

Conflation System MBP

Richard D. Boyce, Richard Chase
Computer Science Department
Central Washington University
Ellensburg, WA, U.S.A.

Abstract *MBPConflate is proof-of-concept software developed to aid in conflation research. The program implements a powerful algorithm for automatic multi-source image conflation that utilizes new Algebraic Systems-based similarity metrics during the feature comparison process. One benefit of integrating Algebraic Systems into the feature comparison process is that more global shape properties can be discovered, preserved, and used for matching. The algorithm presented in this paper clusters the individual attributes from two data sets based on connectivity and neighboring features. The clustered sets are then compared using a hierarchical rule-based technique that includes measures of structural similarity. User selected criteria define a match between features and the user assists feature matching only when ambiguity cannot be resolved automatically. Experiments have shown that the algorithm is computationally efficient and robust.*

Keywords: Linear Features, Conflation, Algebraic Systems, Feature Space, Search Space, Similarity Metric

1 Introduction

Conflation is a process that combines data from two or more images of the same region into a single image.[7] *Multi-Source Image Conflation* involves conflation of data from a combination of multiple sources such as maps, satellite, imagery, aerial photography, or Geographical Information System (GIS) layers. A variety of local and government agencies require powerful geodata conflation tools to efficiently and accurately merge data from many

sources. Their goal is to maximize the accuracy of conflation while minimizing human involvement. Successful conflation systems have been implemented for highly specific applications [6][8]. Yet, for the general conflation task, the available technology requires considerable user interaction[5].

In general conflation breaks down into an iterative three-step process [7]:

1. Find features that match.
2. Re-orient component maps so as to superimpose features (often referred to as registration[1]).
3. Deconflict the attributes belonging to identified feature matches.

In finding features that match many complications may arise. For example:

- the definition of a matching feature may vary depending upon the application and dataset
- feature coordinates may overlap each other yet represent entirely different entities (i.e a road that covers utilities)
- the images may differ in resolution, shape type, and the relative quantity of features represented.

Thus, the metrics used to determine similarity can play a pivotal role in the quality and efficiency of the overall conflation process. A new similarity metric using Algebraic Systems [2][3][5] has been developed which fully

preserves the shape of a feature, does not depend upon coordinate systems and is invariant to scale and rotation. The use of Algebraic Systems to abstract spatial features enables more global shape properties to be discovered, preserved, and used for matching. This paper describes an algorithm and proof-of-concept software (MBPConflate) that integrates Algebraic Systems into the conflation process. The conflation algorithm:

- Incorporates a structural similarity metric based on Algebraic Systems in feature analysis.
- Maintains feature attributes and the z-coordinate throughout processing.
- Creates metadata on and informs the user of 1:1, 1:N, and N:N feature matches.
- Incorporates user selected rules, geometric measures, neighborhood characteristics, and attribute values as well as estimates of significance and certainty to feature matching.
- Allows user to view best matches, uncertain matches, as well as metadata on each match.

In the next section we define the terminology and follow with a detailed description of the algorithm. We then discuss the design, implementation, and effectiveness of the proof-of-concept software, MBPConflate. Finally we summarize the results of the system’s design.

2 Definitions

Deconfliction is “a process in which contradictions in a matching pair’s attributes and/or values are resolved.” [7]

A *point*, p , is an ordered triple $p = (x, y, z)$

A *feature* is defined as a data structure consisting of a continuous series of points and zero or more describing attributes. Let $X = \{X_i\}(i = 1, 2, \dots)$ represent the set of features from the an image. A point in the feature X_i is then represented by p_{ij} .

Let A_i represent a feature from the set of features A . Similarly, let B_i represent a feature from the set of features in B .

A *reference feature*, X'_i , is a feature formed by connecting features in X that have at least one set of common end points and that satisfy a Boolean function $D(a_i)$ regarding their angle of connectivity. We define the set of reference features formed by applying $D(a_i)$ to A and B be A' and B' respectively.

Let $\lambda = \{\lambda_i\}, (i = 1, 2, \dots)$ represent the set of *matching rules*; a set of rules that clearly defines the parameters necessary for a series of points from feature X_i to form a “matching section” with a series of point from feature Y_i .

A *subfeature* is a feature formed from a continuous series of points belonging to a feature X_i .

