ALGORITHM FOR IMAGE INTEGRATION INVARIANT TO DISPROPORTIONAL SCALING

Michael Kovalerchuk, Researcher
Boris Kovalerchuk, Professor
Dept. of Computer Science,
Central Washington University
400 East University Way
Ellensburg, WA, USA 98926-7520
kovalerm@cwu.edu
borisk@cwu.edu

ABSTRACT

Imagery integration has a variety of open questions in image registration, conflation, and search. A new promising approach is based on the algebraic paradigm that differs from the more traditional geometric and topological approaches. In previous research, algebraic methods have been based on algebraic structural analysis of features represented as open polylines called linear features. In this paper we report a new algebraic method based on ratios of areas of closed contours (shapes). It has been proven that this method is invariant to disproportional scaling. The method is applicable to images that contain three or more matching shapes, but those shapes and their actual matches are not known. The method has also undergone some optimization to shorten run time. Computational experiments had shown that the method is capable of integrating images with dozens of shapes in a reasonable time. The algorithm is also a result of the algorithm development technology proposed in this study. The method is implemented as an ArcMap Plug-in.

INTRODUCTION

The development of a universal algorithm that can integrate, register, and conflate images of different and unknown rotations, translations, scaling, resolutions and modalities is an ultimate goal in the area of image integration. The recent survey (Zitova, Flusser, 2003) indicates that in spite of hundreds of algorithms developed this ultimate task of a universal algorithm remains unsolved. There are several reasons for this. One of them is in deficiencies in mathematical foundations. We offer a new promising approach based on the algebraic paradigm that differs from the more traditional geometric and topological mathematical approaches.

In previous research (Kovalerchuk et al, 2001-2004), algebraic methods have been based on algebraic structural analysis of features represented as open polylines called linear features. In this paper we report a new algebraic method based on ratios of areas of closed contours (shapes). It has been proven that this method is invariant to disproportional scaling and invariant to any general affine transform of the images. The method is applicable to images that contain three or more matching shapes, but those shapes and their actual matches are not known.

The algorithm has been developed using both algebraic mathematical paradigm and an **algorithm development technology for conflation (ADTC)** (Kovalerchuk, Schwing, 2004). The concept of conflation was identified in [Cobb et al., 1998; Jensen, Saalfeld et al., 2000; Doytsher et al.,2001; Edwards, Simpson, 2002] and in [Kovalerchuk, Schwing, 2004, Ch.17-19]. This task is an expansion of the imagery registration task [Brown, 1992, Zitová, Flusser, 2003, Shah, Kumar, 2003; Terzopoulos et al, 2003; Wang at al., 2001]. The flowchart in Figure 1 shows the general sequence of steps for the algorithm development methodology. These steps can be looped to get an improved result.

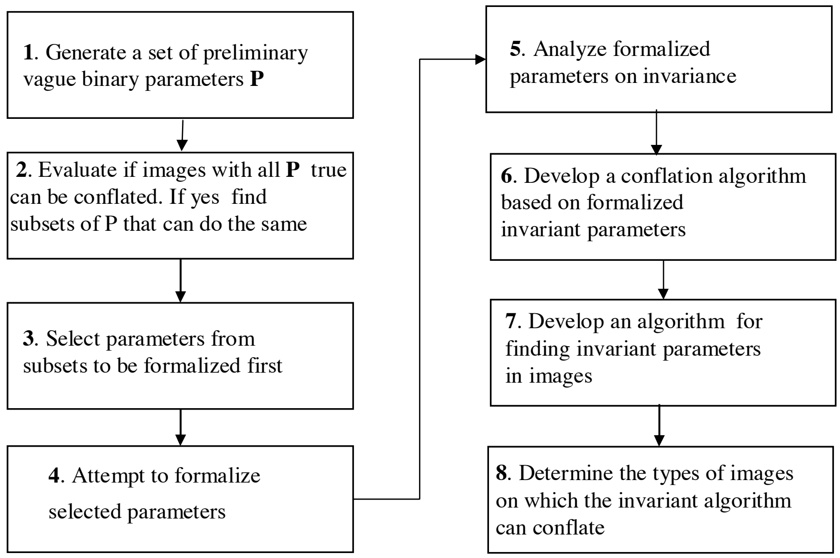


Figure 1. Overall steps of the ADTC technology

PARAMETER INVARIANCE

Invariance to **disproportional scaling** (DPS) is one of the most difficult requirements to meet. Such algorithm can be capable to integrate, register and conflate images that are rotated, shifted and disproportionally scaled. It is assumed that all these factors are not known and the algorithm should discover them from the images.

