Contemporary Feasibility of Image Mosaic Based Vehicle Position Estimation

ALONZO KELLY

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

Keywords: mobile robots, mosaicing, visual tracking, image registration, position estimation, template matching

Abstract

Perhaps one of the simplest mechanisms for navigation in an area is to use real-time imagery to track vehicle motion over a very large, previously stored, high resolution image mosaic of the scene. In structured environments, image mosaics of textured surfaces present a potential for inexpensive, highly repeatable position estimation that is both higher performance and cheaper than existing commercial alternatives. This paper presents a case for the feasibility of such an approach using contemporary cameras and computing devices.

1 INTRODUCTION

Perhaps one of the simplest mechanisms for navigation in an area is to use real-time imagery to track vehicle motion over a large, previously stored, high resolution image mosaic of the scene. This technique will be referred to here as **mosaic-based position estimation**. In one sense, a traditional paper road map is a low resolution feature-enhanced version of such an image.

A little reflection on the ratio of required position resolution to the area of excursion leads to the conclusion that significant memory may be needed to store the mosaic. A little more reflection leaves one wondering if enough processing power can be brought to bear. Issues of lack of texture, repeated texture, occlusion and perspective foreshortening distortion and the difficulty of constructing the mosaic itself also emerge.

Systematic consideration of these potential problems leads to the conclusion that contemporary sensing and computing technology are adequate to the task in suitable environments. This is particularly true when a high frequency position estimation system, called here a primary position estimation system, is available to track motion between lower frequency visual fixes.

1.1 MOTIVATION

In man-made environments, both indoor and outdoor, a robot vehicle introduced into the environment normally moves over a mostly flat floor surface. Walls and ceilings, if present, are also composed of locally flat surfaces punctuated by occasional abrupt shape discontinuities. It is often the case that at least some of these surfaces present at least some areas of local visual texture.

In its oldest sense, a **map** is a mechanism that relates observations to locations. In this same sense, a large image or a collection of images of a scene and their relative or absolute locations constitutes a map from which one may navigate. The only requirements are that there be:

- a mechanism that can be used to determine the location of pixels in the map when they happen to be observed in the environment, and
- a mechanism that can determine camera pose from these pixel locations.

1.2 PROBLEM DESCRIPTION

We are interested here in the specific problem of determining the motion, with respect to a fixed scene, of a camera, and hence of the vehicle to which the camera is attached. Indeed, accuracy of the mosaic, either absolute or relative, need not necessarily be of much concern. Any applications for which motion trajectories can be taught and replayed will benefit significantly just from the repeatability of mosaic-based position estimation.

It will not always be necessary for the scene surface(s) to be a single flat surface or for the camera to move parallel to the surface. However, the following figure illustrates this simplest scenario.

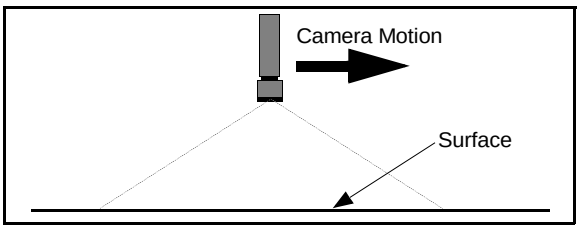

Figure 1: SIMPLEST SCENARIO: Here a camera is erected normal and at constant height with respect to a surface. It moves parallel to the surface - hence variations in foreshortening do not occur.

1.3 TERMINOLOGY

Our proposed system compares immediately acquired imagery with imagery taken previously. Previous imagery may have been acquired milliseconds ago in the current excursion or years ago in some other excursion. We refer to the the most immediately acquired imagery as the **image** and any previously acquired imagery as the **mosaic** or **map**. The map may or may not have been processed into a single coherent block of pixels. It may simply be a list of images and their locations.

In the event the previous imagery comes from the last image acquired, differentials are being integrated, and the

system is operating in **relative** mode (and may be constructing a map). In the event the previous imagery comes from a map calibrated against some reference, and map-referenced position is being generated, the system is operating in **absolute** mode. In the event the system is constructing a map by saving imagery, it is operating in **mapping** mode. When referencing imagery to a preconstructed map, the system operates in **tracking** mode.

