VLSI architecture for Video-Assisted Global Positioning

Anant Utgikar, Guna Seetharaman and H. Le
Center for Advanced Computer Studies, University of Louisiana at Lafayette, LA 70504-4330
{anant, guna, hvle}@louisiana.edu

Abstract

We design and implement an efficient architecture for geometric computation of the global position of an airborne video camera from images of known landmarks by method explained in [12]. A solution based on this analysis, a robust Hough transform-like method facilitated by a class of CORDIC-structured computations is implemented within the framework of terrain navigation. It empowers aerial surveillance systems to navigate effectively when the global position and inertial navigation sensors are out of order. This is particularly useful when the GPS functionality is disrupted by jamming and other techniques. Our architecture exploits parallelism among independent operations and uses pipelining of critical components for superior performance. Double precision division being computationally expensive is performed minimally. Correlation between data is tapped to reduce complexity of flash ADCs, at the cost of few clock cycles once to initialize Hough Voting.

1. Introduction

The perspective-n-point problem is to determine the position of the camera from images of known landmarks, given the intrinsic parameters of the camera. If there are three non-coplanar landmarks, a set of three quadratic equations with three unknowns can be formed [15], that can produce as many as eight solutions. Fischler and Bolles [6] derive a biquadratic polynomial equation with one unknown, that has a maximum of four solutions. For the case of four landmarks, Rives et al. [15] find a unique solution if the landmarks are coplanar by finding the solutions using three landmarks, then verifying them with the fourth one. If the four landmarks are not coplanar, they derive a set of six quadratic equations with four unknowns. Horaud et al. [9] form a biquadratic polynomial equation in one unknown for non-coplanar landmarks. The equation can have as many as four solutions, but some of them can be eliminated. A more general approach to finding solutions for the perspective-n-point problem is to use the least-square techniques [14].

In terrain navigation, the global position of an airborne camera system is usually available through GPS. However, there are circumstances where the position given by a GPS is not accurate due to built-in errors, dysfunctional GPS, or noise caused by intentional R-F jamming. Information gathered from images of known landmarks then can be used to compensate the GPS errors in such cases [4] [17]. Our goal is to design and implement an efficient VLSI architecture for visual navigation as discussed in [12]. We first present the analysis of the problem through constructive geometry and then describe the architecture proposed for implementing it. The main result of the analysis is: for any two landmarks in an image, we can construct a unique toroid in the world space, that constitutes all possible positions of the camera. The toroid can be easily generated using a COordinate Rotation Digital Computer (CORDIC) [2], [1] structured hardware, to register a vote at the appropriate cells of a three-dimensional array of accumulators. For example, in the case of four landmarks, six toroids could be generated, corresponding to the six quadratic equations derived by Rives et al. [15]. An acceptable solution would be identifiable at the cell with a vote of at least three and at most six. The geometric nature of the proposed approach helps understand the span of the subspace (3-dimensional world) to which the expected solution would be constrained. Thus, a finite number of accumulators are sufficient. Moreover, we use a multi-resolution recursive approach outlined in [11] for the implementation. That is our Hough transform-like method for locating the accurate position of the camera when there are more than two landmarks. The corresponding result in 2D is a circle in the plane of landmarks and the observer.

2. The Analysis

2.1. Geometric Analysis

We assume that the operational nature of the imaging system follows geometric optics [5] and perspective projection. Our results are based on the classic theorem of angles on circles [8] which states: Given a circle, and a chord AB the angle subtended by the arc AB at the center is twice that of the angle subtended by it at any point on the circumference along the complementary sector AB. Its corollary states that: The locus of the vertex C of all triangles with a fixed apex angle \( \angle ACB \) and fixed end points \( A \) and \( B \) is the union of two arcs, each a sector of a circle, on either side of
line $AB$. [8].

**Theorem 1.** Let an image of two landmarks and therefore the angle subtended by them at the pupil of the camera, be given. Then, the unknown geolocation is constrained to a unique circle on the plane of imaging. and, it is also constrained onto the surface of a unique toroid in space when the principal plane can not be uniquely fixed. Theorem 1.

**Proof.** Since the image captures aerial view of landmarks, the other circle with larger sector below landmark altitude is ignored. We prove the above claim by constructing the circle and the toroid as described below.

