**Optimal Adversarial Location and Pathway Estimation Using Remotely Sensed Spectral-Terrain Data: A Graphical Modeling Approach**

Josef Affourtit and Nicholas Scott  
Riverside Research

Intelligence and Defense Solutions Directorate  
Dayton Research Center  
Beavercreek, OH 45431

**Abstract**

Military clients often possess high spatial resolution remote sensing data along with a need to accomplish certain objectives stemming from classified mission directives. One problem of significant interest stemming from these directives is the mobile missile problem (MMP). The constant movement of MMPs across variable terrains prompts a need to estimate optimal pathways given remotely sensed ground structural information. This problem can be addressed using optimization-based algorithms such as probabilistic graphical network analysis which when applied to remote sensing data can facilitate decision making in the form of rational environmental state assessment. In particular, reinforcement learning where human value assessment is interjected, can assist in finding optimal pathways where precious resources should be focused. A two-tier processor incorporating both feature extraction and optimization is used for this. In the feature extraction phase, spectral and terrain data are divided into image chips which are distilled in characteristic feature values which in turn are given a human value assessment. The image chip characteristic feature values are nodal states which are part of probabilistic graphical network which are connected via nodal-edge system. Optimal pathways over the value field are estimated in the second stage via maximization of an expected value criterion using reinforcement learning. The spatial optimization results gives leadership rational estimates of what adversaries should do given the accrued remotely sense information and human value assessment.

**Outline**

The following is the outline of the rest of this report:

**Introduction/Motivation**

**Statistical Processing Theory**

**Methodology**

**Results and Conclusions**

**References**

**Introduction/Motivation**

Mobile missile platforms (MMPs) in areas of strategic and tactical importance to military leadership require careful monitoring due not only to their huge destructive power but also to their ability to constantly move over large spatial regions. This mobility has promoted a need to exploit both sensing and algorithmic technology to characterize this motion so as improve counter-adversarial decision making. One method of predictive analysis which is conjectured to be helpful in this problem is optimal pathway estimation of these structures. In this situation, the question confronted is the ability to probabilistically predict the path that one or more MMPs will take given or conditioned on information obtained from remote sensing platforms such as satellite hyperspectral imagery (HSI).

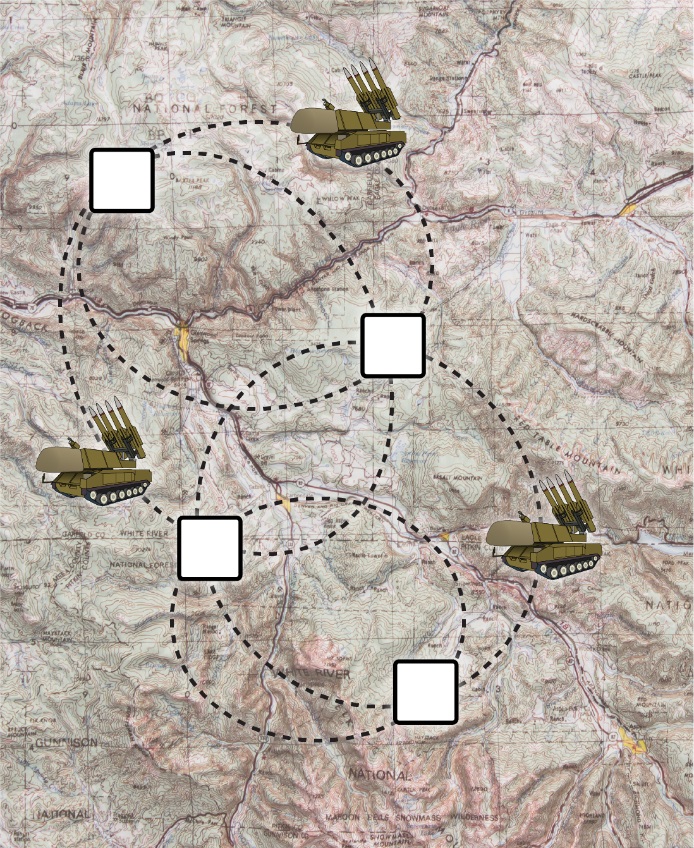


Figure 1: Mobile missile platform along with its stopping points and pathways.

Optimal pathway estimation is based on two tiers of signal and information processing. The first tier is based on feature extraction and value field estimation. In this stage spatial data acquisition and partitioning of the data into sub-regions or image chips is performed. The multidimensional spectral information in the image chips is then distilled into single characteristic spectral quantities which characterize each image chip in an average sense. The distilled image chip characteristic spectral features are transformed into a value field extending over the complete spatial domain of the data set. In the second tier, the estimated value field is transformed into a reward field which in turn is utilized by reinforcement learning allowing for simulation-based estimation of the optimal spatial pathway. This state-action estimator uses a probabilistic network and a trial-error search process which in the end yields the best pathway which ultimately emanates from value field and criteria imposed by military leadership.

**Statistical Processing Theory**

That statistical processing theory is the theoretical basis for the processing of the HSI image data that allows for both the construction of spectral features in the first tier and the estimation of the optimal path in the second tier. A brief synopsis of the estimation theory used in the algorithms is given below

1. *Principal Component Analysis*

Principal component analysis represents the HSI data spectral signatures as linear combinations of a small number of basis vectors. This method finds the projection that stores the largest variance possible in the original data and rotates the set of basis vectors such that maximum variability becomes visible. Geometrically speaking, PCA transforms the data into a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate, the second greatest variance on the second coordinate, and so on. If the data set X is characterized with D spectral dimensions and the aim of the PCA is the d- spectral dimensional reduced representation of the data set, the PCA works as follows: The corresponding d-spectral dimensional output is found by linear transformation, Y = Q X where Q is the matrix of linear transformation composed of the d largest eigenvectors of the covariance matrix and Y is the matrix of projected data set6.

1. *Automatic Target Generator Processor*

The purpose of Automatic Target Generator Processor (ATGP) is to find a spectral dictionary of all the linearly independent spectral signals which comprise the HSI image. The ATGP identifies spectrally distinct signals in HSI images by performing an Orthogonal Subspace Projection (OSP) on the data. In order to find spectrally distinct signals this method first finds the brightest pixel vector in an image, and then uses the OSP on that pixel vector allowing for the creation of subspace comprising a new set of basis vectors which are maximally orthogonal to the pixel vector . The algorithm proceeds to find a pixel which is the brightest pixel vector in the newly generated subspace and then generates the subspace which is maximally orthogonal to the newly generated pixel vector. This process is repeated by performing the OSP on both pixels and finding a third pixel orthogonal to the subspace . This process repeats until a user-defined number of spectrally distinct signals are found, each orthogonal to the newly accrued subspace. Each spectral endmember is the maximally orthogonal to the other spectral endmembers in *U* and it is assumed that the estimated spectral endmembers are linearly independent of each other7. The linear combination of all the spectrally distinct signals can then form any pixel in the image. Mathematically, this concept can be expressed as

**.**  (1)

The value is an abundance value associated with its respective spectrally distinct signal for a given pixel or spectral signature .

1. *K-Means Spectral Clustering*

K-Means represents HSI data as clusters about a user-defined number of endmember spectral signatures. The K-Means algorithm estimates spectrally distinct spectral signal representations of the HSI data by finding endmembers of distinct spectral data clusters, or more simply vectors which represent many other spectral signatures near it in the data. This method works via minimizing a cost function containing the Euclidean distance between two data spectral signals. Working iteratively this algorithm first takes a user-defined matrix containing the user’s initial estimate for endmember spectral signatures, and calculates the Euclidean distance between each estimate and all the data spectral signatures. The algorithm moves the initial estimates towards the closest data spectral cluster, i.e. towards the data spectral signature or vector with the minimal cost8.

Mathematically, the cost function is

(2)

where *N* is the number of spectral data vectors, *m* is the number of clusters (i.e. the number of user-defined estimated endmember vectors), is a spectral data vector or signature from the original data set, and is a user-defined estimated endmember spectral signature. The algorithm assigns every data spectral signature to the nearest cluster that is represented by . It then computes the mean of all the data spectral signatures in a discrete cluster creating new values for the endmember spectral signature. The nearness of the data spectral signatures to are computed again causing a redistribution of the data cluster membership via changing endmember spectral signature . The algorithm finishes upon completing a loop without any reassigning of the endmember spectral vectors, returning a matrix of endmember spectral vectors which are spectrally distinct and describe the data for their respective clusters. These endmembers represent spectrally distinct signals which are in most cases but not definitively linear independent. The linear combination of spectral endmembers is used to form any pixel in the HSI image. Mathematically this can be written as

**.** (3)

The value is an abundance value associated with its respective spectrally distinct signal.

1. *Basic Sequential Algorithmic Scheme*

The Basic Sequential Algorithmic Scheme (BSAS) is a method for reducing the spectral signal signature structure of an HSI data set. It runs a single pass on a given learned target dictionary, creating a more accurate and refined spectral dictionary. BSAS loops through every data vector one time, and compares the vector to its nearest endmember. If the Euclidean distance is larger than a user-defined threshold, then BSAS labels that data vector as an endmember and adds it to the spectral dictionary. If the dictionary already has a maximum number of endmembers (user-defined) or the Euclidean distance is smaller than the threshold value, then BSAS assigns the data vector to its nearest endmember.

For further refinement of the spectral dictionary, BSAS can also re-estimate endmembers values based on the data vectors which have been added to the cluster. Thus, when a data vector is assigned to an endmember the value of the endmember becomes the mean between the two. Additionally, BSAS can merge endmembers that are highly similar to each other, combining similar clusters and creating a new endmember as the mean of all the data vectors in the cluster. In this research, BSAS is used after K-Means and ATGP to refine and simplify the target spectral dictionaries9.

1. *Orthogonal Matching Pursuit (OMP)*

Orthogonal Matching Pursuit (OMP) seeks to identify specific atoms in an overcomplete dictionary which comprise the spectral signals of an HSI data via abundance estimation. The method is a decomposition via the equation Y=WA where Y is a matrix of spectral data signatures, W is an overdetermined spectral dictionary matrix W, and A is a sparse abundance matrix. (A sparse matrix is a matrix in which most of the elements are zero). The OMP used in this research is therefore a [sparse approximation](https://en.wikipedia.org/wiki/Sparse_approximation) algorithm which involves finding the "best matching" projections of spectral signatures in HSI data onto the span of an over-complete (i.e., redundant) dictionary {\displaystyle D}. The basic idea is to approximately represent a spectral signal {\displaystyle f}from Hilbert space H{\displaystyle H} as a weighted sum of finitely many functions {\displaystyle g\_{\gamma \_{n}}} (called atoms) taken from {\displaystyle D}dictionary W. An approximation with {\displaystyle N}P atoms has the form

{\displaystyle f(t)\approx {\hat {f}}\_{N}(t):=\sum \_{n=1}^{N}a\_{n}g\_{\gamma \_{n}}(t)}

where an {\displaystyle a\_{n}}ais the scalar weighting factor (amplitude) for the atom {\displaystyle g\_{\gamma \_{n}}\in D}. Here y is a spectral signature. Normally, not every atom in {\displaystyle D}WWDW will be used in this sum. Instead, orthogonal matching pursuit chooses the atoms one at a time in order to maximally (greedily) reduce the approximation error. This is achieved by finding the atom that has the biggest inner product with the signal (assuming the atoms are normalized), subtracting from the signal an approximation that uses only that one atom, and repeating the process until the signal is satisfactorily decomposed, i.e., the norm of the residual is small, where the residual after calculating {\displaystyle \gamma \_{N}}an {\displaystyle a\_{N}} and is denoted by. If {\displaystyle R\_{n}}ResN+1converges quickly to zero, then only a few atoms are needed to get a good approximation to h. T{\displaystyle R\_{N+1}=f-{\hat {f}}\_{N}}he sparsity problem that orthogonal matching pursuit is intended to *approximately* solve is



{\displaystyle \min \_{x}\|f-Dx\|\_{2}^{2}\ {\text{ subject to }}\ \|x\|\_{0}\leq N,}with {\displaystyle \|x\|\_{0}} is the {\displaystyle L\_{0}} pseudo-norm (i.e. the number of nonzero elements of {\displaystyle x}A). In the previous notation, the nonzero entries of {\displaystyle x}A are {\displaystyle x\_{\gamma \_{n}}=a\_{n}}, and the{\displaystyle \gamma \_{n}}th column of the matrix {\displaystyle D}W is {\displaystyle g\_{\gamma \_{n}}}. Solving the sparsity problem exactly is [NP-hard](https://en.wikipedia.org/wiki/NP-hardness), which is why approximation methods like OMP are used.