1.4 Comparison to Stereo

Many of the problems of mosaic-based position estimation, and their solutions, are common to stereo vision. The difficulties associated with foreshortening in area-based stereo vision are well known. Fundamentally, the same part of the scene is distorted differently from different viewpoints if the part of the scene varies in range relative to the camera. Mosaic-based positioning, however, is a dramatically simpler than stereo vision because:

- disparity is only required at a few (2-10) places in an image pair in order to determine pose error.
- disparity gradient is usually guaranteed to be small and to vary slightly and smoothly over the entire image.
- disparity is a direct measure of the quantity of interest because pose error is directly proportional to it. Hence, output error is a linear - not a quadratic - function of disparity error.

For these reasons, mosaic-based positioning is largely immune to three of the dominant failure modes of stereo: unsufficient texture, forshortening distortion in areas of high disparity gradient, and poor range measurement resolution. Indeed, preliminary experiments indicate quite satisfactory robustness in suitable environments.

1.5 PRINCIPLE OF OPERATION

Normally one real-time image is taken and compared with a previously stored mosaic of images. A small rectangular region of pixels in the image called a **template**, always has a predicted corresponding position in the mosaic. The region surrounding that predicted position in the mosaic is the **search region**.

Some of the notation used in the following discussion is illustrated below. Let the template be of size $w \times w$ pixels called the correlation window size. Also, although the window is rectangular, we will loosely refer to $w/2$ as the correlation window radius. Similiarly, the center of the correlation window will be moved slight amounts in all directions from its nominal position. The rectangular region inside which all such searching is confined is of size $W \times W$ where W is called the search window size. Also, $W/2$ is loosely called the search window radius. The two window sizes are independent. Either can be the larger. w is determined by the amount of texture and W is determined by the current position estimate uncertainty.

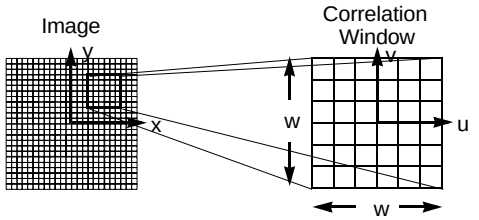

Figure 2: IMAGE AND CORRELATION WINDOW COORDINATES. A local coordinate system (x,y) is attached to the image plane at the central pixel of the reference image and another (u,v) is attached to the correlation window.

The **disparity** is the difference between predicted and actual position in the mosaic, i.e. it is the pose error in pixels. Computation of disparity can be accomplished through maximizing some measure of similarity - such as normalized correlation - by searching a sequence of candidate disparities.

The disparity can be determined by local brute force search for the peak in a crosscorrelation function between the template and the search region in the mosaic. If $T(x, y)$ is the normalized template intensity function and $M(x, y)$ is the normalized mosaic intensity function, then the normalized correlation, computed as a function of disparity (d_x, d_y), comes from the double integral:

$$C(d_x, d_y) = \frac{1}{w^2} \int_{-\frac{w}{2}}^{\frac{w}{2}} \int_{-\frac{w}{2}}^{\frac{w}{2}} T(u, v) M(u + d_x, v + d_y) du dv$$

The figure below presents a template, a search region, and the correlation between them as a function of two-dimensional disparity.

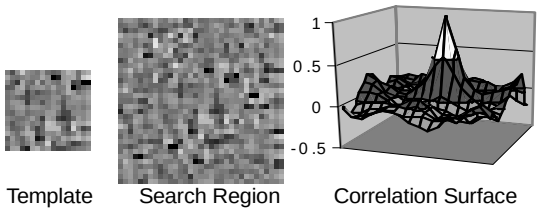

Figure 3: PRINCIPLE OF OPERATION. The translational pose error can be found as the position of the peak in a surface: the correlation as a function of disparity.

The template in the above figure is a 17 X 17 pixel area taken from a concrete floor image. A careful examination shows it is present in the center of the search region. The unprocessed images exhibit very little texture, so they have been enhanced for purposes of presentation. The position of the correlation surface peak is nonetheless clear, and there are no other contenders within the search region.

In stereo, disparity is limited to one dimension by virtue of the epipolar constraint - which itself is due to the fixed spatial relationship between the two views. Here, no such fixed relationship exists. Disparity is unavoidably at least a 3 dimensional quantity in mosaic-based positioning. A guess pose may cause a template to be both positioned and oriented incorrectly in the mosaic. For the moment, however, we will consider translational aspects of disparity only.

1.6 PRIOR WORK

The practice of image mosaicing, of producing larger images from the union of smaller ones registered in the region of overlap, is an old one. The technique of automatic construction of mosaics by computer is also relatively old [1].

Automated mosaicing [2] is often useful in its own right. Applications include station keeping [3], video coding [4], image stabilization [5], and visualization [6]. Only recently have near real-time [7] and globally consistent [8] mosaicing solutions emerged. However, there seems to be little in print on the problem of navigating from mosaics.

