University of Turku Technical Reports, No.2 – June 2014

Model-Based Tracking Initialization in Ship Building Environment

Antti Euranto | Olli Lahdenoja | Rami Suominen | Teijo Lehtonen

University of Turku, Technology Research Center, 20014 Turun yliopisto, Finland,
{ akjeur, olanla, rajusuo, tetale } @utu.fi

trc.utu.fi

ISSN 2341-8028 | ISBN: 978-951-29-5774-3

Abstract

An augmented reality (AR) device must know observer's location and orientation, i.e. observer's pose, to be able to correctly register the virtual content to observer's view. One possible way to determine and continuously follow-up the pose is model-based visual tracking. It supposes that a 3D model of the surroundings is known and that there is a video camera that is fixed to the device. The pose is tracked by comparing the video camera image to the model. Each new pose estimate is usually based on the previous estimate. However, the first estimate must be found out without a prior estimate, i.e. the tracking must be initialized, which in practice means that some model features must be identified from the image and matched to model features. This is known in literature as model-to-image registration problem or simultaneous pose and correspondence problem. This report reviews visual tracking initialization methods that are suitable for visual tracking in ship building environment when the ship CAD model is available. The environment is complex, which makes the initialization non-trivial. The report has been done as part of MARIN project.

Keywords:

Augmented reality, AR, model-based visual tracking, camera pose initialization, camera pose recovery, model-to-image registration problem, simultaneous pose and correspondence problem

Table of contents

1 FOREWORD

MARIN

MARIN (Mobile Augmented Reality Tool for Marine Industry) is a two year (2012 – 2014) research project in Technology Research Center in University of Turku. The project studies the use of augmented reality (AR) in ship building industry.

MARIN device

MARIN is developing a prototype of a mobile AR device. The basic configuration of the device consists of a tablet computer with a touch screen and a video camera. The video camera image together with virtual content is shown on the screen. The screen is also used as an input device for example when the user needs to select features from the video image or from the virtual content.

The device must know its location and orientation to be able to properly register the virtual content to the screen. The location and orientation are determined by comparing the video image to the ship CAD model. The necessary calculations and image processing operations are performed with the tablet computer processor.

An alternative configuration of the device consists of a mobile device as a processing and input unit and see-through AR glasses as a display unit. A video camera is attached to the AR glasses so that the device can determine user’s the location and viewing direction.

The second configuration has some benefits when compared to the first configuration, but the technical limitations of the existing AR glasses have made its implementation difficult. For that reason MARIN has mainly focused to the first configuration.

The purpose of this paper

The initialization of the visual tracking is not a trivial task, especially in a complex environment like a constructed ship. It has been hard to find suitable initialization methods from literature. Therefore, new initialization ideas were gathered in a workshop that was arranged on th13th of June, 2013. This paper first presents the results of a preliminary literature study and then the most promising workshop proposals.

2 TECHNICAL BACKGROUND

Augmented reality (AR) device

An AR device adds virtual content to observer's view. The device may use for example a tablet computer or virtual glasses or see-through AR glasses to display the content for the observer.

Pose

An AR device must know observer's location and viewing direction, i.e. observer's pose, to be able to properly register the virtual content to observer's view. The pose has six degrees of freedom: three position coordinates and three rotation angles.

Pose tracking and initialization

Pose determination usually consists of two phases: initialization and tracking.

Tracking means that the pose estimate is kept up-to-date all the time when the observer moves or changes the direction of the view.

Pose tracking is preceded by pose initialization. The tracking is usually based on an assumption that a new pose estimate can somehow be founded on the previous estimate. However, the first pose estimate must be determined without a prior estimate, i.e. the pose must be initialized.

Pose recovery

If tracking fails, then the pose must be recovered. Recovery resembles initialization, but is often an easier operation, because useful information remains from tracking, for example some older pose estimate.

Model-based visual tracking

Model-based visual tracking is one possible way to determine and track the pose. The technology presumes that there exists an exact 3D model of the surroundings and that the AR device has a fixed video camera. The pose is determined by comparing the video camera view to the model.

The core of the model-based visual tracking is so called feature matching aka 2D-3D matching. It means that certain model features, for example some specific vertices or edges, are identified and located from the camera view and matched to the corresponding model features.

