AN ALTERNATIVE COST FUNCTION TO BUNDLE ADJUSTMENT USED FOR AERIAL PHOTOGRAPHY FROM UAVS

Dale E. Schinstock, Associate Professor
Chris Lewis, Associate Professor
Craig Buckley, Research Assistant
Kansas State University
3002 Rathbone Hall
Manhattan, KS 66506
dales@ksu.edu
clewis@ksu.edu

ABSTRACT

The Autonomous Vehicle Systems Lab at Kansas State University has developed an unmanned aerial vehicle capability for inexpensively performing aerial surveys. Bundle Adjustment (BA) is used in the extraction of terrain information from these aerial surveys. However, the inexpensive, lightweight inertial navigation system on the UAV does not give good initial estimates of the camera poses for the BA algorithm. This leads to an incorrect convergence of the BA minimization in some cases. This paper presents an alternate formulation of BA that uses a reduced set of parameters that are varied in the minimization. For photographic sets covering large areas, but having overlap only between adjacent photos, the search space is significantly reduced when compared to standard BA. The usefulness of the algorithm is demonstrated by generating a digital elevation model of a region covered by 11 photos and an ortho-rectified composite image of this region. While this alternate formulation provides a reduced search space and significant insight about BA itself, the standard formulation seems to provide similar convergence properties in a direct comparison.

INTRODUCTION

The autonomous Vehicle Systems (AVS) Lab at Kansas State University (KSU) performs remote sensing of the nearby Konza Prarie, an NSF long-term ecological research (NSF-LTER) site, using a hand-launched autonomous unmanned aerial vehicle (UAV). This system consists of an RC hobbyist airframe (Sig Kadet Senior) that has been modified to incorporate a Cloudcap Technologies Piccolo II autopilot and remote sensors. The modified payload bay has space to handle devices such as a laser range finder, various cameras, and/or a PC104 computer. The avionics computer triggers the cameras so that the state of the aircraft is recorded when the photos are taken. This autonomous aerial photography system facilitates frequent, low cost imaging of areas for biological and agricultural research. Because the aircraft typically flies at 100 meters or less, very high resolution imagery is possible. The imagery can be digitally post processed into a variety of photogrammetric products including ortho-rectified photo-mosaics and digital terrain models (DTMs).

The autopilot's GPS-aided inertial navigation system is based on lightweight, low-cost inertial sensors, resulting in poor accuracy. Therefore, the initial camera pose (position and orientation) estimates necessary for stereo processing are poor. Bundle Adjustment (BA), [1] is typically used to refine pose estimates. However, our experience with the UAV system and the BA algorithm in a commercial package was very troublesome due to the large errors in the initial pose estimates. BA simultaneously triangulates features and determines camera parameters, including pose. Although BA can be used to estimate intrinsic camera parameters (e.g. focal length) along with extrinsic parameters (pose), intrinsic parameters are typically estimated in a camera calibration process preceding BA. For an in depth discussion of variations in BA see [4].

In BA, a cost function is defined that expresses the errors between the observed coordinates of features in images and the image coordinates found through back-projection of the estimated 3D feature locations. Feature locations and the camera parameters are varied to minimize this cost function. Typically, image blocks taken with an airplane to cover an area have overlap between adjacent images only. This creates a significant number of variables for the algorithm to estimate because the number of features is large. To combat this growth in search variables, an alternate approach has been developed. This alternate BA algorithm optimizes a different cost function. This cost function is the sum of squares of the minimum distance between the projection rays for each feature. Ideally, the projection rays from different camera poses corresponding to a single feature would intersect at the feature’s location. Since the feature locations are implicit in this alternate cost function, the search space contains only the camera parameters.

Both optimization methods depend on the identification of common features (tie points) within different images. To accomplish the identification of tie points we use an automated algorithm called “Scale Invariant Feature Tracker” (SIFT) developed by Lowe [2]. Figure 2 shows an example of tie points from two overlapping images. Miscorrelations of features are rare, but will usually emerge as outliers with during the minimization process and can therefore be recognized and subsequently removed. Correspondences between pairs of images can be combined to create a list of features and their corresponding matches in all other photos.

