

Central Washington University
ScholarWorks@CWU

Computer Science Faculty Scholarship

College of the Sciences

6-13-2013

Guidance in feature extraction to resolve uncertainty

Boris Kovalerchuk

Michael Kovalerchuk

Simon Streltsov

Matthew Best

Follow this and additional works at: <https://digitalcommons.cwu.edu/compsci>



Part of the [Geographic Information Sciences Commons](#), and the [Theory and Algorithms Commons](#)

Guidance in feature extraction to resolve uncertainty

Boris Kovalerchuk^{1,2}, Michael Kovalerchuk², Simon Streltsov³, Matthew Best⁴

¹Central Washington University, ²BKF Systems, ³LongShort Way Inc. ⁴USAF Materiel Command AFLCMC/HBDG

Keywords: Feature extraction, road extraction, trail extraction, guidance, GMTI, Genetic Algorithm, Multi-objective optimization.

Abstract

Automated Feature Extraction (AFE) plays a critical role in image understanding. Often the imagery analysts extract features better than AFE algorithms do, because analysts use *additional information*. The extraction and processing of this information can be more complex than the original AFE task, and that leads to the “complexity trap”. This can happen when the shadow from the buildings guides the extraction of buildings and roads. This work proposes an AFE algorithm to extract roads and trails by using the GMTI/GPS tracking information and older inaccurate maps of roads and trails as AFE guides.

1. Introduction

Automated Feature Extraction (AFE) and edge detection have a long-term history in roads and building extraction from overhead imagery^{1,8}. Content-supported and map-guided feature extraction has been also explored^{1, 2, 4, 6, 9, 10, 13-16, 18, 19}. Related studies that focus not on the road extraction, but on matching and aligning road network with the imagery are known as vector to raster conflation, co-registration, and fusion¹⁴⁻¹⁵. Those fusion systems can benefit significantly from the algorithm proposed in this paper, because it allows moving from a low quality unguided feature extraction step to a guided feature extraction. The trail extraction task was also explored recently^{11, 5, 11, 12, 20}.

This history had shown that often an image itself is not sufficient for unambiguous AFE. Often the imagery analysts (IA) extract features much better than the AFE algorithms do. The fundamental difference is that IAs use *additional information* that is either absent in the image or requires the abilities to extract it from the image. While it was recognized that such information is needed, the actual use of it in AFE algorithms is still limited. Often the direct use of the information that IAs easily use is not feasible in AFE algorithms. The formalization, extraction, and processing of this information can be more complex than the original AFE task, and that leads to the “complexity trap”. This work proposes an AFE algorithm to extract roads and trails by using the GMTI or GPS tracking information as well as older and inaccurate maps of roads and trails as such additional guiding information. The algorithm exploits ideas of multi-objective optimization and Genetic Algorithms (GA). The proposed *Guided Extractor (GUEX) algorithm* expands the capabilities of the MapSnap System¹⁵ that fuses and aligns vector maps and raster images to *guided feature extraction* using GMTI and other sources as guidance.

The reverse task is quite popular where road information is used to improve tracking accuracy²². This reverse task implicitly assumes that the road information is accurate enough, which is not always true. A more promising approach is looping the reverse and direct tasks. At the first iteration in such a loop, inaccurate tracking information is used as guidance to get a better extracted road/trail from the image. At the next iteration, the road/trail extracted at the first iteration is used to improve tracking accuracy. Then this process can be repeated several times to get better consistency. The current common sequential (extract-match) process used in the MapSnap System does not correct the wrong feature extraction result obtained at the first stage. The new guided extractor GUEX algorithm does these corrections in the continuous loop. It generates new extracts by modifying previous ones and tests their structural and distance-based similarity with the generalized structure of the guidance at the different levels of detail and generalization.

Figure 1 illustrates typical challenges of guided feature extraction. The GMTI track contains data (“triangle”) that do not fit the shape of the trail (trail does not contain such “triangle”). Next the track offset from the road and its width does not represent the width of the trail. There are also multiple spots on the image that have similar appearance to the trail. Some of them can represent other trails or “branches” of this trail or can be spurious trails. Usually the number of GMTI or GPS points is less than an accurate vector road/trail would have while still showing a potential direction of the road/trail.



Figure 1. Data example

2. GUEX Algorithm

2.1. Main steps of GUEX algorithm

The main steps (MS) of the Guided Extractor (GUEX) algorithm are:

- MS1: Constructing the vector guiding line(s) for each road/trail from raster guidance. In the current implementation it is an external step—the algorithm assumes that the guiding lines are given. One of the common ways to design guiding lines is using the available tracking algorithms and raw GMTI detections.
- MS2: Structuring and optimizing guidance line (controlling the number and types of line segments).
- MS3: Searching extract candidates using guiding segments in the multi-objective optimization process.
- MS4: Linking best candidates for individual segments to form the extracted road/trail.
- MS5: Forward looking in the extract zone to find a point where guidance lines (tracks) do not follow the road/trail anymore (searching for the deviation point).

Searching extract candidates using guiding segments in the multi-objective optimization process includes:

- Computing Contrast and Uniformity scores for segments, and
- Using these scores in the Guided Extractor optimization process as objective functions and constrains.