1. *Nonnegativity Constrained Least Squares Matched Filtration*

Matched filter processing solves the Y = A\*X matrix equation for the spectral abundances A. Once a target spectral dictionary X is found from K-Means or ATGP, the image chip Y which has maximum variance from PCA is multiplied by the pseudo-inverse of X. Mathematically,

, where is the Moore-Penrose inverse of X, or the pseudo-inverse of X. The resulting matrix A is the abundance associated with each endmember vector, which satisfy each pixel’s linear combination equation for n spectral signatures,

(4)

Physically, the abundance value is the amount of one spectral signal present in a pixel. The nonnegativity constraint allows for physically realizable abundances which are positive11. Using the target spectral dictionaries and matched filter the physical environment of an HSI image can be accurately represented via calculation of abundance images where each image is associated with an endmember spectral signature.

To avoid negative abundance values, which are unphysical in a real-world model, an additional method is required. The method used in this research is Nonnegativity Constrained Least Squares, which solves the matrix equation finding only nonnegative abundances for each endmember.

1. *Laplacian Eigenmap*

Laplacian Eigenmap is a method for linear dimensionality reduction method. The algorithm works by first building a graph based on the k neighboring spectral signatures in the HSI data input space. It then performs a mapping to a new space which tries to optimally preserve the neighborhood structure of the spectral signature data. The mapping component of the algorithm is a linear approximation to the eigenfunctions of the Laplace Beltrami operator on the manifold embedded in the high dimensional vector space.

Given a set of input spectral signature data X where the columns are spectral signatures of dimension D, the adjacency graph of the data is calculated. The algorithm connects data points or spectral signatures with an edge if they are close to each other. The closeness is determined here based on the epsilon neighborhood approach. Weights are assigned to the edges and are calculated based on the use of a heat kernel. For each edge connecting two spectral signatures, the weight is calculated as follows

 (5)

where t is an input parameter that controls the neighborhood scale. The ith column of the data matrix X is the ith object .The algorithm then computes the eigenvectors and eigenvalues for the following generalized eigenvector problem.

 (6)

Where is the Laplacian matrix and M is the diagonal matrix where.

Column vectors are the solution of the generalized eigenvector problem ordered according to the eigenvectors. The mapped data points (i=1, 2, 3, …B) composing a Laplacian eigenface are calculated as follows  where A = (

1. *Q-Learning and SARSA*

Q-learning is based on dividing the system to be modeled into two parts – agent and environment. The agent, or in this case, the defense forces, is the entity which learns, whereas the environment is *where* the agent is learning, and *what* is being learned. Q-learning represents procedural learning methodologies which allow the agent to take action and learn how to accomplish a goal starting from a beginning state. The values associated with pathways provides feedback to the agent as to what is ‘good’ or ‘bad’, but no information is given as to what explicitly should actually be done, as is done in supervised learning. The algorithms are in essence trial-and-error learning methodologies which search for the optimal policy or the best action that should be taken when in a certain state. The optimal policy is obtained through maximization of some parameterization of *cumulative reward* which is accumulated over many state-action transitions from beginning to goal state.

Q-learning problem is defined by the current state s, and the current action “a”, along with a transition function from state s to new state s’, T(s, a) = s’. Here the states are specific nodal locations and actions are *potential nodal locations* which can be transitioned into. There is also an immediate value function R(s, a) ε r where t is a real number for each state-action pair. The value field is derived using the results from the first tier BNN processing. A value function Q(s, a) is defined which gives the discounted cumulative reward for taking action “a” from state s. The value function, in other words, is the expected return when starting in state s and following a policy p thereafter. This can be written as

 (7)

The policy p is a mapping from the states of the environment to actions to be taken when in those states. The subscript t denotes time with Rt, st, and at denoting reward, state, and action at time t.  is the expected value of rewards as transitions are made through different states s and actions “a” over time utilizing policy p. The solution of the Q-learning problem hinges on finding the matrix  that has the *maximum* value in utilizing policy p for Q for the states and actions of the problem.

The Q-learning algorithm used in estimating  is carried out by iteratively changing the state-action values based not only on reward values R, but on future values of Q as transitions are made from different state-action pairs. All values for the value function Q(s,a) are set to 0 initially. A state s1 is then initialized. A random policy is used to select an action a1 while in state s1 from the actions which are available. Note that the actions available in state s1 are simply all the actions which have finite immediate reward values R(s1,a). A transition is made to another state s2 from s1. The Q function is then updated during this transition by adding the immediate reward value associated with state s1 and action a1 to the maximum Q value associated with the new state s2. The immediate reward value R(s1,a1), which is assigned to the agent by the environment, comes from the *randomly* chosen action a1 while in state s1. The maximum Q value is associated with one of the available actions “a” that being in the new state s2 allows. This action is given the label a2. The update rule or policy can be written as

**** (8)

A randomly chosen action while in this new state s2 is performed again which provides a new action a3. The above process then repeats until the goal state is reached or until a set, high number of iterations is attained. This constitutes one episode which provides a matrix of new values for Q(s, a). Q(s, a), a utility function quantifying how good an action “a” is given a certain state s, is equal to the immediate reward for performing an action “a” in state s or r(s, a) plus the best value of rewards to come in the future when another state is transitioned into by a randomly taken action, “a”. The variable γ is a parameter between 0 and 1 which balances the effect of immediate reward versus the relative value of delayed rewards from the future.

The Q-learning algorithm continues with a new episode by producing a new initialization of state. The Q function values from the last episode are used as the new initial values of Q(s, a) during this new episode however. The iterative process above is performed again, and many episodes are cycled through until the Q function converges.

State-Action-Reward-State-Action (SARSA) is similar to Q-learning. The only difference is that the update rule or policy has the form.

** (9)**

With the convergence of Q-learning and SARSA to a result, defense force leadership is afforded the best simulation-based estimate of MMP pathways. This result represents optimized pathways based on uncertainty laden information distilled into a value field. It is the use of a reinforcement learning state-action estimator which allows the optimal filtering of an ensemble of possible pathways affording a solution. With a finite pathway solution set, defense force leadership possesses information from which rational decisions can be made in terms of the allocation of resources to address MMP movement.

1. *Dijkstra’s Algorithm*

For a given graph Dijkstra’s Algorithm finds the shortest path between two nodes. Given an undirected, connect graph, a source node, and a destination node, the algorithm works as follows:

1. Mark all nodes unvisited. Create a set of unvisited nodes.
2. Assign every node a distance from initial node, 0 for initial node and infinity for all other nodes.
3. Calculate distance from all unvisited neighbors to current node. Compare distance with current value and assign the smaller value.
4. Mark current node as visited and remove it from set of unvisited nodes.
5. If the destination node is marked visited or if the smallest distance among the nodes in the unvisited set is infinity, then stop.
6. Else, select the node with the smallest distance and set it as the new current node. Return to step 3.

This method follows the path with the smallest distances. For the purpose of this algorithm this is the smallest Q matrix values. This is not useful for the algorithm’s purpose, so the resulting Q matrix has to be manipulated such that “high value” rewards are represented as the smallest distances. An inverse of each individual nonzero value is sufficient to perform this.

**Methodology**

The signal processing and estimation methodology is a spatial optimization algorithm based on the components shown in Figure 2. Brief explanation of each part is given below.

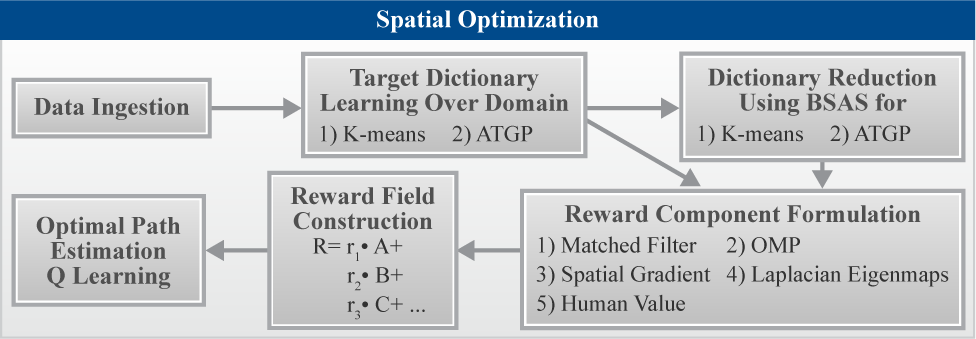


Figure 2: Feature extraction and optimal pathway estimation algorithmic methodology.

1. Data Ingestion, Image Chip Formation, Spectral Dimensional Reduction
   1. Data Ingestion

Using MATLAB 8.6, an HSI image cube is loaded as a three-dimensional matrix. Row and column relate to the 2-D position, or the pixel, in the image and third dimensional coordinate delineates wavelength. The sample HSI data used to test this algorithm has 256 wavelength bands and the value contained at a given row, column, and wavelength band is spectral radiance. A sample HSI data spectral signature is shown below in Figure 3.

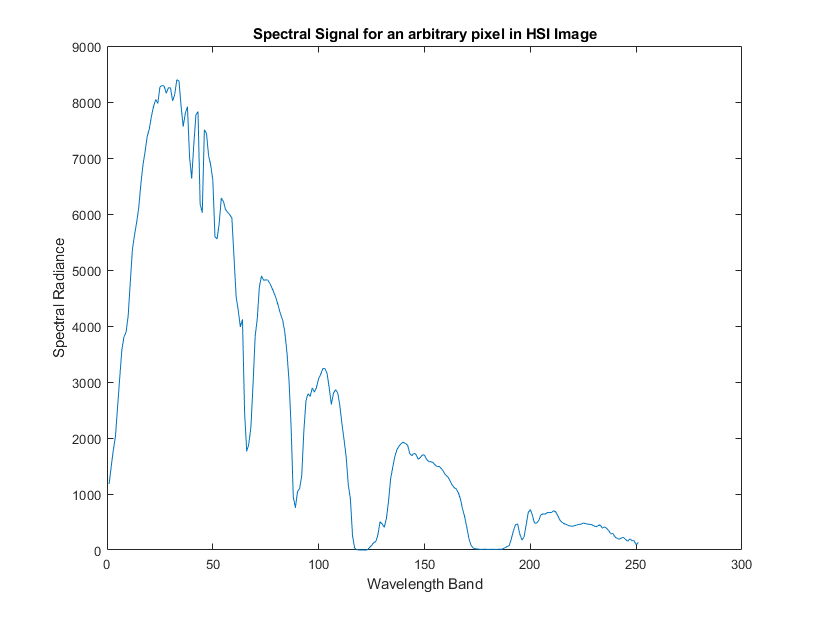


Figure 3: Sample spectral signature from HSI data cube.

* 1. Image Chip Formation

The algorithm cuts an image chip from the entire HSI image for processing.

* + 1. The location of the image chip is based on the iteration counter, moving through the rows of set columns of the image.
    2. The image chips dimensions are predetermined and set to 32x32x25 where each number designates row, column, and spectral depth. This size was optimized to increase the computational efficiency as well as the quality of the resulting dictionary.
  1. Spectral Dimensional Reduction

The HSI image in total contains on the order of pixels, each with 256 different wavelength band values associated with them. This size is intractable and it is also assumed that much of the information in the spectral dimension is noise. Thus, principal component analysis (PCA) is applied to the image chip. The resulting PCA image contains the maximum variance and information in the first few spectral bands. The PCA image is dimensional reduced from 256 wavelength bands (ordered such that the most variance is in the first band) to 20 wavelength bands. This new image contains all the information necessary to perform spectral target dictionary learning.

1. Spectral Target Dictionary Learning

The following unsupervised learning techniques are applied to the dimensionally-reduced image chip: automatic target generator processing (ATGP), K-Means clustering, and the Basic Sequential Algorithmic Scheme.

* + 1. Moving iteratively through the HSI image, the ATGP and the K-means method finds 7 or 13 spectral signatures for each PCA image chip with 20 spectral dimensions.
    2. The signatures are concatenated together into two large spectral dictionaries (one for the ATGP and the other for the K-means clustering) as each image chip is processed. The spectral dictionary size is 20 rows x 2205 or 20 rows X 4095 columns after the entire HSI data set is processed. The value of 2205 (4095) results from 7 (13) spectral signatures for each image chip being appended to the spectral dictionary for 315 image chips which comprise the HSI data.
    3. BSAS is then applied to these overcomplete spectral dictionaries, compressing the 2205 or 4095 spectral signatures into 7 (13). The final matched filter dictionary sizes are 20 rows x 7 columns or 20 row X 13 columns.