PROCESS OVERVIEW

Utilizing BA in dense automatic terrain extraction (ATE) typically involves the following three distinct steps. The first step identifies corresponding features from overlapping images to be used as tie points. The second step, BA, improves the camera pose estimates. The third step performs a dense correlation in the overlapping regions of the images and triangulates a dense 3D point cloud using the correlated pixels, the camera pose estimates, and the corresponding rays from the cameras. It also frequently meshes the features to extract a regularly gridded DTM. For the analyses typically targeted by the AVS Lab, an ortho-rectified composite image is generated in an additional step by projecting the original images onto the DTM.

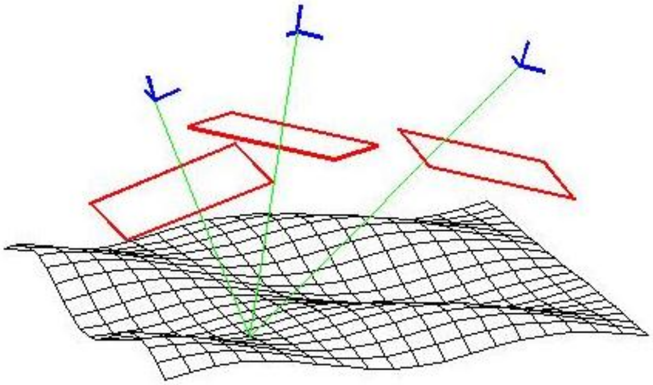

Figure 1. Triangulation of a global feature point with three perspectives

Feature Detection and Correlation

Automatic feature detection and correspondence begins with identifying candidate features within each image that are likely to be easily identified in other images. Pixels that stand out from the surrounding pixels are likely to do the same in other photographs of the same geographic area. The candidate features from each image are then corresponded with features from other images. In our work here SIFT is used for this process. With SIFT, a feature description vector is calculated from the changes in the image surrounding a pixel. This feature description vector is designed to be similar for an object regardless of the scale or zoom factor of the images. Matching feature descriptors between images generates a list of matching or corresponding features and their associated pairs of images. When applied to multiple overlapping images, this process yields a complex list of associated feature/image pairs that must be combined to determine both a global list of features and a list of which images contain which features. Each feature needs at least two perspectives in order to triangulate its 3D position. The figure below shows an example of a pair-wise image comparison. Features are circled with a line between them showing the respective match in the other photo. The lines are parallel since translation dominates the motion between these two images.

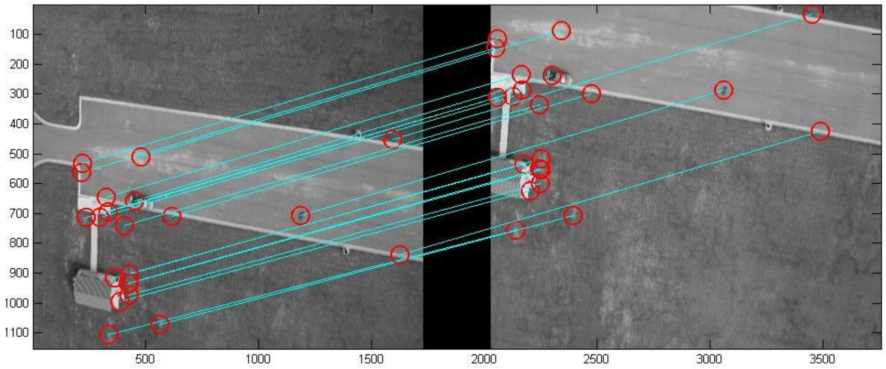

Figure 2. Features (red circles) in the left image correspond to the same feature in the right image.

Bundle Adjustment

BA is an algorithm that operates on all the features and poses as a block to simultaneously refine the pose estimates and estimate the feature locations in 3D. In BA, a cost function is defined that expresses the error between the back-projected 3D feature and the observed feature in image space (*u* and *v* image coordinates). To solve the minimization problem, feature locations and the camera parameters are varied to optimize the cost function. This optimization is frequently accomplished with the Levenberg-Marquardt [3] optimization algorithm. This algorithm will be described in more detail later, however, it is important to understand that like many optimization procedures, it requires repeated computations of a Jacobian matrix found with partial derivatives of the cost function vector with respect to the variables estimated during the optimization. The dimensions and complexity of the Jacobian affect the computational complexity of the algorithm and ultimately limit the size of problem that can be handled. The standard formulation of BA is particularly well suited to image blocks having lots of images and a relatively small number of features. Each feature adds three new search variables to the optimization. Each new image of those same features adds 6 variables. Standard BA is less suited to blocks containing large numbers of features. Image blocks taken with an airplane to cover an area typically have overlap between adjacent images only. This creates a significant number of variables for the algorithm to estimate.

