SUPER-DENSE DIGITAL TERRAIN ELEVATION RECONSTRUCTION THROUGH METHOD OF EPIPOLAR CHARACTERISTICS

Dr. Pascal Monasse, Sr. Researcher
Dr. Lenny Rudin, Director R&D
Dr. Frédéric Cao, Researcher
Cognitech, Inc.
225 South Lake Avenue, Suite 601
Pasadena, CA 91101
pascal@cognitech.com
lenny@cognitech.com
frederic@cognitech.com

ABSTRACT

We propose a novel method for estimating super-dense 3-D geometry of urban scenes and other terrains from an image sequence captured by a calibrated airborne video camera. The flight is assumed to be at constant velocity in a piecewise straight path and the camera is oriented at nadir. Under these conditions, the epipolar lines are stationary. We extract characteristics propagation patterns from the stationary epipolar lines. This is equivalent to the method of characteristics for advection partial differential equations. The geometry of the characteristics is directly linked to the distance between the camera and the observed 3-D scene point. Accumulating and integrating all 3-D estimations over the entire flight sequence permits to build a 3-D mosaic of the scene. This novel algorithm opens up the camera field, making it panoramic. By aligning the epipolar lines profiles delimited by characteristics at different times through a maximal subsequence matching algorithm, the number of elevation posts becomes proportional to the number of spatially distinct pixels in the video data (super-dense condition). The practical applications of this work are done in collaboration with the National Geospatial Intelligence Agency (NGA). Results are demonstrated on NGA aerial video data recently collected for this approach.

INTRODUCTION

The most common technique for estimating the 3-D geometry of an urban scene or other terrain from aerial imagery requires a pair of images from different viewpoints and sufficient overlap. Correspondences between the images, that is the determination of image points recording light rays reflected by the same 3-D point, permit to triangulate the 3-D positions of these points with respect to the viewpoints, provided the system is calibrated (camera parameters and relative positions and orientations known). Constructing a digital elevation model (DEM) of the scene requires a dense set of correspondences, which involves intensive human interaction or automatic detection of correspondences (usually both). Automatic detection is reduced to a one-dimensional search by using epipolar lines, which are lines in one image associated to points in the other image and constraining the positions of corresponding points (Dhond & Aggarwal 1989). The triangulation is all the more accurate as the baseline (distance between viewpoints) is larger. However a large baseline signifies more occlusions and higher disparities, which make automatic correspondence estimation frequently fail. On the contrary a small baseline is favorable to the automatic procedure but has low accuracy. We propose a method that uses the redundancy present in a video footage to detect more reliably the correspondences, while using for accuracy the largest baseline for which automatic detection succeeds.

We assume a push-broom motion at constant altitude and velocity of the platform supporting the airborne video camera. We also assume the camera pointing constantly at nadir. This is the same flight protocol as recommended for Light Detection and Ranging systems (Lidar). In this case, the epipolar lines are stationary and all parallel to the direction of motion. Moreover, the apparent displacement of an imaged 3-D point along the epipolar line is at constant velocity. By stacking together the profile of the video frames along a fixed epipolar line, we obtain a space-time profile image, called epipolar plane image (EPI), where edges crossing the epipolar line produce line segments in the EPI, whose slope is proportional to the inverse of the vertical distance from the 3-D point to the camera (Bolles et al. 1989).

The proportionality factor depends on the platform velocity and internal video camera parameters (optical device internal parameters and frame rate), and is thus assumed known. We call these line segments in the EPI the characteristics, by analogy to characteristics in advection partial differential equations. The detection of characteristics can be done on the EPI for each epipolar line, providing a 3-D geometry restricted to edges. Recovering a dense DEM then involves interpolation between characteristics, for which we propose a dynamic programming approach. Aspects of this technique are covered by a patent application (Cognitech 2005).

DETECTION OF CHARACTERISTICS

Construction of EPI

Up to a rotation of the video frames, we can always assume that epipolar lines are in the *y* direction. By choosing an *x*, we fix an epipolar line. Taking the pixels on this column in the first frame and putting them as the first column of the EPI, and repeating this operation for each video frame, we construct an image where the *x* direction represents the time, or frame number, and the *y* direction represents the position along the epipolar line (Figure 1). This EPI has as many columns as there are video frames. Contrasted edges corresponding to static 3-D features produce line segments in the EPI, whose slope a is linked by the formula $a = f\,d / z$, where f is the focal length of camera expressed in pixel units, d is the displacement of the camera between successive frames, and z is the vertical distance of the 3-D point to the (projection center of) camera. In essence, we have reduced in this particular configuration the problem of establishing correspondences between video frames to the detection of line segments in EPI (Criminisi et al. 2005). These segments are observed as long as the motion from frame to frame is not larger than the extent of the smallest feature in the images. If it is not the case, we call the problem temporal aliasing, because the time sampling interval is not sufficient. Avoiding this default means flying high (but then we lose precision in depth estimation), flying slow (not always technically feasible), or using a video camera with sufficient frame rate.