A *matching section*, M_i , is a set of two subfeatures, X'_i and Y'_i , from features X_i and Y_i such that:

$$\exists \lambda_i \in \lambda : \lambda_i(X'_i, Y'_i) = true$$

A *confidence score* is defined as a scalar value that conveys the degree of similarity possessed by a matching section. A partial list of potential similarity metrics include the weighted average of the number of rules satisfied, proportion of feature area covered by match, level of structural similarity (see [3] for explanation).

The *level of structural similarity*, n , is the level at which $S_a(n) = S_b(n)$ where S_a and S_b are the structural measures of two structured polylines as defined in [3] and n is a positive integer. Two structured polylines are said to be *n-similar* if, for a given integer n , $S_a(n) = S_b(n)$.

A *match record* is a data structure that contains information about a matching segment including the features and points involved in the match. The match record is used to save memory throughout processing.

Let the *bounding cube* of an image or feature be a cube formed using the minimum and maximum x, y, z values of all points within the image or feature. We denote these values min_x, min_y, min_z and max_x, max_y, max_z .

Let the *range*, r , of an image be defined as $r = \max(|\max_x - \min_x|, |\max_y - \min_y|, |\max_z - \min_z|)$.

Unitization involves placing additional points within features using every unit distance along the arclength of a feature. The unit is smaller or equal than the smallest interval in each of two input images. This is done so both images being processed can be analyzed at a similar resolution. For this data we set up the unit as follows: $\text{unit} = r/U$ where U is a scalar integer greater than zero.

3 The Conflation Algorithm

Step one: Let A represent the set of features from the an image, ImA , and B represent the set of features from a different image, ImB . Using the features in A and B , reference features are formed which are then unitized using a common scalar unit so as to describe contours at a finer resolution. This process introduces artificial points to the features which are flagged as such. Let the unitized set of reference features formed from the spatial coordinates of A and B be A' and B' respectively. The new reference features in A' and B' are sorted in non-increasing order by the areas of their bounding cube.

Step two: The algorithm selects a reference feature from A' , A_i . A subset of the reference features in B' whose bounding cubes overlap the bounding cube of A_i the are then searched for those that contain at least one original feature that can form an n -similar structure with an original feature in A_i . The value n is user selected. In the worst case, every reference feature in B' will be checked for matching sections. When a matching segment is found a new match record is initialized with information on the points involved in the match a confidence score. This match record is added to the set of match records generated by this comparison.

Step three: After all matching segments have been located, the set of match records is analyzed for any matching sections that satisfy all

rules in the set of matching rules, λ . If there exists one or more features satisfying all rules, the match with the highest confidence score is chosen and two new features, A'_{matchi} and B'_{matchi} , are created from their points. A'_{matchi} and B'_{matchi} are removed from A' and B' respectively and are placed in the set of matching features M_i . The leftover segments of A'_i and B'_i are used to form new features that replace A_i and B_i in the set of all features being analyzed.

One of the most difficult challenges is to define exactly when two features are said to match. Ambiguity occurs where features are clustered close together or multiple features cross each other multiple times. Employing user selected rules and metrics for judging similarity can decrease the need for user interaction and speed up the conflation process. The algorithm currently incorporates three separate measures of similarity and allows the user to place weights upon their importance:

1. The total coverage of a feature whose points fall within a static threshold when the points are compared with a feature from the base conflation image considered the base of conflation.
2. The percentage of rules that were satisfied during similarity analysis.
3. The level of structural similarity between the two images (see [3]).

Other similarity measures will be incorporated as they are developed.

The application of the similarity metrics is constrained by the choice of rules the user wishes to apply to the conflation task. For maximum flexibility the algorithm allows the user to select from a list of potential rules. For example, the following set of potential rules is currently available:

1. Two features should be examined by the expert if less than X% of their points are within two units of each other.
2. Two features qualify as potential matches if greater than X% of their points are

within two units of each other and there are no other features that have points within the threshold.

3. Two features qualify as potential matches if they are both stable structures as defined in [3] and the level of structural similarity is greater than n .