Certainly, navigating from imagery is a basic technique in robotics [9] [10] but such techniques often deal with the much harder problem of a three dimensional scene - often of unknown geometry. Yet, the notion of determining camera pose in a scene is the sensing half of the visual servoing problem [11].

Perhaps such a lack of emphasis on the problem has stemmed from practical inabilities to store enough imagery and access it fast enough to be useful. The purpose of the paper is to show that these historical restrictions no longer apply.

1.7 OPPORTUNITY

Commercial alternatives for the positioning of vehicles in yards and manufacturing facilities include:

- **wire guidance:** where crosstrack error is measured from inductive pickoffs sensing wires that are literally embedded in the floor of a building.
- **laser guidance:** where a spinning laser beam senses the bearings and sometimes the ranges of retroreflective fiducials mounted on the walls of buildings - and triangulates pose.
- **inertial guidance:** dead reckoning from gyroscopes and accelerometers augmented by occasional position fixes.
- **radio guidance**: forms of indoor GPS are reported to be under development in industrial laboratories.

Mosaic-based positioning promises to compete favorably with these alternatives in suitable environments because:

- Resolution is limited only by optics. It can, in fact, be used on microscopes, cameras, and telescopes.
- Capital cost is limited to that of a camera, lighting, off-line storage, and a capable processor - which may be required for other reasons.
- Installation cost is limited to the labour and time required to map the environment by driving over all necessary guidepaths or areas only once.

Typical conditions in plants, office buildings, and yards, combined with contemporary relatively inexpensive cameras, computers, and off-line storage, present a new opportunity to produce inexpensive and highly repeatable position estimation based on template matching of textured scenes.

Use of such a technique to determine position, requires that:

- the scene have sufficient visual texture which is locally unique enough (i.e. non repetitive) to support template matching, and
- sufficient image storage and computing power must be available.

Following sections discuss these matters in more detail.

2 TEXTURE REQUIREMENTS

Template matching can be surprisingly robust in cases of lack of texture and image noise. This section discusses these matters with regard to typical floor imagery.

2.1 SUBJECTIVE TEXTURE IN MANY MAN-MADE SCENES

The following figure shows images of floor scenes in manufacturing facilities which exhibit typical textured patterns.

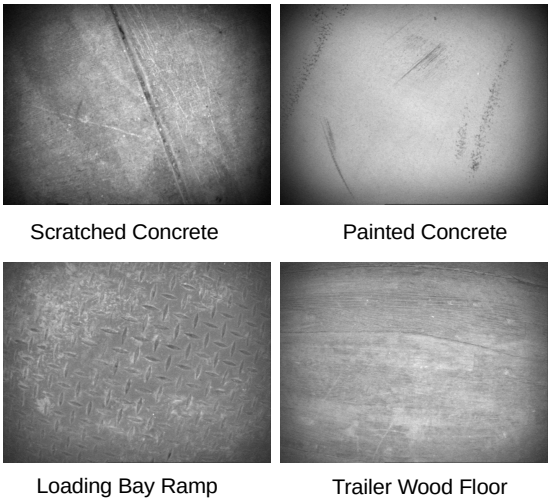

Figure 4: MANUFACTURING PLANT FLOOR IMAGERY. Many man made surfaces exhibit locally unique texture.

In the painted concrete image, just a little wear and tear, when added to the physical texture of the surface, can create enough texture to navigate.

2.2 TEXTURE SIGNAL STRENGTH

A very direct test of whether or not an image or part of an image contains sufficient texture for template matching is to compute the autocorrelation of candidate templates as a function of disparity. A template is suitable for matching when:

- it possesses sufficient texture,
- it has a high cross-correlation surface peak, and
- it has no competitive peaks within a neighbourhood.

All conditions together imply good noise rejection.

2.2.1 SUFFICIENT TEXTURE

Texture in the template implies that a local maximum correlation will exist. Many metrics of texture have been proposed but intuitively, any measure of the “edginess” or gradient magnitude will suffice for a qualitative assessment.

2.2.2 GOOD CROSS-CORRELATION

Good cross-correlation implies a high degree of similiarity between the template and the candidate mosaic region. It is important to recognize that this second condition does not imply the first because even relatively textureless regions may correlate perfectly. Of course, the autocorrelation surface peak always has a value of unity, so it is important to realize that the real issue is the normalized cross-correlation between the image template and the mosaic.

