# Pose Tracking for Mobile Robot Localization from Large Scale Appearance Mosaics

Alonzo Kelly

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890
email: alonzo@ri.cmu.edu, url: http://www.frc.ri.cmu.edu/~alonzo

**Abstract**

*While visual odometry has unbounded error, navigation from pre-existing consistent scene models can generate extremely repeatable position estimates. This paper discusses a new approach to localization motivated by the fact that many man-made environments constain substantially flat, visually textured surfaces of persistent appearance. For this important class of vision-based navigation problems the scene model can be reduced to a 2D surface painted with real textures - in other words, an image mosaic. Straightforward techniques from image-based localization and mosaicking are used to produce a field relevant AGV guidance system based on only vision and odometry. The visual tracking and localization aspects of the approach are described. We show that this approach to localization is able to exceed the speed barriers due to distortion and image overlap that are intrinsic to visual tracking and odometry. Speed can, however, become limited by a new mechanism - the inherent instability of visual tracking when operating in the regime beyond the overlap spped limit. Field trials currently demonstrate that the particular simplifications resulting from a downward looking camera configuration produce a guidance system repeatable to 1 mm throughout a 50,000 square foot facility with an MTBF (corresponding to loss of visual lock) of $10^6$ images or five days of operation.*

## 1 Introduction

Imagine yourself flying over a city in a small airplane. Let the airplane be restricted to level flight and let the terrain below be assumed to be essentially flat. That is, let the terrain undulations be small relative to the aircraft altitude. You can see the ground below through a small viewfinder in the floor. You have a map of the city in the form of a large, high resolution photograph constructed by mosaicking. Your task is to locate yourself, to the nearest building, by matching the views in the viewfinder to the mosaic.

This scenario illustrates the technique of *mosaic-based localization* described in this paper. Replace the viewfinder with a camera; replace the airplane with any vehicle travelling parallel to a mostly flat surface; restrict vehicle motion to the streets; and you have the general idea. This approach to localization has shown itself to be both robust and of high performance in the environments to which it is targeted.

### 1.1 Mosaic-Based Localization

We will represent the environment as an appearance model - in all its photorealistic richness and we will use mosaicking techniques to construct this model. Our technique differs from *visual odometry* [18] in that considerable effort is expended to create a globally consistent model. It differs from *landmark-based localization* in that the scene is represented in an iconic form rather than as a list of landmark locations.

The steps of our mosaic-based approach to localization are:

- Construct a mosaic of an appropriate area.
- Render it globally consistent and store it in persistent memory.
- Subsequently track motion over the mosaic using a visual tracker which computes camera pose.

While straightforward in principle, actual construction of such a system raises such issues as memory capacity, quality of visual texture and processing power. These issues have been discussed in previous papers [12] [13].

Although there are clear alternatives, we will exploit the particular advantages of using floor imagery rather than images of other surfaces:

- the camera can be mounted closer to this surface
- suitable camera to scene geometry is assured
- shielding from ambient lighting is easy

Also, while many other applications satisfy our scene constraints, we will discuss the details of an application to industrial AGVs.

### 1.2 Rationale

Given that a mosaic scene model can be constructed in principle, it still remains to explain why it is even worth such effort. For our purposes, a mosaic is a particularly convenient and appropriate form of prior scene model. This conclusion can be rationalized as follows:

- *Prior Models Enable Higher Tracking Velocities*: When speed exceeds levels at which successive images overlap in the scene, there is no information that can be tracked from image to image. However, referencing a prior model eliminates the image overlap constraint so long as some part of the model remains in view.
- *Global Consistency Imparts Repeatability:* If the model is globally consistent, reported position becomes a one-to-one function of actual position and

the system becomes as repeatable as its fundamental resolution.

- *Iconic Models are Best in Featureless Scenes*: If features are rare, spatially distributed and/or subtle, an iconic representation (rather than a feature-based one) encodes the maximum useful information in terms of providing the best immunity from false correspondence matches and highest spatial (sub-pixel) resolution.
- *Geometry Assumptions Simplify Processing*: Of course, when scene geometry can be regarded as known, algorithms need not recover shape as well as motion, and the distortion of iconic features due to motion can be predicted.