Terrain Extraction

The terrain features triangulated in BA are necessarily limited to high-contrast easily-identifiable features. Usually, these features are too few and insufficiently distributed to reconstruct the entire terrain accurately. Therefore, a final process is necessary to fill in the gaps and extract a detailed terrain model. Additional terrain information is typically extracted by correlating pixel values in one image with those in other images using the adjusted pose information. A best fit estimate of the terrain elevation at any point can be formed by correlating image patches along epi-polar lines for each image pair. Therefore, an elevation estimate can be formed for most pixels in an image even if the coverage of global features is sparse in that area.

ALTERNATIVE COST FUNCTION

The image coordinates of a feature, the camera pose, and the camera's projection model define a ray through space. Ideally, two disparate images of the same feature define two rays that intersect at the 3D location of that feature. Due to noise and imperfections in the projection parameters, the actual rays for any given feature will typically pass close to each other and near the location of the feature, but not actually intersect. Based on this observation, an alternative cost function is defined. This cost function is a sum of squared elements from a cost vector. Any individual element of this vector is the minimum distance between two rays from a pair of images corresponding to a common feature. The cost function is built using all possible pairs in the set of images. This eliminates the actual feature locations from the list of parameters varied in the minimization. To optimize this cost function, the size of the search space is six times the number of camera poses. The size of the search space for standard BA is six times the number of poses plus three times the number of features. The number of features can become very large in a typical image set from aerial photography.

Development of the Cost Function

This section discusses the details of how the individual costs are computed. It is assumed that a feature has been found and corresponded in two images, giving observed image coordinates in both images. It is also assumed that these coordinates have been corrected using the intrinsic camera parameters to account for distortion so that the camera model closely approximates the pinhole projection model[5][7][8]. Using the image coordinates of a feature (*u* and *v*), a unit vector, $\vec{u}_i$, in a global reference frame's coordinates pointing from the Center of Projection (COP) for an image to the feature can be found using the following equations.

$$\vec{U}_i = R_{gc} \begin{bmatrix} u_i \\ v_i \\ f \end{bmatrix} \tag{1}$$

$$\vec{u}_i = \frac{\vec{U}_i}{\left|\vec{U}_i\right|} \tag{2}$$

Here *f* is the focal length of the camera and R_{gc} is the rotation matrix from the camera frame to the global frame. This rotation matrix is a function of the pose of the camera.

Figure 3 is a depiction of the calculation of a single element of the cost function. $\vec{P}_1$ and $\vec{P}_2$ are the positions of two camera frames. The vector describing the minimum distance between the two rays must be perpendicular to both. A unit vector giving the direction of this vector is found with a cross product.

$$\hat{\vec{r}} = \frac{\vec{u}_1 \times \vec{u}_2}{\left|\vec{u}_1 \times \vec{u}_2\right|} \tag{3}$$

A signed magnitude, *c*, of the minimum distance can be found with the projection of any vector from a point on one ray to a point on the other onto the vector of the minimum distance.

$$c = (\vec{P}_2 - \vec{P}_1) \cdot \hat{\vec{r}} \tag{4}$$

This cost element is a function of both the positions and orientations of the two camera poses.

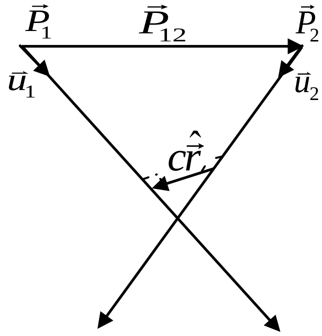

Figure 3. Cost function calculation.

The individual cost elements are assembled into a k by 1 vector, where k is the total number of pairs of rays. $\vec{c} = \begin{bmatrix} c_1 & c_2 & \cdots & c_m \end{bmatrix}^T$, and the scalar cost function is the squared magnitude of this vector, $C = \vec{c}^T \vec{c}$.