The searching for the deviation point follows the same steps, but for the forward look segments (FLS). The FLS segment is a segment that was obtained by extending the extracted road segment to the same direction and tested to be a road segment by GUEX algorithm. The GUEX algorithm for extracting roads using multiple guiding lines is an extension of the basic GUEX algorithm for a single guiding line. It runs sequentially on the several guiding lines.

The algorithm for the main step MS2 (structuring and optimizing guidance line) allows extracting roads using a guiding line at a different (lower) resolution than the road on the raster. This decreases the requirements for the quality of the guiding line. The screenshots of the results are shown in the experiments section.

The algorithm of the main step MS3 (searching extract candidates using guiding segments in the multi-objective optimization process) is outlined below and illustrated in Figure 2. This algorithm uses the concepts of templates for the road, and right and left “shoulders” of the road as shown in Figure 2. These templates are rectangles of adjustable width and length to correspond to the width of the road/trail to be found. The joint road-shoulders template is moved in several iteration is a fixed image area with rotations to find an area in the image that is the best in two optimization criteria R and U defined later. The initial location of the template for the movement is at the center of guidance segment (Figure 2(a)). The initial orientation of this template is also the orientation of the guiding line. The initial length of the is the length of the guiding segment. The final optimized location of the template is illustrated in Figure 2(b). Other parameters of the template such as width are changed in the course of the optimization process.

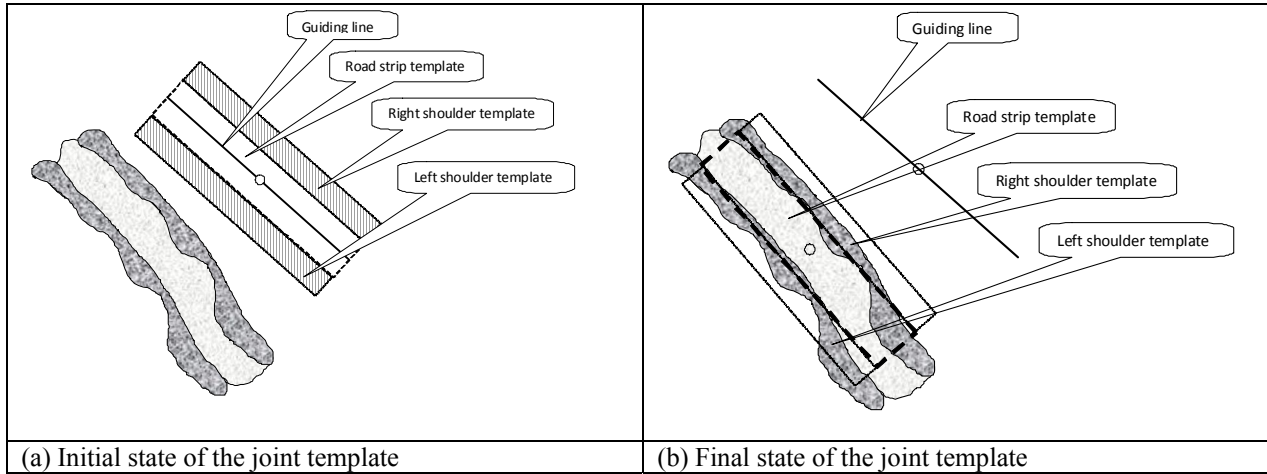


Figure 2. Illustration of GUEX optimization process

2.2. Definitions and optimization task

Below a road template is called a Center Strip Block (CSB) given by its parameter set α , and is denoted as C_{sb} . Similarly we define a left and right side blocks (shoulder templates), L_{sb} , R_{sb} as shown in Figure 2 with respective parameter sets β and γ . Thus, we have a triple $\langle C_{sb}(\alpha), L_{sb}(\beta), R_{sb}(\gamma) \rangle$ that we call a *joint template*. Next let $L(p)$ be the luminance of the point (pixel) p in the image.

We also need to introduce the notation for several other concepts:

A_{icsb} is the average luminance of the center strip block, CSB,

$$A_{icsb} = \left(\sum_{p \in C_{sb}} L(p) \right) / |C_{sb}(\alpha)|$$

A_{llsb} is the average luminance of left side block, LSB,

$$A_{llsb} = \left(\sum_{p \in L_{sb}} L(p) \right) / |L_{sb}(\beta)|$$

A_{rlsb} is the average luminance of right side block, RSB,

$$A_{rlsb} = \left(\sum_{p \in R_{sb}} L(p) \right) / |R_{sb}(\gamma)|$$

Using these averaged values we define:

the left center contrast difference (L_{ccd}),
the right center contrast difference (R_{ccd}), and
the contrast score (T):

$$L_{ccd} = A_{lcsb} - A_{llsb}$$

$$R_{ccd} = A_{ccsb} - A_{lrbs}$$

$$T = L_{ccd} - R_{ccd}$$

Next we introduce the notation:

$N_{MaxPixInBin}$ for the number of pixels in the max pixel bin for the center strip, and
 N_{AllPix} for the total number of pixels in the center strip.