1. Value Field Construction

The Data Ingestion, Image Chip Formation, Spectral Dimensional Reduction, and the Spectral Target Dictionary Learning stage are all parts of a single processing iteration through the HSI data. A second and final signal processing iteration through the HSI image constructs the value fields. Using the same image cutting and PCA technique from the previous stage, five separate value fields are calculated for each image chip using the following methods:

* 1. Matched Filter Based Value Field Estimation

Each dimensionally reduced image chip is reshaped into a 20x1024 matrix, and abundance values are calculated by solving the Moore-Penrose equation for the abundance value estimates. This is done for each PCA HSI pixel using a spectral dictionary composed of spectral endmember signatures from the use of ATGP and K-Means clustering. The abundance estimates for image chip takes the form of a multidimensional abundance imagery cube which is 32 X 32 X NMF,A or 32 X 32 X NMF,A  where NMF,A equals the number of spectral endmember signature which is 7 or 13. The mean value for each abundance image associated with each spectral endmember signatures is calculated giving a characteristic abundance vector associated with an image chip. The spectral endmember signature associated with the maximum abundance value in the abundance vector is selected for each image chip pixel. The spectral endmember signatures are given integer value labels which represent characteristic spectral signature labels for the image chips. These are shown in Figure 4. Integer values provide a label as to which spectral signature in the spectral dictionary had the most abundance for the image chip. For instance, if the spectral signature associated with dirt is the most abundant then the image chip is classified as dirt with the corresponding integer value for dirt.

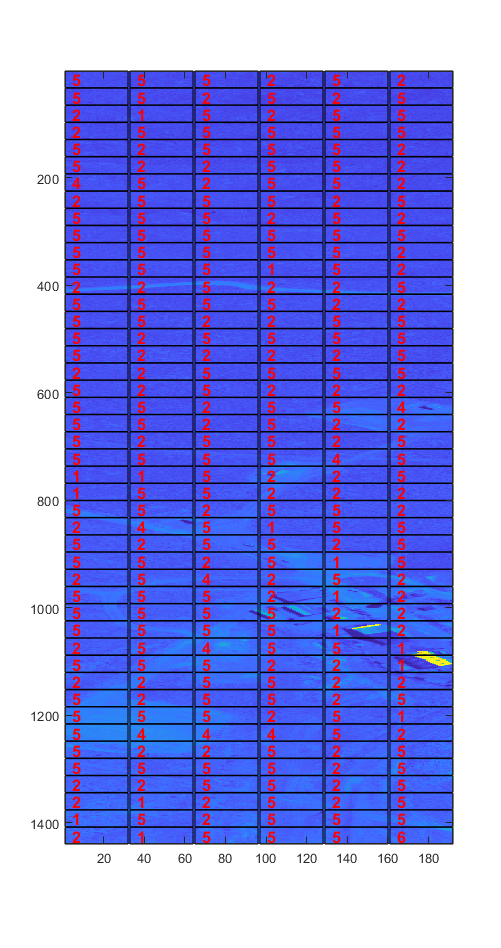


Figure 4: Example characteristic spectral endmember signature labels for image block subset HSI data. Spectral signature classification for 7 component spectral dictionary.

Following the characteristic spectral signature classification for each image chip, values are assigned to each chip based on the Mahalanobis distance classifier for each image chip characteristic spectral signature. The reference spectral signatures used is obtained from calculating the mode of the characteristic spectral endmember signature distribution. In other words, the mode of the characteristic spectral signatures designated by integer values, such as shown in Figure 4 for example, is used to calculated a Mahalanobis distance. The value field is then estimated as being inversely proportional to the Mahalanobis distance for each image chip. Thus, if a reference spectral signal for the Mahalanobis distance representing a roadway was chosen as the reference signal, then all the image chips which possessed characteristic spectral signals for roadways would have high values. An example of a value field calculated using a 7 component spectral dictionary and derived from the use of the ATGP and BSAS is shown in Figure 5. The value field has been smoothed using a Gaussian filter with variance value of 1.5.

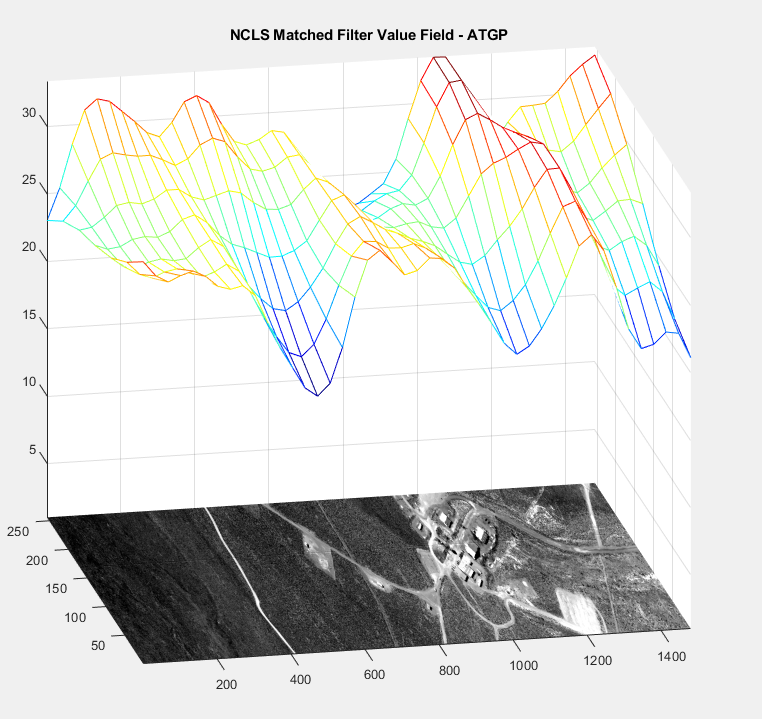


Figure 5: Example of matched filter generated value field using the ATGP.

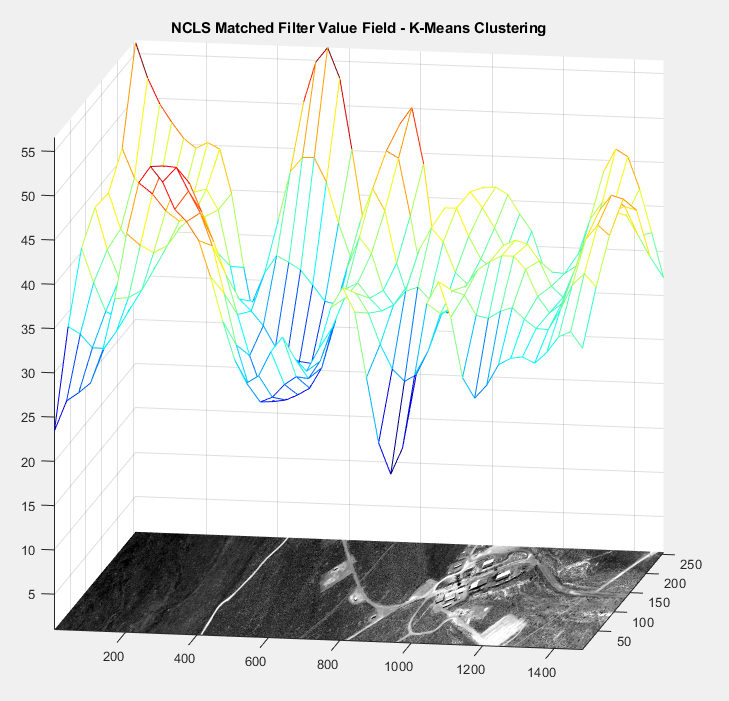


Figure 6: Example of matched filter generated value field using K-means clustering.

* 1. Orthogonal Matching Pursuit Based Value Field Estimation

Following the completion of the first iteration through the HSI data in which the ATGP and K-means clustering are used and prior to use of the BSAS which performs spectral dictionary reduction, two spectral dictionaries derived from the use of the ATGP and K-Means, respectively. These spectral dictionaries represent every spectral signature in the HSI image as each image chip is processed using the ATGP and K-means clustering.

During the second iteration through the HSI image, OMP is applied to each image chip using normalized spectral dictionaries obtained from the ATGP and K-means clustering. During the abundance estimation process associated with OMP, PCA image chips are reshaped to be 2 dimensional matrices, and a user-defined number of spectral signatures which have non-zero values (7) used to determine a multidimensional abundance imagery cube. This cube is 32 X 32 X NOMP,A in dimension where NOMP,,A is the number spectral abundance endmember signatures which is 2205. Because of the invocation of sparsity most of these abundance values are close to 0 where for any pixel only 7 of the abundance values are non-zero. For each pixel in the image chip, the maximal value abundance value is found along with the corresponding spectral endmember signature. The mode of this distribution of pixel spectral signatures for an image chip is found which is taken to be the image chip characteristic spectral signature.

Following the characteristic spectral signature classification for each image chip using the OMP, values are assigned to each chip based on the Euclidean distance classifier for each image chip characteristic spectral signature. The reference spectral signatures used is obtained from calculating the mode of the overcomplete characteristic spectral endmember signature distribution. In other words, the mode of the characteristic spectral signatures designated by integer values is used to calculated a Euclidean distance. The value field is then estimated as being inversely proportional to the Euclidean distance for each image chip. Thus, if a reference spectral signal for the Euclidean distance representing a roadway was chosen as the reference signal, then all the image chips which possessed characteristic spectral signals for roadways would have high values. An example of a value field calculated using a 7 component spectral dictionary and derived from the use of the OMP is shown in Figure 6. The value field has been smoothed using a Gaussian filter with variance value of 1.5.

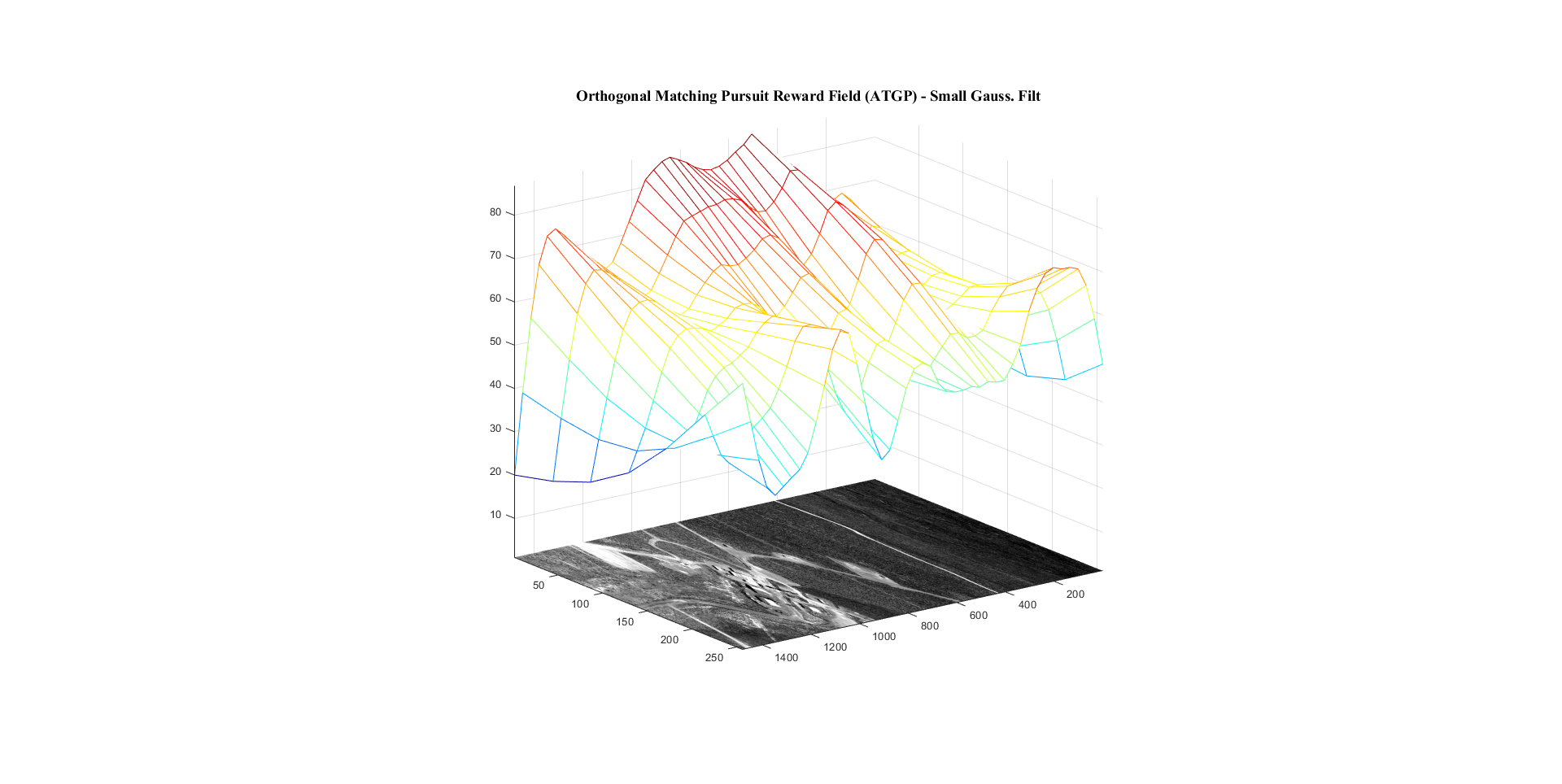


Figure 7: Example of OMP generated value field using the ATGP.