A pose estimate can be calculated if the amount of correspondences n is sufficient. When $n = 3$, then there are 4 possible solutions [1]. When $n = 4$ or $n = 5$, then there are at least 2 solutions in general, but when the points are coplanar and there are no triplets, then the solution is unique. The solution is unique when $n \geq 6$.

The matches should not be lost once they have been created. For that reason the movement of the 2D-features are tracked on video image using optical flow techniques like Lucas-Kanade [2].

An important part of a visual tracking process is the automatic creation of new matches. The additional matches are needed for two reasons. First, the additional matches improve pose accuracy. Second, the new matches are needed to replace matches that have been lost for example because a feature has slid out of camera view or the 2D tracking of a certain feature has failed for some other reason.

Model-to-image registration problem

The model-based visual tracking initialization problem is an example of so called model-to-image aka simultaneous pose and correspondence problem. It is a combination of two problems: the camera pose problem and correspondence problem to find matching object and image features. Either problem is easy to solve, if the other problem has been solved first, but the difficulty comes from the simultaneous nature of the two problems.

Sensor-based pose tracking

Sensor-based tracking is an alternative technology to the visual tracking. The pose can be tracked based on accelerometer and gyroscope measurements, if an initial pose, initial velocity and initial angular velocity are known.

A fundamental shortcoming in sensor-based tracking is that the tracking is relative, i.e. a new pose estimate can only be determined in relation to the previous estimate. Therefore, estimate inaccuracies tend to accumulate and the estimate gradually drifts away from its correct value unless the estimate is calibrated in short intervals using e.g. visual tracking.

3 PROBLEM DEFINITION

An exact definition of the problem to be solved is given below.

3.1 Problem statement 1 – pose initialization

The pose initialization problem to be solved is the following:

Match at least three camera image points to CAD model points

Three 3D-2D matches is the theoretical minimum for pose calculation.

3.2 Problem statement 2 – pose recovery

The pose recovery problem to be solved is the following.

Create so many new 2D-3D matches that pose calculation successes again. Utilize old pose estimate and other available information as much as possible.

The recovery may be needed frequently during the tracking. It is therefore desirable that the procedure is unnoticeable for the observer.

3.3 Assumptions

The following assumptions are made in this paper:

1. The rough position of the observer is known during initialization; see [3]. In other words, it is known in which room the observer is, but the exact location in the room and direction of the sight are unknown.
2. The CAD model of the room is known and available.
3. User may be asked to assist the initialization.
4. The user may be asked to stay in place during initialization, assuming that the initialization takes a reasonable time.

3.4 Manual interaction limitations

Experience has shown that there are certain limitations concerning the user interactions with an AR system. The limitations are to be taken into account when designing initialization methods that require manual interaction or some specific behavior from the user.

1. A user can point features on camera image only with limited accuracy.
2. A user may be asked to stay stationary only a short time.

3. A handheld or a helmet camera is never totally stationary.
4. AR glasses provide only limited possibilities for the user to interact with the model or the world.

4 METHOD REVIEW

The following methods and approaches are either presented in literature or well-known in augmented reality community.

4.1 Manual point-to-point matching

Perhaps the most obvious way to initialize model-based visual tracking is to render the model onto the display and then let the user manually match the image features to the model features, i.e. model vertex points to image points in MARIN case.

A big benefit in the point-to-point approach is that it is robust to changes in the environment. The user can ignore structures that do not match the model, as well as extra objects in camera view. Both could easily confuse an automatic initialization method.

The drawback in this approach is that the pose estimate accuracy depends on how accurately the user is able to locate and point the features in image.

4.2 Manual model alignment

Pose can be initialized manually by rendering the model to the display and then asking the user to move and rotate the model until it aligns the camera view.

Respectively, the rendered model can be stationary, but the user is asked to move to such a location and orientation that the camera view aligns the rendered model.

VTT [4] uses manual initialization in their construction visualization system. First, the user location is determined with GPS or by manual input. Then the user points a known landmark with a crosshair that is shown over camera image and locks the alignment.

This approach is subject to the user interaction limitations in the same way as the point-to-point matching.

4.3 Semi-automatic model alignment

It is possible to automatically search for an accurate alignment after a rough alignment has been found manually or by some other means.