The max pixel bin is a bin with the maximal number of pixels when all pixels from the center strip are distributed to 10 bins in accordance with the luminance level of pixel. We call this 10-bin Uniformity.

In this notation the *Uniformity* measure, U is:

$$U = N_{MaxPixInBin} / N_{AllPix}$$

The GUEX algorithm uses T and U functions defined above as optimization criteria. These functions depend on *joint template*, s and the *raster image*, r . each template is characterized by sizes of its components, the orientation angle and its location. Thus, we have $T(s, r)$ and $U(s, r)$.

In this notation the **GUEX Base Optimization Task (BOT)** is:

$$\text{Arg max}_{s \in \{S\}} T(s, r)$$

Under constrains

$$T(s, r) \geq T_0$$

$$U(s, r) \geq U_0$$

where T_0 and U_0 are predefined thresholds for T and U .

The **GUEX Multi-objective Optimization Task (MOT)** is:

$$\text{Arg Pareto max}_{s \in \{S\}} \langle T(s, r), U(s, r) \rangle$$

Under constrains

$$T(s, r) \geq T_0$$

$$U(s, r) \geq U_0$$

Here Pareto max is a set of all pairs $\Pi = \{ \langle T(s, r), U(s, r) \rangle \}$ that form a Pareto border and satisfy threshold constrains on T and U . In other words, the solution of MOT is a set of pairs $\Pi = \{ \langle T(s, r), U(s, r) \rangle \}$ such that for every pair $\langle T(s_i, r), U(s_i, r) \rangle$ there is no other pair $\langle T(s_j, r), U(s_j, r) \rangle$ for $s_i \neq s_j$ such that $T(s_j, r) > T(s_i, r)$ and $U(s_j, r) > U(s_i, r)$, or $T(s_j, r) = T(s_i, r)$ and $U(s_j, r) > U(s_i, r)$, or $T(s_j, r) > T(s_i, r)$ and $U(s_j, r) = U(s_i, r)$.

The multi-objective algorithm (GUEX-M) consists of two major steps:

MO1: Solving MOT, that is finding a set of all pairs Π and a set of their templates S_Π .

MO2: Generating a final solution from S_Π , that is producing final road templates and extracts.

The step MO1 can be solved by a brute force algorithm that loop through all template alternatives in the vicinity of the guiding segment. There are also algorithmic strategies that help to reduce computation load. These strategies are discussed in the next section.

The first step of MO2 is to analyze S_{Π} that is finding how close locations, sizes and orientations of these road templates. If these templates are close under some threshold or a clustering algorithm then a new road template is produced by using the weighted averaging of them. For instance if the centers of templates s in Π are c_1, c_2, \dots, c_n , then a new center with weights w_i is

$$c = \left(\sum_{k=1:n} w_k \cdot c_k \right) / n$$

The weight w_k are assigned by using values of R , $w_k = T_k / (\sum_{i=1:n} T_i)$, where $\{T_i\}$ are all T values in Π . Similarly are computed, the length, orientation and other parameters of the template.

If templates from Π do not form a single cluster and differ significantly then all of them are analyzed jointly with templates and extracts produced for other guiding segments of the whole guiding line or a set of guiding lines if available. This analysis algorithm tests their consistency such as locating in the close vicinity from each other along with their values of R and U . For the most complex cases the algorithm outputs for the human analysis. In this Pareto approach we avoid premature a priori assigning weights to R and U criteria without knowing actually disposition of extracts.

The multi-objective algorithm GUEX-M has an advantage relative to a more traditional way to solve MOT by combining the heterogeneous criteria T and U to a single cost function with weights assigned to R and U in advance before the actual dispositions of road templates is found. This makes GUEX-M algorithm more meaningful, because it avoids premature assigning weights.

2.3. Decreasing computational load

Below we discuss two strategies to decrease the computation load for the MO1 step. The first one uses the Genetic Algorithm (GA) approach, and the second one uses a modification of the GA approach that takes into account the specifics of the Guiding Extraction task to produce a faster algorithm.

The objective functions T and U are extremely non-linear, non-monotone, and have no obvious analytical form. Thus, the classical optimization algorithms that use linear or quadratic objective functions have little to offer to solve this task leaving a room for the GA algorithms.

The main steps of GA algorithm for the GUEX task are:

GA1. Generate randomly q road templates $\{s\}_1$ (initial population) with different parameters (different sizes and orientations) in the fixed vicinity of the guiding segment and computing $T(s,r)$ and $U(s,r)$ for all of them.

GA2. Generate a new set of templates $\{s\}_2$ (next generation) from $\{s\}_1$ by using a GA method by modifying (recombining and mutating) vectors of parameters of the templates and computing $T(s,r)$ and $U(s,r)$ for these new templates.

GA3. Select (filter) “best” templates from $\{s\}_2$ using $T(s,r)$ and $U(s,r)$ values through a GA fitness-based process by using $T(s,r)$ and $U(s,r)$ values as fitness measures.

GA4. Repeat steps GA2 and GA3 several times to get “best” templates (in $T(s,r)$ and $U(s,r)$ values) from all iterations.

The algorithm terminates after a given number of iterations (generations) or after reaching a satisfactory fitness level for the population of road templates.