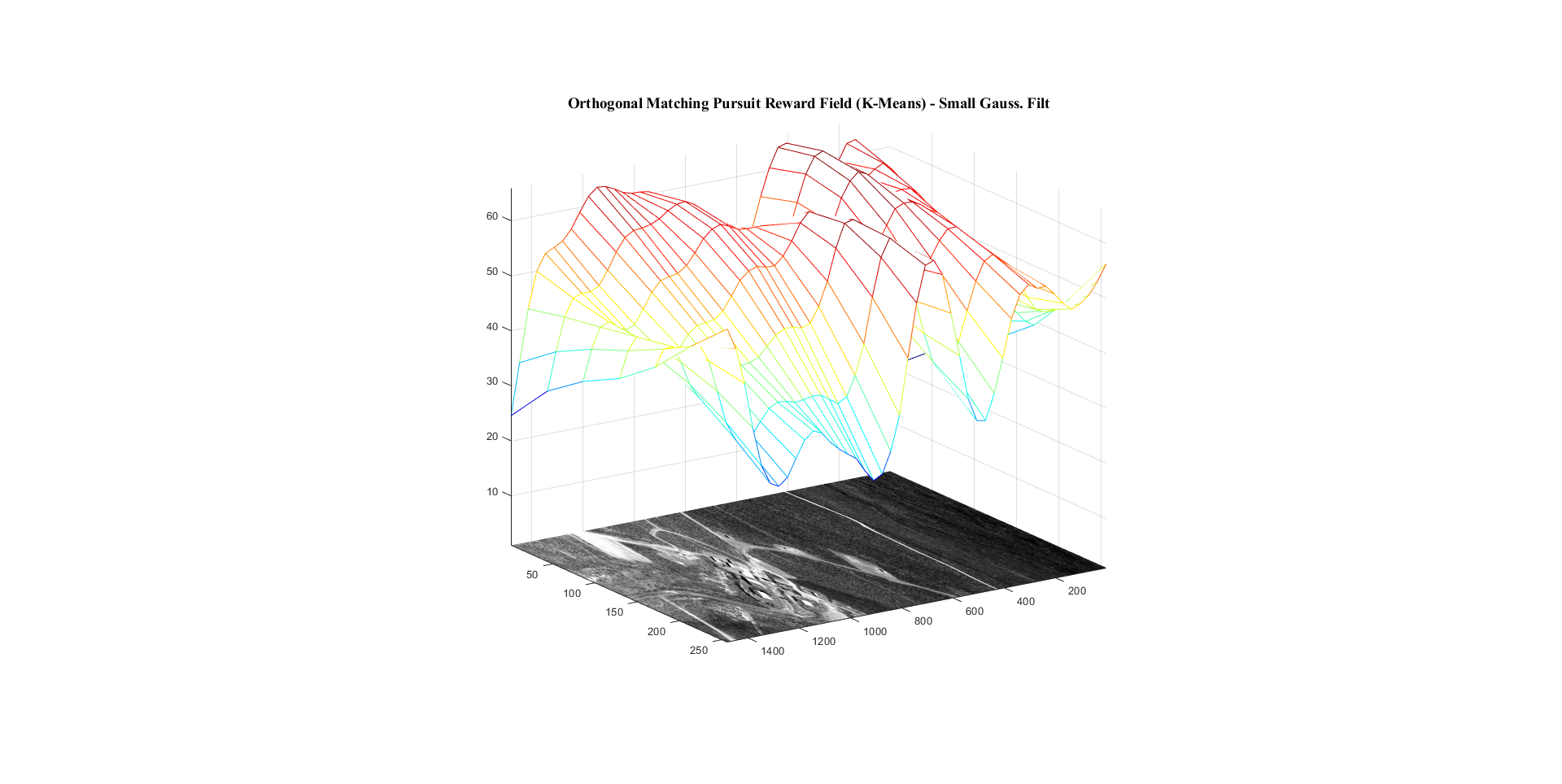


Figure 8: Example of OMP generated value field using K-mean clustering.

* 1. Spectral Spatial Gradient Based Value Estimation

Using the PCA image chips, the gradient is calculated for the first PCA band. The value field is taken to be inversely proportional to the spectral gradient. Thus, a very spectrally uniform image chip would produce a very small gradient value, and the value for that image chip would be high.

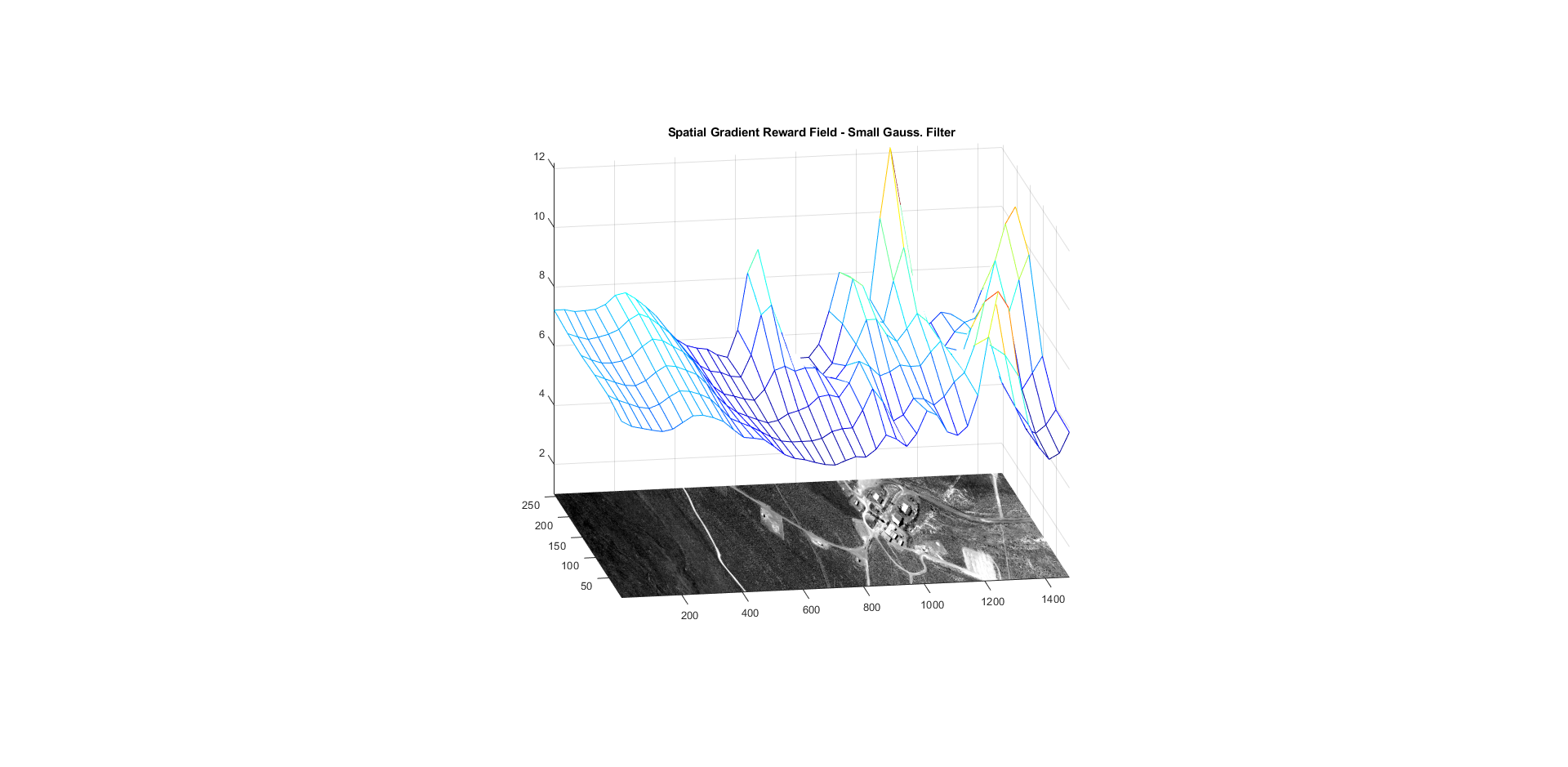


Figure 9: Example of spatial spectral gradient based value field.

* 1. Laplacian Eigenmap Kurtosis

A Laplacian Eigenmap is generated using the PCA image chip, containing only the first 20 wavelength bands. The Laplacian eigenfaces are ordered according to decreasing average kurtosis and the first Laplacian eigenface is extracted. The value for the image chip is taken to be equal to inverse of the kurtosis of the first Laplacian eigenface. Thus, an image chip which does not contain any outliers is given a high value, and a chip that contains an outlier is given a low value.

* 1. Human Value

Possibly the most influential component of value field construction is human intuition. Either by iterating through each image chip in the HSI data or manually selecting image chips, a human value is assigned to each one. A high value indicates an image chip which most likely contains characteristics an MMP would favor. A GUI interface for human value field estimation is shown in Figure 10.

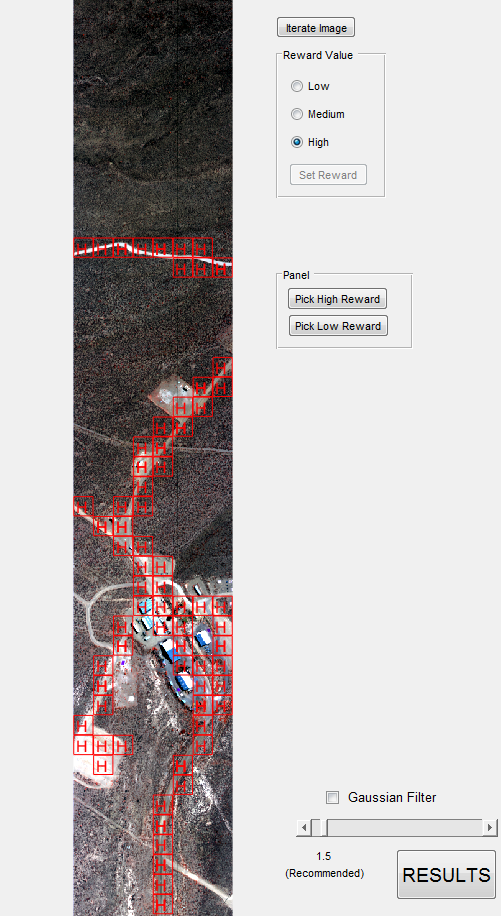


Figure 10: Example of human value field estimation GUI.

1. Q-Learning Optimization

Following construction of the value fields, the spatial optimization algorithm then begins reinforcement learning to determine optimal or most likely pathways that a MMP would take in an HSI image.

* 1. Instant Reward Matrix Construction

For Q-learning and SARSA to accurately predict the most rewarding pathways for an MMP to take, the value fields of the HSI image must be represented as a network of nodes and edges. This graphical network is shown in Figure 11. The image chips are represented as nodes in the graph, and the actions available are represented as edges in the graph. The instant reward matrix required by Q-Learning is a adjacency matrix for the graph seen in the figure. The values for each state (node) represent the rewards of taking an action (traversing an edge) into that state.

Spatial optimal pathway estimation emanates from Q-Learning and SARSA which in turn is contingent on estimates of the instant reward matrix.

The instant reward matrix (R matrix) is the adjacency matrix for the graphical representation of the value field for an HSI image. The structure of adjacency reward matrix is formulated is as follows:

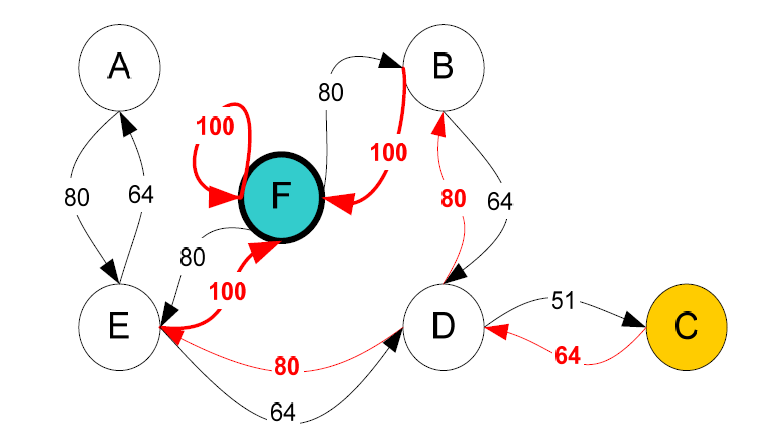
1. Rows denote the State. E.g., the 3rd row is the 3rd image chip in the HSI image.
2. Columns denote Action. E.g., the 3rd column is every action that leads the agent to the 3rd state.
3. A nonnegative value represents a connection between two nodal states.
4. A negative value represents no connection between two nodal states.
5. Values at R(i,j) represent the instant reward associated with taking an action from the nodal state **into** the nodal state.