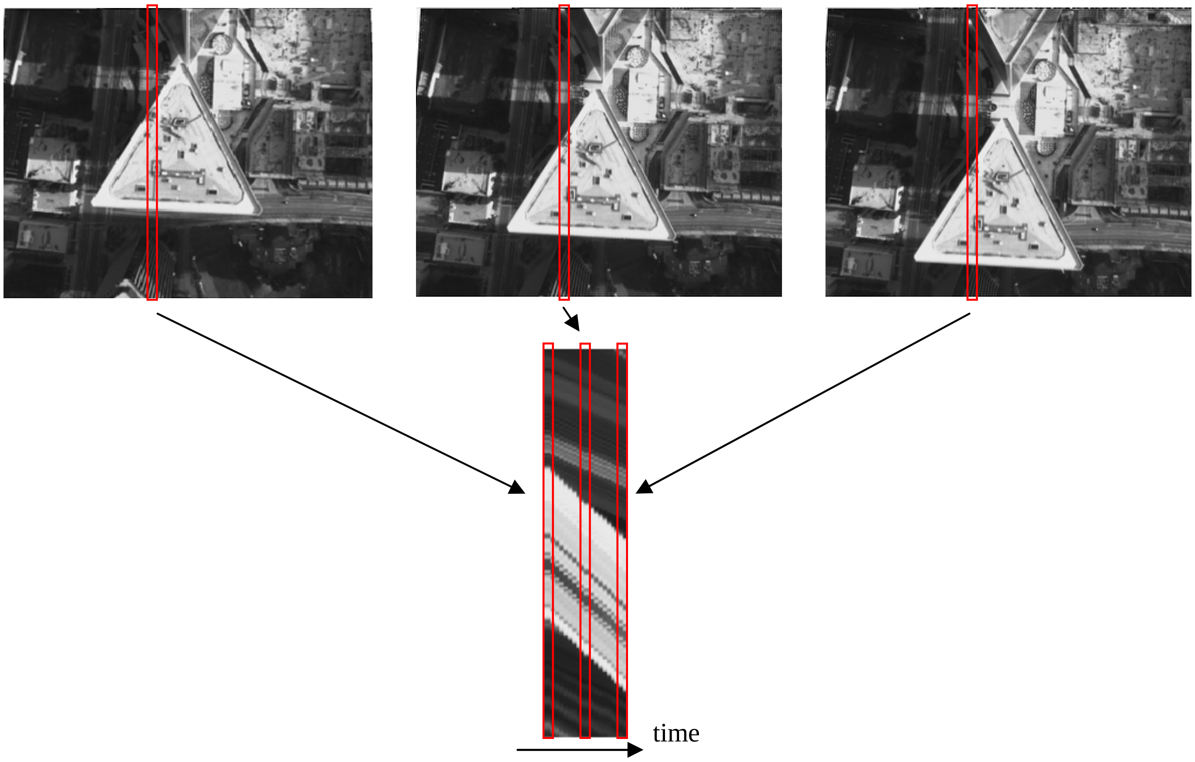
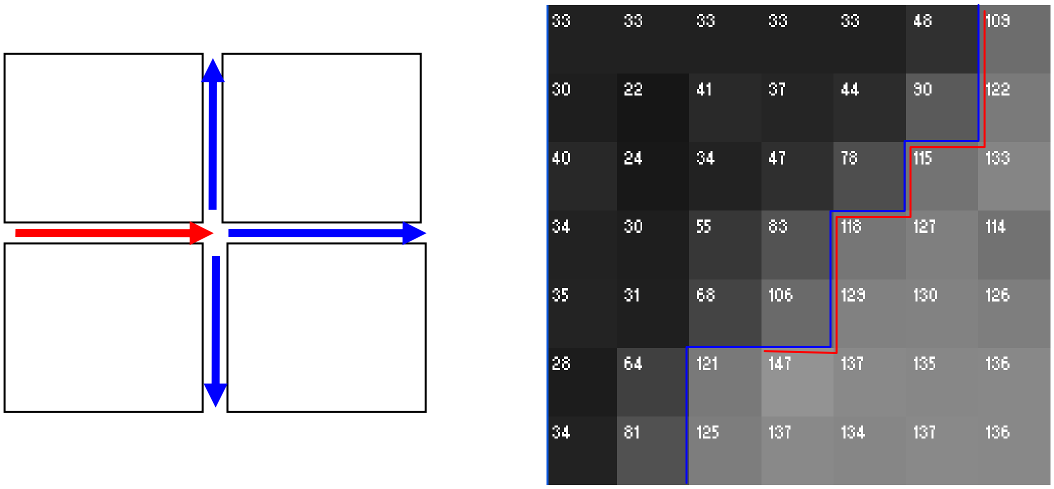


Figure 1. Construction of an EPI. Up: frames 1, 10 and 20 of a video sequence shot over Century City, CA, with one epipolar line marked in red. Down: juxtaposition of the frames contents creates the EPI.

Line Segments Detection

Traditional detection of straight line segments in images relies on edge detection followed by point linking or some Hough transform. This however depends on several parameters, which makes this kind of technique not reliable without manual parameter tuning adapted to the image. Instead of detecting edges, it has been remarked that intensity level lines of images provide good enough approximations of “visual edges”, while having the advantage of being parameter free and already curves (no a posteriori linking is necessary). Level lines are curves separating pixels where all pixels on one side of the curve have higher intensity level than all pixels on the other side. Their detection is straightforward: for a gray level *g*, threshold the image at level *g*, then follow the boundaries of the black pixels, and iterate over all possible *g*. Fast algorithms exploiting the nested nature of level sets are known (Monasse & Guichard 2000). Conversely, if two neighbor pixels have different intensity values, there exists a level line separating them. Instead of extracting first level lines and looking for straight segments on them, we use rather this last property to initiate the search and follow level lines, stopping when the level line deviates from the straightness constraint.

Let us define an edgel (edge element) as an oriented unit segment between two pixels, be it horizontal of vertical. There are three choices for the next edgel on a level line (Figure 2a). We are really following a level line as long as the minimum intensity level on one side of the curve so far is larger than the maximum on the other side. So as we move along the level line, we record and update these two values so as to ensure we stay on a level line. We can even impose a minimum gap between these two values to ensure the level line is contrasted, but in our experiments we have more dense detection with no significant increase of false alarms without this constraint. Moreover, in order to represent a straight segment, such a digital curve must never change direction: if one horizontal edgel is oriented to the left, no other edgel is oriented to the right and vice versa, with the similar constraint for vertical edgels. The straightness constraint is also much stronger: if we encode each horizontal edgel on the level line as a 0 and each vertical edgel by a 1, we get a binary code for the curve called a chain code. Only chain codes that are constant or have runs of successive 0’s (resp. 0’s) of almost equal length separated by isolated 1’s (resp. 0’s) represent chain codes of straight segments (Brons 1974). For example a chain code containing 0011 is not valid, neither is a chain code containing 1010001, since although the 1’s are isolated, there are runs of 0’s of respective length 1 and 3. As we move along a level line, we can record the chain code and stop when the chain code is not valid for a straight segment (Figure 2b).