Example. Let template s_{α} have a parameter set $\alpha = (40, 6, 3, 5, 100, 150)$, where 40 is the length of the template (in pixels), 6 is the width of the shoulder templates (width of the right and left shoulders is assumed to be equal), 5 is the rotation angle of the template relative to the North, and (100, 150) is the (x,y) coordinates of the center of the road template. Steps GA1-GA4 will produce several such templates with relatively high values of objective functions T and U

by modifying the parameter sets for these templates in the GA way. This GA way is not specific for the Guided Extraction task and is not the most efficient for this task. For instance, having parent vectors $\alpha = (40, 6, 3, 100, 150)$ and $\beta = (40, 4, 2, 10, 100, 150)$ the AG algorithm can swap their components and produce new children vectors: $(40, 4, 3, 100, 150)$ and $(40, 6, 2, 100, 150)$ without much justification for the guided road extraction.

A modified GA algorithm will not start from the random set of the templates in the vicinity, but from the specific template (seed): the road template with its center at the center of the guiding segment and oriented along the guiding line. See Figure 2(a). This template has a higher probability to match the road on the raster image r than a randomly selected template. More formally assume that there is a *probability distribution* P that gives the probability $P(s)$ of the road on the raster image to be at the template s location with the same orientation and width. This distribution P is likely unknown, but the *hypothesis* is that the values of the P close to the maximum of P are at the same locations, where objective functions T and U are close to their maximums. While P is unknown, some of its properties can be reasonably identified. For instance, it is reasonable to assume that the direction of the guided segment is close to the direction of the road in the raster. Otherwise it is hard to call such a line a guiding line.

If the object is moving on the road then the GMTI tracking data should mostly preserve a direction of the road with some distortion due noise and differences in resolution. Thus we should have $P(s_i) > P(s_j)$, where s_i follows the orientation of the guiding segment, but s_j follows the perpendicular orientation. These qualitative relations between probabilities are used to construct a faster GUEX specific algorithm that we can call GUEX-S.

Steps of GUEX-S:

GUEX-S1: Start with the road template s_0 with its center at the center of the guiding segment and oriented along the guiding line.

GUEX-S2: Generate randomly q road templates $\{s\}_1$ with different parameters (different sizes), but the same orientation as s_0 in the vicinity of template s_0 and computing $T(s,r)$ and $U(s,r)$ for all of them. The advantage of this step relative to step GA1 is that it is justified by the GUEX task. The randomness is also justified by the fact that the actual location of the road on the raster is unknown and random seeding can narrow the search area. The gradient approach can be too slow and too sensitive to the local noise.

GUEX-S3: Do random or regular minor rotation of templates $\{s\}_1$ relative to the guiding line to produce templates $\{s\}_2$. It is justified by a higher probability of the road in the raster to have the orientation that is similar to the orientation of the guiding segment. A drastically different orientation of the road in the raster (e.g., 90-degree rotation) has a lower probability.

GUEX-S4: Select (filter) “best” templates from $\{s\}_2$ using $T(s,r)$ and $U(s,r)$ values through a GA fitness-based process by using $T(s,r)$ and $U(s,r)$ values as fitness measures.

GUEX-S5: Generate randomly some templates and compute T and U values for them. If these values are much less than for the “best” templates are found, and the algorithm has produced a meaningful result. Otherwise repeat all steps starting with a changed seed template. The algorithm terminates after a given number of iterations (generations) or after reaching a satisfactory fitness level for the population of road templates.

The GUEX-S algorithm is a GA algorithm with operations that differ from traditional GA recombination/crossover and mutation operations. The move of the template along the guiding segment direction can be interpreted as a migration/colonization/extinction operation in terms of genetic algorithms.

2.4. Complexity of the base algorithm

Consider step MS3 of the GUEX algorithm. To solve the optimization task it is looping through n iterations. The complexity of this step of the algorithm is $O(n)$, where n is the number of triples (joint templates) $\langle C_{sb}(\alpha), L_{sb}(\beta), R_{sb}(\gamma) \rangle$. The parameters of each template as a rectangular strip are its center, orientation of the strip (angle to the North or to the orientation of the guiding segment), and length and width of the rectangle. Let k be the number of centers, and let d be the number of directions considered for a given template, then $n=k \cdot d$.

The value k is defined by the size w of the area (window) where it is changed and by the length of the step used to change the center of the template. Similarly the value of d is defined by the angle step and angle limits. In the current computational experiments we used $w \times w = 50 \times 50$ pixel window with one pixel step to change the location of the center of strip starting from the center of the guiding segment. This gives $50 \cdot 50 = 2500$ iterations for each segment. We also

