

VISUALIZING DISTRIBUTED DYNAMIC GEOSPATIAL INFORMATION IN GOOGLE EARTH

Jacek Radzikowski Xu Lu Anthony Stefanidis Matt Rice

Center for Geospatial Intelligence
Department of Geography and Geoinformation Science
George Mason University, Fairfax, VA 22030, USA
{jradziko, xlu5, astefani, mrice4 }@gmu.edu

Commission IV, ICWC IV/II

KEY WORDS: Visualization, spatiotemporal, Google Earth, Sensor Networks

ABSTRACT:

The emergence of Google Earth (GE) as an integrative platform for the visualization of geolocated information is presenting the geospatial community with unique application opportunities and corresponding scientific challenges. In this paper we discuss the extension of GE to function as a four-dimensional (x,y,z,t) virtual spatiotemporal environment through the overlay in it of geo-rectified video feeds, and feeds from geosensor networks deployed in an area of interest. We have created a Virtual model of our University Campus, exported it to GE, and use it to visualize diverse feeds from sensors distributed in our campus. In the paper we present the architecture of a prototype system that uses GE to visualize such sensor feeds. The system allows us to visualize locations and temporal stamps for our datasets, thus enabling a user to select feeds of a specific type for a specific location and time (e.g. video of a building corner at 2:15 and text feeds from a neighboring spot at 2:20). Selected datasets can then be overlaid in GE for visual inspection. In particular we emphasize in particular on issues related to video feeds. We present our approaches to register feeds captured by surveillance cameras located on top of buildings, and video feeds captured by mobile phone cameras. We also discuss the visualization in GE of information extracted from such video feeds (e.g. trajectories of individuals tracked in video). This hierarchical navigation through information presents unique opportunities for visual exploration of geospatial datasets. In our paper and presentation we present theoretical problems and demo our prototype.

1. INTRODUCTION

The emergence of Google Earth (GE) as an integrative platform for the visualization of geotagged information is presenting the geospatial community with unique application opportunities and corresponding scientific challenges, ranging from the visualization of diverse types of geospatial information to the use of GE as a user interface for knowledge discovery in geoinformatics. At the same time, significant technological advances are dramatically improving our capabilities to collect vast volumes of datasets capturing (often in real-time) activities over a wide array of spatial scales. Among these advances we can identify the advent of wide area motion imagery sensors and systems, and the proliferation of narrow-field-of-view full motion video platforms (e.g. UAVs), as key developments enabling the collection of video feeds that can be processed for geospatial information extraction [Keightley and Gale, 2006; Hunn et al., 2008]. Furthermore, the emergence of geosensor networks [Nittel et al., 2008] and volunteered geographic contributions [Goodchild, 2007] is providing us with additional types of geographic data that are increasingly distributed and multimodal in nature, ranging for example from diverse sensor measurements at distinct locations to amateur video and verbal descriptions of complex activities.

With the increased availability of these diverse types of spatiotemporal datasets comes the need for innovative techniques to access, process, and disseminate these datasets and resulting information. Considering video feeds in particular, as they are the main focus of this paper, our main interest focuses on the visualization and dissemination of these video feeds through GE, and on the visualization of information

captured through them, which is primarily object trajectories, extracted through tracking. The typical information extracted from object tracking is the trajectories of targets, which provide the critical support of a variety of applications. For instance, Javed and Shah [2002] apply object tracking in an automated surveillance camera and classified the obtained objects. Kastrinaki et al. [2003] focus on the object tracking and detection techniques for complex traffic monitoring. They evaluate different video processing algorithms with an emphasis on traffic application. Dobrokhodov et al. [2006] developed a system to perform vision-based tracking using small UAVs, estimating target coordinates.

Trajectories represent a novel type of geospatial information, deviating from traditional measurements in terms of content and applications, as they describe information that extends beyond the traditional boundaries of the geoinformatics community. A trajectory for example does not describe only the start and end point of the monitored individual, but also allows the labelling of the target's activities through the identification of patterns of change in it, the application of reasoning techniques, and the prediction of future events. Thus, a variety of concise descriptors of trajectory content have been proposed [Agouris and Stefanidis, 2003] and novel reasoning approaches were developed [Cohn et al., 2003; Gabelaia et al., 2005].

In this paper we address the visualization of spatiotemporal information in Google Earth, focusing primarily on distributed video feeds, as they may be captured from surveillance or amateur cameras in an urban environment. In Section 2 we discuss information visualization in Google Earth. Section 3 addresses information extraction from video feeds. In Section 4

we present experiments performed by our group for geospatial information visualization in Google Earth and provide our future plans and outlook.