### 1.3 Key Assumptions

When cameras are used for vision-based localization, the ability to render a scene permits navigation from real-time imagery [22]. While it is certainly possible to compute unrestricted 3D camera motion in an (even unknown) 3D scene [27], our application will make and exploit several more simplifying assumptions:

- *Persistent Appearance:* The use of a persistently stored model of scene appearance assumes that the actual appearance of the scene will not change significantly over operationally significant periods of time. Exceptions to this assumption are common, but the appearance change needs to be significant and it needs to occur everywhere in order to render the present technique inoperable.
- *2D Scene:* We will use appearance models constructed from real imagery. While completely general 3D polygon models are certainly possible, we will assume that the scene can be represented by a mosaic mapped onto a 2D surface. This assumption applies, at least locally, to most man-made indoor and outdoor environments.
- *Substantially Flat Scene:* While the assumption can be completely relaxed in general (e.g. in computer graphics), we will assume that the scene is flat enough that self occlusion and depth discontinuities cannot occur. This assumption also applies, at least locally, to most man-made indoor and outdoor environments.
- *Restricted Camera Motion:* While arbitrary camera motion is computable, we will restrict motion to be consistent with a camera being mounted under a terrain-following vehicle as shown in Figure 1. Under these conditions, the general problem of "rendering" the scene is reduced literally to that of extracting the pixels in the rectangular region predicted to be in view.
- *Restricted Mosaic Topology:* While not necessary in general, we will confine our attention to environments where vehicle motion is restricted to roadways or guidepaths, rather than regions wider than an image in more than one direction, except at intersections. To do so simplifies considerably the problem of constructing globally consistent mosaics.
- *Primary Position Estimate:* Since we will confine the application to that of vehicles, it is useful and not overly restrictive to assume the availability of an independent estimate of camera motion between frames. This *primary position estimate* can be used to increase reliability and very significantly increase tracking performance and therefore vehicle speed.

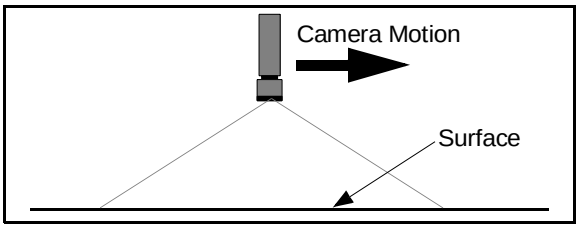


**Figure 1: Simplest Scenario**. Here a camera is mounted normal and at constant height with respect to a surface and it moves parallel to the surface. Assumptions imply that variations in foreshortening over the image do not occur.

### 1.4 Prior Work

While the notion of localization from large scale mosaics seems to be new, its implementation is based upon decades of related computer vision and computer graphics work. Many different techniques have been proposed for localization in general [2]. Certainly, navigating from imagery is a basic technique in robotics [26]. Techniques may use appearance (cameras [1]) or shape (radar [5], sonar [4] or lidar [6]), or both [17].

Automated mosaicking is often useful in its own right. Applications include station keeping [24], video coding [11], image stabilization [19], and visualization [25]. Only recently have near real-time [21] and globally consistent [20] mosaicking solutions emerged.

The literature on determining the motion of a camera and/or the geometry of a scene is extensive. Motion can be recovered from a known scene [29] and this problem is related to visual odometry. Scene structure can be determined from camera motion [28][16]. Shape and motion can also be determined simultaneously [27] and all shape and motion assumptions seem ultimately unnecessary.

Once a camera is permitted to move relative to a scene, one can observe correspondence or flow. For correspondence, the related problem of visual tracking [9][23] becomes important.

It is well-known that relatively few correspondences between the image and the scene are necessary to constrain the relative pose of a camera and a known object or scene [10]. Fairly general 3D solutions for finding the relative pose have been known for some time[8].