used angle interval $[-10^0, 10^0]$ with 5^0 step, that is 5 alternatives $(-10^0, -5^0, 0^0, 5^0, 10^0)$ relative to the orientation of the guiding line segment. It gives $50 \cdot 50 \cdot 5 = 12500$ iterations for each guiding segment in this brute force algorithm. For the guiding line that we used with 7 segments it requires $7 \cdot 12500 = 87500$ iterations. Accordingly running the GUEX for $m=10$ guiding lines requires 875,000 iterations, with computational complexity $O(n \cdot m) = O(k \cdot d \cdot m) = O(w^2 \cdot d \cdot m)$. If w , d and m have the same order of magnitude this will lead to complexity $O(w^4)$. The size of the road strip also impacts the runtime. Currently it is the interval [4,6] pixels with 1 pixel iteration step, that is with $p=3$ iterations of road strip sizes. Thus, gives complexity $O(w^2 \cdot d \cdot m \cdot p)$. The computations for right and left shoulder strips add runtime too. Currently the width of each of these strips is a half of the road strip, that is 2 or 3 pixels. Together they add the same number of computations as the road strip. Thus, $O(w^2 \cdot d \cdot m \cdot 2p) = 87500 \cdot 2 \cdot 3 = 525000$. Next the computation on each strip depends on the size of the strip, $e \cdot h$ (the length e , and the width h of the strip). In the current experiments the e value were in the interval of [20,60] pixels, and width in the interval [4,6] pixels for the road strip and [2, 3] pixels for the side (shoulder) strips. The total width of road strip and two side strips is [8,12] pixels. Taking average length, $e=40$ and with, $h=10$, we get $O(w^2 \cdot d \cdot m \cdot 2p \cdot e \cdot h) = 525000 \cdot 40 \cdot 10 = 210,000,000 = 2.1 \cdot 10^8$ computations. Thus, the algorithm has a polynomial complexity as a function of the number of pixels involved. Unguided brute force extractors make similar computations on all pixels of the raster image. In example above the GUEX uses only about $(10+25) \cdot (40+25) = 10 \cdot 35 \cdot 65 = 22750$ pixels vs. all $400 \cdot 400 = 160,000$ pixels in the image, that is the GUEX uses about 14% of the image pixels.

The algorithm parameters that a user can change in the configuration file are presented in Table 1. The sample values in this table can be used as a default for the novice user.

Table 1. Guided Extractor configuration parameters

<i>Parameter</i>	<i>Sample value</i>
Rotation angle upper/lower limit ($\pm \Delta$ in degrees)	10
Rotation angular step (in degrees)	5
Lower limit on roadstrip width (in pixels)	4
Upper limit on roadstrip width (in pixels):	6
Containment Box X Size (in pixels):	50
Containment Box Y Size (in pixels):	50

3. Experiments

Results of several computational experiments with GUEX algorithm are presented below. In this work we used test data that represent a mountain trail, a rural area and a GMTI track that is offset relative to the trail/road. The goal is to extract the trail/road using the GMTI data as guidance. We used different fragments of the whole image as test data. The input raster data that can be used for this goal are color and panchromatic images in CIB, MrSID, geotiff, georeferenced bitmap and other formats. The expected vector data can be in a simple text format, shapefiles or in other form.

Figure 3 shows a test image with and without guidance. These data including base images in Figures 1,3-5,7-8 are courtesy of Lockheed Martin Co. Figure 4 shows the result that the GUEX algorithm produced where the blue lines indicate extracted trail.