2. VISUALIZING INFORMATION IN GE

2.1 Data visualization using Google Earth

Since the incorporation of KML as the layer description language, Google Earth (and earlier Keyhole Earth Viewer) has been seen as a spatial data visualization tool. Initially these were only satellite images, later more data types have been added. While Google Earth should not be considered a general-purpose GIS application [Andrienko et al. 2007; Goodchild, 2008], it offers broad range of tools aimed at data visualization. The design of the system is based on the client-server model: the application running on the user's computer is just a high-performance rendering engine for data, which is fetched on demand from the server. This design reduces the amount of information that has to be stored on the user's workstation and the centralized management of the data helps to provide up-to-date data to the users.

Google Earth can render two types of data: raster and vector. Raster datasets are used to display satellite images, and photographs. It can be also used to visualize 2D datasets in form of 2D plots overlaid on top of ground imagery. This allows for presentation of the data in its proper context. Vector datasets can represent lines, points or polygons. Lines can be used to visualize linear features, like roads, building outlines or administrative boundaries. Polygons can be used to mark areas, e.g. with active severe weather warnings [Smith and Lakshmanan, 2006].

Points in Google Earth nomenclature are called placemarks. A placemark can represent a place or a location. It can be used to mark a business, an interesting place or a location of a car or an animal at a specific time. The most important feature distinguishing placemarks from other types of vector data is that they can be generalized (Wood et al. 2007). Based on level of details in displayed image, to avoid cluttering of the display, a cluster of tightly packed placemarks can be displayed as one symbol, which expands to show individual placemarks after clicking on it.

All this information is described using an XML-variant language called Keyhole Markup Language (KML) [Google, 2010a]. A KML file can contain vector data, information about presentation of the information (e.g. colors of lines, icons for placemarks, etc), descriptions of presented features and references to other KML files. In the case of raster data, KML contains only metadata, like geolocation. The data itself is stored as a separate image file on a server and the location of the file is stored as a part of the metadata in the KML.

Use of KML files to describe information to be visualized in Google Earth allows for flexible configuration of datasets presented by the application. The default set loaded from GE servers contains satellite imagery for the entire globe and a number of layers predefined by managers of the system. All users of Google Earth share this set. Additionally, each user has an ability to define and load his/her own KML files, containing information about geolocated objects, with spatial footprints specific to user's application. The user-defined files can be loaded from local disk, in which case it will be available only to users sharing access to this storage medium, or on a network

connected server. The latter allows for collaboration between many users located in different parts of the globe.

2.2 Dynamic visualization using Google Earth

Despite its great wealth of capabilities for visualizing geospatial information, Google Earth until very recently had no specific mechanisms for dynamic updates of information. The first mechanism that allowed for a limited dynamism in datasets was through location-based updates. On each viewport update Google Earth could send to the server information describing the extent of the visible area and the server would send in return a KML file containing information about relevant objects. This mechanism allowed not only to limit the amount of the transmitted data, but also to update the dataset on each viewport update.

Recognizing the need for visualization of spatiotemporal information, in version 2.1 Google introduced a temporal extension of the specification of KML. Each object can have temporal information attached in a form of either point in time or a period of time describing the object. Assigning temporal description allows for visualization of time-ordered events or data sets (e.g. sequence of object locations or time-ordered series of satellite images). After opening a KML file with spatiotemporal information a temporal player control panel (Figure 1) is shown on the screen. There, the user can use the time slider to select specific instances in time over the available period, or play the entire temporal sequence by pressing the play button. The speed with which the sequence is displayed, and the play mode (once/repeat) can be set in the player properties window. These two improvements were big steps towards introducing dynamism in the data, but one thing did not change: once the data has been loaded from the server, it could not change.

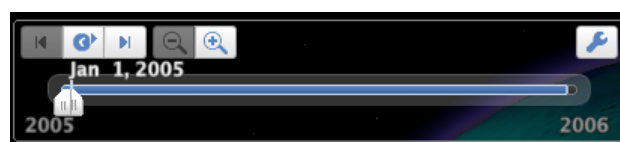


Figure 1. Temporal player in Google Earth

The capabilities for visualization of dynamic spatiotemporal data of Google Earth as a standalone are limited by the descriptive capabilities of KML. The most important limiting factor is lack of scripting capabilities. Google Earth Plug-in is an extension of a web browser, which allows using the Google Earth rendering engine as an element of web pages. The plug-in provides an interface to control the renderer from the browser and to take advantage of the browser's scripting capabilities (Fig. 2).