33	33	33	33	33	48	109
30	22	41	37	44	90	122
40	24	34	47	78	115	133
34	30	55	83	118	127	114
35	31	68	106	129	130	126
28	64	121	147	137	135	136
34	81	125	137	134	137	136

(a) (b)

Figure 2. Level lines and line segments: (a) In the block of 4 pixels, if the current edgel is the red one, the 3 possible next edgels on the level line are in blue. (b) A level line (in blue) and a straight segment on it (red), of chain code 01101011. All pixels at the immediate left of the level line have intensity at most 106 while pixels on the immediate right of the level line have intensity at least 109.

The slope of a characteristic in an EPI is computed as the displacement in *y* divided by the displacement in *t*, that is the number of frames in which it can be tracked. So it will be all the more accurate as the segment is longer in the *t* direction, which means also a longer baseline for the correspondences. Therefore we record only as characteristics the straight segments on a level line that have a minimum time span. The threshold can be established from the required minimal accuracy in z of the 3-D estimation. To favor the detection of longer characteristics and to take account of noise, we do not enforce a strict straightness constraint on the whole chain-code. Instead, we fix some $l_{\min}$ and require every subpart of the chain-code of length $l_{\min}$ be valid for a straight segment. To prevent the detection of too curved lines caused by this relaxed condition, we check also that the curve remains within 2 pixels of the chord joining its endpoints.

Redundancy Elimination

Because edges in an image are always slightly blurred, when a characteristic is detected there is also a slew of other characteristics nearby sharing some edgels. They all represent variants of the same 3-D point, and it is better to keep only one of these competing characteristics. Preferably the longest in a group of competing characteristics should be stored and the others removed. However, the contrast across the characteristic is also important, as characteristics that are right on the edge must be more reliable. We choose to measure the contrast of a characteristic as the median of the differences of gray intensities between pairs of neighbor pixels separated by an edgel of the characteristic (Cognitech 2005). Other possibilities would include the minimum of these differences or their average (Desolneux et al. 2001). We measure the meaningfulness of a characteristic by its probability to have a contrast at least as high as observed. We assume that for a contrast level $c > 0$, $H(c)$ is the probability of having a contrast at least c across an edgel. An empirical estimation of H can be computed by computing the histogram of intensity level differences for all neighbor pixels and normalizing by the number of neighbor pixels. Given a characteristic C of n edgels, if the contrast across edgels are assumed to be i.i.d. random variables according to law H, the probability of having at least the observed median contrast c is the half-tail of the binomial distribution:

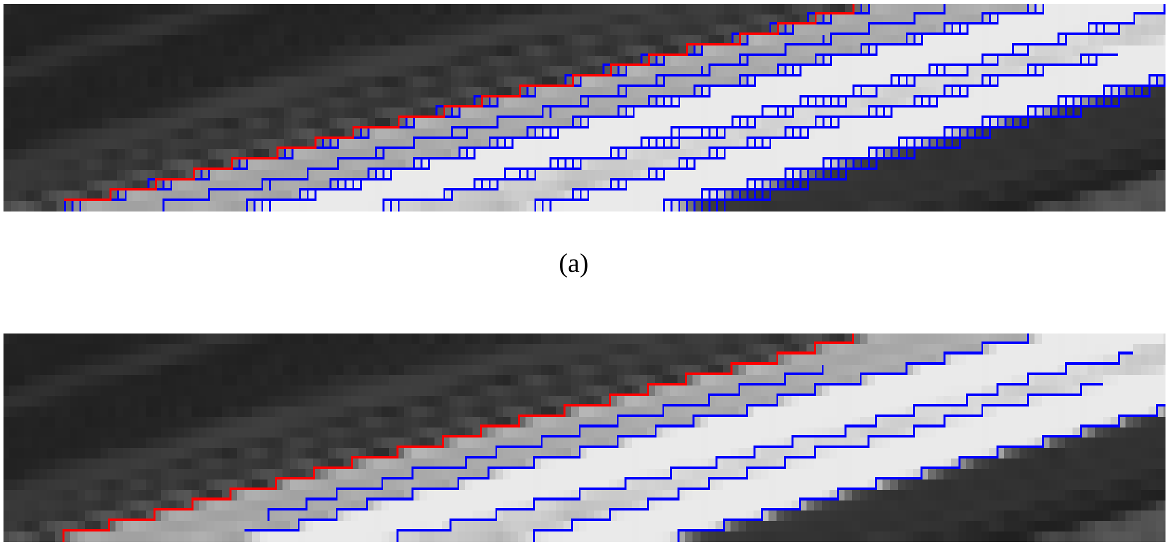
$$P(C) = \sum_{k=n/2}^{n} \binom{n}{k} H(c)^k \left(1 - H(c)\right)^{n-k} .$$

The lower this probability, the more unlikely the contrast is due to chance, and the more reliable the characteristic. We consider two characteristics to be competing with each other if they share at least one edgel. We want to keep the maximal subset of non-competing characterstics that minimizes the sum of probabilities. This problem being NP-complete, we use the following greedy algorithm:

1. Sort the segments by increasing probability.
2. Set all edgels unmarked.
3. For each segment S in the list, taken sequentially:
    - Remove from S all edgels already marked, resulting in smaller sub-segments.
    - Keep only the longest sub-segment of S.
    - Mark the remaining edgels of S.

In the end, only one characteristic may contain a given edgel, so that we get a non-competing set of characteristics. After this procedure, only the characteristics of sufficient length may be retained.

An example of the resulting simplification on an EPI is in Figure 3. The number of characteristics is reduced to the essential straight line segments.



(a)

(b)

Figure 3. Comparison of straight line segments and maximally contrasted straight line segments. (a) All detected straight line segments in an EPI, one of them marked in red. (b) The maximally contrasted straight line segments, with the red one winning over the competing red straight line segment in (a).

