aircraft and Vehicles, 2008
10. M. A. Fischler, R. C. Bolles: *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,* Readings in computer vision: issues, problems, principles, and paradigms, 1987, pp. 726-740
11. K. Schönherr, B. Giesler, A. Knoll: *Vehicle Localization by Utilization of Map-based Outline Information and Grayscale Image Extraction,* Proceedings of the International Conference on Computer Graphics and Imaging, 2010



$$t_1 : W_1\left(s_{i,t}\right) = \frac{n_{\text{gef}}}{\sum_{j=0}^{n_{\text{gef}}} l_j}$$

(a)

$$t_1 : W_2\left(s_{i,t}\right) = \frac{\left(\frac{n_{\text{gef}}^2}{n_{\text{ges}}}\right)^2}{\sum_{j=0}^{n_{\text{gef}}} l_j}$$

(b)

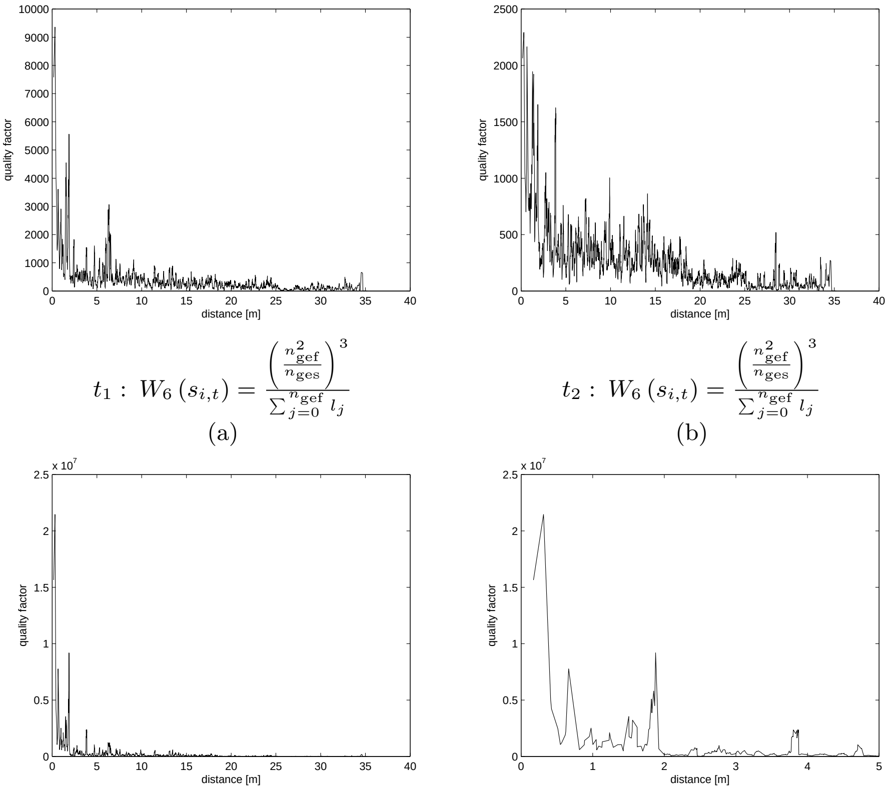
$$t_1 : W_3\left(s_{i,t}\right) = \frac{\left(\frac{n_{\text{gef}}^2}{n_{\text{ges}}}\right)^4}{\left(\sum_{j=0}^{n_{\text{gef}}} l_j\right)^2}$$

(c)

$$t_1 : W_4\left(s_{i,t}\right) = \frac{\left(\frac{n_{\text{gef}}^2}{n_{\text{ges}}}\right)^4}{\sum_{j=0}^{n_{\text{gef}}} l_j}$$

(d)

Fig. 6. The diagrams show the results of the different mathematical methods for calculating the quality factor. The x axis of the coordinate system represents the distance values relating to ground truth position. a) - d) Smoothed quality functions



$$t_1 : W_6\left(s_{i,t}\right) = \frac{\left(\frac{n_{\mathrm{gef}}^2}{n_{\mathrm{ges}}}\right)^3}{\sum_{j=0}^{n_{\mathrm{gef}}} l_j}$$

(a)

$$t_2 : W_6\left(s_{i,t}\right) = \frac{\left(\frac{n_{\mathrm{gef}}^2}{n_{\mathrm{ges}}}\right)^3}{\sum_{j=0}^{n_{\mathrm{gef}}} l_j}$$

(b)

(c)

(d)

Fig. 7. The diagrams show the result of the time analysis. a) Smoothed qualitiy function at time t_1 b) Smoothed qualitiy function at time t_2 c) Multiplyed quality functions of t_1 and t_2 d) Zoomed-in on diagram c