One practical approach is such that the system tries in background to find a perfect fit between the model and the video image at the same time the user is doing the manual alignment. When a fit is found, the alignment is locked and tracking starts.

The automatic alignment procedure can be based on an assumption that a certain model feature corresponds to some nearby image feature. In an ideal case each model feature could be matched with the nearest image feature, but that is not a realistic assumption in practice. Instead, an optimal set of correspondences must be found using some suitable optimization technique.

In ARTESAS system [5], user moves either the real or the virtual camera until a 3D line model roughly overlays the corresponding object in the image. The camera pose estimation is based on the minimization of the distance between a projected 3D line and a 2D line in the image. The minimization is done by creating control points on the projected line and then searching gradient maxima for the control points along orthogonal direction of the projected line. The problem with this approach is that there are usually multiple gradient maxima near the projected line. Therefore, ARTESAS system decides the final match by creating a certain error function and minimizing it.

Another line segment based system [6] uses a similar strategy. It projects the model on the image from the initial pose guess. It then matches the projected lines to image lines by minimizing a certain correspondence error function and calculates a new pose estimate based on the matches. Each line matched is weighted according to its correspondence error. The two steps are repeated until the pose estimate converges.

4.4 Model matching

Model matching [7] is a problem type, where the intention is to match the corresponding parts of two geometric models to each other. Model matching can be divided to different sub classes, for example 2D-2D, 3D-3D and 2D-3D matching. The model matching is interesting from initialization point of view, because the initialization in visual tracking can be seen as a special case of the 2D-3D matching.

Model matching methods typically start by first searching for some easily identifiable parts from the models. Then they hypothesize a match between some parts that have a similar appearance. If the hypothesis leads to a dead end, then the method backtracks and tries another hypothesis.

A so called view-class approach is often useful when solving a 2D-3D matching problem. A view-class represents all such views of a 3D model that are similar by some criteria. For example, a cube has six faces. One, two or three of those faces can be visible at the same time. In this case a view-class could consist of all the views where the same faces are visible. In this way all the infinite number of cube views can be represented by a small number of view-classes.

The benefit of the view-class approach is that it allows the original 2D-3D matching problem to be solved by comparing the actual view to each view-class in turn. Thus, the original 2D-3D matching problem can be solved by solving a set of easier 2D-2D matching problems.

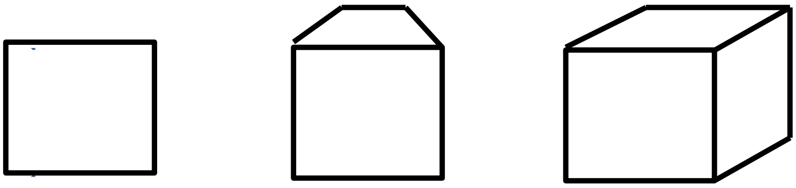

Figure 1. Some of the view classes of a cube.

Two examples of model matching systems, TRIBORS and RIO, are explained below.

TRIBORS [8] system selects line segment triplets from the model, see Figure 1. The triplets are selected so that they can be recognized with high probability. Then nine parameters are calculated for the selected triplets. The parameters consist of segment lengths, distances and angles. The matching between image and model is done by searching triplets from the image and comparing their parameters to the model triplet parameters.

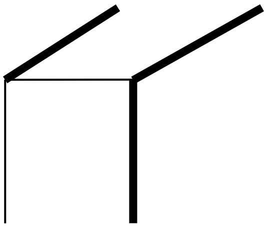

Figure 2. Line segment triplet.

RIO system [9] can recognize multiple 3D objects from intensity images. RIO objects can consist of planar, cylindrical and threaded surfaces. The edge image is used to obtain straight-line and circular-arc segments from which the high-level recognition features are constructed. RIO employs ten such features: ellipses, coaxial arcs, parallel pairs of line segments, triples of line segments and different line segment junctions. In addition, RIO utilizes binary relations over the features.

RIO encodes the relations with a 4-tuple and uses the tuple as an index to a hash-table. The hash-table element lists all the model-views that have the relationship as part of it. Each view in the list then gets a vote. At the end, the view that got the most votes is selected as hypothesis. The hypothesis is verified by computing a transformation from the selected 3D model to the image.