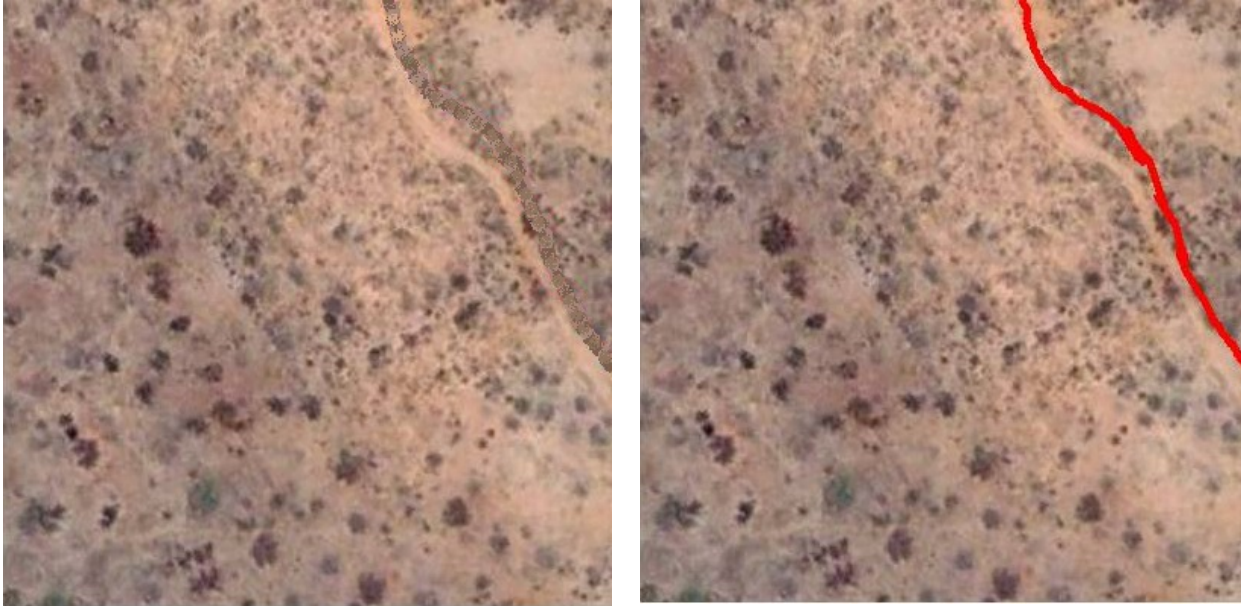


Figure 3. Test image 1 with and without guidance

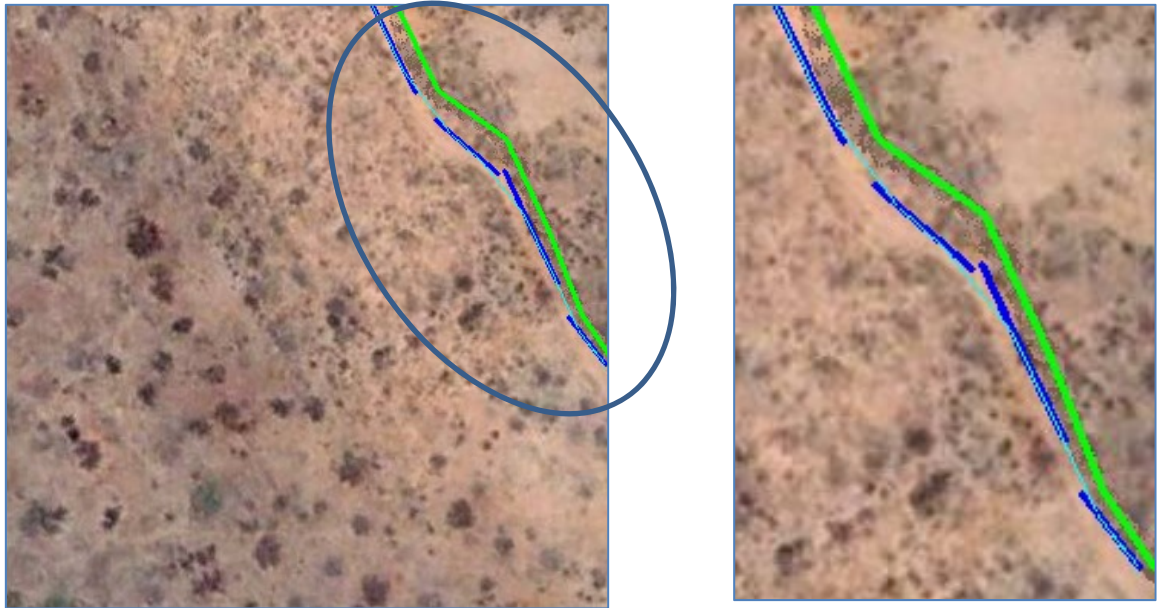


Figure 4. Test image with guidance and trail extraction results. Vectorized guidance – green line; intermediate extracts - dark blue lines, final extract- light blue line. A zoomed fragment is on the right.

Figure 5 shows the expanded area and results that the GUEX algorithm produced on this expanded area.

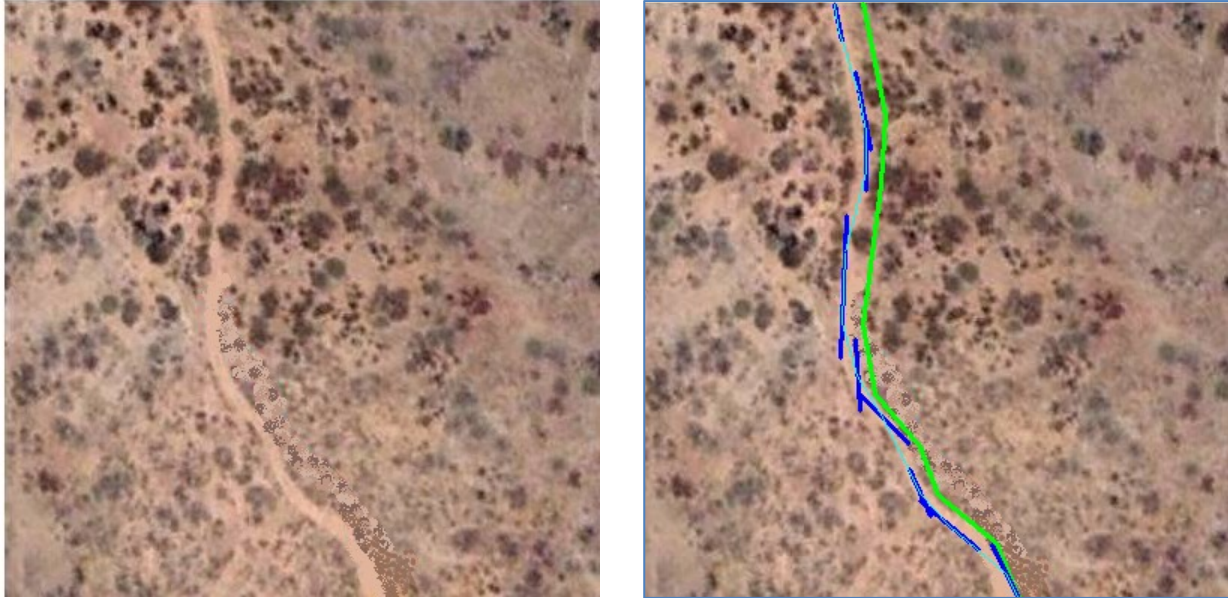


Figure 5. Test image 2 (expanded area) and results that the GUEX algorithm produced on this expanded area. Vectroized guidance – green line; intermediate extracts - dark blue lines, final extract- light blue line

Figure 6 shows another test image and results that the GUEX algorithm produced on this image. These data are courtesy of LongShortWay Inc. The guidance line was created from GMTI data. The GUEX algorithm and the program correctly found a matching road where people actually were walking on the road. The GUEX also identified the area where people diverted from the road.