To be able to build an invariant algorithm we need to discover parameters that will be invariant under disproportional scaling. Few invariant parameters are known. Affine Moment Invariants (AMI) (Zitova, Flusser, 2003) are among them, but AMI are not very robust to the noise in image and have some other limitations.

Other parameters that are typically invariant if they exist in both images are road intersections, and building corners. Extraction of both depends on image quality and if there are too many of them then there is a computational challenge to match them. If images have very different resolutions then finding matching road intersections and building corners may not be possible at all. Thus the algorithms based on these parameters may not be quite universal.

Algorithms based on AMI are potentially more universal. Such algorithms require only three matched full closed contours in each image. The algorithm that we present in this paper requires the same three matched closed contours in each image to exist.

The most fundamental difference of our method from AMI is that parameters that we use have clear and intuitive interpretation in contrast with AMI that is hard to interpret. Our parameters are **area relations and area ratios** that we discuss below. Next our parameters are more robust to noise than AMI. Another contrasting property of our algorithm is that we use as parameters the relations between two different shapes in the image. AMI algorithms use features of each single shape in the image. They are later forced because of this to use relatively arbitrary metrics for similarities between shapes. We use actual relations that exist between shapes in the image that have direct meaning in the image and direct interpretation in the image also.

Angles between linear segments of the features and lengths of those segments vary under affine transforms significantly, but their relations, >, <, = are more robust. However, these relations are not complete affine invariants under disproportional scaling, but some **relations between areas are invariant** as Figures 2 and 3 show. In Figure 2, area S_1 is equal to area S_2, $S_1=S_2$. In addition, angles A, B, and C, D are equal, A=B, C=D too.
Figure 3 presents the same image after disproportional scaling, where X coordinate was multiplied by $k_x>1$ and Y coordinate is not changed, $k_y=1$. Relations between angles A and B have been changed, A`< B`. Also relations be-

tween angles C and D are changed, C`<D`. In contrast, the relation between areas $S`_1$ and $S`_2$ is not changed, $S`_1=S`_2$. This can be proved by noticing that bounding boxes U_1 and U_2 around rhombuses S_1 and S_2 are not changed and each rhombus occupies a half of its bounding box. More formally we have $U`_1= k_x k_y U_1$, $U`_2= k_x k_y U_2$, and using property $U_1=U_2$, we conclude that $k_x k_y U_1= k_x k_y U_2$ and therefore $U`_1= U`_2$. Next, $S`_1= U`_1/2$ and $S`_2= U`_2/2$ hence $S`_1=S`_2$.

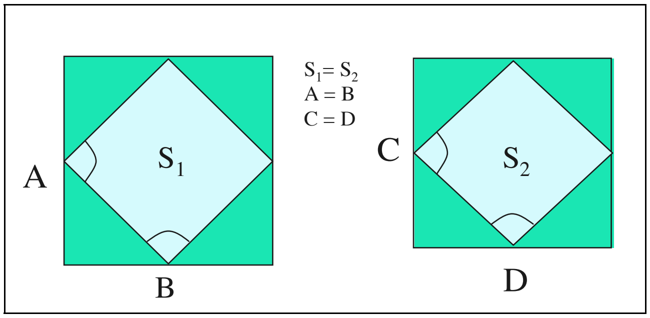


Figure 2. Original image

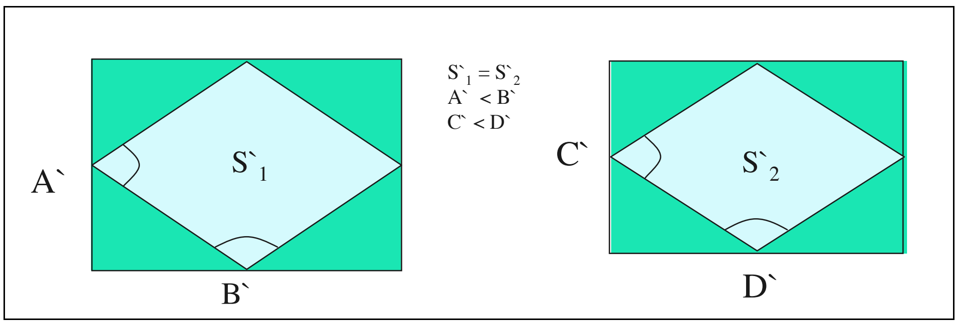


Figure 3. Image after disproportional scaling

Thie proof is also valid for the general case when $k_x \neq 1$. We can notice that rotation and translation do not change relations between angles as well as between areas. Thus, area relations are not changed under translation, rotation and disproportional scaling.