2.2.3 UNIQUENESS

Uniqueness implies a low probability of false matches. A good measure of this third criterion is the difference in height between the highest and the second highest autocorrelation peak in the search neighbourhood. When the difference is large, it should be possible to correctly match the template even in the presence of substantial random noise because significant noise would be necessary to lower the higher peak and/or raise the lower peak.

2.2.4 EXAMPLE

The following figure presents a quantitave assessment of the suitability of a concrete floor image. The original image shows the nicks, cracks, and scratches that might be expected due to normal wear and tear. Point imperfections, which are particularly valuable for localization, are apparent as well.

The texture enhanced image is generated by replacing each pixel by its suitably scaled, normalized deviation from the mean of its local neighbourhood thus:

$$I' = (I - \mu_I)/\sigma_I$$

For this operation, a 9 X 9 window centered at each pixel was used.

The texture score image is generated from "edginess" per unit area. This is computed from a suitably scaled sum of all intensity gradient magnitudes in a local region thus;

$$I'' = \frac{1}{w^2}\sum\left[\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2\right]$$

For this operation, a 17 X 17 window centered at each pixel was used. The reader may detect refinements implemented to identify and discourage one-dimensional texture. These are beyond our scope here. One can verify that whiter pixels correspond to features of higher texture due to imperfections in the floor.

The autocorrelation score image is generated from the difference of the highest and the second highest autocorrelation peak in a search neighbourhood surrounding the pixel. Let C_1 be the correlation score of the highest peak (which is unity for autocorrelation). Let C_2 be the correlation score of the second highest peak (or it could be simply the highest score outside the radius of the highest peak). Then the image is a scaled version of:

$$I''' = (C_1 - C_2)$$

For this operation, a 17 X 17 search window centered at each pixel was used. For the test image, there are no pixels for which $(C_1 - C_2) < 0.2$ and fewer than 1% have $(C_1 - C_2) < 0.4$.

The conclusion that can be drawn is that this image is likely to be an excellent image for the purpose of template matching - especially when augmented by a search for high texture regions because the differences in peak correlation scores is virtually everywhere large.

A preprocessing search for high texture pixels is a good idea for two reasons:

- it will shortly be shown that noise immunity is correlated with high texture
- in the following image for a 17 X 17 search region, computation of a texture score for a 160 X 120 image requires 70 milliseconds, whereas the autocorrelation score requires a full 180 seconds.

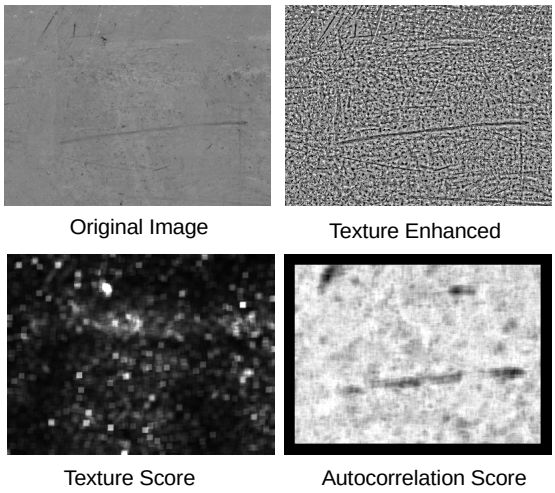

Figure 5: ANALYSIS OF TYPICAL FLOOR IMAGE. While it may subjectively seem that this concrete floor image lacks sufficient texture, it is actually acceptable at every single pixel.

2.2.5 ROBUSTNESS TO RANDOM NOISE

It is possible to test the assertion that the above image is acceptable by deliberately introducing noise and cross-correlating the corrupted image with the original to see how the peak location changes.

The figure below illustrates the result of cross-correlating a pristine mosaic against a noisy image. A 17 X 17 search window is used, and a peak that is mislocated by 8 pixels is deemed a false positive. The threshold of 8 is chosen because the radius of the correlation peak is related to the size of the correlation window.

The curve labelled "texture threshold" depicts the highest texture score of any false positive at the indicated injected noise level (as determined from its deviation from ground truth). The curve labelled "percent above threshold" is the number of image pixels whose texture score remains above this threshold.

Hence, no region whose texture score exceeds the threshold generates a false positive. Such regions are "safe" for matching at the indicated noise level. Clearly:

- as the threshold rises, the percentage of regions above threshold must lower.
- the fact that random noise of magnitude 60 counts (out of the available 255 intensity levels) is required to render any texture threshold useless predicts a high level of noise immunity.