SUPER-DENSE 3-D GEOMETRY

The preceding steps can only provide 3-D geometry estimation at edges not parallel to the epipolar direction. This is what we call a dense estimation, as it potentially computes the geometry on all edges, that is everywhere where there is information available. The uniform parts between edges cannot be readily matched because of the aperture problem. They can be only guessed from nearby edges, thus it is an interpolation problem.

Matching as a Search for a Longest Common Subsequence

In practice, not all edges provide a characteristic. In particular, details moving with an apparent frame to frame displacement larger than their extent along the epipolar line do not create one level line in the EPI, a phenomenon called time aliasing (Figure 4). Matching these while interpolating is important. Consider a given EPI and frames at times t_1 and t_2. The characteristics passing through both times t_1 and t_2 separate the epipolar line in a partition at time t_1 and a corresponding partition at time t_2, since characteristics do not cross, as they are defined as parts of level sets. If we consider the sequence of pixel values in a segment of this partition, we get $a_1a_2...a_m$ at time t_1 and $b_1b_2...b_n$ at time t_2. Notice that m and n need not be the same, this happening only if the delimiting characteristics are parallel. We want to interpolate the delimiting characteristics by new characteristics satisfying the constraints that characteristics do not cross. The best interpolation is the one that matches intensity levels with the least error. The cost of matching a_i to b_j is $d(i,j)=|a_i-b_j|$ if we use only intensity level, or some Euclidean distance if we have color information. We look for a correspondence f between indices that is non-decreasing. We can represent the correspondence f as a path in the rectangle $[0,m]$x$[0,n]$ between the points $(0,0)$ and (m,n) with a diagonal segment from $(i-1,j-1)$ to (i,j) if f matches index i in a to j in b and horizontal or vertical segments in-between, that is for indices not matched (Figure 5). The former represent matches while the latter represent occlusions (a pixel having no match in t_2) or disocclusions (a pixel having no match in t_1).

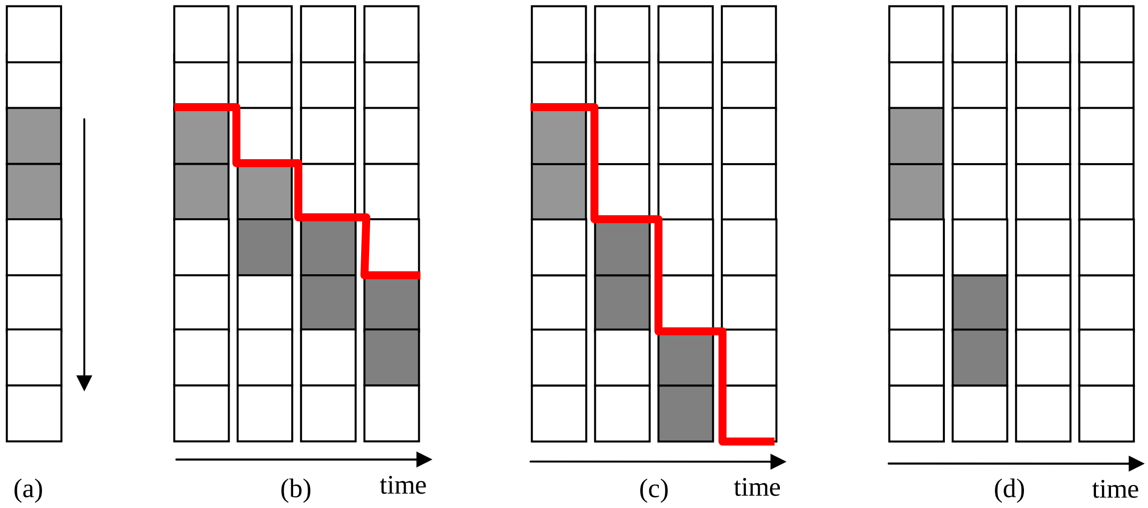


Figure 4. Occurrence of time aliasing depending on frame rate. (a) Pixels of an epipolar line, the arrow indicating the direction of motion. (b) At sufficient frame rate, there is no time aliasing and a characteristic is detected. (c) At the critical frame rate, a characteristic can still be detected. (d) At an insufficient frame rate, the level set in the EPI is broken into several components and no characteristic is detected.

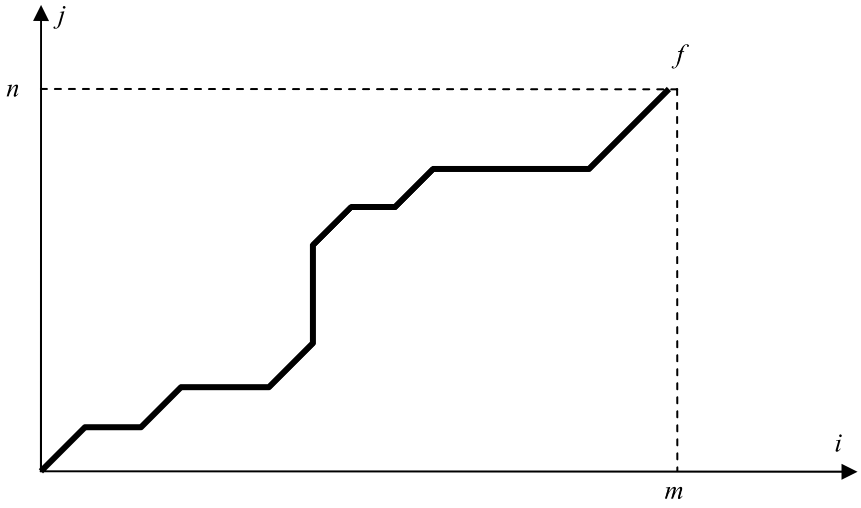


Figure 5. The longest common subsequence problem can be viewed as finding a path with lowest cost between (0,0) and (*m*,*n*) with only diagonal (meaning match), horizontal (meaning occlusion) and vertical (meaning disocclusion) increments.

We have to put a cost to occlusions and disocclusions, otherwise the best subsequence match is the one that occludes everything, at cost 0. To put a cost commensurate to the cost of a matching, we can for example take the difference of the averages of intensity levels on the epipolar line.