Let the points $A$ and $B$ be the two landmarks observed by the camera $C$ from an unknown geolocation. Then, the plane $ACB$ is the principal plane as shown in Figure 1. Let $\theta$ be the angle $\angle ACB$, which can be computed from the video image. Details of the imaging model will be shown in Section 2.2. Also, let $n_p$ be the normal of the principal plane $ACB$. And, let $L$ be the length of the line segment $AB$. From the geolocations of $A$ and $B$ compute the midpoint $M$ on $AB$. Draw the bisector of $AB$, the line perpendicular to $AB$ passing through $M$. Locate the center $O$ on the bisector at a distance $h$ from $M$; and, compute the radius $R$ as:

$$h = \frac{L}{2} \frac{1}{\tan \theta}; \quad \text{and} \quad R = \sqrt{\left(\frac{L^2}{4}\right) + h^2}$$

(1)

Also, $X_M = 0.5 \ (X_A + X_B)$. The direction cosines $l$ of the line segment $AB$, and $p$ that of its bisector are:

$$1 = \frac{X_B - X_A}{|X_B - X_A|} \quad \text{and} \quad p = n_p \times 1$$

(2)

Then, trace the locus of all feasible positions of the camera $C$, by tracing $X_C(\alpha)$, $\left(\theta - \frac{\pi}{2}\right) \leq \alpha \leq \left(\frac{\pi}{2} - \theta\right)$:

$$X(\alpha) = X_M + h p + (R \cos \alpha) l + (R \sin \alpha) p$$

(3)

Then, based on the Theorem of Thales[8] we prove that the circle is unique.

When the principal plane is not known, any plane can be used to construct the circular arc first, which is then rotated about the axis $AB$ to create the toroidal surface. For example, consider the unit vector $e_3 = \frac{1}{\sqrt{3}}(1, 1, 1)$. If $l \neq e_3$ then let $n_B = e_3$; else, let $n_B = \frac{1}{\sqrt{3}}(-1, 1, 1)$. Then compute, $n_p = n_B \times l$, $p = n_p \times 1$, and $q = l \times p$. Then, for any $(\alpha, \beta), -\pi < \beta \leq \pi$ and $\left(\theta - \frac{\pi}{2}\right) \leq \alpha \leq \left(\frac{3\pi}{2} - \theta\right)$, compute $X(\alpha, \beta)$ on the surface of the toroid as:

$$X(\alpha, \beta) = (V_\alpha \cos \beta)p + (V_\alpha \sin \beta)q + U_\alpha l + X_M$$

(4)

where

$$U_\alpha = R \cos \alpha \quad \text{and} \quad V_\alpha = h + R \sin \alpha$$

(5)

This completes our proof.

These computations follow a CORDIC structure; hence, they have been implemented in hardware to rapidly generate the circle and the toroid. For example, the value of $X(\alpha)$ can be computed as follows, using a recursive algorithm for Equation (3).

$$\begin{align*}
\alpha_0 & := 0; u_0 := 0; v_0 := 0; \\
\left[\begin{array}{c}
u_k \\
v_k
\end{array}\right] & := \left[\begin{array}{c}
1 & \delta \\
-\delta & 1
\end{array}\right] \left[\begin{array}{c}
u_{k-1} \\
v_{k-1}
\end{array}\right] \\
\alpha_k & = \alpha_{k-1} + \delta \\
X(\alpha_k) & = X_O + hp + u_k l + v_k p
\end{align*}$$

(6)

(7)

(8)

(9)

where $\delta \approx \sin \delta$ is chosen as $\delta = 2^{-8}$, to perform the computation rapidly in hardware as a bit shift rather than a floating point multiplication. [1].

### 2.2. Imaging Model

The notation used in this paper is as follows: An object $A$ located at $X_A$ as in Fig.2, is imaged by a camera $C$ kept from a geoposition $X_C$. Perspective imaging is assumed. The image point $x_A$ is on the $z = -f$ plane of the camera coordinate system $(x, y, z)$; however, the derivations will consider $z = f$ plane [16]. The angle $\theta_Z$ and the vector $X_C$ denote the instantaneous orientation and geolocation of the camera. Also, $\theta_A$ describes the line of sight of $A$.
from the current location \( X_C \) of the camera \( C \). The angle \( \phi_a \) and the vector \( x_a \) are used to represent the direction and position of \( a \), the image of \( A \), measured in the 3-D frame of reference of the camera. The uppercase subscripts are used to indicate quantities that are prone to sensor inaccuracies; and, lowercase subscripts will be used to denote quantities which are measured using the observed image.

**Lemma 1.** Let \( A \) and \( B \) be two objects in the field of view of the camera \( C \) which is located at \( X_C \). Then, the angle \( \angle ACB \) subtended by the 3-D line segment \( AB \) at the pupil of the camera \( C \) can be measured accurately without any explicit knowledge of the exact geoposition and orientation of the camera.