Perhaps one reason for this level of noise immunity in the face of little absolute texture is the fact that perspective foreshortening distortion is not an issue for a surface normal to the camera optical axis. Hence, the correlation window size w can be increased to levels that impart high noise immunity without suffering from distortion problems.

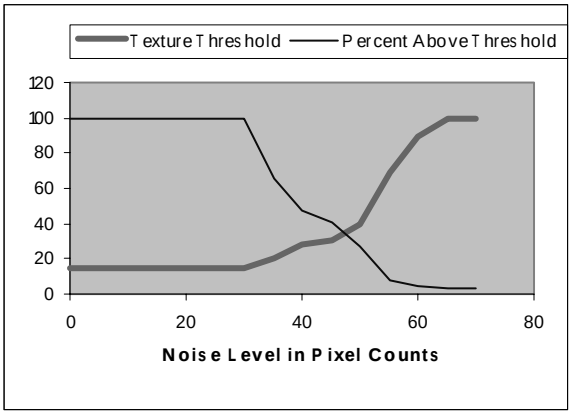

Figure 6: NOISE IMMUNITY. For a maximum grey level value of 255, it takes random noise wih a standard deviation of 60 counts before the highest texture regions in the image generate false positives.

This analysis concentrates on random noise. Systematic image corruption is another matter entirely. Such errors are analyzed in a later section.

3 STORAGE REQUIREMENTS

When operating in absolute mode, it is necessary to have the entire map that could be traversed available in off-line memory. Estimates of storage requirements come down to a determination of surface-projected pixel size and of the surface area that must be covered. For a pixel size δ of 2.5 mm, the following graph shows that one-tenth of a square kilometer of area (roughly 1000 ft by 1000 ft) can be stored in 10 GByte of memory. This amount of storage corresponds to a single contemporary DVD ROM. Memory requirements drop quadratically if the pixel size can be increased.

In a realistic manufacturing setting, vehicles are limited to specific guidepaths perhaps a meter wide, so the graph also indicates that 100 Km of guidepath (10^5 square meters) can be stored in only 10 GByte of memory. Due in part to demand for multimedia content on the internet, such memory capacity is already inexpensive. Many applications could already store usefully large areas in solid state flash disks - similiar to those used in consumer digital cameras.

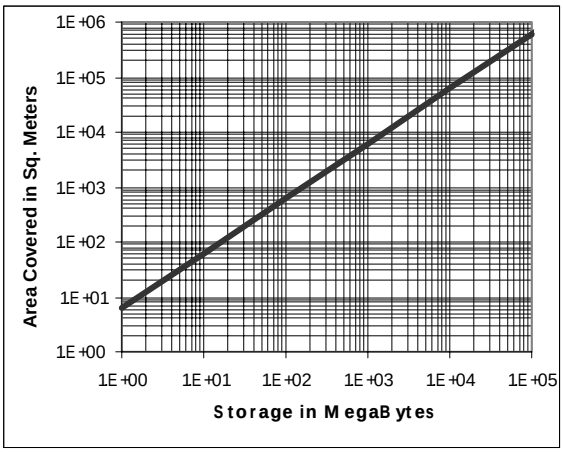

Figure 7: STORAGE REQUIREMENTS. Up to 10 Km of guidepath can be stored at 1 mm resolution in 1 GByte of memory.

4 PROCESSING REQUIREMENTS

4.1 PROCESSING SPEED

It remains to be shown that a vehicle moving at a given speed can have its position continuously updated by a mosaic-based positioning system.

4.1.1 PROCESSING REQUIRED TO LOCALIZE N TEMPLATES

Let us suppose that normalized cross-correlation is the mechanism used and that the time required to normalize the image can be neglected. For a template of size $w \times w$, it clearly takes at least w^2 operations to compute its cross-correlation coefficient with another template of equal size. If various overhead operations such as array indexing are included, let there be K operations required per pixel in the template to arrive at Kw^2 operations per template correlation.

If an entire region of size $W \times W$ is to be searched, the careful reader can verify that it is not possible to reuse any computations, so the required processing to search for a match to a single template is KW^2w^2. Computation in frequency space may be a more efficient alternative but it has not, as yet, been explored.