Since we will render predicted imagery, this work is also peripherally related to image-based rendering [3][15][30]. This problem is itself related to visual tracking in that the motions and deformations of all regions of the image are being predicted.

## 2 Performance and Reliability Analyses

This section explores some figures of merit that are particularly relevant to a mosaic-based approach to localization. After some simple analyses, it becomes clear that our scene geometry assumptions, a prior model, and odometry aiding lead to levels of tracking performance that are not

possible in visual odometry and visual tracking. Several different regimes of operation exist which indicate clearly why mosaic-based localization achieves relatively high levels of tracking performance.

## 2.1 Projective Mapping

In mapping quantities in the scene to their associated quantities in the image plane, scene geometry dependence and projective loss of information can, of course, complicate matters. Figure 2 indicates the simplest case of motion confined to a single axis parallel to the image plane and defines notation for this section.

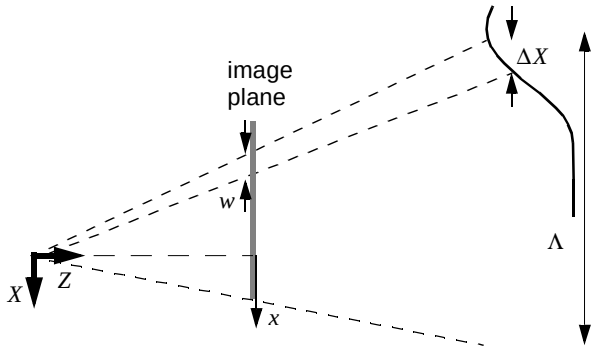


**Figure 2: Simplified Projection**. Notation for a projection from 2D to 1D.

Let $x$ denote the image coordinate and $X$ denote the corresponding scene coordinate while $Z$ denotes depth. If the focal length is $f$, the basic projective mapping to the image plane is elementary:

$$x = (f/Z)X$$

In this restricted case, if depth is constant over a small motion in the scene, differential motions and hence velocities scale linearly from the scene to the image:

$$\Delta x = (f/Z)\Delta X$$

So, depth in units of focal length is an important scaling parameter. Limits on the feature velocities that can be tracked give rise to corresponding limits on associated camera-to-scene relative velocity in the direction parallel to the image plane.

## 2.2 Performance Attributes Related to Geometry

Of course, one of the many implications of the above mapping is that points at different depths have different image velocities, and therefore, depth gradient in a template implies distortion as the template moves. If we also allow depth to vary across a template, the total differential is:

$$\Delta x = (f/Z)\Delta X - (f/Z^2)Z_X\Delta X$$

and a dependence on depth gradient in the scene $Z_X$ (infinite at occluding boundaries) is thereby introduced. Dividing by a small time increment it becomes clear that while the first term indicates template motion due to camera motion, the second indicates a distortion effect.

### *2.2.1 Distortion Speed Limit*

When the range to features varies substantially over an image, severe limits on speed of tracking can be introduced by the distortion and/or occlusion of features from frame to frame. A small patch of scene of width $\Delta X$ projects onto an image template of width:

$$w = (f/Z)\Delta X$$

From the second term above, we can derive that, as the template moves across the image due to camera motion at speed $V$, for a time period of $T_{cyc}$, its width changes by:

$$\Delta w = (f/Z^2)Z_X V T_{cyc}$$

Hence, the change in template size in relation to size is:

$$\frac{\Delta w}{w} = \left(\frac{Z_X}{Z}\right)\frac{VT_{cyc}}{\Delta X} = \left(\frac{Z_X}{Z}\right)\left(\frac{u}{w}\right)T_{cyc}$$

where $u$ is the template velocity in the image plane.

If distortion is to be reduced in order to make feature matching easier, the only way to do so for given geometry is to reduce the nondimensional $u/(w/T_{cyc})$ which represents the template velocity in units of templates per cycle. If template size is determined by texture content or available computation, this can only be accomplished by reducing speed, or cycle time.