**Proof.** We observe from Figure 2, that

\[
\tan \theta_A = \frac{X_A - X_C}{Z_A - Z_C} \quad (10)
\]

We assume that the video imaging system has already been calibrated, and its intrinsic parameters such as the location of the optical center within the given image, pixel dimensions, and focal length have been determined and made available. Also, we assume that the exact location of the point \( a \), the image of \( A \) has been determined by standard image processing techniques at the best possible accuracy. Then, \( \phi_a \) can be computed from the video-image based measurement of \( x_a \). Thus,

\[
\phi_a = \arctan \left( \frac{x_a}{f} \right) \quad (11)
\]

which satisfies the expression,

\[
\phi_a = \theta_A - \theta_Z \quad (12)
\]

Also,

\[
\theta_A - \theta_B = \phi_a - \phi_b \quad (13)
\]

where \( m_{ab} \) is computed from \( x_a \) and \( x_b \) and treated as a single measurement derived through the image. Thus, even though \( X_C \) and \( \theta_Z \) are not explicitly known, the difference \( \theta_{AB} = (\theta_A - \theta_B) \) could be measured indirectly from the image. Moreover, the differential nature of the quantity \( \theta_{AB} \) makes it insensitive to the orientation jitters of the camera.

### 2.3. The Hough Transform Approach

Hough transform [10] is a popular method for 1) Finding a way to parameterize the patterns of interest. 2) Mapping the points in the image space into the parameter space which is clustered so that a voting technique can be used to determine the most likely values of the parameters of the patterns appearing among the given set of points.

We divide the \( XYZ \) space into a three-dimensional grid of accumulative cells, called Hough bins. To find the camera position in the grid, discretize all the toroids over the grid, and find the cell that has the highest number of votes. The discrete values of each toroidal surface can easily be generated using equations (4) and (5), with \( \alpha \) and \( \beta \) ranging from 0 to \( 2\pi \).

If the coordinate system \( XYZ \) is too large, we may use output of faulty or inaccurate GPS as a first order approximation. Thus, the search space can be limited into a relatively small volume for which the clustering technique is deemed effective. We also employ a multi-resolution approach to speed up the computation: using an iterative “coarse-to-fine” accumulation and search strategy [11].

Errors of discretization need special consideration in our case because the toroids are generated by incrementing the angles \( \alpha \) and \( \beta \), while the space is divided along the \( X \), \( Y \), and \( Z \) axes. A discussion of discretization, selection of incrementing procedure, and error analysis, is provided by Ballard [3].

### 2.4. Special Configurations

There are special cases in which we can simplify the results of Section 2 so that faster computations can be achieved. Given an image of three colinear landmarks, \( L_1 \), \( L_2 \), \( L_3 \) and the orientation of the principal plane of imaging, the geolocation of the camera is uniquely determined. Since, two uniquely defined circles say \( C_{12} \) and \( C_{23} \) can be constructed on the principal plane by applying Theorem 1, to the landmark-pairs \( L_1 \), \( L_2 \) and \( L_2 \), \( L_3 \) respectively.

**Theorem 2.** Given an image of three colinear landmarks, the geolocation of the camera is constrained to a circle on a plane perpendicular to the line passing through the landmarks; and, both the circle and the plane are unique.
Proof. Let $L_1, L_2$ and $L_3$ be the landmarks; and, let $l$ be the direction cosine of the line $L_{123}$ passing through these landmarks. Let $n_P$ be the normal to a plane passing through the colinear landmarks. One such normal can be uniquely identified as in Theorem 1; also $p = n_P \times l$. Let $G$ be the geolocation of the camera uniquely resolved. Then $d_2 = L_{12}$ and $d_3 = L_{13}$ are the pairwise distances of the landmarks $L_2$ and $L_3$ respectively from $L_1$. Then, $X_G = X_{L_1} + (sx_g)l + (sy_g)p$, and, $s = \frac{d}{l}$ is the scale factor. Now rotate the plane around the axis $L_{123}$ by $2\pi$ radians. It will produce a circle of radius $sy_g$ contained in a plane perpendicular to $L_{123}$; and, the plane is at a distance $sx_g$ from the landmark $L_1$. Since $G$ is unique and $L_{123}$ is fixed, both the circle and the plane of the circle are unique.