In order to determine the location of a vehicle robustly, it will likely be necessary to process more than one template. Let's assume that N templates are necessary. Hence, computation of a position fix will require:

$$flops = NKW^2w^2$$

If a position fix is required every T_{cyc} seconds, then the required computational power devoted to template matching is:

$$f_{cpu} = \frac{flops}{T_{cyc}} = \frac{NKW^2w^2}{T_{cyc}}$$

This can be written in terms of the number of correlation operations ("correlation ops") required per unit time thus:

$$f_{corr} = \frac{f_{cpu}}{K} = \frac{NW^2w^2}{T_{cyc}}$$

It can also be written in terms of the number of template correlations required per unit time thus:

$$f_{templ} = \frac{f_{cpu}}{Kw^2} = \frac{NW^2}{T_{cyc}}$$

4.1.2 Processing Required for Tracking

Regardless of whether or not the mosaic itself is accurate, there is a clear requirement for the system to be able to track its motion over the map - in order not to get lost. To do so requires that the area searched per unit time equal or exceed the tracking error that has accumulated per unit time. Here, the tracking error is the difference between the predicted position of a template and its actual position in the map.

Let us assume that a dead reckoning system is used to compute an estimate of where the camera is, and that the estimate becomes the center of the template search region. A conservative estimate of error buildup as a function of time is as a percentage α, called the error gradient, of travelled distance s.

$$\varepsilon_{DR} = \alpha s$$

Such an error model is admittedly a simplification but it renders the following analysis tractable. The general conclusions do not depend on the error model chosen as long as it is monotonically increasing with time.

If this error accumulates after a time period T_{cyc}, then it an be related to speed V thus:

$$\varepsilon_{DR} = \alpha V T_{cyc}$$

Also, it is likely that the map is not perfect and that, in the worst case, errors due to slight mislocation ε_{MAP} of regions are in exactly the same direction of the dead reckoning error so the errors add.

$$\varepsilon_{TOT} = \varepsilon_{DR} + \varepsilon_{MAP}$$

If we equate this total tracking error error ε_{TOT} to the radius $W/2$ of the search region, we obtain a relationship between the template correlation rate required at a given vehicle speed when the CPU is dedicated to template matching (because we assumed the variable part of the error accumulated over exactly one cycle time).

$$f_{corr} = \frac{NW^2w^2}{T_{cyc}} = \frac{4N(\alpha V T_{cyc} + \varepsilon_{MAP})^2 w^2}{T_{cyc}}$$

The factor of 4 was introduced because the search radius is half the search window size. This relationship is plotted below as a function of speed for various values of the intrinsic map error.

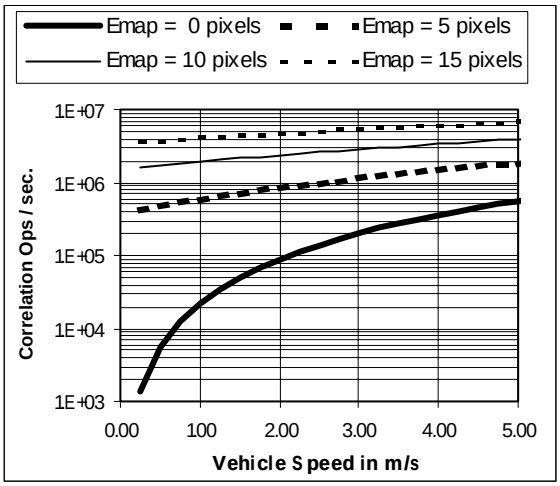

Figure 8: PROCESSING REQUIREMENTS. Increase with vehicle speed. From bottom to top, the 4 curves are for ε_{MAP} of 0, 5, 10, and 15, pixels of size 2.5 mm.

The rest of the values in the formula are given in this table:

Table 1: Processing Speed Model Parameters

Symbol	Value	Symbol	Value
N	4	α	0.01
T_{cyc}	0.1 secs	w	17 pixels
δ	2.5 mm		

Clearly, map error can dominate the processing requirements at low speeds. Experiments have indicated that a 300 MHz Pentium II can perform 10^7 correlation operations per second. On this basis, the mosaicing system could support speeds up to 5 m/s at 10 Hz update rate.

4.1.3 UPDATE RATE FOR MINIMUM PROCESSING REQUIREMENTS

The dependence of processing requirements on speed is nonlinear. As cycle time is increased, the rate nominally goes down because the number of searches conducted per second is smaller. However, due to build up of error over time, the size of those searches increases. Hence, the relationship is a rational polynomial in T_{cyc}. The former expression can be differentiated to yield an expression for the minimum:

$$f_{corr} = \frac{4Nw^2[(\alpha V T_{cyc})^2 + 2\alpha V \varepsilon_{MAP} T_{cyc} + (\varepsilon_{MAP})^2]}{T_{cyc}}$$

Therefore:

$$\frac{df_{corr}}{dT_{cyc}} = 0 \Rightarrow T_{cyc} = \varepsilon_{MAP}/(\alpha V)$$

Hence the minimum occurs intuitively when the dead reckoning error equals the map error - when the gains in one are exactly cancelled by the losses in the other. The following figure shows the variation in processing requirements with cycle time.