If no matching section satisfies all of the rules in λ , all features referenced by the set of match records are placed in the set of ambiguous matching features, Q , for user-assisted deconfliction. Steps two and three are repeated iteratively until no further matching sections can be found in A' and B' . The result at this stage is four sets realized as six arrays:

- a) A' , containing only the unique portions of features from A
- b) B' , containing only the unique portions of features from B
- c) $M_{A'}$, the subset M containing the components of the matching segments from ImA's reference features ordered as found.
- d) $M_{B'}$, the subset M containing the matching segments from ImB's reference features ordered as found.
- e) Q_A , the subset Q containing ambiguous matching segments from A'
- f) Q_B , the subset Q containing ambiguous matching segments from B'

Step four: User-assisted deconfliction of all ambiguous matches proceeds via the user interface. The deconfliction process allows the user to resolve ambiguous matches manually using visual and metadata generated by the algorithm. Similar to step two, matching segments are removed from the search space and remaining segments replace the original features. Upon completion of the deconfliction process, conflation proceeds from step two. Thus, the algorithm may iterate through steps two through four several times until only sets A' , B' , and M remain.

These three sets contain within them sufficient information to demonstrate the structural similarity of conglomerate features formed in step one. They also contain sufficient data to determine the relationship of original features in A to B . The original attributes associated with each spatial coordinate in A and B are maintained throughout processing so that feature attributes can be used in conflation. However, the algorithm does not utilize feature attributes in the conflation process at this time.

4 The Proof-of-Concept: MBPConflate

4.1 Purpose of the Software

As mentioned previously, MBPConflate is proof-of-concept software written to aid in conflation research. MBPConflate provides the following functionality:

- automatic image registration of images including raster images that meet the preconditions of an included raster to vector conversion algorithm.
- automatic image conflation of vector images or raster images converted to vector format.
- tools that assist validation of new conflation similarity metrics.
- easy integration of new conflation techniques as they are developed.
- portability to other operating systems.

4.2 Software Structure

Due to its experimental nature, the development of robust image conflation algorithms is an incremental process. Thus, we designed the software so that, as improved techniques in any of the three processing portions of conflation became available, they could be quickly integrated into MBPConflate. The core algorithms

used in MBPConflate occupy the Model portion of a Model-View-Controller architecture. The model portion is written in ANSI C/C++ and is portable to other operating systems. It is separate from the controller portion, which was written using MFC 7.0. We designed the Model portion of MBPConflate to be easily updated as better techniques emerge.

Figure 1 shows a class diagram of the model portion of MBPConflate. The conflation algorithm is implemented by the *conflation_control*, *preprocessing*, *correlation*, and *conflation* classes. The MBPConflate model includes the classes involved in the conflation task as well as the supporting classes, *vector_image*, *line*, and *match_record*.

The *conflation_control* class serves as both an interface and a controller for the conflation process. The *preprocessing* class contains methods for creating reference features and unitization. Future classes derived from the *preprocessing* class may add preprocessing of feature attributes to aid in conflation.

Image registration is performed through the *correlation* class and its methods while the *conflation* class and its methods identify matches in both original and reference features. The *conflation* class works in conjunction with the *match_record* to build the arrays of matching and unique features mentioned in the description of the conflation algorithm.

Currently MBPConflate supports ESRI Shape Files as input but other file formats can be added with a small amount of effort. A new class can be written that inherits from the *vector_image* class and adds the ability to load images from different sources file formats. The base *vector_image* class can load raster images preprocessed to contain only features of interest, ESRI Shape Files, and to text file formats that we have written for research purposes.

4.3 User Interface

The user interface was written using OpenGL and MFC©. Figure 2 is a screenshot of the user interface showing two vector images to be conflated. The MBPConflate user interface en-

ables the user to perform the following tasks via mouse clicks or keyboard shortcuts:

- view, manipulate, correlate and conflate image files
- view visual and textual results of conflation
- ascertain the quality of conflation by viewing best matches, uncertain matches, and metadata on matches
- store or print the results of conflation for later viewing

5 Summary

MBPConflate has been used to conflate images from both raster and vector formats. In one test, a vector image consisting of 1497 features was conflated to another image containing 407 features. The program easily identified the matching and unique segments in both images in a matter of seconds. Conflation was validated by careful inspection of the visual and textual output of the program. Text output from MBPConflate links each feature with its attributes. We have used information gathered from comparing the output of MBPConflate with the results of human assisted conflation to develop more robust techniques. The design of MBPConflate has made integrating new techniques efficient. The use of structure as a similarity metric has improved the efficiency of the conflation algorithms as well as broadened the usefulness of the program.

