INTEGRATION OF PHOTOGRAMMETRIC AND SPATIAL INFORMATION SYSTEMS

Daniel EDWARDS*, Justin SIMPSON*, Peter WOODSFORD**
*** US Army TEC, Virginia, USA**
Daniel.L.Edwards@tec02.usace.army.mil, Justin.R.Simpson@tec02.usace.army.mil
****Laser-Scan Ltd., Cambridge, UK**
peterw@lsl.co.uk

KEY WORDS: GIS, Integration, Object-oriented, Photogrammetry, Topology

ABSTRACT

The technologies of Digital Photogrammetric Workstations (DPW) and Geographic Information Systems (GIS) have traditionally developed separately. The paper describes the integration of a DPW with an Object-Oriented GIS, and details the resulting synergies and efficiency gains. A flowline of seven steps is reduced to three. An implementation of 2.5D topology (3D geometries, with topological relationships in the projected z=0 plane) is described with examples of the utility of this approach.

The principles of such an integration are detailed with an analysis of the roles of the DPW and GIS. The mode of operation for feature extraction and edit is described, together with the benefits of real-time topology and verification in collecting and maintaining 3D linear and area networks. The model includes the representation of such man-made features as road overpasses and tunnels. The Object-Orientated GIS provides a strong framework for knowledge-based validation methods. The paper concludes with some directions for future developments.

1 INTRODUCTION

During the 1990's, two complementary technologies, Geographic (or Spatial) Information Systems (GIS) and Digital Photogrammetric Workstations (DPW), have risen to a level of prominence. A DPW produces spatial data that needs to be managed. A GIS manages, but needs spatial data. Despite their natural co-dependence, these two technologies have developed independently and, to date, largely remain separate.

Currently, within a DPW, 3D features are delineated and attributed using interactive, non-topological digitising software. In contrast, the current state of GIS technology is largely based on topological storage, access, and manipulation of 2-D feature data and attributes. We use the term 2.5D for models in which coordinates are 3D, but topological relationships are enforced on the projection of the geometries in the z=0 plane. The 2.5D model allows for multiple z-values (unders and overs) although some data products (e.g. DIGEST VPF) cannot accommodate to these.

The current method for mapping topological features from stereo imagery and subsequent storage involves the following steps:

1) Input, rigorous registration, and block adjustment of aerial imagery at a DPW;
2) Creation of compatible feature schemas on both DPW and GIS;
3) The interactive 2.5D (X, Y & Z) delineation of spatial features at a DPW by an image analyst through either:
   a) Digitisation of spaghetti edges (linear features) or spaghetti edges with label points (areal features); or
   b) Digitisation of linear edges with explicit placement of nodes (linear features) or explicit multiple delineation of shared boundaries (areal features);
4) Export of the 2.5D feature dataset to a GIS, usually via conversion to a neutral format, and then import to the GIS where the 2D topology is built. Both the elevation coordinate (Z) and DPW feature identification (FID) is usually *lost or ignored* during this process;
5) Errors are detected in the topological build are sent back to the DPW for correction (usually without a compatible or useful FID);
6) Errors must be manually identified and corrected at a DPW; and
7) Repetition of previous steps 3 through 6 until the dataset is successfully built and topologically correct in 2D.

Clearly these seven steps are not only cumbersome and inefficient, but are also symptomatic of much greater problems.

The separation of the DPW compilation process from the GIS analysis process is a flawed model of knowledge acquisition, validation, and utilisation. Creating knowledge from data not only produces new knowledge, but also new data. Various tools available within a GIS could be used to assist an analyst in compiling new data. Terrain analysis based on physical knowledge and localised rules can serve an important role during feature compilation. Topological analysis, editing, and final validation must take place within the stereo collection environment.

The emergence of object-oriented technology further underscores the flaws of the current model. Objects and methods are needed during both compilation and analysis. Methods, running concurrently with compilation, can be an important tool in the collection and validation of both spatial and semantic data. Methods can be used to generate both new data and new knowledge.

The authors propose a tight integration of a DPW and a GIS that will reduce the current seven steps to three. Unique strengths of each system are united and redundant functions are eliminated.

Definition of the roles and responsibilities of the DPW and GIS is necessary if they are to engage in a co-operative venture. This is a risky business – drawing lines in the sand that can be quickly over-washed by changes in technology. Nevertheless, duplication of services can be consolidated. The division of labour will be discussed later in the paper.