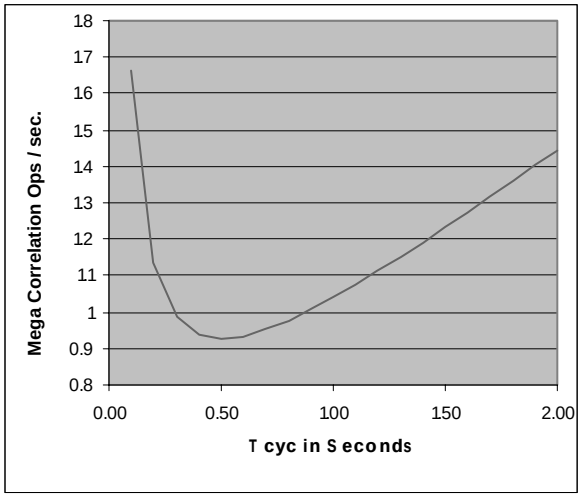

Figure 9: MINIMUM PROCESSING REQUIREMENTS. This curve is for ε_{MAP} of 5 pixels of size 2.5 mm. Processing requirements at 2.5 m/s for 1% of distance dead reckoning error are minimum at T_{cyc} = 0.5 secs.

Note that minimum error accumulation happens for minimum cycle time but there are limits on how low the cycle time can be reduced - even for a perfect map, because of the speed of the digitization hardware. 30 Hz is typically the fastest possible rate available.

4.1.4 PROCESSING REQUIRED FOR ON-LINE MAPPING

When building maps, speed is limited by an entirely different mechanism - images must overlap at the operating speed and cycle time if a single pass is to produce complete coverage. If the image height is H, and β is the fraction of overlap required, the maximum speed possible is:

$$V_{max} = \frac{(1-\beta)H}{T_{cyc}}$$

For an overlap of 30%, a height of 0.3 meters, and 10 Hz update, this gives 1.8 m/s as the maximum vehicle speed. When operating at any speed, the formula of the last section applies to determine the necessary processing power. Hence, provided the computer can support the vehicle speed in tracking mode, it can be supported in mapping mode.

Here, if building maps on-line, there is a clear motivation to decrease cycle time as much as possible. Otherwise imagery can be stored at frame rate and processed off-line.

4.1.5 SAFE DISTANCE AFTER LOSS OF VISUAL LOCK

A mosaic-based position estimation system implements a visual lock on the mosaic. Hence, the longer the delay or distance between position fixes, the larger the error grows. At some point the system will be unable to recover from temporary loss of visual lock.

Much of the analysis so far has relied on the assumption that the template matching system is required only to damp errors in dead reckoning or, more generally, errors in any primary position estimation system. The dependence of processing requirements on the error gradient α is quadratic, and it has a typical value of 0.01. Hence, the use of a primary system can be expected to reduce processing requirements by a factor of 10^4, or equivalently, increase permissable speed by a factor of 100.

The analysis of the impact of random image intensity error might be expected to model system performance in the face of dust and small dirt particles that occlude individual separated floor pixels in whole or in part.

Another important form of image error is the occlusion, replacement, or obliteration of all texture in a relatively large region. This error might occur when a liquid is spilled, when floor coverings are replaced, or when a floor is painted. It might also occur when the system is obliged to move over an area that has been unmapped (in order to avoid an obstacle on the guidepath for example).

Regardless of the source of the error, its worst case net effect is to render a position fix impossible over that floor region until it is restored close to its original state, or remapped. A more immediate issue, however, is whether or not the system can recover visual lock after leaving the area.

One practical implementation of mosaic-based position estimation dedicates a fixed amount of processing power, searches a fixed large search radius W each cycle, and therefore cycles at a more or less fixed frequency. Although this approach may often waste processing time searching unnecessarily, the extra search radius comes into play immediately after driving over an unrecognizeable region.