The creation of the instant reward matrix, R is done in 4 steps:

1. Initialize a square matrix, with dimension size determined by number of nodes, to values of negative infinity (-∞).
2. Loop through the number of nodal states and if the current node has connections, set those connections values to 0. E.g., if the nodal state is a top left corner image chip then it is connected to the right, underneath, and diagonally to the bottom right. It’s connection to the neighboring right nodal state would be R(nodal\_state, nodal\_state+1) = 0. This is because if the agent is currently at this “nodal\_state”, it can take the action to “nodal\_state+1”.
3. Loop through the number of nodal states, and check the Value Field for values at each respective nodal state. If the value field is nonzero for the current nodal state then assign a reward value to each neighboring nodal state’s action which leads **into** the current nodal state. E.g., if the 3rd image chip has a value of 30, then the image chip immediately to the right (the 4th image chip), would receive an “instant reward” of 30 when taking an action from the 3rd image chip. So the representation in the R(s,a) reward matrix would be: R(4th chip, 3rd chip) = 30. In other words, the instant reward matrix, R(state, action), emanates from the value field. Each nodal state element of R possesses a reward attained by of virtue of mapping **into** that nodal state given the end result, Q(s,a). This idea is shown in Figure 12.
4. Finally, given a user determined end state, assign the maximum instant reward value for any nodal state’s action into that end state. E.g., R(321st image chip, 322nd image chip) = maximum value.



Figure 11: Node and edge graphical network for HSI data.



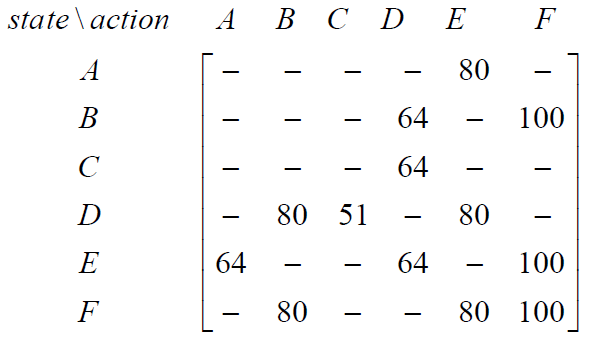


Figure 12: Illustration of the construction of the instant Reward matrix R(s,a).

* 1. Q-Learning

The instant reward matrix is akin to a guideline for performing Q-Learning. The only other parameters needed for performing reinforcement learning are the transition function T(s,a) and the learning rate γ. In this work two types of transition functions are used. The first one consists of a functional rule that allows nodal state transitions from downward, and diagonally only. No nodal transitions laterally along a row or backwards are allowed. The second transition function consists of nodal state transitions in any direction. The learning rate γ determines the value associated with importance of future decisions in the trial and error process reinforcement learning process and is a constant value with the bounds (0≤γ≤1). If γ ~ 0 then minimum value is placed on future seeking decisions, whereas γ ~1 implies there is no discounting for future decisions. The resulting Q(s,a) matrix is an updated reward matrix which holds the optimal path for the MMP to move through the graph.

Following the use of the first transition rule the maximum action policy is used to determine an optimal pathway. The freedom of transition afforded by the second transition rule does not allow for optimal pathway estimation but rather causing internal pathway loops which do not allow for beginning to end pathway estimation. The Dijkstra algorithm is used in the case to allow for beginning to end shortest pathway estimation from the Q matrix.

**Results**

A series of graphical user interfaces, value field maps, and optimal pathway imagery plots serve to illustrate the fundamental points emanating from the application of the feature extraction and spatial optimization algorithms.

1. *Graphical User Interfaces for Image Chip Feature Extraction Analysis*
2. *Principal Component Analysis and Spatial Gradient Analysis*

**

Figure 13: GUI illustrating PCA and Spatial Gradient for HSI image

A GUI was created to illustrate the preprocessing step of HSI data which is shown in Figure 13. The GUI allows for iterating through each image chip in the HSI image (image chips are depicted here as green squares superimposed onto the HSI on the left-hand side) and the application of PCA. The spectral signature for the image chip following PCA, a sample HSI spectral signature, the image chip undergoing PCA, and the spatial gradient map of the image chip are shown in the first row of the GUI. The first 12 PCA spectral band images are plotted in the second and third rows. The first Principal Component wavelength band contains the most signal variance in the image where a clear image of a building roof top is shown. The twelfth Principal Component wavelength band depicts image noise primarily. The appearance of noise near the 12th band is a clear demonstration of why the first 20 PCA spectral bands are used in the subsequent analysis rather than using all 256 available spectral bands.

1. *Abundance Estimation Maps from ATGP Dictionary Learning and Matched Filter; Abundance Estimation from K-means Dictionary Learning and Matched Filter*

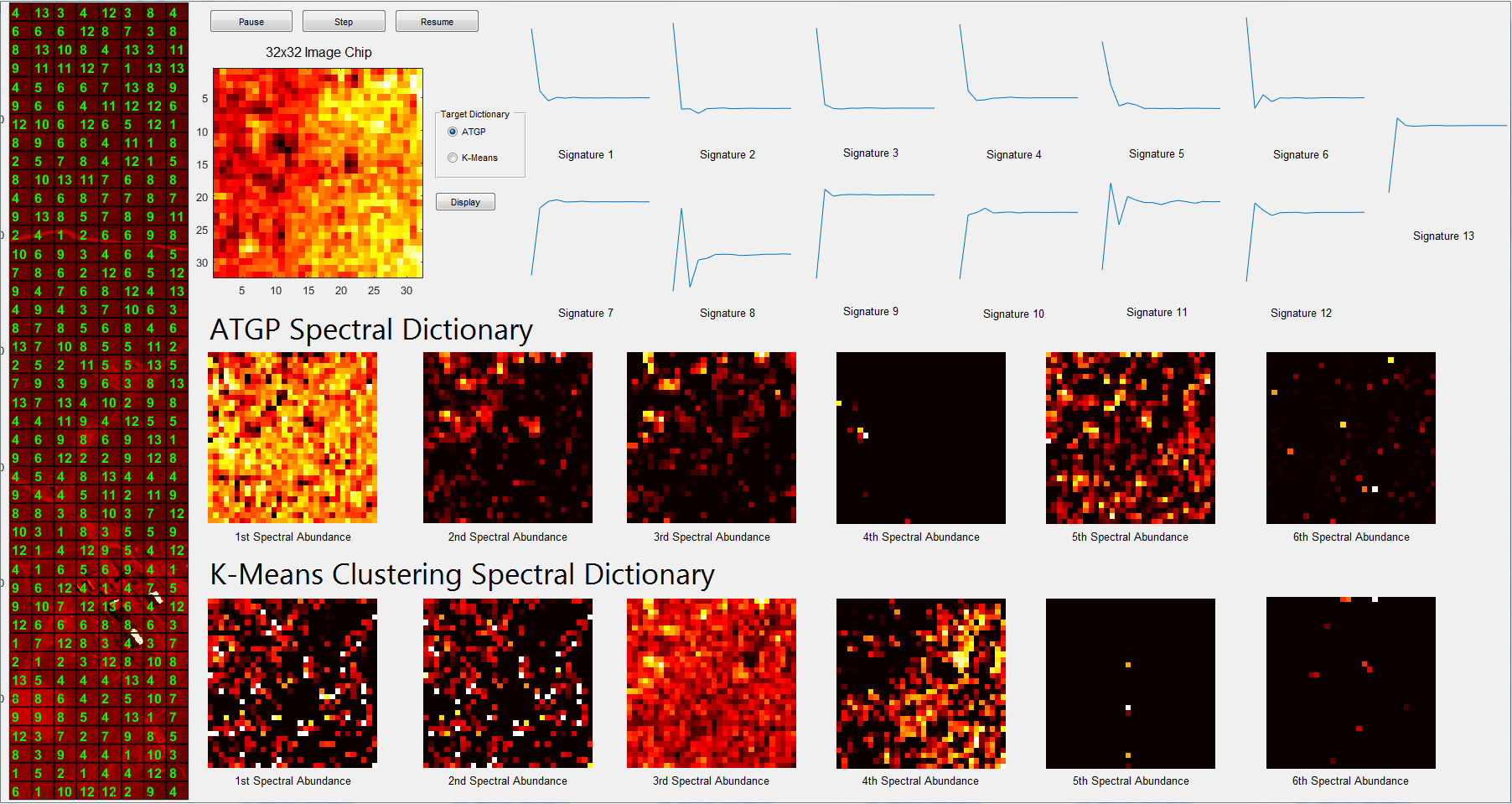
**

Figure 14: GUI illustrating abundance estimation based spectral labeling of image chips using ATGP and K-means clustering spectral dictionaries.

Iterating again through each image chip, the abundance values associated with each spectral signature are calculated. Shown on the bottom of the Figure 14 are the first 6 spectral abundances for both ATGP and K-Means. The spectral signature dictionary is show at the top of the figure. The individual pixels of the abundance images represent the abundances of the respective signatures in the dictionary for a pixel. The most abundant spectral signature in the spectral dictionary then becomes the label or characteristic spectral signature for that image chip. Each image chip, in other words, is assigned a label which in Figure 14 is a number between one and thirteen corresponding to one of the spectral signatures in the spectral dictionary. The characteristic spectral signature labels shown in Figure 14 on the left hands side are based on the abundance estimation which uses the ATGP spectral dictionary signatures and the nonnegativity constrained least squares matched filtration shown at the top.

Different spectral dictionaries varying in the number of elements were used in calculating characteristic spectral signatures for image chips. Characteristics spectral signature labeling of the HSI image using a 7 element spectral dictionary obtained from the use of the ATGP and the matched filter is shown in Figure 15.

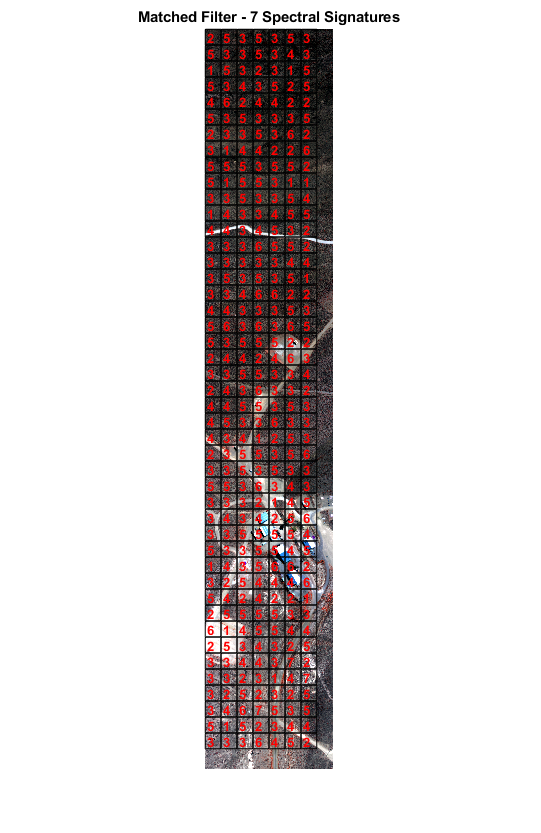


Figure 15: Characteristics spectral signature labels from nonnegativity constrained least squares matched filtration Matched Filter using ATGP spectral dictionary.

1. *Laplacian Eigenmap Analysis for Local Anomalies*

The Laplacian Eigenmap Kurtosis (LEK) GUI is illustrated in the Figure 16. The complete HSI data is shown with the image chips superimposed on the left-hand side. The image chip which is the Laplacian eigenmap is applied is shown to the right of the HSI data. Laplacian eigenfaces capture local structures smaller than a set scale threshold providing images associated with low variance eigenvectors which exhibit outlier kurtosis. A large kurtosis means there exists a sharp peak in the image chip’ Laplacian eigenface, thus there is something anomalistic in the projection of the HSI data in the direction the Laplacian eigenmap eigenvector. In this work, the first Laplacian eigenface is used in the estimation of the LEK. Sample Laplacian eigenfaces are shown in Figure 15 on the right hand side. The Laplacian Eigenmap manifold using the first and the second Laplacian eigenfaces is shown below the image chip.