Above we considered only simple relations "=" and ">" between areas. Other **area functions** can also be invariants. The r**atio of areas** S_1/S_2 is also invariant under disproportional scaling, because $k_x k_y S_1/k_x k_y S_2 = S_1/S_2$. The consideration above can be converted into a formal theorem statement. Let F be an affine transformation that combines disproportional scaling transformation K with scaling coefficients (k_x, k_y), $k_x \neq 0$, $k_y \neq 0$, translation T and rotation R, $F=K \cdot T \cdot R$, where K,T and R are transformation matrixes. Let also G_1 and G_2 be two closed regions and $S_1=S(G_1)$ and $S_2=S(G_2)$ be their areas respectively, where S( ) is an operator that computes area of the region G_i.

Theorem: An affine transformation F of the image does not change the relation between area ratios,

$$S_1/S_2 = S(G_1)/ S(G_2)= S(F(G_1))/S(F(G_2)),$$

that is F is an **isomorphism** for area ratio $S(G_1)/ S(G_2)$.

The proof follows from the considerations that preceded the theorem. If images are rotated and shifted then the theorem is true as well because rotation and shift do not change areas at all. Thus we can first rotate and shift images to made them horizontal and with the common origin. After that our previous reasoning about disproportional scaling can be applied.

For a general affine transform F

$$x`=m_{11}x+m_{12}y+m_{13}$$
$$y`=m_{21}x+m_{22}y+m_{23} \quad (1)$$

and arbitrary shapes we can explain the idea of the proof of this theorem in the following way. Any shape can be interpolated by a set of rectangles such as horizontal black rectangles shown in Figure 4. Then we can apply a general affine transform F to the image that will transform each rectangle into a parallelogram.

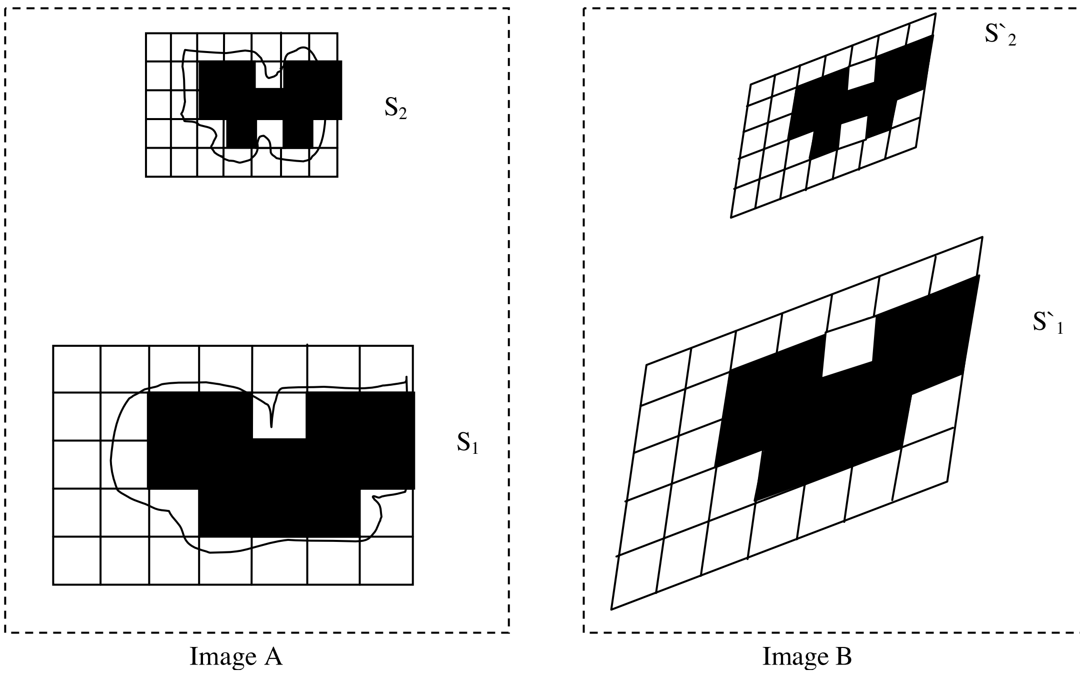


Figure 4. General affine transform

Figure 4 explains the idea of the proof for a general affine transform F. We have black shapes S_1, S_2 and shapes $S`_1$ and $S`_2$ after applying F to the image A that contains both S_1 and S_2. The ratio S_1/S_2 is equal to 12k/11 by counting black squares in S_1 and S_2, where k is a ratio of sizes of squares. In this example four small squares form a large square, thus k=4. In image B the ratio $S`_1/S`_2$ is equal to m12/11 again by counting black distorted squares in $S`_1$ and $S`_2$, where m is a ratio of sizes of distorted squares. Here four small distorted squares form a large distorted square, thus m=k=4.

INVARIANT CONFLATION / REGISTRATION ALGORITHM

In this section we describe an **Area Ratio Conflation Algorithm** (**ARC** algorithm for short). The invariance of the area relations found in the previous section is the base for this algorithm.