Of course, the expression also shows that distortion is eliminated if there is no depth gradient across the template. When geometry is such that features can be tracked across a significant portion of the image in a single cycle, another limit comes into play. This elementary observation is important here because our scene geometry assumptions allow us to break the distortion speed barrier.

### *2.2.2 Overlap Speed Limit*

In many visual tracking applications, tracking features in successive frames implies a fundamental speed limit induced by the geometric constraint of overlapping fields of view. If the image width at the feature depth is $\Lambda$, and $\beta$ is the fraction of image overlap required for matching, the camera speed $V$ must satisfy:

$$V < \left(\frac{\Lambda}{T_{cyc}}\right)(1-\beta)$$

The velocity of the camera in units of images moved per cycle is thereby limited to a value somewhat less than unity. If no feature is to be skipped over, the calculation must be performed for the feature at the minimum depth - whose velocity in the image is highest.

This observation is important because the use of prior models such as mosaics allow us to also break this speed barrier when tracking.

### *2.2.3 Geometric Instability*

Exceeding the overlap speed limit has important implications on the stability of tracking. Suppose for simplicity

that depth is constant, and that the camera moves parallel to the image plane. The heading can be determined from the positions of two features sufficiently separated in the image. Let one feature be mislocated by an error $\Delta X_k$. The resulting effect on the computed heading is:

$$\Delta\theta_k = \Delta X_k / \Lambda$$

Suppose further that the camera travels a distance $S$ before another visual fix is attempted. At this point, the error of the original fix causes a position error normal to the direction of travel at the new position of:

$$\Delta X_{k+1} = (S/\Lambda)\Delta X_k$$

Since the nondimensional ratio of distance travelled to image width exceeds unity beyond the overlap speed limit, visual tracking becomes *unstable*. Hence, errors of acceptable magnitude in a given visual fix can cause loss of visual lock in the next cycle. Such errors might be due, for example, to false feature matches or mosaic distortions.

Pragmatic strategies to manage this issue include redundant sensing of intervening motion, estimates of accrued error, higher update rates, consistent mosaics, robust matching, and outlier rejection in pose determination.

## 3 Mosaic Tracker Design

The overall architecture of our position estimation system is as shown in Figure 3.

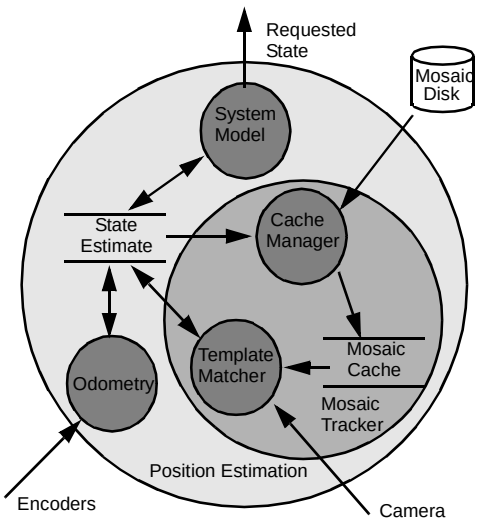


**Figure 3: Position Estimation System.** The system model integrates the equations of motion to interpolate to the instant of time requested. The odometry system and mosaic tracking system provide two complementary estimates of state. For large scale mosaics, a cache is needed to store part of the mosaic in RAM.

As long as the system is operating, camera imagery is acquired as fast as processing can manage. Simultaneously, an odometry thread of execution continues to read the wheel encoders to provide position estimates between image acquisitions.

Position estimates between encoder readings are supplied by integrating the equations of motion under a constant linear and angular velocity assumption. This system model runs continuously whether or not state requests have been received.

The inner circle delineates the mosaic tracker. The job of the mosaic tracker algorithm is the core problem of mosaic-based localization. It must determine where the camera is over the map. Following the localization literature, this section will use the more generic term *map* to mean the mosaic.