4.5 Natural feature matching

Natural image feature detectors like SIFT [10], SURF [11], Harris corner detector [12], Shi-Tomasi [13] and FAST [14] have some desirable properties that make them potentially useful in pose initialization:

- Many detectors are able to produce descriptors for the features, allowing easy and efficient matching of the features e.g. between images
- The detected features are typically scale invariant, robust to viewing angle and robust to light conditions
- The detectors are quite efficient and many of them are able to operate in real time

An initialization approach using natural features is part of the “Adaptive Feature Tracker” [15]. The article assumes that feature descriptors for the model have somehow been calculated in advance. Initialization is then done by finding the best feature descriptor matches between the image and the model.

4.6 Key frames

Key frames are photographs that have been taken from a tracked object in advance. The usage of key frames is a common technology that can be applied in many different ways to pose tracking, initialization and recovery both in model-based and point cloud based applications.

Common to all key frame methods is that that the pose determination is based on similarity comparisons between the current camera view and the key frames. The similarity comparisons are usually done using natural features that are detected using well-known techniques like SIFT or SURF. For efficiency reasons, the detection from the key frames is done in advance and the feature descriptors are stored so that efficient comparisons between the key frames and the video image are possible.

A typical point cloud based system is described in [16]. Tracking starts with an offline phase when photographs are taken from the target from different positions. Then natural features are detected from the photographs with SIFT algorithm to establish multi-view correspondences. Then SFM (structure from motion) method is used to estimate camera pose together with the 3D locations of the SIFT features. In the online phase, for each video frame appropriate candidate key frames are selected and the video frame features are matched to the candidate key frame features. Camera pose can be calculated using the 2D-2D correspondences between the video frame and the selected key frames and the known 3D locations of the key frame features. The special contribution in the system is an algorithm that finds an optimal subset from the original set of key frames and fast key frame recognition algorithm to select the candidate key frames.

Another example of key frames is an AR based industrial repair guidance application [17]. The application has a learning phase where the key frames are created. Markers in known 3D locations are used to calculate camera pose of the key frames and Lucas-Kanade to carry out the 2D-2D matching of features between the selected key frame and the camera view. It is mentioned that the approach has two drawbacks. First, the approach is vulnerable to changes in the illumination conditions. Second, even a small pose change may cause that the key frame features can not be found from the actual camera view.

The earlier mentioned ARTESAS system [5] uses key frames for pose recovery. The key frames are automatically created and saved during tracking. A set of SIFT features and their corresponding 3D model points are stored along each key frame. If pose recovery is needed, then the most alike key frame is searched from a database with help of a histogram comparison. Exact locations of the model 3D features on the current image are found by matching the SIFT features between the key frame and the current image.

4.7 Hypothesize-and-test

The hypothesize-and-test is a classic approach to solve the image-to-model registration problem [18]. The procedure is started by identifying probable model features from image. Then a small set of image feature to model feature matches are hypothesized and a pose estimate is calculated from these matches. Then the rest of the model features are projected to image from the hypothesized pose. If sufficient similarity is observed between the identified and projected features, then the pose is accepted. Otherwise, a different hypothesis is created and the process is repeated. Hypothesize-and-test can be based on RANSAC [19].

4.8 SoftPOSIT algorithm

The softPOSIT [18] [David et al 2004] algorithm is another approach to solve the image-to-model registration problem. SoftPOSIT combines the iterative softassign algorithm [20, 21] for computing correspondences and the iterative POSIT algorithm [22] for computing object pose under a full-perspective camera model. It treats all matches identically throughout the search for an optimal pose. The writers say that the complexity of the algorithm is better than any other algorithm for the image-to-model problem.

4.9 Cone culling

Cone culling [15] is a technique that can be used to automatically create new feature matches if an approximate pose is known. Thus, it is suitable for pose recovery.

Cone culling is started by detecting a potential model feature from the camera image. An imaginary cone is then drawn from the estimated pose to the direction of the detected feature. The width of the cone should correspond to the uncertainty of the pose estimate. It is easy to test, which model features are inside the cone. One of these features is then possibly matching the image feature. The final decision must be based for example on the appearance of the image feature.

5 BRAINSTORMING PROPOSALS

This chapter presents the brainstorming proposals. They are listed in the following order: equipment improvements, algorithmic proposals, manual initialization proposals and miscellaneous proposals.