Figure 6. Test image 3 and results that the GUEX algorithm produced on this image (green line -guidance, dark blue line - intermediate extract, light blue line - final result)

The next experiment with test image 1 was conducted to test the GUEX capabilities in extracting a trail using coarse guidance with 2-3 points only. Figure 7(a) shows guiding line with 2 points (a single segment), and Figure 7(b) shows the result for the guiding line with 3 points (2 segments).



(a). 2-scale results for the coarse guidance line that is a single straight segment.

(b). 4-scale results for the coarse guidance line with two straight segments.

Figure 7. 2-scale results and 4-scale results. Green- the guiding line, blue- the extracted line, magenta-forward looking segments, and yellow – preliminary detected split segments among the forward-looking segments.

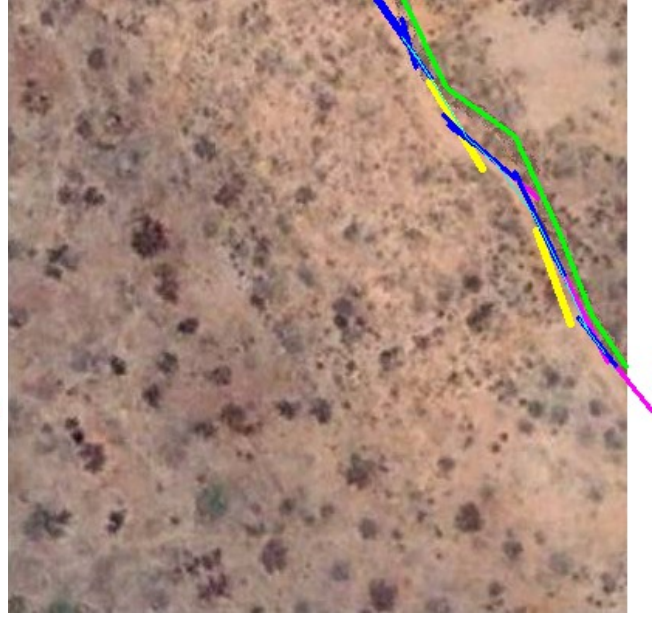
In case shown in Figure 7(a) the GUEX algorithm worked at two-levels and provided the best result from two scales. At the first scale, the whole guiding line was used as a seed template. At the next level (scale) GUEX split the guiding line into two half-segments in the middle of it and used each of them as templates. The GUEX found the matching trail segment for the second segment. This 2-scale result is incomplete and would need to be expanded to a 4-scale run. Splitting such segments leads to a 4-scale process with 4 sub-segments. In general, it can lead to 2^k segments, with the computation time growing proportionally to the number of sub-segments. Such 4-scale result is shown in Figure 7(b) where both the upper and the lower segments were divided into two sub-segments. This 4-scale run extracted both the upper and the lower parts of the trail.

Figure 8 shows the GUEX results for more refined guiding lines, that less coarse than shown in Figure 6. In the case shown in Figure 8 each of 4 original segments was split into 2 sub-segments in the middle. This is again a 2-scale process, where the original scale was converted to a scale with the doubled number of segments

While Figure 8(a) shows the results for a single guiding line that consists of 4 straight segments, Figure 8(b) shows the results for the same guiding data, but split into two separate sequential guiding lines (each line with 2 original straight segments). Thus, the implementation of the GUEX algorithm is capable of extracting roads and trails using multiple guiding lines that can be structured differently.



(a) Two-scale results for guidance as a single guiding line with 4 original straight segments.



(b) Two-scale results for guidance given as two separate sequential guiding lines (each with 2 original straight segments)

Figure 8. Two-scale results for less coarse guiding line. Green- the guiding line, blue- the extracted line, magenta- forward looking segments, and yellow – preliminary detected split segments among the forward-looking segments.

4. Conclusion and future work

This work had shown the feasibility of guided extraction of roads and trails using guiding lines from GMTI, GPS data and maps of relatively low quality and high uncertainty. On the algorithmic side the novelty of this work is in the combination of the template approach with the multi-objective optimization and genetic algorithms specific for guided feature extraction. The work in progress is the incorporation of the guided Extractor GUEX algorithm and software into the vector to raster fusion system MapSnap¹⁵. The future work is using other sources of guiding information such as Lidar, DEM, and others to guide the road/trail extraction and then to improve the tracking accuracy using roads and trails extracted in this way.

5. Acknowledgement

Partially supported by US Air Force Materiel Command AFLCMC/Hanscom AFB Contract FA8721-12-C-0019.