Two raster images are conflated by finding at least 3 matching uniquely sized regions (areas) in both images and using the center points of those regions as reference points for an affine transform calculation. There are several possible formalizations of the concept of "**matching uniquely sized regions (areas)**".

In the ARC algorithm, we use the area ratio S_i/S_j as the matching characteristic, because of its invariance shown in the previous section. If two areas in the original image have a ratio, say 0.3, then the same ratio between them should remain the same under any affine transform. Thus, in the second image, we can compute areas of regions, their ratios S_i/S_j and search for a 0.3 ratio among them. If only one such ratio was found then centers of these regions give us two tie (control) points for building an affine transform. Finding a third region S_m in the both images with the equal ratios S_i/S_m in both images provides the third tie point needed for an affine transform. This basic idea is adjusted for the cases where more than one matching triple found. An additional uniqueness criterion is introduced in the algorithm based on the analysis of additional ratios.

Suppose there is an image that contains a large lake of some size and a small lake whose size is ⅓ of the size of the large lake. This size ratio (⅓) is invariant to affine transformations. The ratio precision needs to be adjusted to the scale of least precise image. Ratios ⅓, ½ and ¼ could match 0.336 0.52 0.27 if images are of different scales. The algorithm uses a matching threshold for these cases.

This logic of the algorithm requires: (1) an algorithm for computing area ratios and for matching ratios and (2) an algorithm for region extraction from the image. The first algorithm called the Ratio Algorithm and the second algorithm called Vectorizer are described below. The development of the second algorithm is the goal of the Step 7 of the ADTC technology.

The ratio algorithm starts from a set of regions $\{G_{1i}\}$ for image 1 and a set of regions $\{G_{2i}\}$ for image 2 extracted by the Vectorizer algorithm. The *Ratio algorithm* computes areas for each region in both images, $S_{1i}=S(G_{1i})$, $S_{2i}=S(G_{2i})$ as a number of pixels inside of the region. Next this algorithm computes two matrixes V_1 and V_2. Elements of matrix $V_1=\{c_{ij}\}$ are $c_{ij}=S_{1i}/S_{1j}$ Elements of matrix $V_2=\{q_{ij}\}$ are defined similarly, $q_{ij} = S_{2i}/S_{2j}$. We assume that all areas S_{1i} and S_{2i} are positive.

The area ratio algorithm consists of several steps presented below:

Step 1: Extract shapes (closed contours) in each image.
It can be done using many different methods. We developed a simple Vectorizer algorithm that sharpens images and finds regions using a flood-fill method from computer graphics [Angel, 2000] that starts from a seed point and looks recursively at colors of adjacent pixels including diagonal neighbors until all neighboring pixels of the same color are found. The set of these pixels is considered a single region. The number of the pixels in the region is considered its area/size. A set of all extracted regions is ARC algorithm input. Now we work on more sophisticated feature extraction algorithm that can be more specific for this registration task and noise robust.

Step 2: Compute areas of each shape, order areas and select n largest shapes. For practical implementation we use n=35.

Step 3: Build a matrix as shown in Tables 1 and 2.

Table 1. Matrix of shape size ratios in Image 1

	$S_{11}=6$	$S_{12}=4$	$S_{13}=2$	$S_{14}=1$
$S_{11}=6$	1	4/6	2/6	1/6
$S_{12}=4$	6/4	1	2/4	1/4
$S_{13}=2$	6/2	4/2	1	1/2
$S_{14}=1$	6/1	4/1	2/1	1

Table 2. Matrix of shape size ratios in Image 2

	$S_{24}=7$	$S_{23}=6$	$S_{21}=4$	$S_{22}=1$
$S_{24}=7$	1	6/7	4/7	1/7
$S_{23}=6$	7/6	1	4/6	1/6
$S_{21}=4$	7/4	6/4	1	1/4
$S_{22}=1$	7/1	6/1	4/1	1

In this example matrix V_1 shown in Table 1 is computed for regions with areas $S_{11}=6$, $S_{12}=4$, $S_{13}=2$, $S_{14}=1$ in image 1 and matrix V_2 is computed for areas $S_{21}=4$, $S_{22}=1$, $S_{23}=6$, $S_{24}=7$ in image 2 (see Table 2).

Step 4: Search for common parts with at least tree rows and columns in each.
We can notice that there is a common part in Tables 1 and 2 shown within the bold borders.

Step 5: Select a common part out of the set of found common parts. Some optimization algorithm can be used for such selection. Otherwise it can be a random choice.