I) Stereo camera

A stereo camera, or two cameras in a fixed distance, can be used to estimate distances to the features that are seen on camera image. This would provide more information both for initialization and tracking.

II) Depth camera

A depth camera aka RGB-D camera provides color information as well as the estimated depth for each pixel. Hence, the camera knows the distance to the target. The most well-known depth camera is Microsoft's Kinect.

III) Laser rangefinder

A laser rangefinder can be used to measure distances to objects, e.g. ship structures. This would provide supporting information both for initialization and tracking

IV) Compass

A compass can provide information about the orientation of the camera. Utilizing compass is tempting, because it is available in most mobile devices

The problem in a ship environment is that a compass is expected not to work properly inside a ship. Another problem is that the orientation of the ship itself should be known before compass information can be utilized.

It should be noted that there could be some benefit of compass even though the accuracy were poor, because even a 45 degree accuracy would be enough to specify, which wall the user is looking at.

V) Line intersections to be used for corner detection

One of the main challenges in initialization and pose recovery is the detection of model features from video image. The detection should be automatic or at least semi-automatic. If the tracking is corner-based, then corner detectors like Harris detector can be used. However, these detectors do not necessarily work reliable enough in ship environment for example because the physical corners in many cases are not sharp enough. Therefore, a more robust way to find the corners might be to detect line segments and calculate their intersection points.

VI) Rejection of messy image areas

Another (see above) challenge in automatic feature detection is messy image areas where a lot of features appear near each other. It is difficult to decide reliably the correspondences between the model and the image features in such areas. In addition, in many cases the messy areas are caused by an extra object on image and thus the features are not model features at all. If possible, it would be better to avoid the messy image areas during feature detection.

VII) Feature selection support

(a) Automatically detect and highlight image features that probably correspond to model features.
(b) Specially mark the feature that is the nearest to the current position of the pointer.
(c) Snap to the nearest detected feature when user selects an image point.

These three techniques help to circumvent the user interaction limitations. They can be utilized for example in manual point-to-point initialization.

VIII) Region of interest (ROI)

Search image features only from a local environment around the pointer.

There are two benefits in the proposal. For the first, it helps avoiding false feature detections as the user can select only distinct image areas. For the second, it improves computational efficiency, because the feature detection is one of the most time consuming operations in tracking and it would be enough to do it only in a small area.

IX) Specially marked environment key points

Mark some key points in ship structures so that the system can be easily and reliably recognize them from camera image

For example, the corners or the edges could be marked with some bright tape.

X) Marker based tracking

Use marker-based tracking instead of markerless tracking. Marker-based tracking is very mature technology.

6 EVALUATION

The proposals are evaluated according to the following criteria.

Criteria	**Explanation**
Applicability I / R / T	I = Applicable to **Initialization** R = Applicable to **Recovery** T = Applicable to **Tracking** Tracking was not in the main scope of the workshop, but can take advantage of some of the proposals.
Functionality	Does it solve the initialization or recovery problem? The comment "complementary" means that the proposal can be used to boost some method, but is not a complete method itself.
User aspects	What kinds of actions are required from user?
Operability	The system's ability to stay operable during the initialization. For example, can the camera be moved?
Effort?	How big effort is it to implement the proposal?

Method	**Applicability**	**Functionality**	**User aspects**	**O perability**	**Effort**
(chapter 5.1) Manual point-to-point matching	I R	Yes, but pose estimate will not be accurate.	Troublesome to use in a complex environment.	Restricted mobility.	Small
(chapter 5.2) Manual model alignment	I R	As above	As above	Yes	Small
(chapter 5.3) Semi-automatic model alignment	I R	Yes	Less troublesome than above methods, because alignment need not be so accurate.	Restricted mobility.	Doable, but not trivial.
(chapter 5.4) Model matching	I R	Yes in principle, but further development is needed. Ship construction environment is too complex for the view-class approach.	No user intervention needed.	No or restricted mobility. System might be inoperative during initialization. Initialization may take a remarkable time.	Major
(chapter 5.5) Natural feature matching	I R	It is an open question, how to find natural features from a model and how to create descriptors for them.	No user intervention needed.	Probably yes.	The open questions must be solved.
(chapter 5.6) Key frames	I R T	Yes, well-known technology.	Creating the key frames is a remarkable extra effort for the user.	Restricted mobility.	Medium