MINIMIZATION

The cost function defined in the previous section is a sum of squares of individual elements. A Levenberg-Marquardt (LM) algorithm [2][6], which is commonly used in bundle adjustment, is a likely method to use in the minimization of this type of cost function. At each iteration step of the minimization, the LM algorithm updates the state vector $\vec{K}$ with a step:

$$d\vec{K} = \left(J^T J + \lambda I\right)^{-1} J^T \vec{c} \tag{5}$$

Here, J is the Jacobian of the cost vector, $\vec{c}$, with respect to the state vector $\vec{K}$.

$$J = \frac{\partial \vec{c}}{\partial \vec{K}} \tag{6}$$

For the cost function in the previous section, the state vector is composed of the camera poses for all images. Each camera pose has three elements of position and three elements of orientation.

$$\vec{K}_i = \begin{bmatrix} X & Y & Z & \omega & \varphi & \kappa \end{bmatrix}^T \tag{7}$$

The camera position, $\vec{P}_i$, in (4) is composed the first three elements of $\vec{K}_i$ and the rotation matrix, R_{gc}, in (1) is a function of the last three. The full state vector is a concatenation of the individual poses.

$$\vec{K} = \begin{bmatrix} \vec{K}_1^T & \vec{K}_2^T & \cdots & \vec{K}_n^T \end{bmatrix}^T \tag{8}$$

An analytical Jacobian is required for the minimization algorithm. The analytical development was completed with a symbolic software package and is much too large to give here.

The terms $J^T J$ and λI in (5) represent a trade off between an approximation of the Hessian and the gradient respectively, where λ adjusts the tradeoff between Steepest Gradient Descent and a Gauss-Newton type algorithm. Minimization starts with a large initial value for λ because Steepest Gradient Descent is more efficient far from the minimum. The minimization is terminated either when a maximum number of iterations is reached, the norm of the gradient $\left\| 2J^T c \right\|$ falls below a defined threshold, the cost falls below a defined threshold, or λ reaches a maximum value.

The size of J is $k \text{ x } 6*m$, where k is the number of pairs and m is the number of poses. The size of the inverse in (5) is therefore $6*m \text{ x } 6*m$. In standard BA this inverse would be $(6*m+3*n) \text{ x } (6*m+3*n)$, where n is the number of features. While the sparse nature of the matrix can be utilized in the inversion, this still represents a very significant reduction in the complexity of the minimization problem.

POTENTIAL PROBLEMS

It is well known that in BA the scale of the scene cannot be determined without some sort of external measure. Typically this is handled in commercial packages with ground control points. Ground control points are features with fixed 3D positions. It can also be handled by providing additional cost elements derived from the deviation of the estimated camera positions with the initial positions. This is useful if the accuracy of the initial position estimates is good. These scaling problems are an issue in both standard BA and with our alternate cost function. However in our alternative algorithm there is more of a concern because the size of the cost function is reduced simply by reducing the size of the scene and because it provides a few less constraints when features are seen in more than two cameras.

Figure 4 demonstrates a scaling issue that occurs when all of the true camera positions are collinear or nearly collinear. In this figure, A, B and C are three camera locations and a, b, c and d are feature locations. It is assumed that the scale between A and B is fixed and that the ray intersections from A and B occur at/near the features. It is also assumed that the true position for C is along the line of A and B. If the camera pose for C is oriented correctly then it may slide along the line of the cameras with zero cost. The rays from A and B for a particular feature form a plain. If C is collinear with A and B then C's ray will also lie in the same plane provided that C is oriented correctly. It will therefore intersect the rays from A and B, providing zero cost irrespective of C's position along the true line of the cameras. This is true for all features. In standard Bundle Adjustment if the scale were fixed between A and B (with GCP for example) this would propagate to C if there were features common to all three, because the 3D feature location is used to position C. However, this is not true without features seen by all three cameras. Standard BA does provide a little more constraint.