### 3.1 Mapping and Localization Modalities

The position estimator is coordinated with a mapping process which operates in a number of useful modes.

#### *3.1.1 Visual Odometry*

When the camera is continuously moving over an unknown area, provided successive images overlap, visual odometry, perhaps aided by the encoders, can be performed. At such times, evolving position error is unbounded.

#### *3.1.2 Automatic Mapping*

While traversing unknown areas, the acquired imagery can optionally be added to the mosaic. If it is, the position reported for the location becomes repeatable. While this process can produce mosaics automatically, they are not guaranteed to be globally consistent unless they are (both apparently and actually) acyclic. At such times, the overlap speed limit applies. Rendering large scale mosaics globally consistent is normally an off-line process due to its excessive computational requirements.

#### *3.1.3 Simultaneous Localization and Mapping*

When both visual odometry and automatic mapping are being performed, the system is performing a restricted form of simultaneous localization and mapping [14].

#### *3.1.4 Automatic Map Updates*

In principle, an image which appears sufficiently different from expectations, but is nonetheless confidently positioned, can be used to overwrite the data in the map to reflect a change in appearance.

#### *3.1.5 Automatic Mode Switching*

It is possible for the system to automatically switch from one mode to another. Conceptual logic is as follows:

```
if (the current image has minimal
    overlap with the mosaic)
    {
    add it to the mosaic;
    }
else if (the current image looks
    very different from the mosaic,
    but is confidently positioned)
    {
    overwrite the mosaic with the
    new image;
    }
else
    {
    compute pose and discard the image;
    }
```

Of course, the last case is the mosaic tracking case where repeatable position estimates beyond the overlap speed limit are achievable.

### 3.2 Pose Refinement

The fact that a mosaic can be constructed means that image registration and pose determination are equivalent problems. Many approaches to a solution are possible. Anticipating future work, we have used a Kalman filter [7] which determines the planar pose which aligns a set of corresponding planar point features. While the assumption can be easily relaxed to an affine transform, we presently expect several features to move as a rigid unit. To compute the pose of the image, attach a model frame $M$ to an arbitrary location on it. Similarly, attach a world frame $W$ to an arbitrary location on the mosaic.

The predicted positions of the point features with respect to the model frame ${}^{m}\underline{r}$ come from their positions in the image. The observed positions of the corresponding features ${}^{w}\underline{r}$ come from their positions in the mosaic. The problem is to find the pose $\rho = (a, b, \theta)^T$, or associated transform ${}^{w}_{m}T(\rho)$ which best aligns the corresponding points. The situation is summarized in Figure 4:.

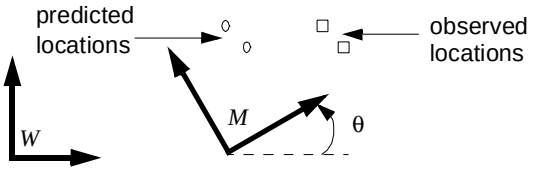


**Figure 4: Observer Formulation**. Given a set of point positions ${}^{m}\underline{r}$ expressed in the model frame $M$, and a corresponding set of point positions ${}^{w}\underline{r}$ expressed in the arbitrary frame $W$ called "world", find the best pose of frame $M$ with respect to frame $W$ which brings the points most nearly into coincidence.

The prediction equation, or *observer*, tells us how to predict the locations of the points in the mosaic (world frame) from their locations in the image (model frame).

$$ {}^{w}\underline{r} = [{}^{w}_{m}T(\rho)]\,{}^{m}\underline{r} $$

If we denote the Kalman state vector thus:

$$ \underline{x} = (a, b, \theta)^T $$

then this relationship is of the form $\underline{z} = h(\underline{x})$ of the standard observer where the pose is the state vector, the model locations of the points are constant, and the prediction is the world locations of the points.

## 4 Results

A working version of a mosaic-based navigation system-system has been in operation for about 4 years. It represents a free-ranging automated guided vehicle exhibiting excellent repeatability in a facility where no infrastructure has been installed to supports its operations.The system functions like GPS, laser or wire guidance in that it provides a position fix, when requested, to be used to damp the growth of errors that unavoidably occurs in a primary position estimation system such as odometry.