The most recent major release of Google Earth introduced an extension of the KML standard: *tours*. Tours allow creating presentations of geospatial information in the form of flyovers over terrain and objects, and smooth transitions between them. While not fully dynamic per se (each tour has to be pre-designed), tours introduce a very important extension, introducing true dynamism into the visualization: periodic updates. KML features in the Earth environment can be modified, changed, or created during a tour, including size, style, and location of placemarks, the addition of ground overlays, geometry, and more [Google, 2010b].

Our approach to real-time visualization of raster data takes advantage of the refresh mechanism available for links to external resources. In our application we use it to periodically reload an image draped over the ground, but it is similarly possible also to use the same mechanism to enforce periodic updates of any external KML files that describe the area of interest.

The mechanism of periodic updates opens a way to visualize asynchronously changing data. However, it has some important shortcomings. The most important one is the refresh rate. The update period is fixed and can be modified in multiples of full seconds. This means that currently the fastest refresh rate is one update per second. While this may be sufficient to visualize dynamic scenes, it still does not meet the requirements to display full motion video.

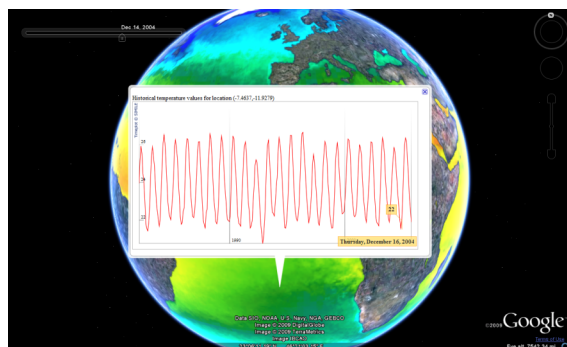


Figure 2. Visualizing a temporal sequence for a specific location on top of Google Earth: sea surface temperature variations (3-month averages) as ground overlay, and a graph showing monthly SST averages for the period 1982-2005 for a South Atlantic location.

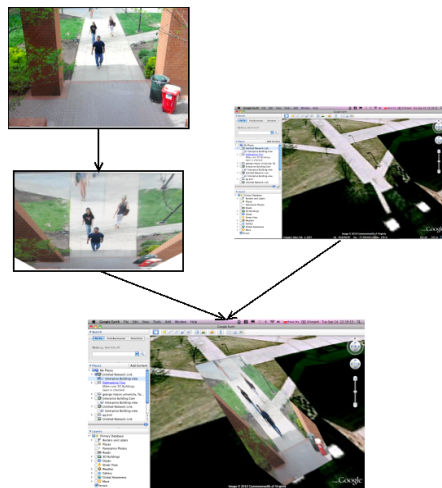


Figure 3. Video feeds (top left); the corresponding scene (top right) and registered video overlaid over the ground surface (bottom)

3. SPATIOTEMPORAL TRACKING

Video feeds captured by surveillance or amateur cameras over an area of interest can be easily geo-registered using either a manual process (whereby the user selects a minimum of 3

points to solve for the transformation relating the video plane to the ground), or by using any of the well established automated image registration approaches [Zitova and Flusser, 2003]. Using this georegistration information video snapshots can be overlaid on Google Earth (Fig. 3).

Object tracking in video feeds allows us to precisely track the movement of individual or objects in these feeds. In a simple manner it may proceed through background subtraction and subsequent target tracking. If the number of moving objects in a scene is sufficiently low, the background scene image can be generated by averaging (or median filtering) a sequence of frames taken with a static sensor (see Figure 4).

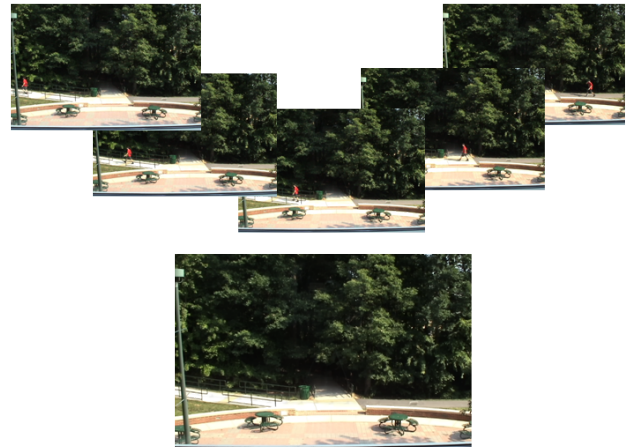


Figure 4. Sample video frames (top) and resulting background composite image (bottom). Notice the removal of the walking person from the background composite image (bottom).