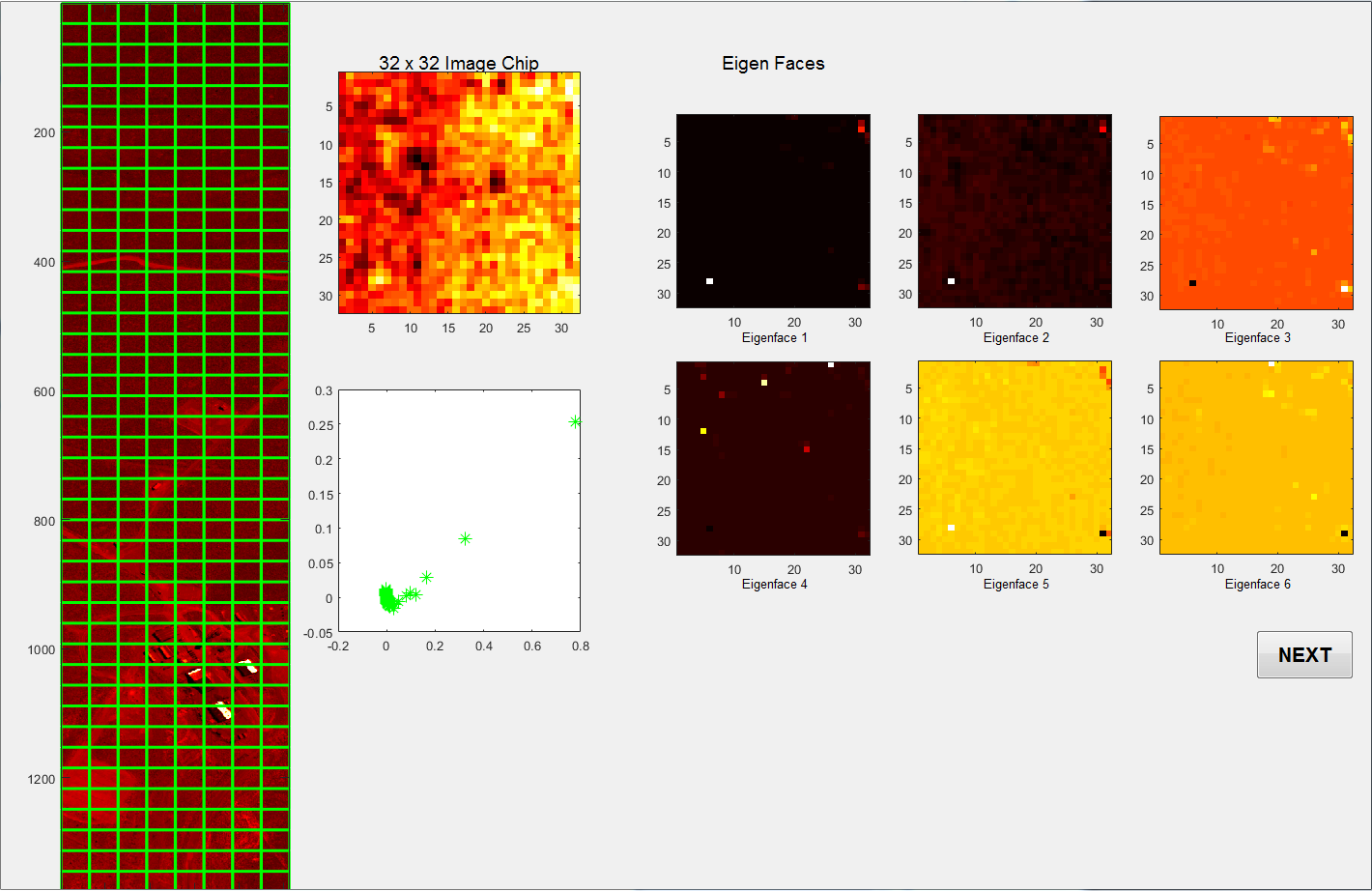


Figure 16: Laplacian Eigenmap GUI for the detection of local anomalistic features.

1. *Value Field Maps Examples*

Examples of value fields which are necessary components for the estimation of reward fields are shown below.

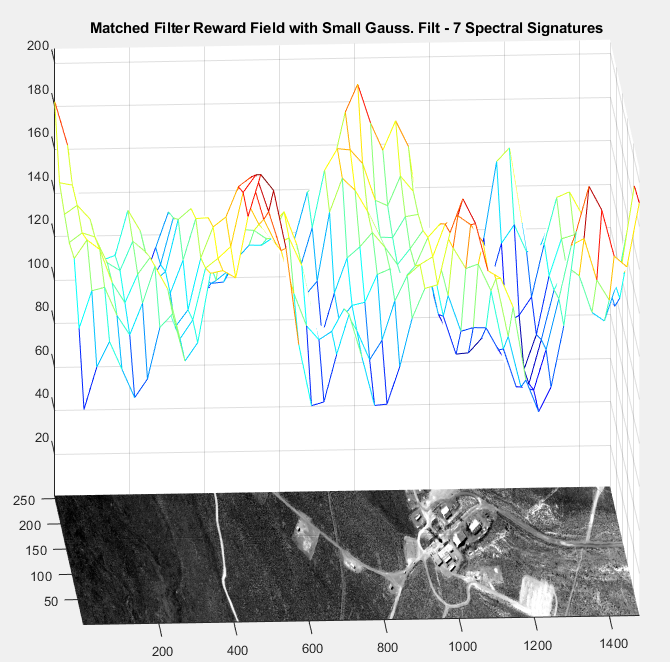


Figure 17: ATGP dictionary and matched Filter based value field.

Figure 17 shows the value field superimposed on top of the HSI dat. The value field is calculated based on the characteristic spectral signatures which are in turn estimated from the use of the 7 element ATGP spectral dictionary and the matched filter. The median PCA spectral signature for the entire HSI data is calculated first. This global HSI data characteristic spectral signature is the signature for dirt. The Mahalanobis distance (MD) of the median HSI data signature and each characteristic spectral signature obtained through the ATGP dictionary and matched filter abundance estimation is calculated. The value field for the HSI data set is taken to inverse of the image chip (MD). The resulting values are shown in Figure 17. The value field is extremely noisy exhibiting image chip value field inconsistencies which are attributed to the HSI data not being atmospherically corrected. For example, image chip in the area comprising dirt have various values suggesting different characteristic spectral signatures for the image chip. With better HSI data it is believed that the resulting value field would be smoother, and more accurate.

The value field obtained from the use of the LEK is shown in Figure 18. This figure clearly shows that the detection of a high kurtosis object, an ‘anomalistic road’ yields a local minimum or depression in the value field. This low value field area in turn could means that an adversary who is aware of such an anomalistic feature would possibly steer clear of the area.

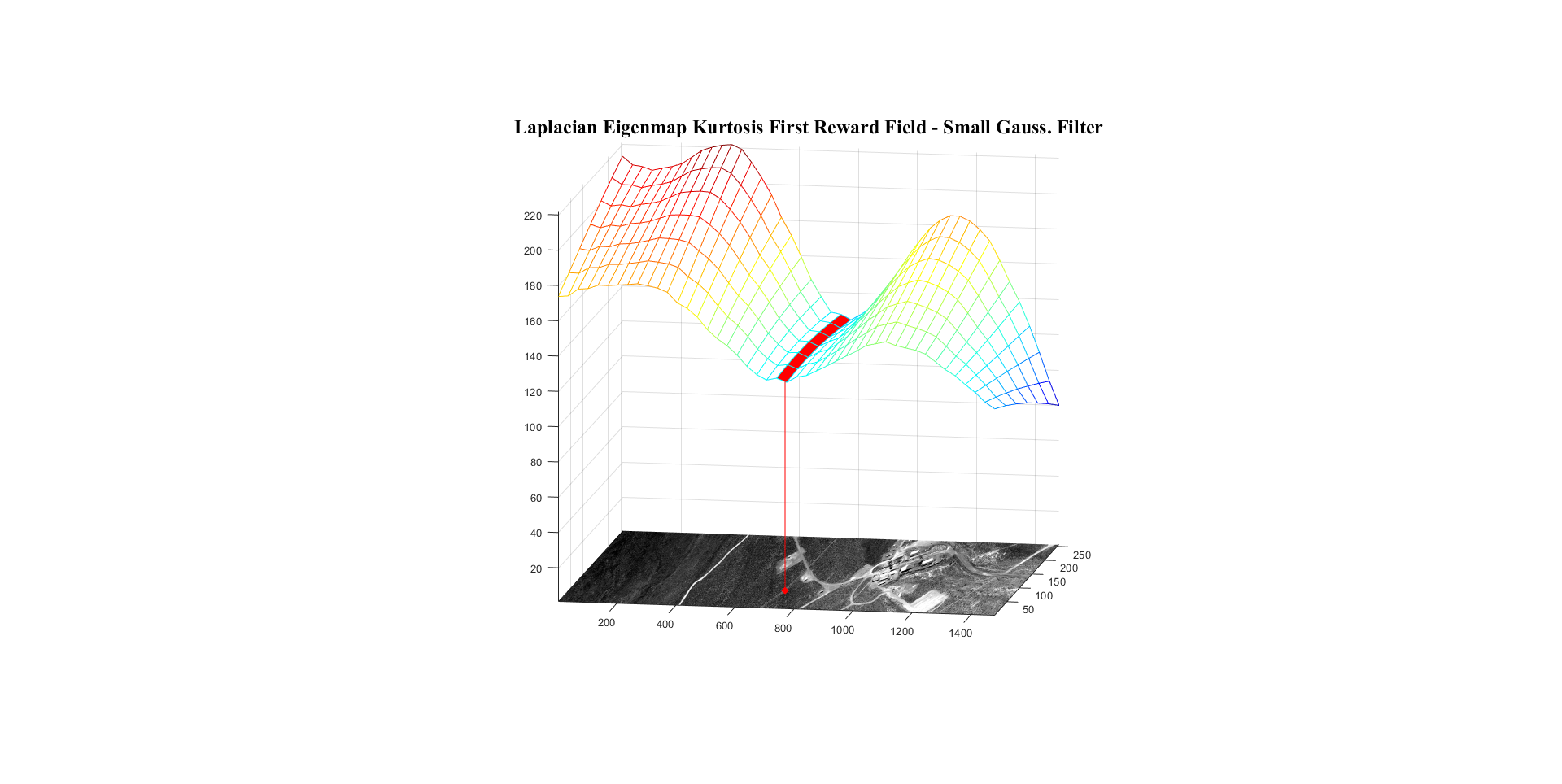


Figure 18: LEK value field illustrating detection of an ‘anomalistic road’ and it

resulting low value.

1. *Q-Learning and SARSA based optimal pathway estimation*

The output results of Q-learning and SARSA are depicted as pathways overlaid on top of HIS images. When applied to a toy case with no value field, Q-Learning finds an optimal path from image chip 1 to image chip 91 as shown in Figure 19. Note that the second transition function T(s,a) mention above is used which allows complete freedom of transition in any direction. Without a value field which provides more relative value to one image chip than another, logically the optimal path is moving down the rows without traversing any columns, then cutting diagonally to the right to finish at state 91. A simple proof of this path being the shortest would be explained using Pythagoreans Theorem.

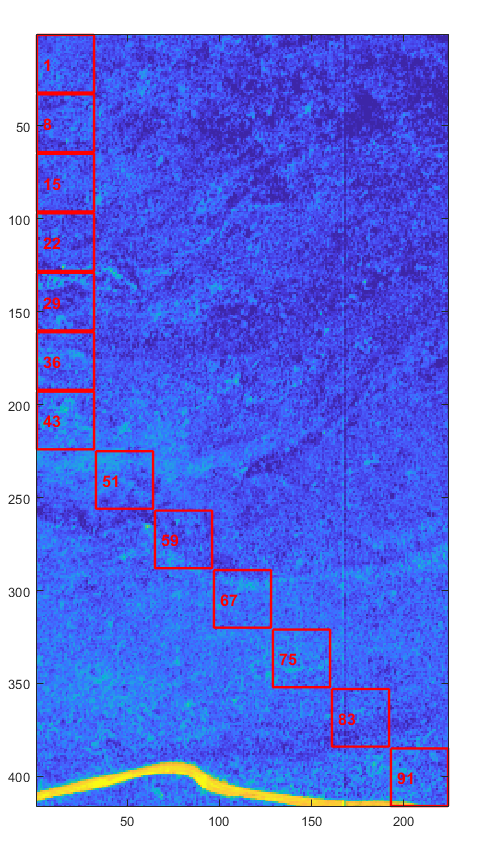


Figure 19: Q-Learning Optimal Pathway Toy Example

In Figure 20, the agent is tasked to move from image chip 1 (green) to image chip 98 (red). Q-Learning finds an optimal path that is the most rewarding for the agent to take. When a value field is used in the creation of the instant reward matrix for Q-Learning, the resulting path tends to follow the steepest gradient of the reward field. Given a large γ=0.9 value, the learning rate parameter allows for the Q-learning algorithm to behave less greedily, and the agent, in principle, moves less opportunistically and more prudently since future decisions are taken more into account. This result is important for predicting MMP movement, because it is assumed that an MMP motion would exhibit some degree of intelligence about its local terrain movement and will move accordingly.