Step 6: Identify shape match using a selected common part.
In the example the common part identifies matching shapes in two images, that is

Image 1: (S_{11}, S_{12}, S_{14})
Image 2: (S_{23}, S_{21}, S_{22})

Step 7: Compute centers of gravity of each matched shape

Image 1: Shape centers (c_{11}, c_{12}, c_{14})
Image 2: Shape centers (c_{23}, c_{21}, c_{22})

Step 8: Compute affine transform F between shape centers found in step 7 and find vector of coefficients (m_{11}, m_{12},m_{13}, m_{21}, m_{22},m_{23}) of that transform.
Step 9: Repeat steps 5-8 for every common part found in step 4.

Step 10: Cluster vectors {(m_{11}, m_{12},m_{13}, m_{21}, m_{22},m_{23})} into similarity groups, compute the number of vectors in each cluster and set up the highest priority to the largest cluster with descending priorities for other clusters. Compute centers of each cluster and find vectors that are closest to the centers.

Step 11: Apply the affine transform found in step 10 to image A and transform it to image B

Step 12: Compute shape discrepancy for each transformed shape $F(S_{11})$, $F(S_{12})$, $F(S_{13}$) and shapes (S_{23}, S_{21}, S_{22}) as the number of pixels that do not match. It can be done using the XOR operation described below. The difference of two regions is XOR (exclusive OR) of pixels of regions G and G`. The **absolute difference** of regions G and G`, Δ(G,G`) is computed as the number of pixels in the difference of regions G and G`:

$$\Delta(G,G`)=S(XOR(F\,(G),G`))$$

and the **relative difference** of the regions is

$$\rho(G,G`)=\Delta(G,G`)/(S(G)+ S(G`)).$$

The total difference between three matched regions {G} and {G`} of images Im1 and Im2 found by ARC algorithm is

$$\mu(\{G\},\{G`\})= \mu(G_1,G`_1) + \mu(G_2,G`_2)+ \mu(G_3,G`_3).$$

Similarly the relative difference of matched regions is

$$\rho(\{G\},\{G`\})=[\Sigma_{i=1,2,3}\Delta(G_i,G`_i)] /[\Sigma_{i=1,2,3} (S(G_i)+ S(G`_i))].$$

The maximum of relative difference is 1 and the minimum is 0.

Step 13. Repeat steps 11-12 for every cluster found in Step 10.

Step 14: Evaluate discrepancy. Find the best transform from Step 13. The **conflation quality** can be evaluated by visual inspection of the conflated images and by a computational procedure based on the absolute and relative difference between matched regions found in Step 13. For instance if discrepancy is less then 10% of each shape $F(S_{1i})$ then it can be considered as an acceptable match. In general this threshold should be task-specific.

ALGORITHM OPTIMIZATION AND NOISE ROBUSTNESS

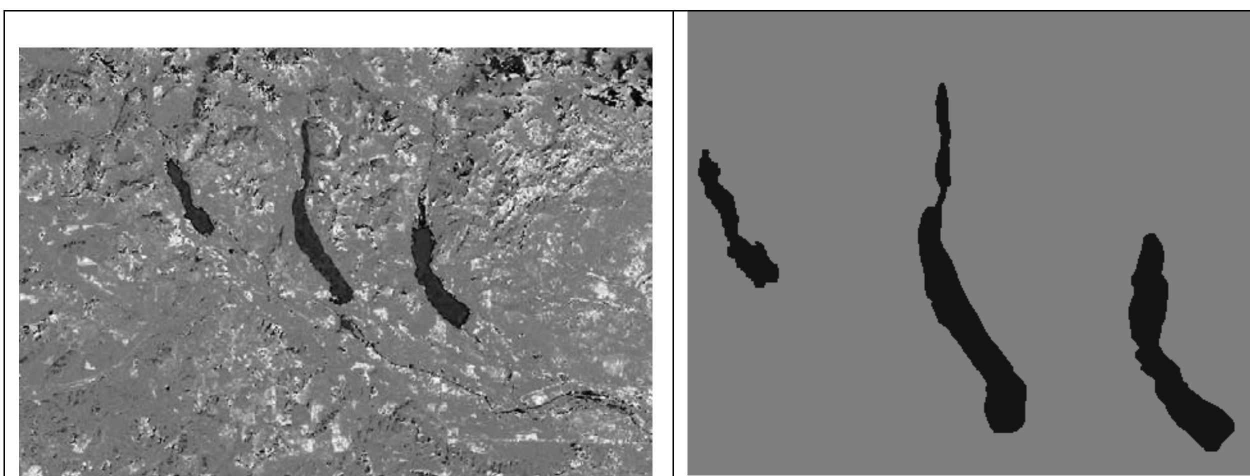
Step 4 that searches for the largest common part in two tables with at least tree rows and columns. Such search is not very computationally challenging because it does not run on the raster images but on a few characteristics extracted from the raster images such as ratios of areas of shapes found in the images. Having *n* shapes we need to generate two tables *n x n* for each of two images such as shown in Tables 1 and 2. Then for each element of these tables we need to find a matching (equal element) in another table. In the worst case such search takes $O(n^2)$ operations to check all elements of the second table for an element from the first table. Having $O(n^2)$ elements in the first table itself the total search complexity is $O(n^2)\ O(n^2) = O(n^4)$. After finding all equal elements we need to search for pairs of them that represent three shapes, because some pairs can represent two, three, or four shapes. In the worst case all elements of both matrixes are equal, that is we have $O(n^4)$ pairs. Each pair can be tested for the three shapes involved separately of other shapes. Thus, total we need $O(n^4)+ O(n^4) = 2O(n^4)$ computations, that is also $O(n^4)$. This number of computations is much less then is typically taken by methods based on direct manipulations with pixels.