Using this stationary background view, difference images may be generated when subtracting subsequent frames from it. In this image, pixels are marked as moving or stationary according to a comparison of their values to a threshold value. Threshold selection is typically based on a statistical analysis of image content, to ensure that the radiometric difference between the compared instances is sufficiently large. Morphological operations can be performed to eliminate noise (e.g. waving trees, flags), ensuring that only sufficiently large connected components (representing the target) remain. A trajectory can be generated by linking the centers of the tracked blobs (see Figure 5).



Figure 5. The trajectory of the moving person detected in the video feed of Fig. 4, overlaid on the background frame.

In order to improve the accuracy of detection, different object representation methods are developed. Some use global features:

Leibe et al. [2005] represent human as a whole, and detect the pedestrian in the crowd. Some use local features: Wu and Nevatia [2006] use parts of human body (legs, torso, head-shoulder, etc.) to track multiple partially occlude humans in a multi-view system. Lersudwichai et al. [2005] apply the local feature, specifically the face, to detect human, and present an algorithm to track multiple people with partial and total occlusion. Also there are some novel human representations, for example the path-length, introduced by [Yoon et al. 2006], is defined as the normalized length of the shortest path from the top of head to a given pixel inside a human silhouette.

Other tracking methods are also popular. As the objects can be represented in a feature space, and the tracking problem is reduced to match the moving target to match candidates. Mean-shift approach estimates the similarity between the target and candidates using the Bhattacharyya coefficient [Comaniciu et al., 2000]. The problem of multiple object tracking can be simulated by particle filters, and they have been widely used in video tracking [Hue and Perez, 2002; Kembhavi et al. 2008].

4. EARLY EXPERIMENTS AND OUTLOOK

We have generated a virtual model of our GMU Fairfax Campus, comprising approximately 100 buildings in 150 acres. We have already released approximately half of these buildings in GE and will be releasing the remaining buildings by the end of the year. We used this dataset as a reference for our experiments, together with various video feeds captured in our campus.

Video frames are draped in GE as shown in Fig. 3 as ground overlays. Currently, GE allows these overlays to be refreshed once per second. Accordingly, using the current GE configuration we are able to display one frame per second from our video feed onto GE. It is anticipated that this refresh rate will improve in the future.



Figure 6. Object detection and tracking. The green rectangles show the detection results (upper body detection), and the green curve shows the trajectory of the person.

In terms of computational complexity, early experiments we performed for this proposal (in order to generate the prototype from which Fig. 3 was captured) support the notion that *quasi real-time performance is possible from-video-capture to GE-display*. Rectification of a 640x480 30fps video stream consumes about 40% of time of a P4 processor running at 3.0GHz clock speed. Considering the above-stated maximum frequency to refresh ground overlays in GE is once per second,

we can estimate that a low-end off-the-shelf PC will be able to handle rectification of video streams from at least 50-60 sources and their display in GE. If the video cameras are mobile (e.g. roaming over the area of interest) these estimates will have to be revised to take into account the time needed to estimate the orientation matrix. And of course more robust computing resources can be used to improve this performance as needed.

Regarding trajectory extraction from our videos, in Fig. 6 we see an example of tracking an individual in a video feed using the Viola-Jones algorithm [Viola and Jones, 2001]. Trajectories extracted in this manner are the important information extracted from video analysis.

Having collected object trajectories over a longer period of time, we can compute traffic patterns over the observed area. Each trajectory can be assigned width proportional to the size of the tracked object (widths can be assigned by classes of objects: car's trajectory will be wider than pedestrian's trajectory, but it is not necessary to assign different width for each person). Then, by integrating the trajectories over the observed region, we can compute volume of traffic for communication. The more trajectories pass over an area, the higher traffic is traffic intensity. Using traffic intensities computed for different times of a day, week or month, it is possible to identify and compare traffic patterns for these periods of time. Computed traffic patterns can contain not only information about traffic intensity, but also about traffic direction. This is very valuable information that can help better design communication tracts.

Overlaying individual trajectories and computed traffic patterns on the map of existing communication tracts, it is possible to identify anomalous behaviour, like pedestrians crossing street in wrong places, cars going against the traffic direction on a one-way street, people cutting through a traffic lawn.