3. Implementation

3.1. System Architecture

We illustrate the implementation of the positioning system in the two-dimensional case. The coordinates of the landmarks and angles subtended by them at the camera, are fed as inputs to the architecture. Usually it is known which way the camera taking the picture is facing (front or back). Hence, the loci of both points at which given angles are subtended by landmarks (circles for 2D, by Thm.1) can be drawn and their point of intersection determined in a 2D square grid. Grid size and location is so chosen as to circumscribe one of the circles completely and the other partially, as shown in Fig.3. This is necessary and sufficient since any intersection of the two loci must fall within the grid. Better resolution of the intersection point would be obtained by letting circle with smaller radius fall entirely in the grid since it determines the grid size. Digitized circle gives rise to symmetric structure for complete circle and a partial curve for other, depending on degree of overlap. The grid is divided into many identical square cells, depending on degree of resolution. The coordinates of the center point of cell in the grid that has maximum votes are output.

Note that to generate a circle completely, only one half of the portion of circle lying in the first quadrant needs to be uniquely generated. As seen in Fig.4, rest of the circle can be obtained by mirroring points on the curve in this region, over the coordinate axes (x, y) and the grid diagonals (y=x, y=-x). Subsequent points on the circle are generated using Givens transform with a step size small enough to ensure that next point is not so far away as to vote to a cell farther than cells adjacent to current cell. Mapping from set of points on the circle to set of cells in the grid is many to one and only one of these many would vote. Without any loss of generality, only the first point reaching a previously unvisited grid cell votes. Thus at the end of voting process for both circles in a fine grid, only the cells that receive votes from both the circles will have two votes and the others exactly one, as in Fig.3. In a coarse grid or with a high degree of overlap between the circles, a bunch of cells may obtain 2 votes, all of which need to be examined at higher resolution.

3.2. Hardware Design Issues

We use CORDIC rotation and vectoring modes [2] to compute sine, cosine functions and lengths of line segments joining landmarks respectively. The lengths of segments can also be stored in a lookup table. This is done in as many steps as the precision of CORDIC angles (in binary arctangent base) in the inbuilt lookup table of the architecture. At each iteration $i$ in the CORDIC operation,

$$x_{i+1} = x_i - y_i.d_i.2^{-i} \quad (16)$$
$$y_{i+1} = y_i + x_i.d_i.2^{-i} \quad (17)$$
$$z_{i+1} = z_i - d_i.tan^{-1}(2^{-i})$$ \quad (18)

where for rotation mode $d_i = -1$ if $z_i < 0$, +1 else and...
for vectoring mode \( d_i = 1 \) if \( y_i < 0 \), -1 else. Each CORDIC operation introduces a gain in the computed magnitude. In the rotation mode, \( x_0 \) is initialised to \( 1/gain \) so that at the end of CORDIC rotation, unscaled sine and cosine is obtained. However, the projections of landmark vector are not known and hence must be compensated for the gain. Since the above analysis is valid for input angles in range \(-\pi/2\) to \( \pi/2\) an initial rotation of 0 or \( \pi \) is performed for other values.

Next we compute unit vectors orthogonal to landmark line segments and coordinates of points on circles where grid diagonals would cut the circles \( (R_i/\sqrt{2}) \). This is achieved using streamlined divider modules, which use less storage than normal dividers. From these, the difference in coordinates of the centers of two circles is found to obtain relative coordinates for both circles in the grid.

The forward plane rotation [7] is used for generating circles (centered at the origin, with possibly different radii) in three lifting stages each of which uses simple shift and add operations as described below. The plane rotation matrix can be decomposed as:

\[
\begin{bmatrix}
\cos \alpha & \sin \alpha \\
-\sin \alpha & \cos \alpha
\end{bmatrix} = \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ u & 1 \end{bmatrix} \begin{bmatrix} 1 & p \\ 0 & 1 \end{bmatrix}
\]

\[
\text{where } p = (\cos \alpha - 1)/\sin \alpha \text{ and } u = \sin \alpha.
\]

Observe that the incremental angular step size, \( \alpha \) is chosen such that multiplication by \( p \) reduces to a bit shift. In stage 2, \( u = 2p/(1+p^2) \) can also be implemented as a bit shift if \( p \) is very small compared to 1. This approximation produces an error of \(< 1\% \) per \( 2\pi \) cycle, whose exact magnitude increases with \( p \). This is a tradeoff to avoid floating point multiplication for every point. Thus, consecutive points on circle can be computed in three steps involving bit shifts dictated by \( p, u \) and \( p \) respectively as in Fig.[5]

This implies a delay of two clock cycles (with one or two stages executing at each edge) to evaluate next point on the same circle. Observe that at any instant, only one stage of the transform is (two register sets are) active. So we can interleave the computation for the two circles such that coordinates of points corresponding to any particular circle reach the output registers on every alternate clock cycle. The angle counter is decremented only on alternate clock cycles.