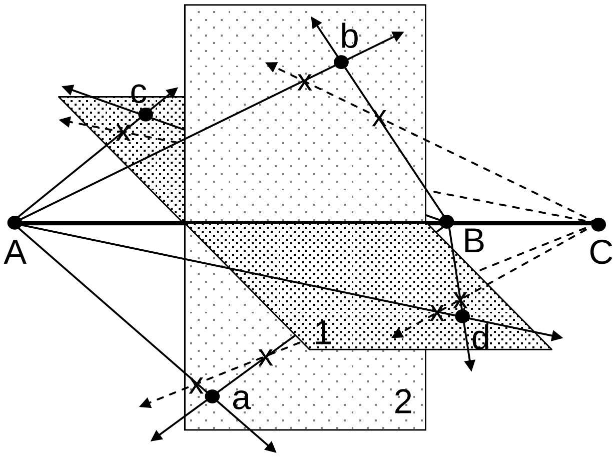

Figure 4. Problem with collinear camera positions

The solution to this problem is to provide scale from external measures such as control points or weighting the movement of the camera positions from the good estimates provided by the GPS in the avionics. It is also beneficial to avoid the use of a single line of images, instead using images from overlapping transects. In the next section we will show results generated with no control points, but the issue has been addressed using the last two methods discussed above.

RESULTS WITH AERIAL PHOTOGRAPHY

This section uses 11 overlapping photos from the Konza Prairie to demonstrate the practical viability of the alternative cost function. The DTM shown in Figure 5 was created by averaging the 3D locations of the tie points used in the minimization. It is fairly ragged. This is partially true because current efforts have mostly focused on the development and analysis of the proposed BA algorithm. Surface fitting and visualization have only been studied in a limited manner. The above DTM was created by generating a grid of points in the North and East directions and calculating an elevation for each point on the grid based upon the average of the nearest tie points used in the minimization. Commercial packages typically use a second pass algorithm for dense feature extraction to fill in the gaps between the sparsely distributed tie points. The crude method has created a model with acceptable resolution in some locations and stair stepping in others.

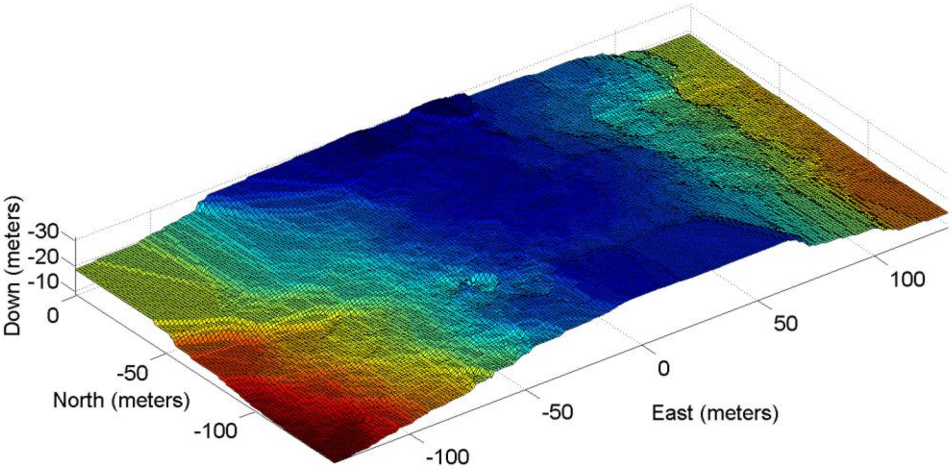

Figure 5. Generated DTM based upon global feature locations

Figure 6 shows an otho-rectified image generated using the DTM in Figure 5. It is a composite image based upon all 11 photos projected onto the DTM. A corresponding pixel value from the camera pointing most directly at the DTM location is assigned to that location. This ensures that the pixel values closer to the center of the image dominate over pixel values on the edge of another image for any point on the DTM. The black sections are areas of the DTM not contained in any image's perspective. The road and ravines are the most notable objects for continuity. Each traverses multiple image segments with very little misalignment. The lines between image segments can be seen by color changes caused by illumination differences.