Figure 7 shows pedestrian traffic patterns on Mason campus. Green colour denotes normal traffic, red colour indicates anomalies. The intensity of the colour serves as an indication of the traffic intensity: light colour corresponds to small number of trajectories identified by that area, while strong, opaque colour indicates heavy traffic.

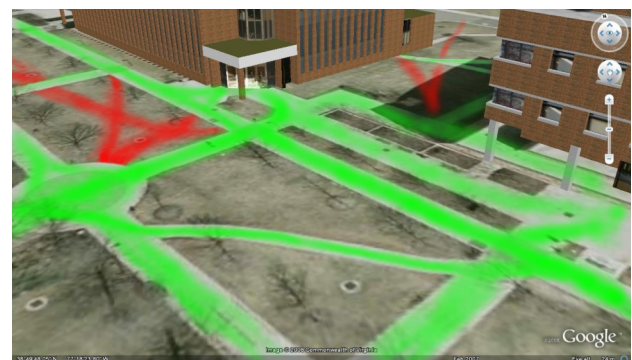


Figure 7 Pedestrian traffic patterns on Mason campus

Google Earth is also suitable for visualizing layers of information overlaid in it, in order to support multi-source spatiotemporal analysis through information availability visualization. More specifically, Fig. 7 shows a chart inside a GE popup window at a specific location. We have visualized three different types of datasets that are available for this location: trajectories crossing it (top), video feeds with a footprint that includes it (middle), and text feeds originating

from or relating to it (bottom). The plots on the left side of the popup window show the number of available information units (vertical axis) as a function of time (horizontal axis). Consequently, a peak indicates very large amounts of data available at a certain instance, while a valley represents relatively low amounts of data available. The vertical dashed line on the left side of the plot is a time marker. The user can position the marker at any point in time. After the user positions the time marker, a query is sent to the server. The server returns to the client a list of information sources relevant for this moment. The thumbnails of the sources (video streams, static images, snapshots of web pages) are presented to the right of the plots, as scrollable stacks. Accordingly from a database point of view we are looking at queries issued over distributed databases (the information visualized in the plots is typical metadata information), the plots are generated and displayed in a popup window.

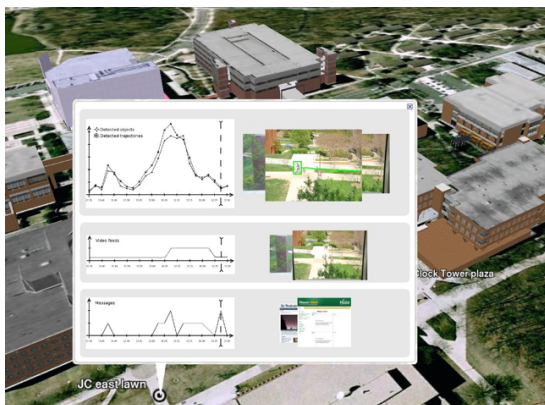


Figure 7. Visualizing dataset availability for an area of interest (JC East Lawn/ GMU Fairfax).

While we display 3 types of datasets in this early experiment, in principle this can be extended to show numerous types of information available for this location, (e.g. sensor measurements). This visualization allows us for example to select all datasets for a specific time instance (if this instance becomes of critical importance), making this a portal to all available information for the area of interest.

Above we presented an overview of some of our activities related to visualizing spatiotemporal information in Google Earth, focusing primarily on distributed video feeds, as they may be captured from surveillance or amateur cameras in an urban environment. They demonstrate the suitability of GE Earth for such visualization, and also the potential offered to use it as a GUI to access distributed spatiotemporal datasets, and support knowledge discovery operations. If we combine this potential with the tremendous popularity of GE among the public at large, extending well beyond the traditional geospatial community, we can easily perceive the opportunities presenting themselves for using GE to visualize and access volunteered information

ACKNOWLEDGEMENT

This work was supported by the National Geospatial-Intelligence Agency through a NURI grant NMA 401-02-1-2008 and NURI grant NMA HM1582-10-BAAA-0002, and the National Science Foundation Award 0429644.