Dynamic Programming Algorithm

If we denote by C(*i*,*j*,*k*,*l*) the cost of the best increasing path from point (*i*,*j*) to (*k*,*l*), the fact that the cost is additive ensures that C(0,0,*m*,*n*)=C(0,0,*i*,*j*)+C(*i*,*j*,*m*,*n*). This says that if we know the best path from (0,0) to (*i*,*j*), it is sufficient to look for the best path from (*i*,*j*) to (*m*,*n*) and concatenate the paths to get the global solution. This is the paradigmatic case for a dynamic programming algorithm (Hirschberg 1977): we have

$$C(0,0,i+1,j+1) = \min(C(0,0,\mathrm{i},\mathrm{j})+d(i,j),\ C(0,0,i+1,j)+c,\ C(0,0,i,j+1)+c)$$

if $d(i,j)$ is the cost of matching a_i to b_j, and c is the cost of occlusion/disocclusion. In other words, it is easy to update the best path if we know its left, down and left-down neighbors. Starting from (0,0), we use a region growing approach to get to (m,n). The complexity of this algorithm is $O(mn)$.

Notice that this algorithm can be applied globally to an epipolar line, without partitioning by characteristics. However, experience shows that the result is much improved if anchor points at detected characteristics are used.

The maximum principle, stating that the slope of interpolated characteristics must be comprised between the slopes of delimiting characteristics, helps prevent aberrant interpolations. Also a more sophisticated way to compute the matching error $d(i,j)$ is to compare not only the initial and final pixel a_i and b_j, but instead to compute the variation in the the EPI on the segment joining both pixels. The EPI should be the most homogeneous possible on such a segment. So for example the variance can be chosen as a measure of error.

DEM CONSTRUCTION

Depth from Characteristics

Consider a world point of coordinates $(X(t), Y(t), Z(t))$ with respect to the camera position at time t. Since the camera moves at constant altitude, we have $Z(t)=Z(0)$, if we take as time origin the instant of the first frame. If we denote by $(x(t), y(t))$ the imaged point at time t, we have $Z(x\ y\ 1)^T = K(X\ Y\ Z)^T$, with K the calibration matrix given by

$$K = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}.$$

by taking the time derivative of this linear equation, we get $Z(x'\ y'\ 0)^T = K(X'\ Y'\ 0)^T$. The norm of the vector $(X'\ Y')$ is the known velocity V of the camera. The image point (x,y) moves along the epipolar line with a slope v, so we can write $(x',y') = v\ (\cos\theta, \sin\theta)$, where θ is the angle of the epipolar lines with respect to the x-axis in the image. This leads to the equation

$$V = Zv\left(\frac{1}{f_x^2}\left(\cos\theta - s\frac{\sin\theta}{f_y}\right)^2 + \frac{\sin^2\theta}{f_y^2}\right)^{1/2}.$$

we see that Z is inversely proportional to v, with the proportionality factor depending on the platform velocity V, the camera parameters f_x, f_y and s, and the direction of epipolar lines θ. Notice that if s=0 and f_x=f_y=f, we get $fV = vZ$, the formula given in the introduction.

Geo-referencing

The DEM estimated through the methods of characteristics permits for example to create the ortho-rectified image of the scene (for standard ortho-rectfication reference, see Novak 1992, Abib et al. 2007). For inclusion into a GIS, it needs also to be geo-referenced, that is the geodetic coordinates of each pixel must be recorded. Also, the DEM records vertical distances to the camera, not to the ellipsoid representing Earth. As explained above, the recovered X, Y and Z are in the coordinate system linked to the camera position and orientation at time 0, with (X,Y) being the image plane and Z the view direction. There is a rigid motion between the world system of coordinates (Earth Centered, Earth Fixed, or ECEF) and the camera coordinate system. This motion consists of a rotation R followed by a translation T. The translation vector T is given directly by GPS coordinates at time 0. Since we are at nadir, the optical axis of the camera (0,0,1) is sent to the inner normal to the Earth at the latitude and longitude of the projection center at time 0. Moreover, the direction of the epipolar lines $(\cos\theta, \sin\theta, 0)$ is sent to the motion direction, available through GPS. Knowing the image of two independent vectors by a rotation R completely determines it.

EXPERIMENTS

We illustrate the automatic extraction of a DEM from a video sequence of 20 frames over Century City, CA. The Basler digital video camera of resolution 640x480 at 30 frames/s was supported by a helicopter flying at 300 m altitude.and speed 30 m/s. The horizontal angle of view of the camera is 40°. Frames of the original sequence are in Figure 6.

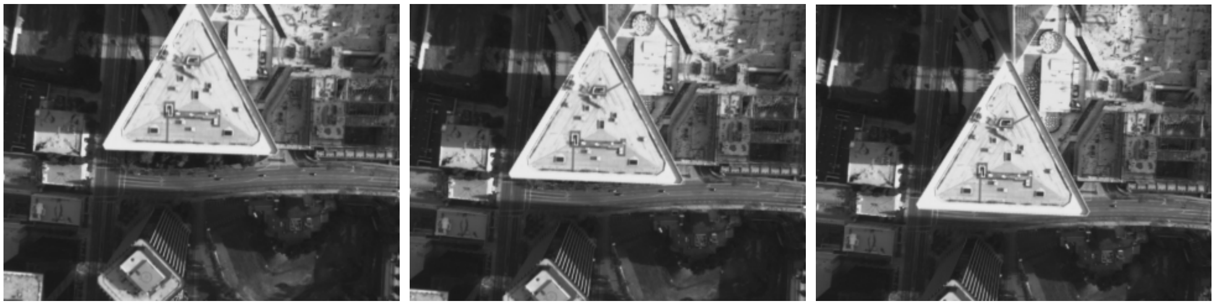
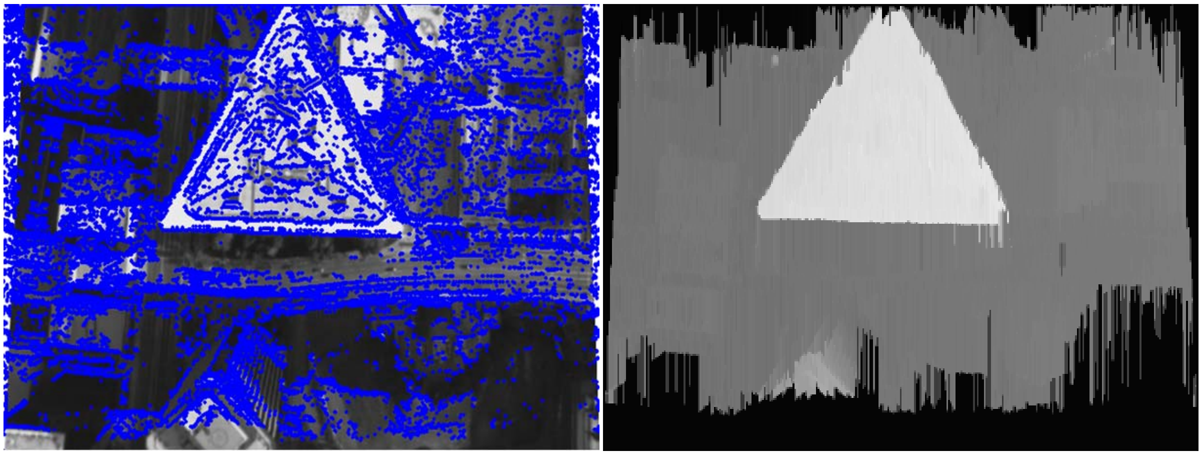