Traditional GIS technology must evolve before a DPW and a GIS can work in harmony. Changes are required in the spatial, topological, and semantic models of data. The spatial extensions force an evolution in the topological models, and these, in turn, must be integrated into future representations of semantic information.

In the spatial domain, features must be represented in three dimensions. For editing, validation, and analysis what is stored in the GIS must be identical to what is displayed at the DPW. Three-dimensional spatial representation requires an evolution of the topological model. Traditionally, topology is built after features are projected to the plane Z=0. If features are collected and stored in a three-dimensional format, then three-dimensional adjacencies and intersections should be represented in the topological model. Ideally a full three-dimensional topology is needed to capture such man-made features as bridges, tunnels, and buildings.

Today semantic information is simplistic, consisting of feature code and attribution. In the future, feature creation and validation must use feature-specific knowledge that is tightly coupled with 3D geometry and topology.

2 THE GIS-DPW INTEGRATION

The GIS and DPW are two mature technologies that have developed independently and that show no sign of merging [Edwards, D. and Simpson, J.2000]. Given the high cost of merging two mature technologies into one, it seems most efficient for the GIS and DPW to be integrated by having two separate systems that communicate through an interface. To define a standard interface in this paper is hasty, since experience gained through future integration is needed before an interface is properly defined. Here we hope to discuss the first syllables of the dialogue, which should begin as efficiently as possible.

In this paper we shall discuss the integration through an interface, outlining the roles and responsibilities of the GIS and the DPW and discussing the requirements of the interface. These guiding principles will be used to organise the integration of the two systems:

- Functions should reside where needed;
- Use the interface for simple things; and
- Minimise commercial disruption.

As a starting premise, either the GIS or the DPW should act as the manager, communicating through the application-programming interface (API) of the other. Since each system should be expected to have both an API and managerial software, which should be the manager?

If the DPW acts as manager, continuity of operations is preserved. Certain functions such as project definition, source selection, image import, and image registration must be performed on a DPW before data are collected. Keeping the DPW in control through the collection and verification stage reduces the operator's dependence on a second user interface. However, this communication would require major changes to the user interface of the DPW. For example, a DPW would need a new suite of menus in order to invoke a GIS query from a DPW. A DPW would have to develop the look and feel of a GIS.

If the GIS were manager, minor changes are needed in the user interface, since the GIS already supports data input, edit, verification, and query. The use of the API is simple - largely being a response to DPW mouse events and retrieving X, Y, and Z. Access to both objects and methods of the GIS is direct. The GIS contains managerial software, and it is in the GIS where management is needed. A disadvantage is that the user must learn a second user interface.

While either design is possible, letting the GIS act as manager, while communicating through the API of the DPW, minimises disruption to the both the DPW and GIS manufacturers. This basic architectural relationship will be developed in this paper.

2.1 GIS-DPW major functions

Before discussing the division of responsibilities, let us list the major functions that a GIS-DPW should perform. Then we shall discuss these functions in light of how they change in the merger between the GIS and the DPW.

	Function	DPW	GIS
1	Source selection	X	
2	Project definition	X	X
3	Schema definition		X
4	Image input and photogrammetric registration	X	
5	Elevation extraction	X	
6	Elevation editing	X	
7	GIS-DPW connection	X	X
8	Communication through an API	X	X
9	Feature extraction and edit in a stereo environment	X	X
10	Look and feel of a stereo environment and seamless functionality		X
11	Real-time topological verification		X
12	Rule-based feature verification		X
13	Display and verification of features over imagery	X	
14	Access to query and other GIS functions		X
15	DPW in stand-alone mode interface; and	X	
16	GIS in stand-alone mode		X

2.2 DPW-unique functions

Certain DPW functions, which precede feature extraction in a flow of work, do not require a connection with a GIS. These functions should remain unchanged. Source selection, project definition, image input and registration, elevation extraction and editing, orthophoto creation, and verification of features over imagery are examples of DPW functions that are not affected by the GIS-DPW union. If a DPW is acting in stand-alone mode, it must be able to define its own collection schema.

2.3 GIS project definition

Inevitably, the GIS and the DPW need similar, but not identical, parameters in order to define the project. To reduce duplication, the GIS should read the project definition of the DPW through the API, in order to extract as much commonly needed information as possible. The shared information should include image dates, image source, security classification, projection, horizontal datum, vertical reference surface, DEM footprint, DEM size, DEM spacing, image centre, image pixel size and units, image rows and columns.