Recall our elementary expression for processing requirements:

$$f_{corr} = \frac{NW^2w^2}{T_{cyc}}$$

Therefore, the time required to search a region of size W is:

$$T_{cyc} = \frac{NW^2w^2}{f_{corr}}$$

Let the total error in the primary position estimate again be determined from the fixed map error and the accumulation of dead reckoning error over the time spent in the unrecognized region:

$$\varepsilon_{TOT} = \varepsilon_{DR} + \varepsilon_{MAP} = \alpha s + \varepsilon_{MAP}$$

If we set this error equal to the search region radius, we compute the situation when the system is on the threshold

of not being able to find itself. Solving for the distance travelled leads to the largest distance that can be travelled without a position fix before the system gets lost:

$$s = (\varepsilon_{TOT} - \varepsilon_{MAP})/\alpha = (W/2 - \varepsilon_{MAP})/\alpha$$

The following figure illustrates this relationship for a pixel size of 2.5 mm.

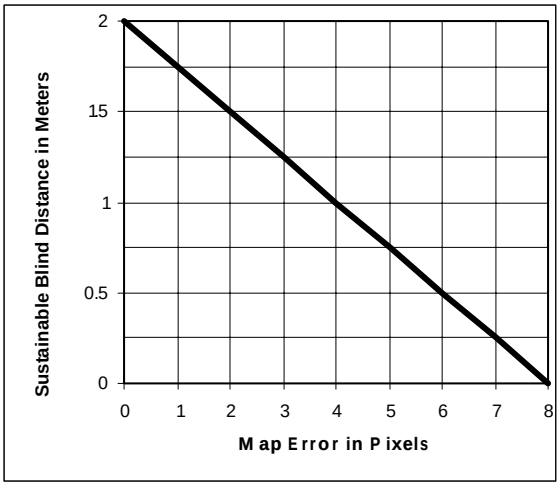

Figure 10: SUSTAINABLE EXCURSION OVER UNKNOWN REGION. For a search radius of 8 pixels, a pixel size of 2.5 mm, and a map error of 2 pixels, 1.5 meters of excursion over unknown, unmapped, or changed surface can be accomodated.

5 CONCLUSIONS

This paper has presented a case for the feasiblity of a very straightforward new approach to vehicle position estimation based on tracking a camera pose during motion over a prestored mosaic. The analysis has lead to the following conclusions:

- Image mosaic-based position estimation is feasible on large scale mosaics today. It can be expected to produce submillimeter repeatability of vehicle position estimates at speeds up to 5 m/s and update rates of 10 Hz. These numbers represent one set of design decisions for automated guided vehicles whereas designs for microscopic and planetary scales are equally feasible. Designs for highway speeds of automobiles are also feasible.
- A primary reason for its feasibility is the ability to engineer a system for surface normal erection of a camera moving parallel to the surface. Mounting a device underneath a vehicle pointed directly downward, for example, achieves such geometry. Such geometry makes straightforward template matching feasible - eliminating perspective foreshortening distortions. Eliminating distortion preserves highest noise rejection in template matching. Extensions to manage some degree of surface roughness based on determining affine transform parameters rather than camera pose are also clear.
- Another reason for its feasibility is a 100-fold increase in vehicle speed or a 10,000-fold decrease in required computing when a primary position estimate can be used to seed the search for a template match.
- Relatively high update rates are advisable because computational requirements tend to be lower overall.
- Correlational template matching on typical floor imagery is quite robust to noise. Since texture can be quantified efficiently to prequalify templates and since only a few templates per position fix are required, it can be a rare event when a particular image cannot be used to generate a fix.

A mosaic based positioning system for a 40,000 square foot test facility has been in operation by the author for over three years. This paper has concentrated on the original feasibility analysis. Sequel papers will discuss mosaic construction, tracking, and the design and performance of the system.

6 References

[1] P. J. Burt and E. H. Adelson, A Multiresolution Spline with Application to Image Mosaics, *ACM Transaction on Graphics*, Vol 2, pp. 217-236, 1983.

[2] Q. Zheng and R. Chellappa. A computational vision approach to image registration. *IEEE Transactions on Image Processing*, 2(3):311-325, July 1993.

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

[4] 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.

[5] 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).

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

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

[8] H. S. Sawhney, S. Hsu and R. Kumar, Robust video mosaicing 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

[9] S. Atiya and G. D. Hager, Real-Time Vision Based Robot Localization, *IEEE Transactions on Robotics and Automation*, vol 9, no. 6, pp. 785-800, 1993.

[10] B. K. P. Horn and E. J. Weldon, Direct Methods for Recovering Motion, *Int. J. Computer Vision*, vol 1, pp 51-76, 1988.

[11] G. Hager, S. Hutchinson, and P. I. Corke. A Tutorial on Visual Servo Control. *IEEE Transactions on Robotics and Automation*, 12(5):651-670, Dec 1996.