The justification for iterative process of the algorithm is based on the following observations. We can take randomly three shapes in one image and three shapes in the other images matched by their area ratios, but it can be accidental. It is possible for the shapes that are really very different to have the same area ratios. However if we have many different pairs of three shapes matched by ratios with almost identical affine transforms (located in the same cluster on parameters of the transforms) then it is less likely that this match is accidental.

Why do we not build a single transform using, say, 10 shapes that match? If some of these matches are accidental they will corrupt a transform. Next matches can be real not accidental, but they may not be exact because of **data noise**, data error, and physical differences in images. For instance, one image is taken in the spring but another taken in the fall where a reservoir is almost empty and fall shape is quite different from the shape of the same reservoir in the spring when it is full. Next, if images are of different modalities, e.g., radar, optical or multispectral then shapes of the same reservoir can be quite different. Below we show our experiments with data of different modalities: SRTM and Landsat.

SOFTWARE AND COMPUTATIONAL EXPERIMENT

Figures 5-7 show results of our successful computational experiments with Area Ratio Conflation algorithms described above and implemented as a Plug-in to ArcGIS. It is important to notice that we used images of very different modalities: SRTM and Landsat. The screenshot in Figure 5 shows a Landsat image before its co-registration with SRTM . In fact, this is a hue component of the original Landsat image shown as a grey scale image. Figure 5 (b) shows the largest closed contours/areas in the same image.



(a) Hue component of the original Landsat image shown as a grey scale image	(b) Largest closed contours/areas in the sharpened Lansat image

Figure 5. Landsat image

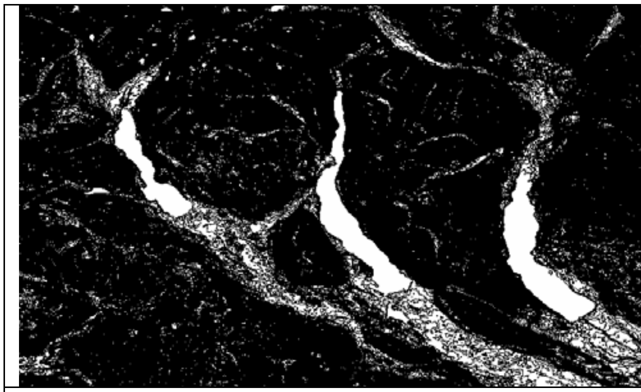
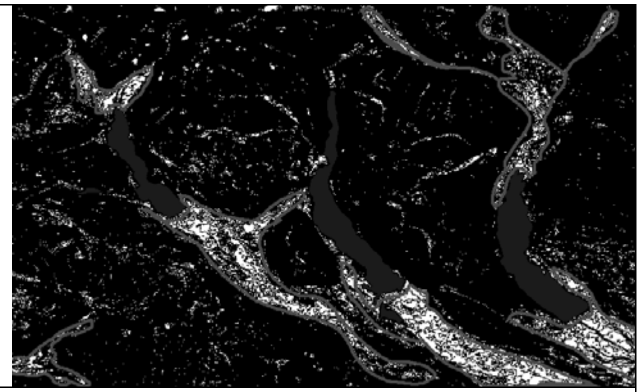
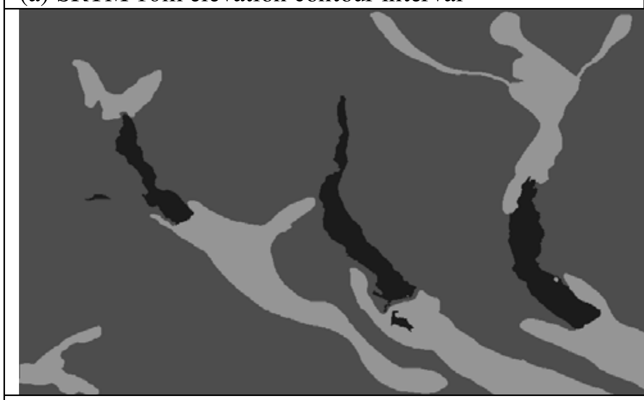
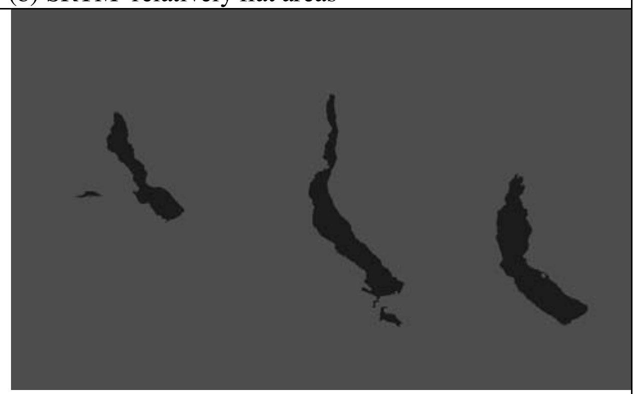
	