Method	**Applicability**	**Functionality**	**User aspects**	**O perability**	**Effort**
(chapter 5.7) Hypothesize-and-test	I	Yes	No user intervention is needed.	Restricted mobility.	Medium / Major
(chapter 5.8) SoftPOSIT algorithm	I	Yes	No user intervention is needed.	Restricted mobility.	Medium / Major
(I) Stereo camera	I R T	Open question, how to utilize the depth information. Only complementary. Extra HW needed.	No user intervention needed?	Yes	Medium / Major
(II) Depth camera	I R T	Open question, how to utilize the depth information. Only complementary. Extra HW needed.	No user intervention needed?	Yes	Medium / Major
(III) Laser range-finder	I R T	Open question, how to utilize the depth information. Only complementary.	User should point different objects with the range finder.	Yes	Medium
(IV) Compass	I R	Functionality inside a ship and usefulness debatable. Only complementary.	N/A	Yes	Small

Method	**Applicability**	**Functionality**	**User aspects**	**O perability**	**Effort**
(V) Line intersections for corner detection	I R	Yes. Only complementary.	User may have to manually adjust some tuning parameters.	N / A	Small
(VI) Rejection of messy image areas	I R T	Yes Only complementary.	N/A	Yes	Small / Medium.
(VII) Feature selection support	I (R)	Yes Only complementary.	N / A	Yes	Small / Medium
(VIII) Region of interest (ROI)	I R T	Yes Only complementary.	N / A	Yes	Small
(IX) Specially marked key points	I R	Yes Only complementary.	Marking the key points is a remarkable effort.	N/A	Small
(X) Marker-based tracking	I R	Yes.	Placing the markers is a remarkable effort.	N/A	Small, solutions are available.

7 CONCLUSIONS

7.1 Basic tablet computer initialization solution

Manual point by point matching (5.1) is straightforward to implement and works reliably. It is a good choice for example when the system is being developed and the initialization is not the main interest. The drawback in the method is that in some cases the matching requires considerably effort from the user.

A remarkable benefit in the point-to-point method is that it is robust to changes in the environment. Therefore, it would be good to include it to all systems as a backup method.

The point-by-point method can be made easier to use by applying some of the auxiliary techniques

The **feature selection support** techniques (VII) make the initialization more accurate and easier for the user. Harris corner detector and the line intersection proposal (V) can be used for automatic feature detection.

Region of Interest (ROI) approach (VIII) makes the automatic feature detection techniques to work more reliably and computationally more efficiently.

Rejection of messy image areas (VI) is useful, but probably not needed if ROI approach is used.

7.2 Advanced tablet computer initialization solution

Initialization can be made more user-friendly by using the **semi-automatic model alignment method (5.3)**, especially if the background fitting strategy is used. The implementation may not be trivial, especially if the manual alignment requirement is loose and the device must be kept fully operable all the time.

7.3 AR glasses initialization solution

The basic tablet computer initialization method, point by point matching, relies on the touch screen and is therefore not applicable for AR glasses. Most probable the semi-automatic alignment suits best for glasses.

7.4 Hypothesize-and-try & softPOSIT

Hypothesize-and-try and softPOSIT methods could provide opportunity for fully automatic initialization. However, the functionality of the methods in a ship building environment should be verified before their real feasibility could be evaluated.

7.5 Pose recovery

The best pose recovery strategy is to try to avoid the need for the pose recovery altogether by continuously adding new matched point.

If the pose is lost anyway, then the **cone culling** method (5.9) can be used to match image features to model features. A difficulty in this approach is that there may be several model features in the direction of a detected image feature and it may not be trivial to decide, which one of them matches the image feature.

If the **semi-automatic model alignment** (5.3) is used, then alignment procedure can be applied using the last known pose as a starting point.

Gyros and acceleration meters (Chapter 2 / Sensor-based tracking) could be used to estimate pose changes during periods when the tracking is failing.

7.6 Backup initialization solution

Specially marked key points (IX) can be considered if the initialization and tracking do not otherwise work reliably enough.

If everything else fails, then **marker-based tracking** (X) is a well-known and reliable technology.

8 REFERENCES

[1] V. Lepetit and P. Fua, ”Monocular Model-Based 3D Tracking of Rigid Objects: A Survey”,
Foundations and Trends in Computer Graphics and Vision, vol. 1, No 1 (2005) 1–89.