Voting process uses hardware modules that employ a variation of flash ADCs that exploit the inherent correlation between consecutive votes eliminating computationally expensive division by grid resolution for each data point. It is not required to compare a data point with each cell boundary, but just the ones adjacent to it since step size ensures vote in existing or adjacent cell. Voting each point in the grid requires finding first cell to vote and an adjacent cell subsequently. Initially we employ binary logarithmic search to determine in \( O(\log N_{\text{grid}}) \) cycles (\( n \) clock edges for a \( 2^n \times 2^n \) grid), the first cell to be voted in the grid for each circle. Each of 8 points on the circle lying partially in the grid must pursue this search independently, since information about voting of a cell by a point cannot be used to conclude about vote of any other point on the circle. Observe that for the circle falling completely and symmetrically in the grid, mirror cell in the grid is identical to the cell that mirror point would vote to. For either circle, points mirrored over diagonals vote to a grid cell, only if it is different from cells voted by points being mirrored.

Voting process is further optimized in ADCs by accounting for the movement along either coordinate axes separately. Each data point, mirrored or otherwise should fall in the grid range for voting, indicated by a output valid signal from flash modules. Example output of voting scheme for 16-cell grid is shown in figure [3]. The block(s) with maximum votes are potential candidates for the intersection of the two circles. These need to be examined with higher degree of resolution [smaller angle increment and mapping coordinates of this (these) block(s) to the entire grid]. By applying this procedure repetitively the resolution of the coordinates of the desired point can be improved.

Voting at higher resolution needs mapping new Cartesian coordinate range to polar system for circle generation using (7). Radii of circles are still the same, but since loci may intersect at any angular coordinate, new angle for each corner of the new Cartesian range (with respect to center of circle, by CORDIC inversion of tangent) must be computed, whose extremes define the new angular range. Only circle generation and voting is required in subsequent iterations, as earlier computations do not change. Note that the 3-stage bit shift approximation of plane rotation continues to be valid since \( p \) goes down with increased resolution. These iterations must be performed until the required resolution is obtained.

### 4. Experimental Results

We conducted a set of experiments to validate the basic concepts. The tests were carried out on a Newport-Optics Optical Test Bench of size \( 9' \times 3' \times 6' \) fitted with 4 fully calibrated video cameras. The test scene was made of points in the three-dimensional space. We measured the location of...
Based on the experimental and simulation results, the following observations can be made.

1. The focal length of the camera should be short to obtain more reliable results.

2. The farther the distance between two landmarks, provided they can both appear in an image frame, the more reliable the measurement.

3. The closer the landmarks to the camera, the more accurate the results.

We found that our observations agree in general with the results of Weng et al. [21] and Lee [13]. Details on the algebraic methods for error estimation and analysis are discussed in [21]. The Schematics generated by Synopsys Design Analyzer Tool are shown in Figs. [6], [7] and [8]. The power and area obtained for the three modules, Flash converter, Streamlined division and Main GPS unit have been listed in Table below:

<table>
<thead>
<tr>
<th>Module</th>
<th>Power (mW)</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main GPS</td>
<td>1565</td>
<td>6544.73</td>
</tr>
<tr>
<td>Streamlined Divider</td>
<td>68.8</td>
<td>649.63</td>
</tr>
<tr>
<td>Flash Converter</td>
<td>391.89</td>
<td>1884</td>
</tr>
</tbody>
</table>

5. Conclusions

A rigorous geometric analysis of visual landmark based terrain navigation has been presented. The new insights introduced in this paper may be summarized as an image of two landmarks helps to constrain the unknown geoposition of the aerial observer onto the surface of a unique toroid; an image of three collinear landmarks reduces the same onto a unique circle, whose plane is perpendicular to the axis passing through the landmarks. The Hough transform-like approach is facilitated by a class of CORDIC-structured computations based on the results of our analysis that provides a fast and robust technique for compensating the GPS errors in locating the position of an aerial camera system. An architecture for geometric determination of camera location in landmark based navigation has been presented. We also identify techniques to improve throughput, a step closer to building an ideal real-time system. The design tightly integrates CORDIC computation and Hough transform with flash ADCs and three-stage circle generation and has been implemented successfully to yield high-resolution Cartesian coordinates of the observer. With N cell x N cell voting grid, the resolution of coordinates improves at each iteration approximately by a factor of N. Number of cycles has $N_{\text{cordic}} + N_{\text{divider}}$ one-time computations besides iterations of $\log N_{\text{grid}} + (\pi / (4.\text{step}))*2$ steps which depend on resolution desired.

Acknowledgement. First Author is grateful to (αβ-ical) Prof. Chaudhari, Desai, Fernandes, Lagu, R. Rao and Advisor, Sahasrabudhe of IIT-Bombay.

References