(a) SRTM 10m elevation contour interval	(b) SRTM relatively flat areas
	
(c) SRTM sharpened flat areas	(d) SRTM flatest areas

Figure 6. SRTM data in a visual form (10m elevation contour intervals)

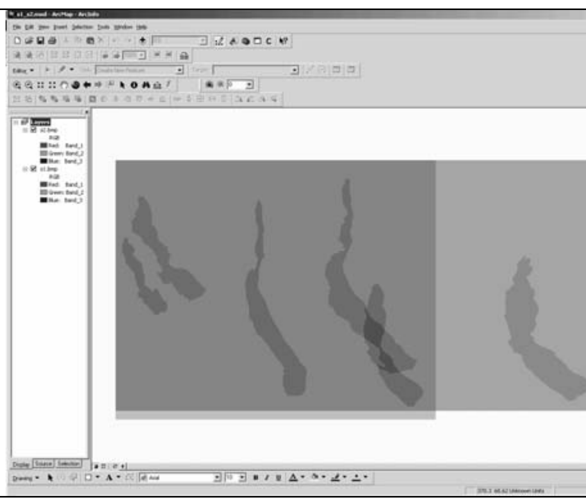
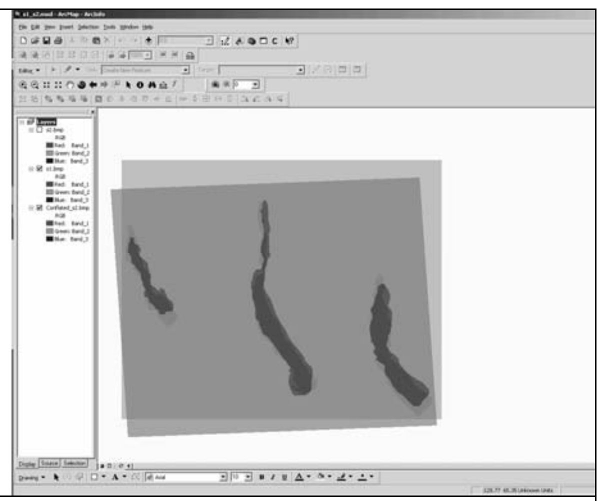
	
Input images for ARC algorithm	Co-registration output

Figure 7. Images from Landsat (Figure 5b) and SRTM (Figure 6d) co-registered

Other computational experiments with ARC algorithms had shown that the method is capable of integrating images with dozens of shapes in a reasonable time.

GENERALIZATION

The ARC algorithm is a part of general Algebraic Framework (Mal'cev, 1973; Kovalerchuk et al. 2001-2004) The matrix representation used this algorithm is important because it permits us to convert the situation to a generic **algebraic system framework**, with algebraic systems $\mathbf{A_k}$=<A_k, R_k, Ω_k>, where signature Ω_k contains the operator $V_k(a_i,a_j)$ represented as a matrix V_k and handles the conflation problem uniformly. From this point uniformity permits us to use a single and already implemented algorithm to search for matching features in the images. It does not matter for the algorithms in algebraic form whether elements of A_k are straight-line segments, polylines, areas, or complex combinations, or some other features. Elements of A_k also can be numeric characteristics of image components such as a size of region i in image k, S_{ki} and their ratios used in ARC.