Figure 6. Frames 1, 10 and 20 of a video segment over Century City, CA.

The direction of motion of the camera is oriented up relative to the image, so features move down in the video. The camera was tilted slightly forward, so a rectification of the view direction is performed first: it is a planar projective transform that, after application to the video, makes epipolar lines parallel. The points in frame 1 belonging to the detected characteristics visible for at least 10 frames are displayed in Figure 7a. Notice that most edges generate a characteristic, except where they are vertical.



(a) (b)

Figure 7. (a) Points of the first frame in Figure 6 belonging to a characteristic. (b) Estimated DEM.

The resulting DEM is illustrated in Figure 7b. The black parts along the boundary are there because they delimit the first and last point on a characteristic for each epipolar line. Notice that the two highest buildings are clearly visible. Some other rooftops are also discernible in the DEM, but correspond to much lower buildings. Since epipolar lines are processed independently, errors between adjacent epipolar lines provoke the vertical patterns. The triangular building, the Century Plaza Tower, is estimated at height 176 m, whereas the ground truth (Emporis 2007) is 174 m. The 3-D rendering of the estimated DEM with the video frame as applied texture is illustrated in Figure 8. Notice that two facades of the tower are transparent. This is due to the fact that the video stops before they become visible.

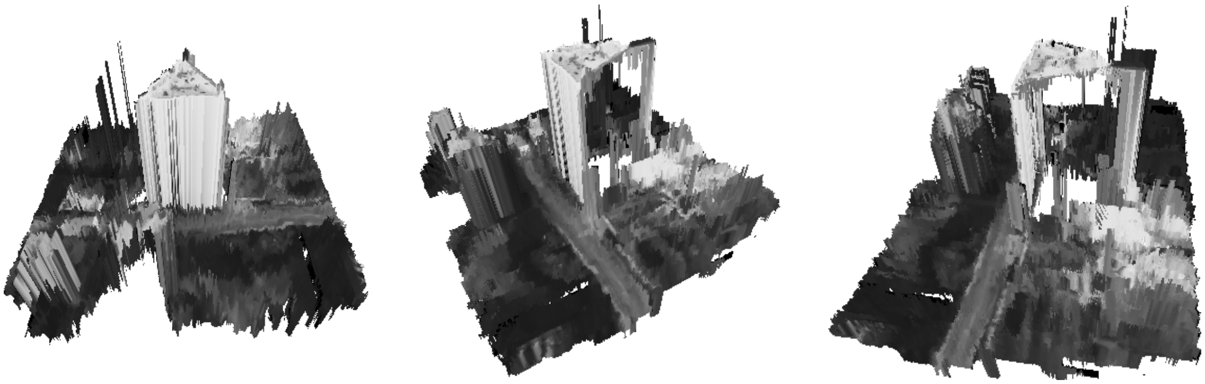


Figure 8. 3-D rendering of the estimated DEM with video frame used as texture.

Another test video segment is shown in Figure 9. The same Basler video camera was flown over downtown Los Angeles, CA, at an altitude of 400 m and a speed of 50 m/s. The epipolar lines are parallel but at 10° off vertical.