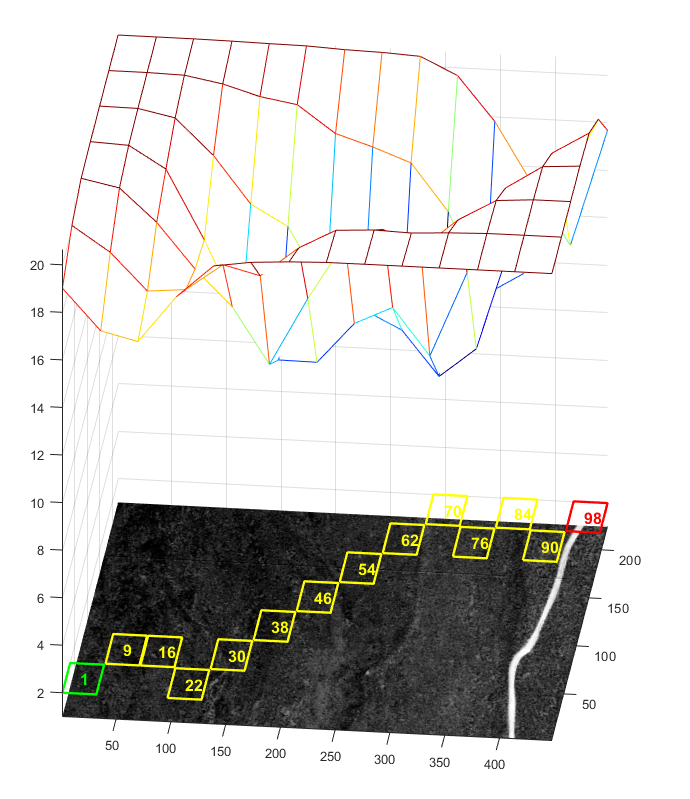


Figure 20: Q-Learning optimal pathway based on superimposed value field.

Using an entire HSI image as another example for illustrating the influence of the value field on optimal pathway estimation, optimal agent or MMP movement is shown for γ=0.9 case. Because of complete freedom of transition used for T(s,a) function, optimal paths found by first taking the most rewarding action from the initial state, and then the most rewarding action from that next state until the current state becomes the end state never provided a complete path to the end state. The agent would simply move between two states repeatedly around a local maximum. In this toy example, these singularities were avoided by imposing the aforementioned transition rule. Under this rule, two important heuristically nodal state transition laws are used when creating the graph to represent state-actions available to the agent. First, the agent was not allowed to move along its current row, i.e., traverse the columns of its row. Second, the agent could not move to a row which the path already traversed. The resulting path followed the value field sufficiently, although the results were influenced by an unrealistic set of nodal state transition rules.

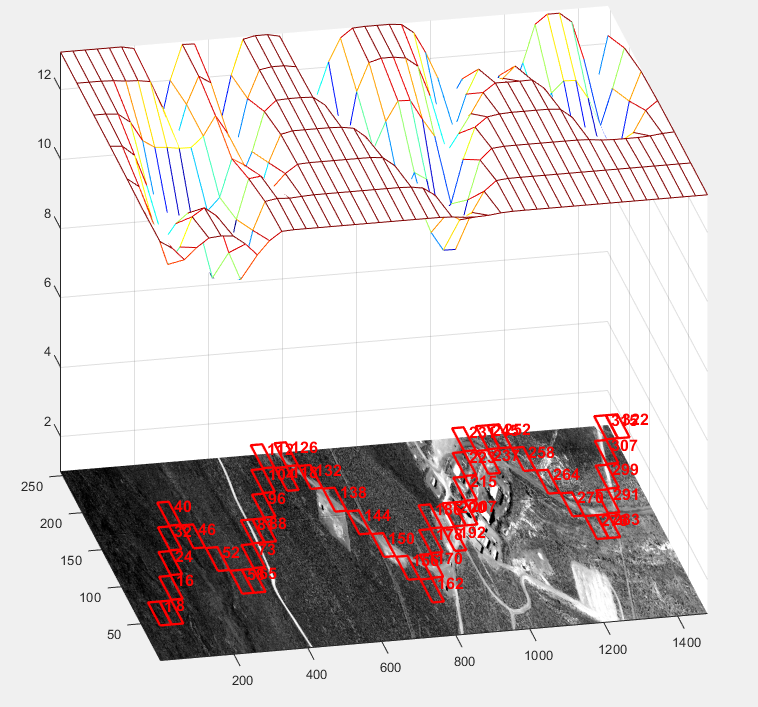


Figure 21: Q-Learning optimal pathway using heuristic nodal state transition rules for HSI image

Using a value field in the creation of the instant reward matrix, Q-Learning teaches the agent which actions are most rewarding in a forging a path from state 1 to state 322. Figure 22 shows an example of Q-learning when there are no heuristic nodal transition rules being imposed. The agent is free to move within the bounds of the HSI image in any direction. This then requires the use of a new path-finding algorithm based on the resulting Q matrix from Q-learning. Using Dijkstra’s algorithm to find the least distance optimal path, the resulting optimal path is shown in red in Figure 22. Note that the Q matrix stores larger values to represent rewarding actions to be taken by the agent and smaller values to represent less rewarding actions. In Dijkstra’s algorithm these values would represent large distances and the opposite effect would result. Thus, the Q matrix must be manipulated such that a rewarding action is represented by a small distance. Taking the reciprocal of each value in the Q matrix accomplishes this, and the result is that a small reward becomes a value close to 1, and a very large reward becomes a value closer to 0.

An optimal pathway that follow the value ridge is the result. Note that the downward dip near the lateral coordinate of x = 200 is avoided since the least number of steps is focused on.

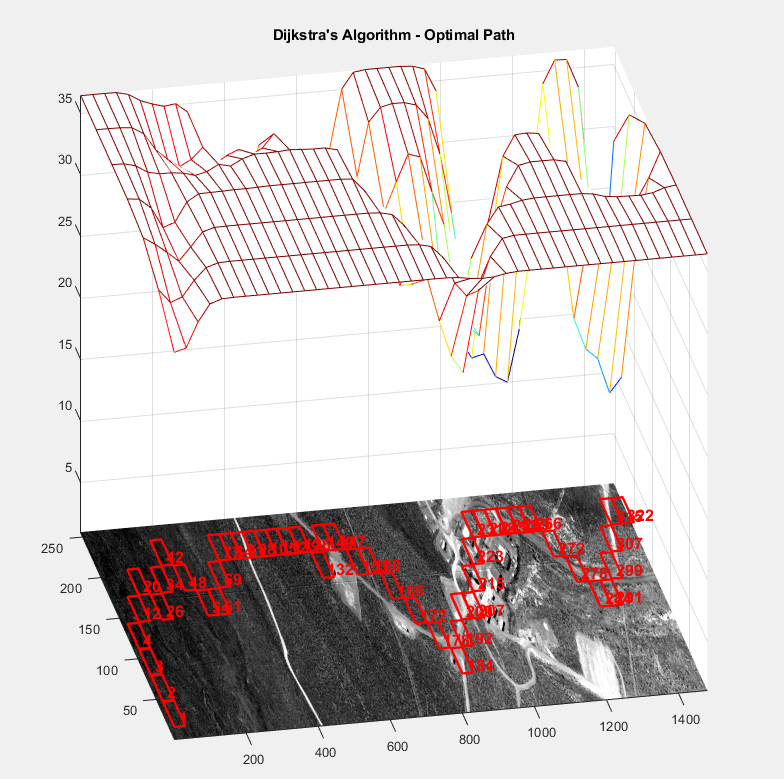
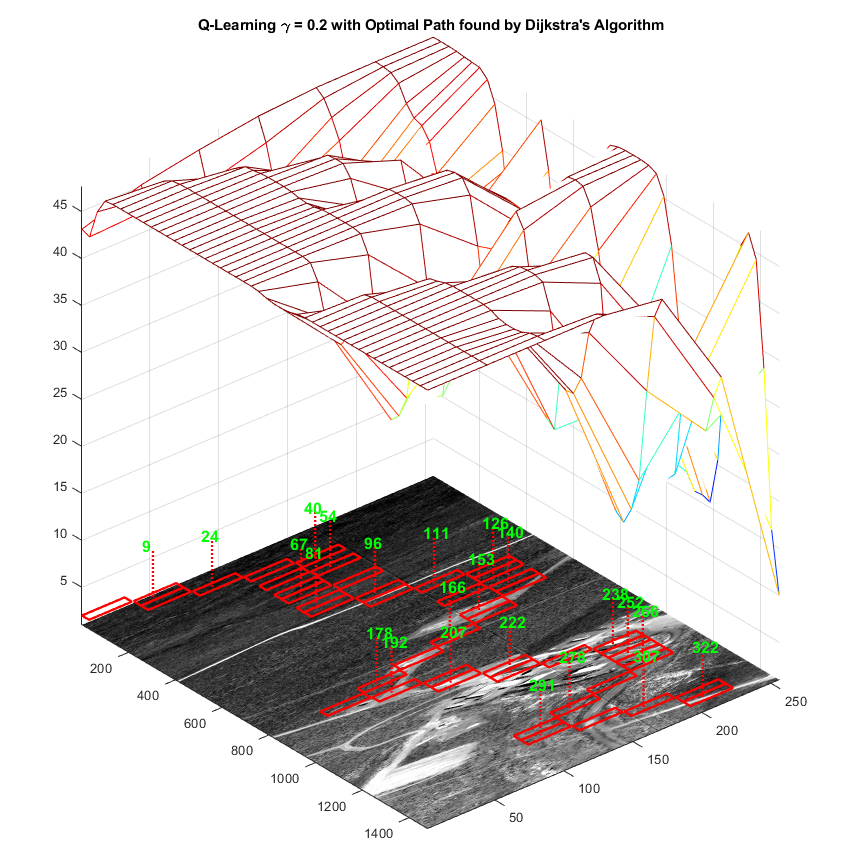
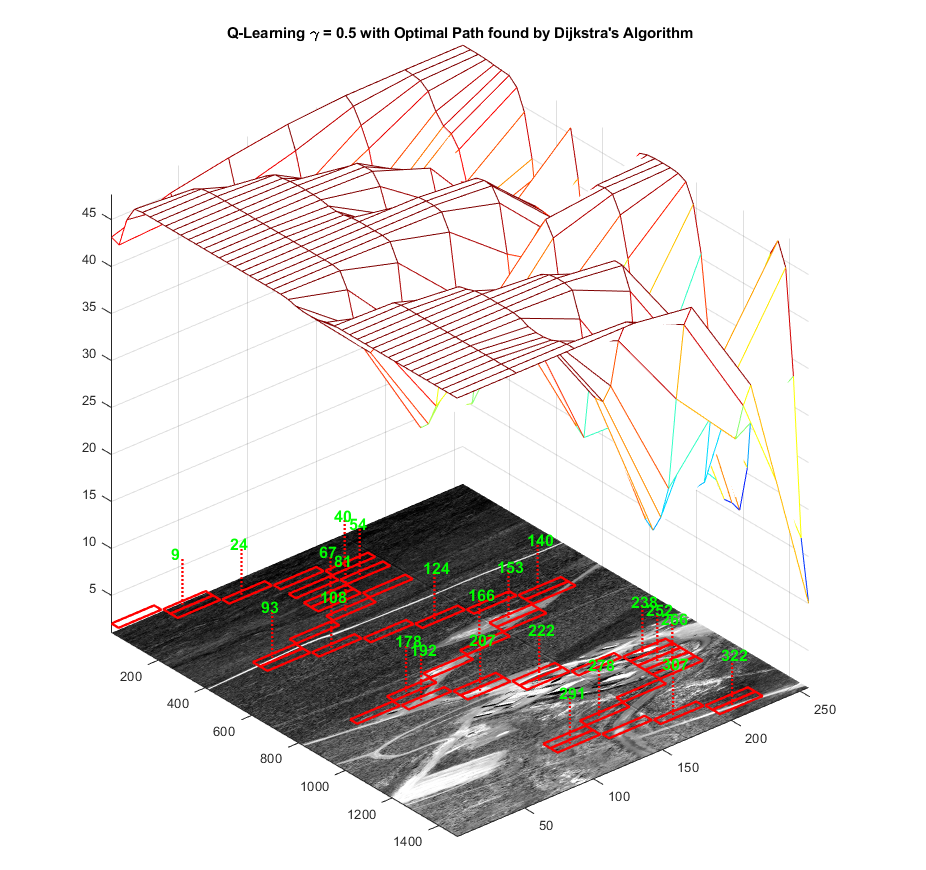


Figure 22: Q-Learning Dijkstra’s optimal pathway for HSI image.

Figure 23 a-c shows the superimposed value field used in the creation of the instant reward matrix, the HSI data, and the optimal pathway obtained using Q-learning and Dijkstra’s algorithm. Q-Learning taught the agent which actions were most rewarding in a path from state 1 to state 322. Note that in this example there are no heuristics imposed. The agent is free to move within the bounds of the HSI image, which then requires the use of Dijkstra’s path-finding algorithm on the resulting Q matrix estimated from Q-learning. Increases in learning rate γ cause the more exploration of the nodal state field since the future decision are discounted less. Evidence of nodal state field exploration can be seen at x= 200 designated by the ‘star’ symbol. When learning rate γ= 0.5 or 0.9 the downward dip at x= 200 exists but is absent at the small learning rate γ=0.2.