The system has been in operation in our 40,000 square foot facility at the National Robotics Engineering Consortium and it has also been tested in two others. It has been installed on tug, unit load, and forked AGVs - including the tug AGV shown in Figure 5.

**Figure 5: Tug AGV.** This automated guided vehicle is one of three with mosaic positioning installed.

A network of guidepaths has been mapped and rendered geometrically consistent for our testing purposes. This map is shown in Figure 6. Our installation process calibrates cameras to produce images of ideal geometry so that one vehicle can generate a mosaic on behalf of all. The short horizontal segments of the map are areas where the vehicle interfaces with racks and loads.

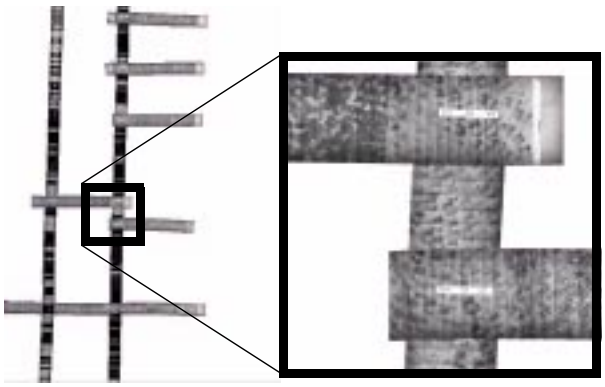

**Figure 6: Network Mosaic and Exploded View of Component Imagery.** This network of images covers part of our test facility floor.

On this particular mosaic, the system has operated for four years producing 1 mm repeatability at speeds sometimes exceeding 15 mph - more than safety regulations would allow outside our controlled laboratory setting. We have recently achieved a milestone of 40 hours of error-free operation in order to demonstrate commercially relevent levels of reliability.

We have observed excellent noise immunity in the template correlation algorithm used to match features. It operates robustly in the face of months of cumulative dust and grime which can hide the underlying floor texture that was

originally mapped. This level of noise immunity is partly due to the ability to use very large feature templates when the scene is flat, and partly due to the excellent noise rejection performance of cross-correlation.

## 5 References


[1] C. S. Andersen, S. Jones, J. Crowley, “Appearance Based Processes for Visual Navigation”, Proc SIRS’97, pp227-236, 1997.

[2] J. Borenstein, B. Everett, and L. Feng, "Navigating Mobile Robots: Systems and Techniques." A. K. Peters, Ltd., Wellesley, 1996.

[3] E. Chen and L. Williams. “View Interpolation for Image Synthesis”, In Proc. SIGGRAPH 1993.

[4] J. L. Crowley, “World modelling and position estimation for a mobile robot using ultrasonic ranging”, Proc IEEE Int. Conf. on Rob and Aut., 1989.

[5] H. F. Durrant-Whyte, “The design of a radar-based Navigation System for Large Outdoor Vehicles”, Proc IEEE Int. Conf. on Robotics and Automation, Aichi, Japan, pp 764-769.

[6] T. Einsele, “Real-Time Self-Localization in Unknown Indoor Environment Using a Panorama Rangefinder.”, In IEEE/RSJ International Workshop on Robots ans Systems, IROS 97.

[7] A. Gelb, Ed., Applied Optimal Estimation, MIT Press, Cambridge MA, 1974.

[8] D. B. Gennery. Visual tracking of known three-dimensional objects. *Int’l Journal of Computer Vision*, 7(3):243-270, 1992.

[9] G. Hager, and P. Bellhumeur. Efficient Region Tracking with parametric models of illumination and geometry. *IEEE Journal of Patt. Recog and Machine Intelligence* 20(10),1025-1039, Oct 1998.