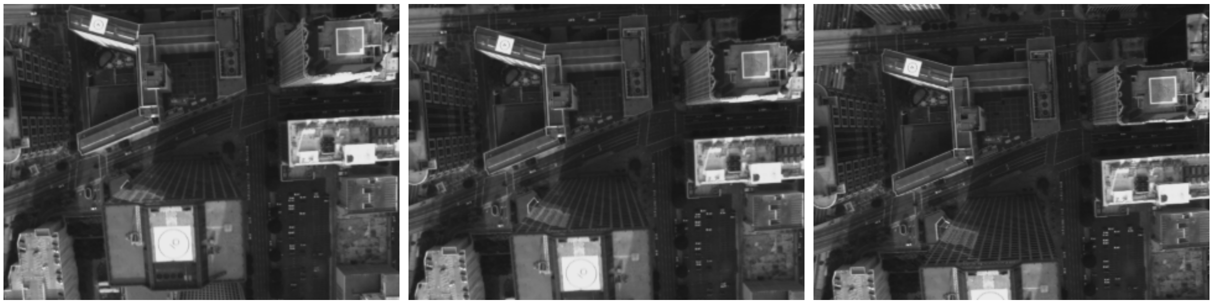
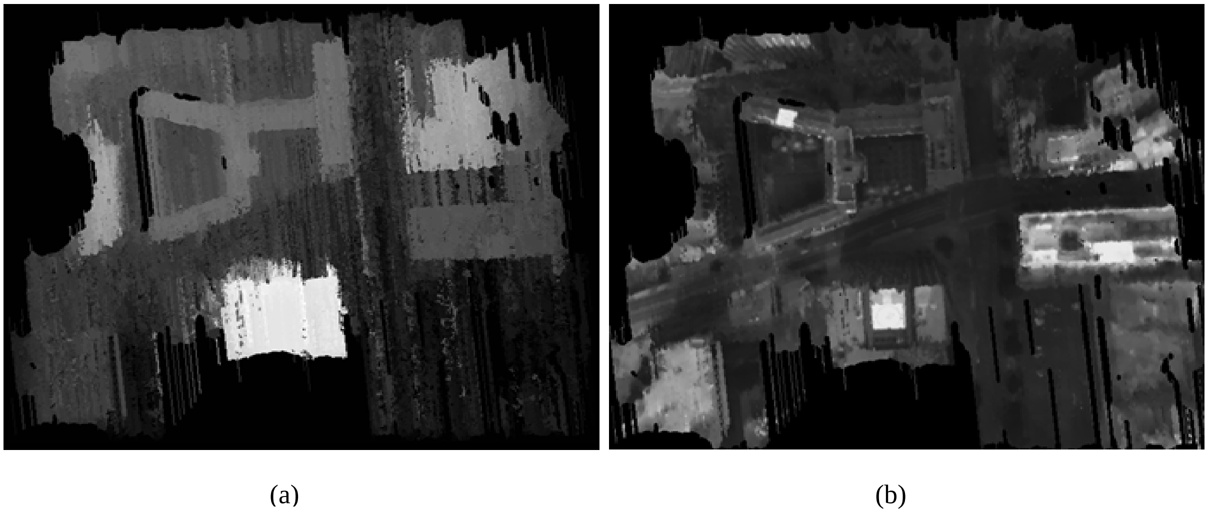


Figure 9. Frames 1, 10 and 20 of a video segment over downtown Los Angeles, CA.

The reconstructed rectified DEM and ortho-mosaic are shown in Figure 10. An ortho-mosaic is constructed by extending the characteristics and intersecting them with a fixed frame number, as if these were points tracked through that instant, even though they are not visible because of the limited angle of view (Figure 11). For example, the high building at the bottom of the video is fully visible in the first frame but then gets occluded. On the contrary, the hotel structure in the center of the video is partially occluded in the beginning but becomes visible in the last frame. Both are reconstructed in the DEM and the ortho-mosaic in Figure 10.



(a)

(b)

Figure 10. (a) Computed ortho-rectified DEM from the video over downtown Los Angeles, CA. (b) Computed ortho-mosaic from the video over downtown Los Angeles, CA.

The building in the upper right area of the frames, the Figueroa Tower, is estimated at 110 m, while the ground truth is 109 m. The highest building, in the central bottom, the Ernst & Young Plaza, is measured at 175 m. The ground truth is 163 m. The discrepancy in this measure is due to the presence of an elevated helipad on the building, which is not measured in the ground truth.

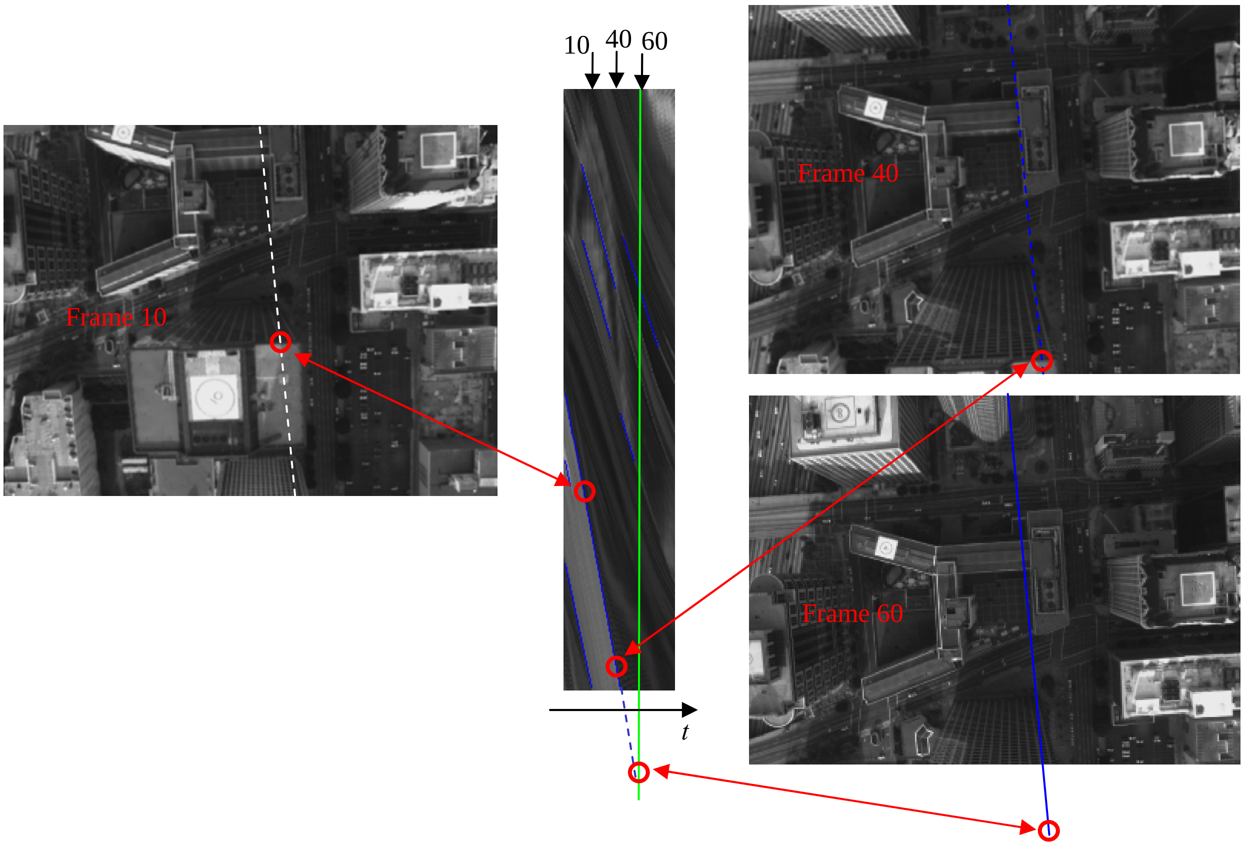


Figure 11. Construction of a 3-D mosaic from video. An epipolar line is shown in three different frames of the video. A point on this epipolar line generates a characteristic in the EPI, which gets occluded shortly after frame 40 because of the limited angle of view. By extending the characteristic, we can find the position of this point in the geometry of frame 60, even though it is outside the image frame.