1. ****

****

**b)**

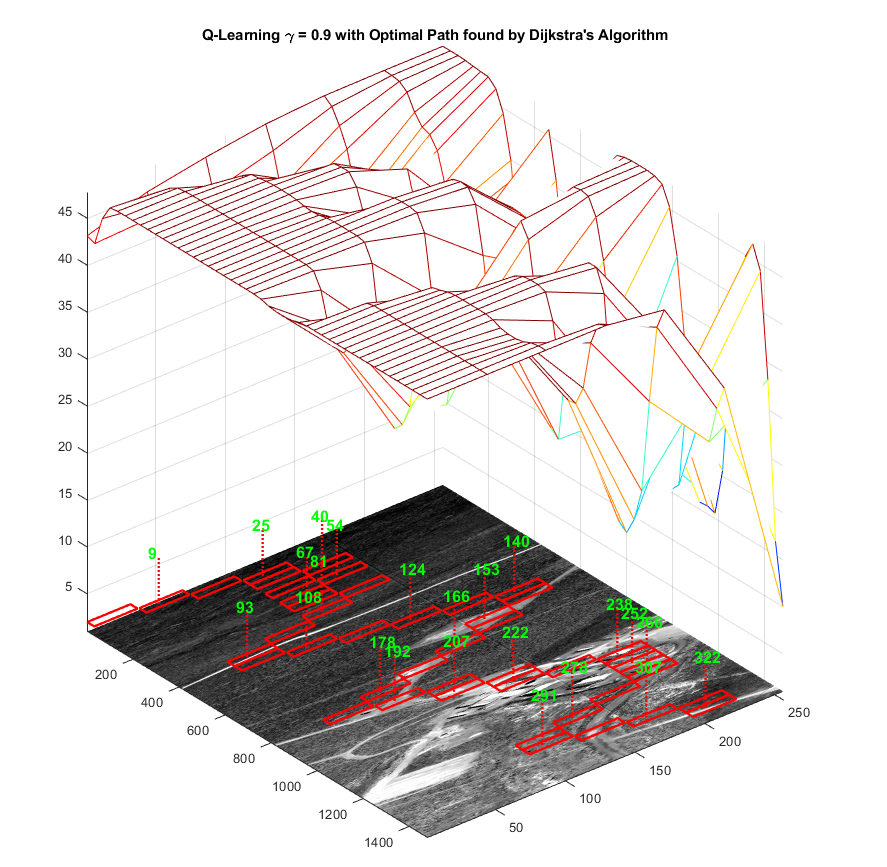
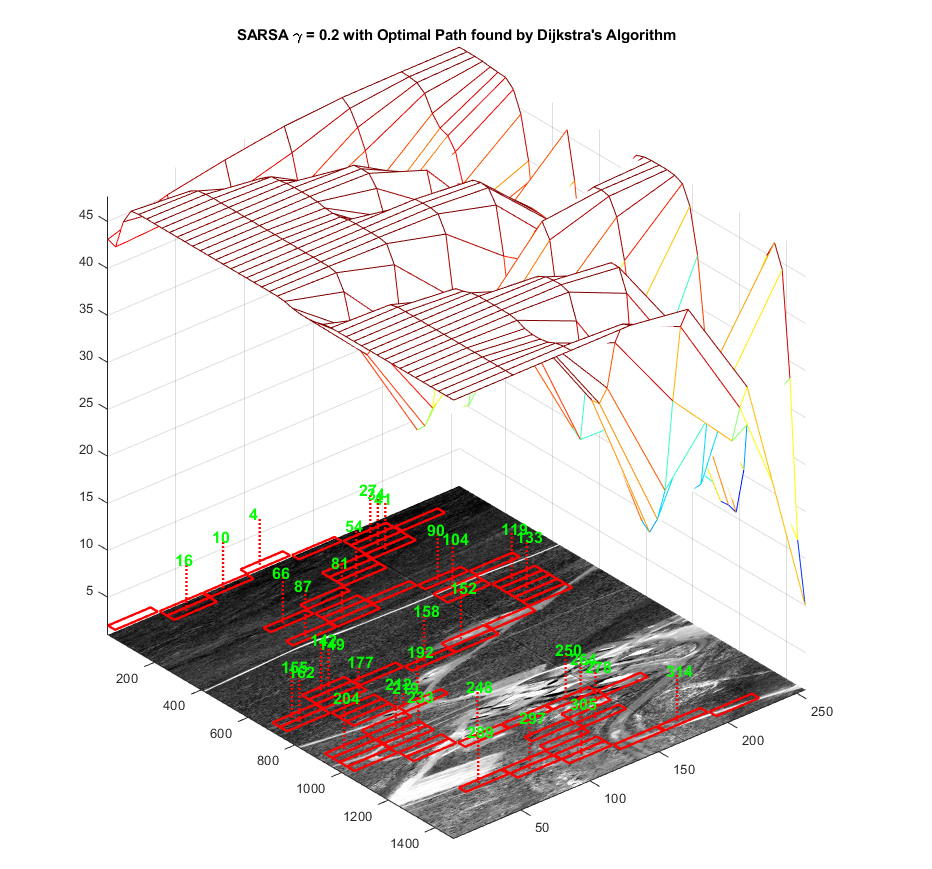
**c)**

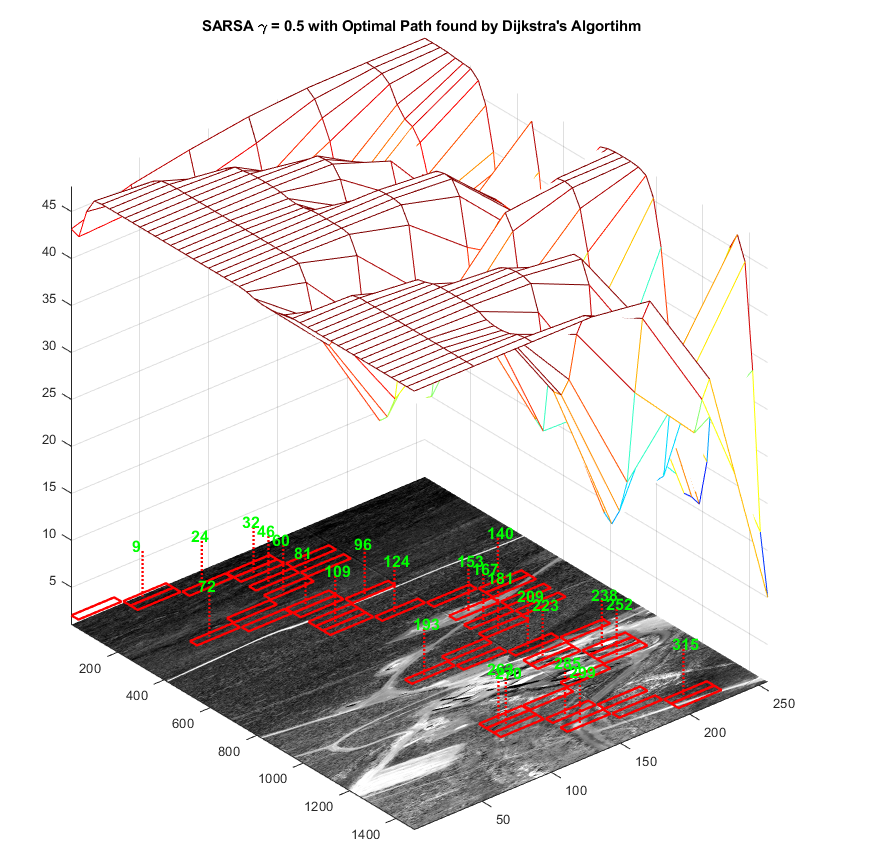
Figure 23: Q-Learning Dijkstra’s optimal pathway for HSI image for variable γ.

Figure 24 a-c shows the superimposed value field used in the creation of the instant reward matrix, the HSI data, and the optimal pathway obtained using SARSA and Dijkstra’s algorithm. SARSA taught the agent which actions were most rewarding in a path from state 1 to state 322. Note that in this example again there are no heuristics imposed. The agent is again free to move within the bounds of the HSI image, which then requires the use of Dijkstra’s path-finding algorithm on the resulting Q matrix estimated from Q-learning. Compared to Q-learning SARSA is far from optimal but it is ‘safe’ and conservative pathway which takes less risks.This accounts for the winding tortuous path observed in Figure 24a when compared to Figure 23a. Figures 24b and 24c also show large amounts of nodal state meandering when compared to Figures 23b and 23c as SARSA attempts to negotiate and ‘safe’ but non-optimal pathways. Qualitatively speaking it seems that Figure 24a where the learning rate is low has more nodal state exploration than Figures 23b and 23c which is inconsistent with the low learning rate value. However, without more rigorous parameterization of image nodal exploration, it is essentially impossible to comment more on this issue. Figures 24 b-c qualitatively show ‘safe’, non-optimal pathways where the gradients of the reward field are avoided. This can be seen at x = 600 and x = 1200. In these situation pathways of less risk which are conservative can mean a much longer pathway.



**a)**

**b)**

****

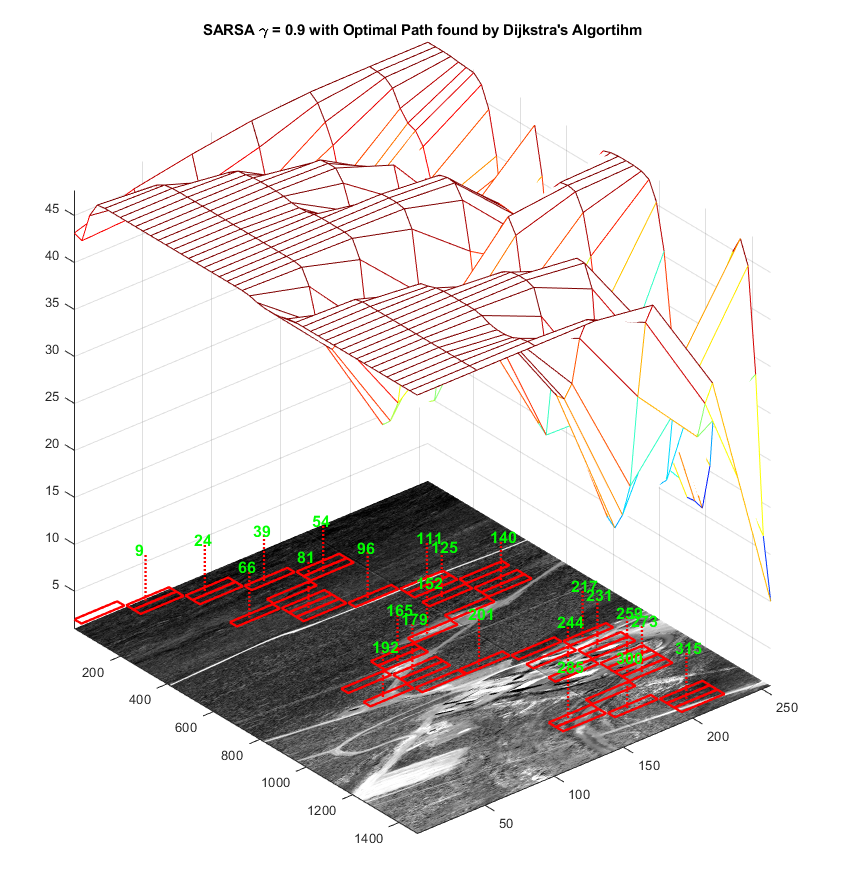
**** **c)**

Figure 24: SARSA Dijkstra’s optimal pathway for HSI image for variable γ.

**Conclusions**

With large amounts of HSI data accrued from large expanses of space, intelligence agencies have the opportunity to develop algorithms for feature extraction and state-action estimation for decision making capabilities. The reinforcement learning used in this report is an example of model free optimal path estimation, which can be applied to the MMP problem to find an optimal pathway between states which may be used for mission planning.

We claim that a meaningful value field can be extracted from the HSI data’s features which are then mapped into an instant reward matrix used in reinforcement learning to find an optimal pathway between states.

No model can tell a decision maker was should be done, the models simply give information. These models form a rational basis from which important evaluations, assessments, and decisions can be made.

**References**

Marsland, S., Machine Learning: An Algorithmic Perspective, Chapman and Hall/CRC Press, Machine Learning and Pattern Recognition Series, 2015, 437 pp.

Schwartz, H. M., Multi-Agent Machine Learning, John Wiley and Sons, 2014, 241 pp.

Stone, J. V., Bayes’ Rule, A Tutorial Introduction to Bayesian Analysis, Sebtel Press, 2013, 170 pp.

Sucar, L. E., Probabilistic Graphical Models: Principles and Applications, Springer-Verlag, 2015, 253 pp.

Sutton, R. S., and Barto, A. G., Reinforcement Learning: An Introduction, MIT Press, 1998, 322 pp.

Teknomo, K. Q-learning Tutorial, Revoledu.com, 2005, http://people.revoledu.com/kardi/tutorial/ReinforcementLearning/Q-Learning.htm, 29 pp.

Theodoridis, Pikrakis, Koutroumbas, Cavouras, Introduction to Pattern Recognition: A MATLAB Approach, Academic Press, 2010, Hard Clustering Algorithms, 169-178 pp.