2.4 GIS schema definition

Both systems can develop their own feature schema. However, since the GIS will ultimately store the features, it is natural that the schema definition takes place within the GIS to best utilise existing resources. The use of objects and inheritance needs to be expanded in the schema definition to handle 3D features.

2.5 GIS-DPW connection and High Level API

The GIS initiates a connection with the DPW through its Graphical User Interface (GUI). This connection enables the GIS to communicate with the DPW through its API. Through the API the GIS must be able to read the project definition file of the DPW and establish the space in which the DPW returns coordinates.

2.6 Low Level DPW API

The DPW API should provide the following generic functions (based on the L-H Systems SOCET SET/Laser-Scan LAMPS2 integration [L-H Systems, 2000;Laser-Scan, 2000]):

- ability to detect mouse button and keyboard events
- requests to draw the screen (redraw/zoom/pan)
- plotting of 3D points and lines in various styles, colours, sizes and widths
- support for rubber band lines (one and two anchor points)
- ability to read and set cursor position, and toggle cursor between systems
- ability to get window extents

2.7 DPW in stand-alone mode

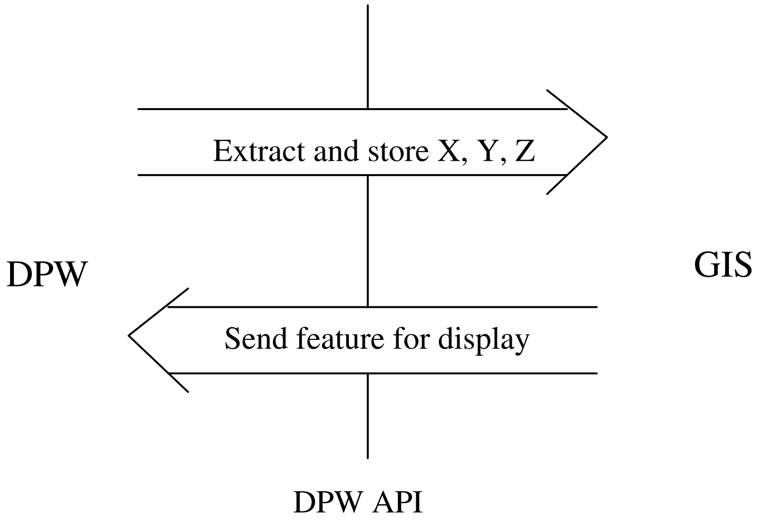
An operator, who is not extracting features, should be able to use the DPW in stand-alone mode. Typical functions that are well suited to stand-alone mode are source selection, project definition, image input and registration, DEM creation, DEM editing, orthophoto creation, and viewing features over elevations.

2.8 GIS in standalone 2D and 3D mode

It is desirable that the user interface for the GIS in 3D feature extraction mode should be consistent with the 2D standalone case. Operators are increasingly likely to rotate between feature extraction/update from stereo imagery and from other sources, so consistency of interface is needed.

3 FEATURE EXTRACTION AND EDIT IN THE STEREO ENVIRONMENT

The full suite of 2D digitising tools of the GIS should be extended so that they operate seamlessly in the stereo collection environment. The DPW is used as a 3D digitiser that performs the rigorous conversion from image coordinates to ground coordinates. Within the GIS the existing feature types of point, line, and area need to be expanded to store a Z with each X and Y. In addition, a new feature type, volume, needs to be added to the above list and to be managed. The operator collects features, being unaware of the interplay between the GIS and DPW. The following diagram illustrates the transparency of the connection between the DPW and the GIS.