The 3-D rendering of the estimated DEM with the ortho-mosaic as applied texture is illustratred in Figure 12.

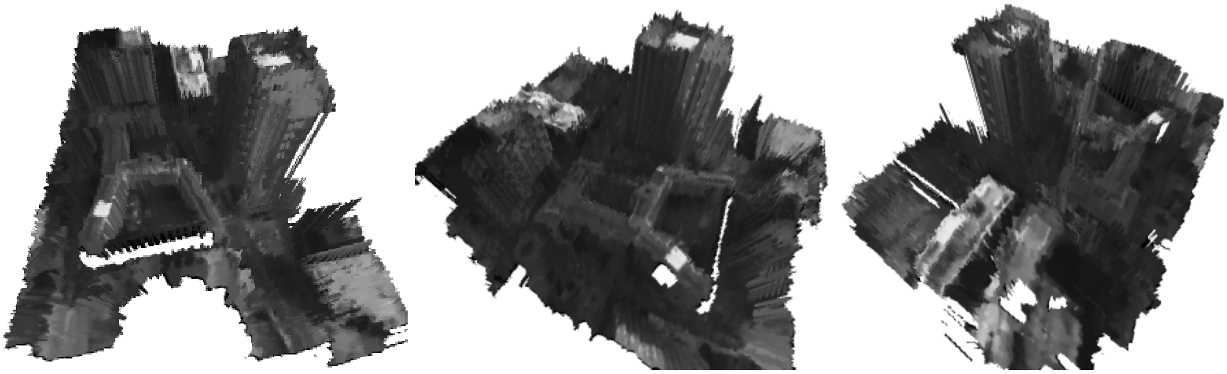


Figure 12. 3-D rendering of the estimated DEM with applied ortho-mosaic as texture.

CONCLUSION

We have presented a novel method for computing the 3-D geometry of a scene in four steps, provided the camera satisfies a flight protocol (push-broom motion at constant altitude and velocity). We take first time-slices of the video volume along epipolar lines. The straight line segments in these images, called characteristics, are then detected and provide a 3-D estimation along edges. This geometry is interpolated by a longest common subsequence algorithm to give a super-dense depth estimation. Finally, the depth map is ortho-rectified and geo-referenced by using in-flight GPS data. The technique presented here provides a low cost alternative to lidar systems, and has the advantage of using a passive captor.

Future possible improvements include the removal of the dependency on two fixed times t_1 and t_2 for the interpolation by dynamic programming. Indeed, only features producing characteristics through times t_1 and t_2 are taken into account for initiating the longest common subsequence search. However, all detected characteristics can be extended so that we could know the position the 3-D points would have at times t_1 and t_2. This would involve taking care of not extending the characteristics beyond their visibility limits, that is detecting occlusions and disocclusions. This happens when two extended characteristics intersect. As characteristics cannot cross, that means that one hides the other. Typically, the characteristic of highest slope, corresponding to a point closer to the carmera, should be the occluding one. Future research may also focus on regularizing the depth map by introducing dependency between adjacent epipolar lines. For example, some anisotropic filtering of the depth along the edges would smooth the DEM.

REFERENCES

Bolles, R.C., Baker, H.H. and Marimont, D.H. (1989). Epipolar plane image analysis: an approach to determining structure from motion. *International Journal of Computer Vision, 1*(1): 33-49.

Brons, R. (1974). Linguistic methods for description of a straight line on a grid. *Computer Graphics and Image Processing, 3*(1): 48-62.

Criminisi, A., Kang, S.B., Swaminathan, R. Szeliski, R., and Anandan, P. (2005). Extracting layers and analyzing their specular properties using epipolar-plane image analysis. *Computer Vision and Image Understanding, 97*(1): 51-85.

Desolneux, A., Moisan, L. and Morel, J.M. (2001). Edge detection by Helmholtz principle. *Journal of Mathematical Imaging and Vision 14*(3): 271-284.

Dhond, U.R. and Aggarwal, J.K. (1989). Structure from stereo – a review. *IEEE Transactions on Systems, Man and Cybernetics, 19*(6): 1489-1510.

Emporis (2007). Emporis buildings – International database about buildings, construction and the real-estate industry. Web site at http://www.emporis.com.

Habib, A.F., Kim, E.M., and Kim, C.J. (2007). New methodologies for true orthophoto generation. *Photogrammetric Engineering & Remote Sensing, 73*(1): 25-36.

Hirschberg, D.S. (1977). Algorithms for the longest common subsequence problem. *Journal of the ACM, 24*(4): 664-675.

McGlone, J. Chris (2004). *Manual of Photogrammetry*. *Fifth Edition*. American Society of Photogrammetry and Remote Sensing, Bethesda, Maryland.

Monasse, P. and Guichard, F. (1999). *Scale-space from a level lines tree*. Lecture Notes in Computer Science 1682, Springer, pp. 175-186.

Monasse, P. and Guichard, F. (2000). Fast computation of a contrast invariant image representation. *IEEE Transactions on Image Processing, 9*(5): 860-872.

Novak, K. (1992). Rectification of digital imagery. *Photogrammetric Engineering & Remote Sensing, 58*(3): 339-344.

Cognitech (2005). USPTO patent application regarding the subject has previously been filed on behalf of Cognitech, Inc., Pasadena, CA