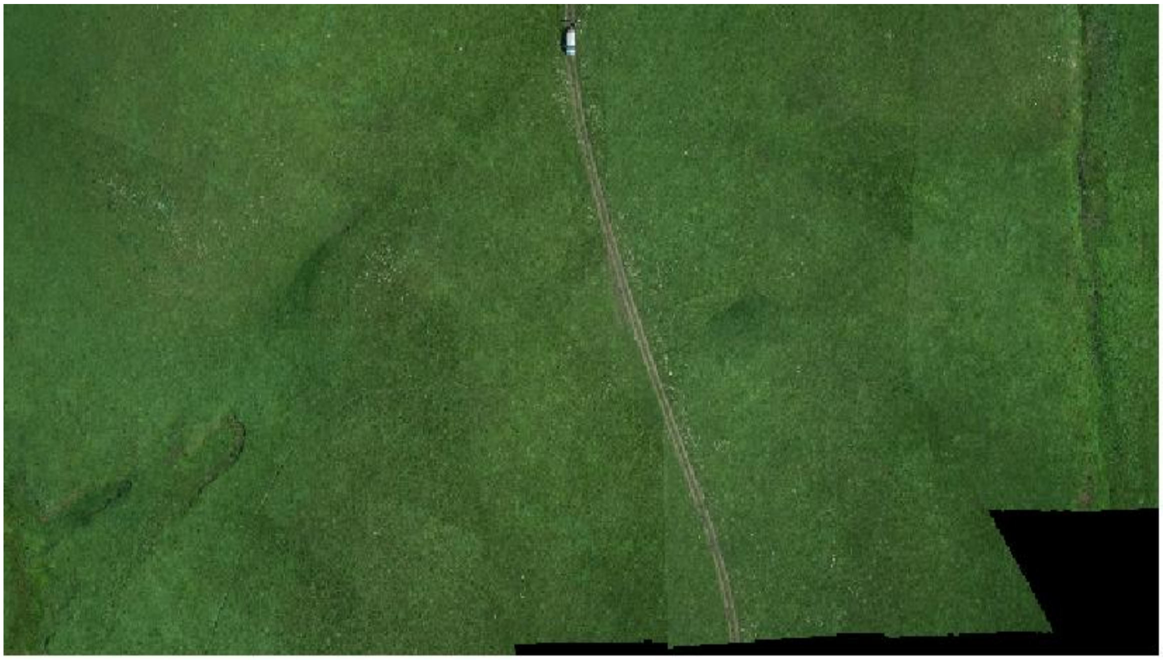

Figure. 6 Ortho-rectified image based upon camera perspective and the generated DTM

CONCLUSIONS

An alternate cost function for BA has been presented and demonstrated to be capable of reconstructing terrain from photographic surveys. The original motivation for exploring the use of this alternative formulation was that commercial photogrammetric software was unable to converge on a solution using the navigation data from a small UAV as the initial estimates of camera poses for the minimization algorithm. We found that our alternative formulation did however converge to a solution, and could in fact be used to seed the commercial software which would then converge to a similar solution. This led the authors to believe that the reduced search space in the alternate formulation resulted in a more robust formulation of BA. However, subsequent studies using the same minimization algorithm (a LM implementation) with the different cost functions do not support this. In fact the convergence properties seem to very similar. While this alternate formulation provides insight about BA, the standard formulation does not suffer from the problem rewarding a reduced scale for the scene. Therefore it is probably better suited for most applications.

REFERENCES

[1] M. Lourakis and A. Argyros, “The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm,” CVRL ICS, Heraklio, Crete, Tech. Report FORTH-ICS/TR-340, Aug. 2004.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 60, 2 (2004), pp. 91-110

[3] D. Marquardt "An Algorithm for Least-Squares Estimation of Nonlinear Parameters." SIAM J. Appl. Math. 11, 431-441, 1963.

[4] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, “Bundle adjustment - a modern synthesis,” Springer-Verlag, 2000.

[5] J. Heikkilä and O. Silvén. “A Four-step Camera Calibration Procedure with Implicit Image Correction,” University of Oulu, Finland.

[6] K. Madsen, H. Nielsen, and O. Tingleff, “Methods for Non-Linear Least Squares Problems,” Technical University of Denmark, April 2004.

[7] J. Heikkilä. “Geometric Camera Calibration Using Circular Control Points,” University of Oulu, Finland.

[8] R Tsai, “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses,” IEEE: Journal of Robotics and Automation, Vol. RA-3, NO. 4, Aug. 1987