In this integration, the seven steps of feature mapping, listed in the introduction, are reduced to the following three steps:

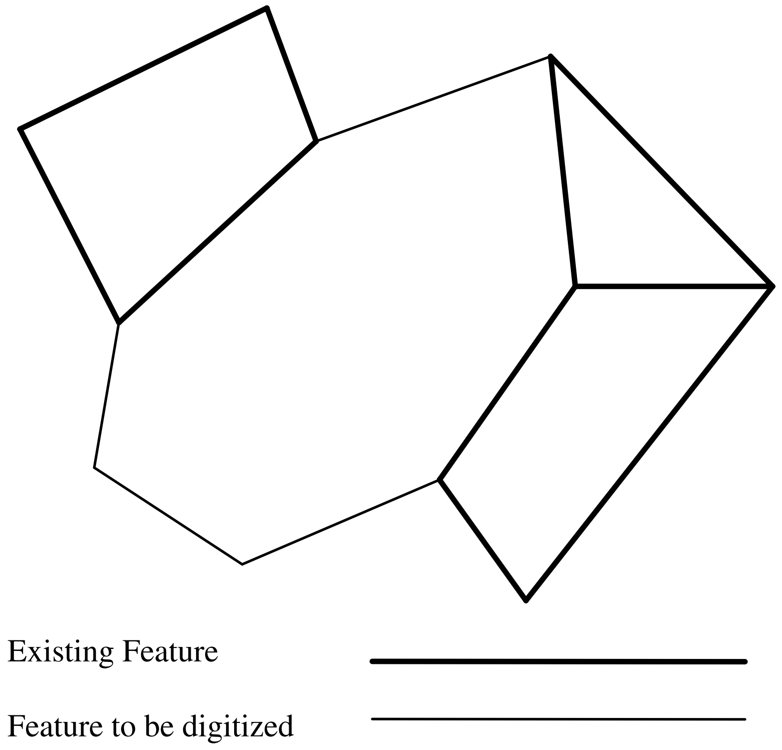
1) Input, rigorous registration, and block adjustment of aerial imagery at a DPW;
2) Creation of compatible feature schemas on both DPW and GIS; and
3) Digitise features with automatic placement of both nodes and shared boundaries.

Topology is built as features are collected, and errors are reported as they are detected.

To illustrate the seamless integration of these systems, consider the collection of linear and areal 2.5D features. For linear features, the operator selects the feature in the GIS environment. Next the operator selects the feature type of “line”. After this second mouse click, control is passed to the DPW stereo screen where the operator digitises the feature, with the feature being drawn over the imagery as it is collected. As the feature is mapped, nodes are automatically placed. Intersections of 3D features are detected automatically. Other methods are called for the purposes of verification and inference. Errors are flagged. Another interesting possibility is that a gridded urban road network can be digitised in a serpentine fashion, with a minimal number of mouse clicks, since nodes are automatically created.

For areal features, the operator selects the feature in the GIS environment. Next the operator selects the feature type of “area”. Again, after this second mouse click, control is passed to the DPW stereo screen where the operator digitises the feature with the feature being drawn over the imagery as it is collected. When the operator terminates the feature, the feature is automatically closed. Methods are invoked and errors are flagged.

An important tool in areal feature collection is the sharing of common boundaries with features that are already digitised. This automatic tool increases efficiency since is very difficult for an operator to re-digitise part of a boundary in 3D. Automatic shared boundary creation is illustrated below:



4 REAL-TIME TOPOLOGY

At the present time, major GIS vendors have not modeled three-dimensional topological relationships. The utility of having a 3D topological model of terrain features must be balanced against the reality of today's proven topological models. To achieve 3D representation of terrain features, GIS vendors need to improve current topological models, but vendors need to enhance tools for both feature collection and feature modelling. For example, it is useful to model the trunk of a tree, even if only its canopy is visible in imagery. Likewise, there is growing interest in modelling interiors of buildings.

The importance of representing features must be considered in light of the cost of developing new software. In due course, the market place will resolve today's blurry distinctions between dreams and requirements. In the meantime, we identify a representative sample of what is desirable:

- Storage of X, Y, and Z.
- Storage of 3D lines and their intersections.
- Representation of surfaces, whose projection to the X-Y plane is one-to-one, including the adjacencies, intersections and islands contained therein.
- Representation of features that can be modeled by placing a rubber sheet over them. For example, represent the adjacencies between the ground, and faces of a simple building. Similarly represent the adjacencies between the ground, tree trunk, and tree canopy. Represent cliffs.
- Representation of such man-made features as road overpasses and tunnels.
- Full 3D representation.

5 VERIFICATION

Real-time feature verification in the stereo editing environment is one of the chief advantages of the union between a DPW and a GIS. A general treatment of the benefits of the Object lifecycle concept in this regard is to be found in [Woodsford and Arctur, 1996]. Major sub-categories of validation are topology, knowledge-based feature validation, knowledge-based cross-feature validation and knowledge-based feature creation.