REFERENCES

- Agouris, P. and A. Stefanidis., 2003. Efficient summarization of spatiotemporal events. *Communication of the ACM*, 46(1), pp. 65-66.
- Andrienko, G., N. Andrienko, P. Jankowski, D. Keim, M.-J. Kraak, A. MacEachren and S. Wrobel. 2007. Geovisual analytics for spatial decision support: Setting the research agenda. *International Journal of Geographical Information Science*. 21(8): 839-857.
- Cohn, A.G., D.R. Magee, A. Galata, D.C. Hogg, S.M. Hazarika, 2003. Towards an architecture for cognitive vision using qualitative spatio-temporal representations and abduction. *Spatial Cognition III*, 2685, pp. 232-248.
- Comaniciu, D. and V. Ramesh, 2000. Mean shift and optimal prediction for efficient object tracking. In: *Proceedings of International Conference on Image Processing*, Vancouver, Canada, Vol. 3, pp. 70-73.
- Dobrokhodov, V.N., I.I. Kaminer, K.D. Jones, and R. Ghabcheloo, R., 2006. Vision-based tracking and motion estimation for moving targets using small UAVs. In: *Proceedings of the American Control Conference*. Minneapolis, USA, WeC01.3, pp. 1428-1433.
- Gabelaia, D., R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyashev, 2005. Combining spatial and temporal logics: Expressiveness vs. complexity. *Journal of Artificial Intelligence Research*, 23, pp. 167-234.
- Goodchild, M., 2007. Citizens as sensors: the world of volunteered geography, *GeoJournal*, 69: 211-221.
- Goodchild, M., 2008. The use cases of digital earth. *International Journal of Digital Earth*. 1(1):31-42.
- Google, 2010a. KML Overview. <http://code.google.com/apis/kml/documentation/>.
- Google, 2010b, Touring in KML, <http://code.google.com/apis/kml/documentation/touring.html>
- Hue, C., and P. Perez, 2002. Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic System*, 38(3), pp. 791-812.
- Hunn B.P., K. Schweitzer, J. Cahir, M. Finch, 2008. IMPRINT analysis of an unmanned air system geospatial information process, Army Research Laboratory, Report ARL-TR-4513.
- Javed, O., and M. Shah, 2002. Tracking and object classification for automated surveillance. In: *Proc. 7th European Conference on Computer Vision, Lecture Notes In Computer Science*, Copenhagen, Denmark, Vol. 2353(IV), pp. 343-357.
- Kastrinaki, V., M. Zervakis, and K. Kalaitzakis, 2003. A survey of video processing techniques for traffic applications. *Image Vision Computing*, 21(4), pp. 359-381.
- Keightley, D.E. and K.L. Gale, 2006. Bandwidth-smart UAV video systems in distributed networked operations, *IEEE MILCOM*, Washington DC, pp. 1-5.

Kembhavi, A., W. Schwartz, and L.S. Davis, 2008. Eighth International Workshop on Visual Surveillance, in *Workshop at the 10th European Conference on Computer Vision*. Marseille, France.

Leibe, B., E. Seemann, and B. Schiele, 2005. Pedestrian detection in crowded scenes. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Diego, USA, Vol. 1, pp. 878-885.

Lerdsudwichai, C., M. Abdel-Mottaleb, A.N. Ansari, 2005. Tracking multiple people with recovery from partial and total occlusion. *Pattern Recognition*, 38, pp. 1059-1070.

Nittel S., A. Labrinidis and A. Stefanidis, 2008. *Advances in GeoSensor Networks*, Lecture Notes in Computer Science, Springer, Vol. 4540 (272 pages)

Smith T.M., V. Lakshmanan, 2006, Utilizing Google Earth as a GIS platform for weather applications, The 86th AMS Annual Meeting (Atlanta, GA)

Viola, P., and M. Jones, 2001. Rapid Object Detection using a Boosted Cascade of Simple Features. *IEEE Conference on Computer Vision and Pattern Recognition*.

Wood J., J. Dykes, A. Slingsby, and K. Clarke, 2007, Interactive Visual Exploration of a Large Spatio-Temporal Dataset: Reflections on a Geovisualization Mashup, *IEEE Trans. On Visualization and Computer Graphics*, 13(6).

Wu, B., and Nevatia, R., 2006. Tracking of multiple, partially occluded humans based on static body part detection. In: *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, New York, USA, Vol. 1, pp. 951 – 958.

Yoon, K., D. Harwood, and L.S. Davis, 2006. Appearance-based person recognition using color/path-length profile. *Journal of Visual Communication and Image Representation*, 17, pp. 605-622.

Zitova B., and J. Flusser, 2003. Image registration methods: a survey, *Image and Vision Computing*, 21(11): 977-1000.