[10] ,R. Horaud, B. Conio, O Leboulleux, and B. Lacolle, An Analytic solution for the perspective 4-point problem. Computer Vision, Graphics, and Image Processing 47(1),:33-44, 1989.

[11] M. Irani, P. Anandan, and S. Hsu. mosaic-based representations of video sequences and their applications. *Proc. Intl. Conf. on Computer Vision*, pages 605-611, 1995.

[12] A. Kelly, “Contemporary Feasibility of Image Mosaic Based Vehicle Position Estimation.” Proceedings of IASTED International Conference on Robotics and Applications, Santa Barbara, October 1999.

[13] A. Kelly, “Mobile Robot Localization from Large Scale Appearance Mosaics.” To appear, International Journal of Robotics Research, November 2000.

[14] J. J. Leonard and H. F. Durrant-Whyte. “Simultaneous Map Building and Localization For An Autonomous Mobile Robot”, *In IEEE/RSJ International Workshop on Robots and Systems, IROS 91*, Pages 1442-1447, 1991.

[15] S. Laveau and O. Faugeras, “3D Scene Representation as a Collection of Images”, in Proc. ICPR, 1994.

[16] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter based algorithms for estimating depth from image sequences. International Journal of Computer Vision, 3(3):209-236, September 1989.

[17] J. Niera, J. D. Tardos, J. Horn, and G. Schmidt, “Fusing Range and Intensity Images for Mobile Robot Localization”, IEEE Trans on Rob. and Aut., Vol 15, No 1, p 76, 1999.

[18] C. Olson, L. Matthies, M. Schoppers, M. Maimone “Robust Stereo Ego-Motion for Long Distance Navigation”, in Proc. Conference on Computer Vision and Pattern Recognition

[19] L. Wixon, J. Eledath, M. Hansen, R. Mandelbaum, D. Mishra, Image Alignment for Precise Camera Fixation and Aim, Proc. Conference on Computer Vision and Pattern Recognition (CVPR ‘98).

[20] H. S. Sawhney, S. Hsu and R. Kumar, Robust video mosaicking through topology inference and local to global alignment, In Proc. European Conference on Computer Vision, ECCV, Freiburg Germany vol 2, pages 103-119, June 1998

[21] H. S. Sawhney, R. Kumar, G. Gendel, J. Bergen, D. Dixon, V. Paragano, VideoBrush: Experiences with Consumer Video Mosaicking, Fourth IEEE Workshop on App. of Comp.Vision, WACV, Oct 98

[22] M. Schmitt, M. Rous, A. Matsikis, K. F. Kraiss, “Vision based self-localization of a mobile robot using a virtual environment”, Proc IEEE Int. Conf. on Rob. and Aut., Detroit, MI, May 1999.(CVPR ‘00).

[23] J. Shi, and C. Tomasi, “Good features to track”, In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR94), Seattle, June 1994..

[24] Richard L. Sparks, Stephen M. Rock, and Michael J. Lee, Real-Time Video Mosaicking of the Ocean Floor. *IEEE Journal of Oceanic Engineering*, Vol 20, No. 3, July 1995.

[25] R. Szeliski. Image mosaicking for tele-reality applications. IEEE Wkshp. on Applications of Computer Vision, pages 44-53, 1994.

[26] R. Talluri and J. K. Aggarwal, “Position Estimation Techniques for an Autonomous Mobile Robot - a Review., Handbook of Pattern Rcognition and Computer Vision, pp 769-801, World Scientific.

[27] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization Method. Int J. Computer Vision, 9(2):137-154, 1992.

[28] S. Ullman, The interpretation of visual motion, MIT Press, Cambridge Ma, 1979.

[29] J. Weng, T. Huang, and N. Ahuja, “3D motion estimation, understanding, and prediction from noisy image sequences”, IEEE Trans Pattern Analysis and machine intelligence, 9(3), p 370-389, May 1987.

[30] T. Werner, R. D. Hersch, and V. Hlavac, “Rendering Real-World Objects Using View Interpolation”. In Proc ICCV, Boston, 1995.