Topological checking within the stereo editing environment is central to feature validation. Performing topological validation at collection time, by identifying and correcting a problem within the stereo environment, saves considerable time. For example, when closing an areal feature, an operator can not always position the cursor so that the initial and final values of X, Y, and Z coincide. A topological check can force these two sets of values into agreement. As a practical matter, linear features, which should intersect in 3D, rarely do, due to the inability of an operator to position vertices to insure an exact Z-match at the intersection. The GIS should compute the X and Y values at the intersection as well as interpolate two Z-values. If the interpolated Z-values differ by greater than a predetermined tolerance, then the crossing should be treated as an overpass and the operator should be notified. If the Z-values are within the tolerance, they should be given a common value. Topological primitives such as edges are central to shared boundary validation.

Knowledge-based feature validation uses such tools as query and geometrical descriptors to assist the operator. Methods are particularly useful here. Feature-specific rules, based on the collection specification, can be invoked automatically as soon as a feature is collected. For example, insure that collected rivers flow downhill, and that collected lakes are flat. Only collect forests of a sufficiently large area. Rice paddies do not exist in certain parts of the world; a feature mislabelled as a rice paddy should be flagged at collection time so that an operator can quickly re-examine it in stereo imagery.

In cross-feature validation, topological properties such as intersection and adjacency can point to unnatural relationships. For example, a road passing through a lake should be flagged as a possible error. Two lakes of differing elevations should not be adjacent; either a dam should be present or an elevation is incorrect. Differing sensors will be used for feature mapping. The resulting features from different sensors will not always agree in spatial or thematic content, creating a need to validate features across sources. For example, a river from one source can lie on a steep slope that is derived from another source.

In knowledge-based feature creation, rules are applied to existing features to infer and store new features. For example, a road crossing over a river can be flagged, with a bridge automatically placed at the intersection. An operator can then visually verify that the inferred bridge is valid or, otherwise, correct the error.

6 FUTURE DIRECTIONS

This discussion has been an initial glimpse into the potential of this union. Lessons learned from initial implementation will suggest refinements, while creating demands, unforeseen today, that will suggest future directions to be followed by the joint technology. But from a broader perspective, more general research and development is needed in the following areas (ordered by priority):

1. *Robust OO feature representations* need to be established for the physical world features to be mapped. This involves a solid object-based modelling of the features to be mapped and their cross relationships to each other. The physical world to be mapped needs to be modeled into parent-child relationships and inheritance properties need to be capitalised upon. This effort will likely change what features and attributes are mapped and how they will be collected.
2. *Unique OO compilation specifications* for different standard mapping products need to be developed.
3. *Feature-specific methods (including inference)* that are exclusive to specific feature types need to be developed and tested. Research and development in *spatial data mining* will play a significant role in the explication of feature specific and cross feature knowledge.
4. The *resolution and integration of stereo mapping tools and simulation modelling tools* in order to efficiently collect and manage volume features such as buildings.
5. A *true 3D topological model* needs to be developed and implemented in OO GIS technology.
6. Techniques are needed to perform *cross-source feature validation and inference*. Features in one source can reinforce, augment, create or discredit features in another.
7. Techniques need to be developed to *maintain consistency* between DTM data and extracted/refined feature data of geomorphological significance.
8. Research needs to be done in the *management of multi-source/multi-scale representations of feature*s. This includes the automatic resolution of spatial and semantic differences (linking) between alternate representations of the same feature. Note that at present Generalisation algorithms tend to ignore the third dimension.

Acknowledgements

This development has been carried out as a collaborative research project between the US Army Topographic Engineering Center, Alexandria, Va. USA and Laser-Scan, (Sterling Va. USA and Cambridge UK).

References from Other Literature

Edwards, D. and Simpson, J.2000, "Topological Photogrammetric Mapping in 2.5 Dimensions", to be presented at ASPRS, Washington DC, May 2000

Woodsford, P. A. and Arctur, D. 1996, “Data Conversion and Update in the Object Paradigm”, GIS/LIS '96 Conference, Denver, USA, November 1996 and at http://www.lsl.co.uk/papers/datacon.htm

References from Websites

Laser-Scan 2000, http://www.lsl.co.uk/products/lamps2.htm

L-H Systems 2000, http://www.lh-systems.com/products/laser-scan1.html

[Original 2000-03-26, Revised 2000-03-28]