6 Acknowledgements

This research has been supported by a University Research Initiative grant from the National Imagery and Mapping Agency. We would like to thank E. Frank, and M. Kovalerchuk for programming and computational support and S. Sento and P. Brennan for providing data.

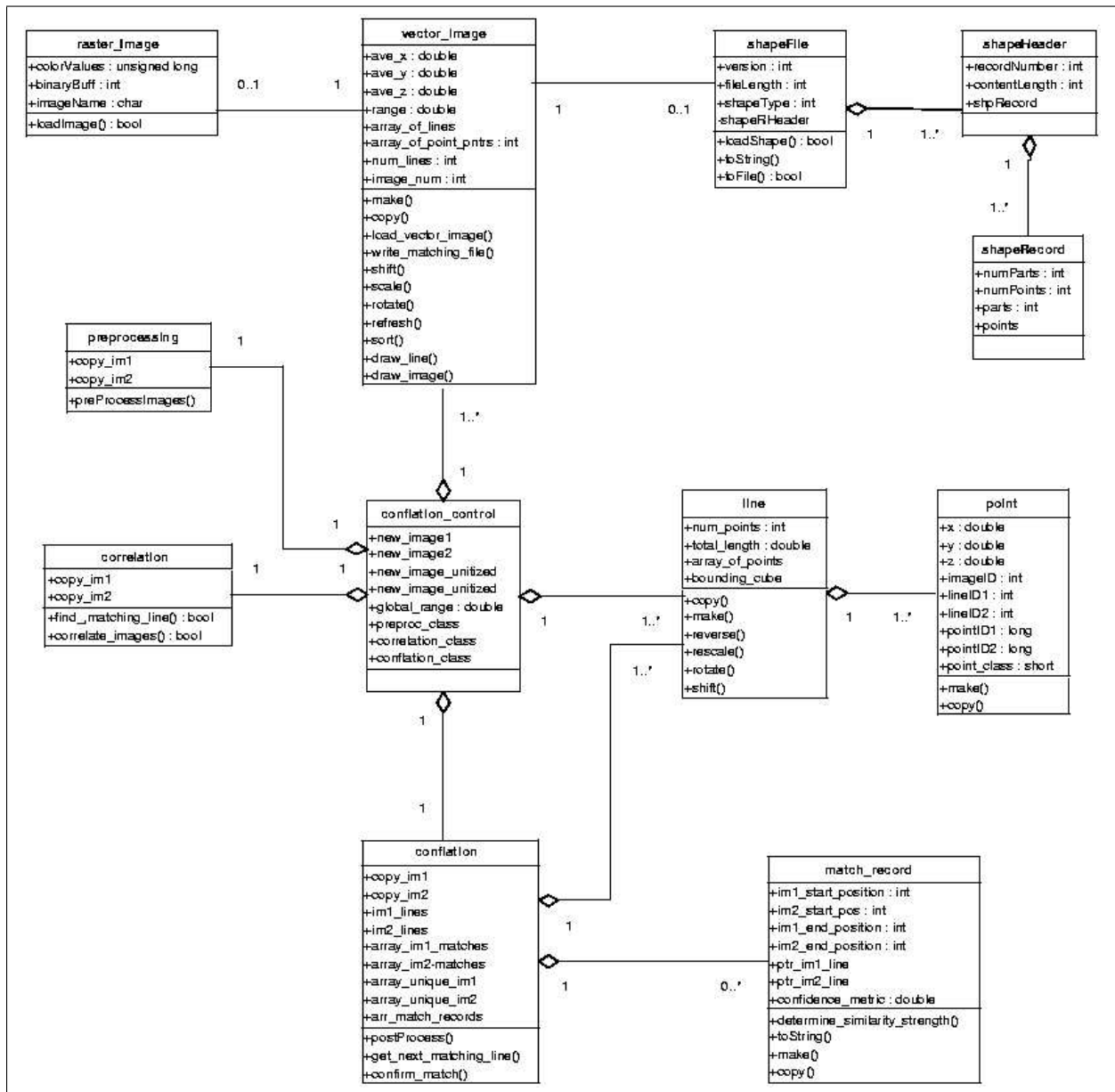


Figure 1: UML Diagram of System

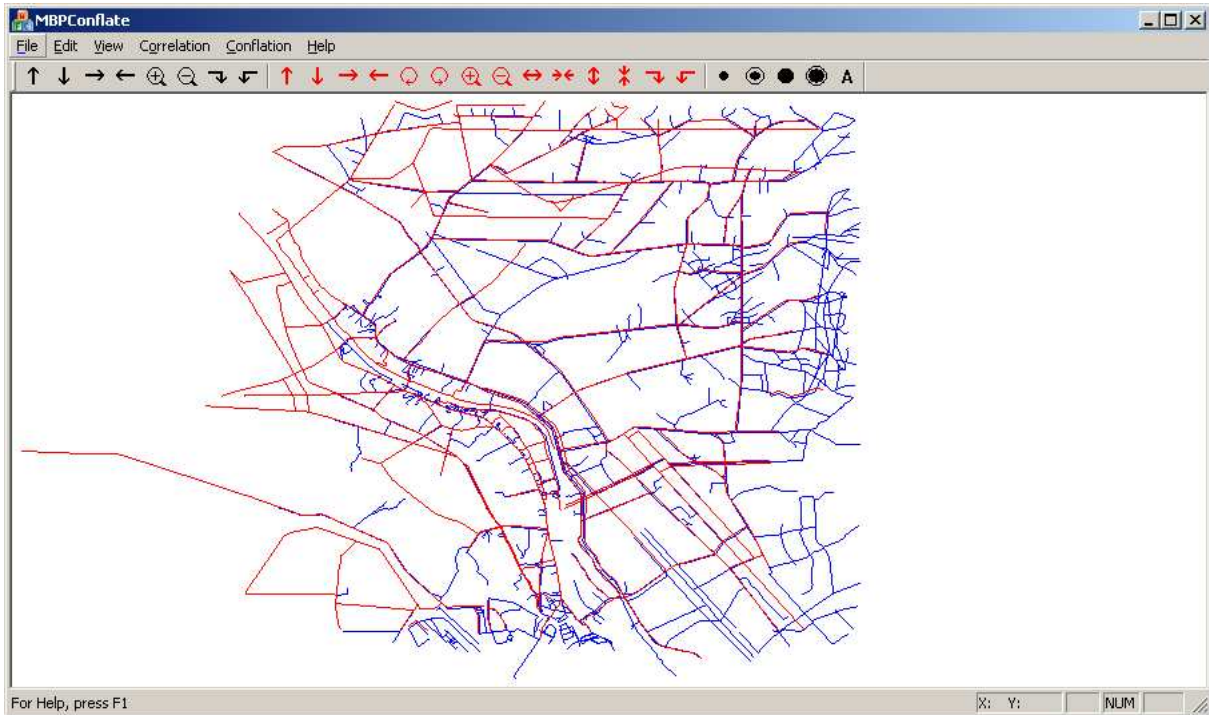


Figure 2: MBPConflate User Interface

References

- [1] Brown, L. *A Survey of Image Registration Techniques*. ACM Computing Surveys. Vol 24 (1992):325-376.
- [2] Kovalerchuk, B. Shcwing, J. Algebraic Relational Approach for Geospatial Feature Correlation. in: *Proceedings of International Conference on Imaging Science, Systems, and Technology (CISST'2002, June 24-27, 2002) Las Vegas, pp. 115-121, 2002*.
<ftp://ftp.geology.cwu.edu/pub/sumner/method.pdf>
- [3] Kovalerchuk, B. Sumner, W. Schwing, J. Image Registration and Conflation Based on Structural Characteristics CISST 2003. Las Vegas, 2003. (This volume)
- [4] Bern, M., et al. Emerging Challenges in Computational Topology, 1999
<http://citeseer.nj.nec.com/bern99emerging.html>
- [5] Kovalerchuk, B. Sumner, W. Algebraic Relational Approach to Conflating Images in: *SPIE International Conference AeroSense 2003*. SPIE 2003 (in print)
- [6] Cento, S. Automated Conflation and Update of Geospatial Feature Data Bases, 1997.
<http://gis.esri.com/library/userconf/proc97/proc97/abstract/a471.htm>
- [7] Cobb, M. Rahimi, S. Ali, D. Paprzycki, M. Petry, F. A Knowledge-Based Multi-Agent System for Geospatial Data Conflation in: *Journal of Geographic Information and Decision Analysis 2002*. Vol. 6, No. 2, pp 67-81
- [8] Saalfeld, A. Conflation: Automated Map Compilation in: *International Journal of GIS*. 2(3):217-228. 1988.