CONCLUSION

Imagery integration has a variety of open questions in image registration, conflation, and search. A new promising approach is based on the algebraic paradigm that differs from the more traditional geometric and topological approaches. In this paper we presented a new algebraic method based on ratios of areas of closed contours. This method is invariant to disproportional scaling and any general affine transform. It is applicable to images that contain three or more matching shapes. The algorithm discovers such matching shapes. Computational experiments had shown that the method is capable of integrating images with dozens of shapes in a reasonable time. The algorithm has been developed using two methodologies: Algorithm Development Technology for Conflation and Algebraic Relational Invariant paradigm. Also the algorithm is implemented as an ArcMap Plug-in.

This research has been supported by a grant from National Geospatial-Intelligence Agency.

REFERENCES

Brown, L. A Survey of Image Registration Techniques, ACM Computing Sur-veys,vol.24 (4),pp. 325--376, 1992, citeseer.nj.nec.com/brown92survey.html

Chen, C. Knoblock, C. Shahabi, C. Thakkar, S, and Chiang, Y. Automatically and accurately conflating orthoimagery and street maps, In: Proceedings of the 12th ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS'04), 2004. http://www.isi.edu/info-agents/papers/chen04-acmgis.pdf

Chen, C. Shahabi, and C. Knoblock, C. Utilizing road network data for automatic identification of road intersections from high resolution color orthoimagery, In: Proceedings of the 2nd Workshop on Spatio-Temporal Database Management, STDBM'04, 2004. http://www.isi.edu/info-agents/papers/chen04-stdbm.pdf

Cobb, M. Chung, M,.Foley, Petry. F., Shaw, K., and Miller, H., A rule-based approach for the conflation of attributed vector data, GeoInformatica, 2/1, 1998, 7-36.

Doytsher, Y. Filin, S., Ezra, E. Transformation of Datasets in a Linear-based Map Conflation Framework, Surveying and Land Information Systems, Vol. 61, No. 3, 2001, pp.159-169.

Edwards D., Simpson J. Integration and access of multi-source vector data, In: *Geospatial Theory, Processing and Applications*, ISPRS Commission IV, Symposium 2002, Ottawa, Canada, July 9-12, 2002.

Edwards, D., Simpson, J. Integrating, Maintaining, and Augmenting – Multi-source Data through Feature Linking , In: *OEEPE/ISPRS Workshop: From 2D to 3D; Establishment and Maintenance of National Core Geospatial Databases,* 8-10 October 2001, Hannover; OEEPE, Frankfurt am Main, Germany, 2002,

Ghosh, S.K. Analytical Photogrammetry, Pergamon Press, New York, 1998.

Jensen J., Saalfeld, A., Broome, F., Price, K., Ramsey, D., and Lapine, L. Spatial Data Acquisition and Integration, 2000, NSF Workshop GIS and Geospatial Activities. Accessed June 2001. http://www.ucgis.org/research_white/data.html

Kovalerchuk B., and Schwing, J., Mathematical support for combining geospatial data, In: Proceedings of International Conference on Imaging Science, Systems, and Technology (CISST'2001), Las Vegas, 2001, pp. 485-491.

Kovalerchuk B., and Schwing, J., Algebraic relational approach for geospatial feature correlation, In: Proceedings of International Conference on Imaging Science, Systems, and Technology (CISST'2002), Las Vegas, 2002, pp. 115-121.

Kovalerchuk B. and Sumner W., Algebraic relational approach to conflating images, In: Algorithms and technologies for multispectral, hyperspectral and ultraspectral imagery IX. Vol. 5093, International SPIE military and aerospace symposium, AEROSENSE, Orlando, FL, April 21-25, 2003.

Kovalerchuk, B., Sumner, W., and Schwing, J., Image Registration and Conflation Based on Structural Characteristics, In: Proceedings of International Conference on Imaging Science, Systems, and Technology (CISST'2003), Las Vegas, 2003, pp. 239-245.

Kovalerchuk, B., Sumner W., Curtiss, M., Kovalerchuk, M., and Chase, R., Matching Image Feature Structures Using Shoulder Analysis Method, In: Algorithms and technologies for multispectral, hyperspectral and ultraspectral imagery IX. Vol. 5425, International SPIE military and aerospace symposium, AEROSENSE, Orlando, FL, April 12-15, 2004.

Kovalerchuk B., Schwing J. (eds.), Visual and Spatial Analysis: Advances in Data Mining, Reasoning and Problem Solving (Eds.), Springer, 2005.

Shah, M., Kumar, R., (Eds.) Video Registration, Kluwer, 2003

Terzopoulos D., Studholme, C., Staib, L., Goshtasby A., (Eds) Nonrigid Image Registration, Special issue of Computer Vision and Image Understanding Journal, vol. 89, Issues 2-3, pp. 109-319, 2003

Wang, J., Chun, J., and Park, Y.W. GIS-assisted image registration for an onboard IRST of a land vehicle. Proc. SPIE Vol. 4370, p. 42-49, 2001

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