[2] B. Lucas and T. Kanade, ”An Iterative Image Registration Technique with an Application to Stereo Vision”, International Joint Conference on Artificial Intelligence, pages 674–679, 1981.

[3] S. Helle, M. Kaustinen, S. Korhonen and T. Lehtonen, “Indoor Localization Solutions for a Marine Industry Augmented Reality Tool”, University of Turku, Technical Reports, No 1, 2013.
Available at http://www.doria.fi /handle/10024/93484.

[4] VTT, #Augmented Reality for Building Construction (AR4BC)”.
http://virtual.vtt.fi/virtual/proj2/multimedia/videos/2010/003-mobile-augmented-visualization.html

[5] R.Koch, J-F. Evers-Senne, I. Schiller, H. Wuest and D. Stricker.
http://biecoll.ub.uni-bielefeld.de/volltexte/2007/11

[6] V. Kang and S. Eiho, “3D Tracking Using 2D-3D Line Segment Correspondence and 2D Point Motion”, Published in J. Braz et al. (Eds.): VISAPP and GRAPP 2006, CCIS 4, pp. 367-380, 2007.
@Springer-Verlag, Berlin Heidelberg.

[7] L. Shapiro and G. Stockman, “Computer Vision”, Prentice-Hall 2001, pp. 479 – 526.

[8] K. Pulli and L. Shapiro, “Triplet-Based Object Recognition Using Synthetic and Real Probability Models”, Proceedings of ICPR, Wien, Austria, August 1996.

[9] M. Costa and L. Shapiro, ”3D Object Recognition and Pose with Relational Indexing”, Computer Vision and Image Understanding 79,pp 364–407, 2000.

[10] D. Lowe, ”Object recognition from local scale-invariant features”, Proceedings of the International Conference on Computer Vision, vol 2. pp. 1150–1157.

[11] H. Bay, A. Ess, T. Tuytelaars and van L. Gool, ”SURF: Speeded Up Robust Features”, Proceedings of the 9th European Conference on Computer Vision, Springer LNCS volume 3951, part 1. pp. 404–417.

[12] C. Harris and M. Stephens, ”A combined corner and edge detector”, Proceedings of the 4th Alvey Vision Conference. pp. 147–151, 1988.

[13] J. Shi and C. Thomasi, “Good Features to Track”, IEEE Conference on Computer Vision and Pattern Recognition.

[14] E. Rosten and T. Drummond,”Machine learning for high-speed corner detection”, ECCV'06 Proceedings of the 9th European conference on Computer Vision - Volume Part 1, Pages 430-443.
Springer-Verlag Berlin, Heidelberg ©2006

[15] J. Herling andW. Broll, “Markless Tracking for Augmented Reality”
In book Borko Furht (ed.), ”Handbook of Augmented Reality”, Springer New York, Dordrecht Heidelberg London 2011, pp 255 – 272.

[16] Z. Dong, G. Zhang, J. Jia and H. Bao, ”Keyframe-Based Real-Time Camera Tracking”, ICCV, 2009, pp. 1538–1545.

[17] J. Platonov, H. Heibel, P. Meier and B. Grollman, “A mobile markerless AR system for maintenance and repair”, ISMAR 2006. IEEE and ACM International Symposium on Mixed and Augmented Reality, 2006, pp 105-108.

[18] P. David, D. DeMenthon, R. Duraiswami and H. Samet, “SoftPOSIT: Simultaneous Pose and Correspondence Determination”, International Journal of Computer Vision, 59(3):259–284, 2004.

[19] M. Fischler and R. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, Comm. of the ACM **24** (6): 381–395.

[20] St. Gold and A. Rangarajan, “A graduated assignment algorithm for graph matching”, IEEE Trans. on Pattern Analysis and Machine Intelligence 18(4) 1996:377–388.

[21] S. Gold, A. Rangarajan, C-P. Lu, S. Pappu and E. Mjolsness, “New algorithms for 2D and 3D point matching: Pose estimation and correspondence”, Pattern Recognition 31(8):1019–1031.

[22] D. DeMenthon and L. Davis, “Model-based object pose in 25 lines of code”, International Journal of Computer Vision 15(1/2):123–141 (1995).