6. References

- [1] Agouris, P., Stefanidis, A., and Gyftakis, S., “Differential snakes for change detection in road segments,” *Photogrammetric Engineering & Remote Sensing*, 67(12), 1391–1399 (2001).

- [2] Baumgartner, A., Steger, C. Mayer, H., and Eckstein, W., "Automatic road extraction based on multi-scale, grouping, and context," *Photogrammetric Engineering & Remote Sensing*, 65(7), 777–785 (1999).
- [3] Barzohar, M., Cooper, D.B., "Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7), 707–721 (1996).
- [4] Bold, M., Weiss R., and Riseman, E., "Token-based extraction of straight lines," *IEEE Transactions on System, Man, and Cybernetics*, 19(6), 1581–1595 (1989).
- [5] Cao L., Stow D., Kaiser J., Coulter L., "Monitoring cross-border trails using airborne digital multispectral imagery and interactive image analysis techniques," *Geocarto International*, 22(2), 107-125 (2007).
<http://www.tandfonline.com/doi/pdf/10.1080/10106040701204073>
- [6] Dolan, J., Weiss, R., "Perceptual grouping of curved lines," *Proc. SPIE* 1002, 1-9 (1989). doi: 10.1117/12.960295.
- [7] Doucette P., Kovalerchuk, B., Kovalerchuk, M., Brigantic R., "An evaluation methodology for vector data updating," *Proc. SPIE* 7334, 73341F (2009).
- [8] Doucette; P., Grodecki, J., Clelland R., Hsu A., Nolting J., Malitz S., Kavanagh C., Barton S., Tang M., "Evaluating automated road extraction in different operational modes", *Proc. SPIE* 7334, 73341A (2009).
- [9] Gruen, A., Li, H., "Road extraction from aerial and satellite images by dynamic programming", *ISPRS Journal of Photogrammetry and Remote Sensing*, 50 (4), 11–20 (1995).
- [10] Hu X., Tao V., "Automatic extraction of main road centerlines from high resolution satellite imagery using hierarchical grouping," *Photogrammetric Engineering and Remote Sensing*, 1049-1056 (2007).
http://www.asprs.org/a/publications/pers/2007journal/september/2007_sep_1049-1056.pdf
- [10] Kaiser J, Stow D., Cao L., "Evaluation of remote sensing techniques for mapping transborder trails," *Photogrammetric Engineering & Remote Sensing*, 70(12) ,1441–1447 (2004).
- [12] Kim A.M., Olsen R. C., "Detecting trails in lidar point cloud data", *Proc. SPIE* 8379, 837906 (2012). doi:10.1117/12.918631; <http://dx.doi.org/10.1117/12.918631>
- [13] Kovalerchuk, B., Perlovsky L., Kovalerchuk M., "Modeling spatial uncertainties in geospatial data fusion and mining," *Proc. SPIE* 8396, 83960N (2012). doi:10.1117/12.920878
- [14] Kovalerchuk, B., Perlovsky, L., "Integration of geometric and topological uncertainties for geospatial Data Fusion and Mining," *Applied Imagery Pattern Recognition Workshop (AIPR)*, 1-8, 2011. doi: 10.1109/AIPR.2011.6176346
- [15] Kovalerchuk, B., Doucette P., Seedahmed, G., Tagestad, J., Kovalerchuk, S., Graff, B., "MapSnap System to Perform Vector-to-Raster Fusion," *Proc. SPIE Proc* 8053, 805306 (2011).
- [16] Kovalerchuk, B., Schwing J., (Eds). [Visual and Spatial Analysis: Advances in Visual Data Mining, Reasoning and Problem Solving], Springer, 600 p. (2005).
- [17] Lu, S., "Off-road Driving and Wildlife Trails Extraction from High Resolution Satellite Imagery: A Curvelet Transform Based Approach", University of Twente, NL (2012).
http://www.itc.nl/library/papers_2012/msc/gem/shaoqing.pdf
- [18] McGlone, C., "Automated Image-Map Registration Using Active Contour Models and Photogrammetric Techniques", *Proc. SPIE* 1070, 109 (1989). doi:10.1117/12.952498; <http://dx.doi.org/10.1117/12.952498>
- [19] McKeown, D.M., Denlinger, J.L., "Map-guided feature extraction from aerial imagery", Carnegie Mellon University, Technical Report (1984).
<http://repository.cmu.edu/cgi/viewcontent.cgi?article=2494&context=compsci>
- [20] Morris S., Barnard, K., "Finding trails," *Proc. Computer Vision and Pattern Recognition*, 1 – 8 (2008).
- [21] Park, J. M., Lu, Y., "Edge detection in grayscale, color, and range images", *Wiley Encyclopedia of Computer Science and Engineering*, 1–16, (2008). doi 10.1002/9780470050118.ecse603
- [22] Yang, C., Bakich, M., Blasch, E., "Nonlinear constrained tracking of targets on roads," *Proc. Information Fusion* (2005). doi 10.1109/ICIF.2005.1591860
- [23] Ziou D., Tabbone S., "Edge detection techniques: An overview", *International Journal of Pattern Recognition and Image Analysis*, 8(4), 537–559 (1998).