

# MODERN EIGENVALUE SOLVERS FOR SPECTRAL IMAGE SEGMENTATION

C. JAMESON \*, A. JUJUNASHVILI \* AND A. KNYAZEV \*

**Abstract.** The spectral properties of the laplacian and normalized laplacian matrices are considered. These matrices are related to the graphs of the images. The behavior of the condition numbers of these matrices in the subspaces orthogonal to the first eigenvectors is studied. The question of time needed for calculation of some eigenvalues is investigated.

**Key words.** Graph matrix, Fiedler's vector, eigenvalue, eigenvector, condition number, spectral partitioning, NCuts partitioning

**AMS(MOS) subject classifications.** 65F15, 65F35, 05C50, 15A18,

## 1. INTRODUCTION

The purposes of this paper are: survey of some graph bisection methods and compare them to spectral bisection; computational study of the behavior of MATLAB function `eigs` on the large matrices of the garphs of images; comparison between Fiedler partitioning and normalized cuts partitioning; discovering the ways for improving of the results that are given in this work.

## 2. THE GRAPH PARTITIONING HEURISTICS

In general, graph partitioning (bisection) algorithms could be divided in

- Geometric,
- Structural (combinatorial),
- Refinement, and
- Multilevel algorithms.

Geometric algorithms use the coordinates associated with a graph's vertices. Among these algorithms are coordinate bisection, recursive inertial bisection, circle bisection. For these algorithms see, for instance, [4], [5] [13], [19], [20].

To the structural algorithms belong recursive level-structure bisection, Farhat's greedy algorithm, greedy graph growing algorithm of Karypis and Kumar, spectral algorithms (including Hendrickson and Leland's algorithm, which uses several eigenvectors). For this class of algorithms see, for instance [6], [8], [9], [11], [15], [16], [22], [24].

Refinement algorithms consist of the Kernighan-Lin algorithm, a greedy steepest descent, Fiduccia-Mattheyses algorithm (see [17], [7], [15]).

For the multilevel algorithms see, for instance [1], [3], [14], [16].

## 3. THE GRAPH MATRICES

Let  $G$  be an undirected finite graph. By  $V = V(G)$  and  $E = E(G)$  we denote its vertex and edge sets. Denote by  $n$  and  $m$  the numbers of vertices and the number of edges. Let  $w$  be a

weight function, that is

$$w : V(G) \times V(G) \rightarrow R^+,$$

which assigns a nonnegative real weight  $w(u, v)$  to each pair  $u, v$  of vertices (the weight will be denoted also by  $a_{uv}$ ). Then, let's denote by  $d_u$  the degree of vertex  $u$ . In the unweighted case  $d_u$  is defined as the number of edges of  $G$  incident to  $u$ . In the weighted case we set

$$d_u = \sum_{v \in V} w(u, v).$$

Given a graph  $G$ , its (weighted) adjacency matrix  $A(G) = [a_{ij}]$  is an  $n \times n$  matrix with rows and columns indexed by  $V(G)$  whose entries  $a_{ij}$  are defined as above. Let  $D(G) = \text{diag}(d_i)$  be the diagonal matrix with the vertex degrees on the diagonal. The difference

$$L = L(G) = D(G) - A(G)$$

is called the Laplace matrix (Laplacian) of  $G$ .

By definition,  $L$  is a real symmetric matrix. Therefore, it has  $n$  real eigenvalues  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  (repeated according to their multiplicities).

#### 4. SPECTRAL PARTITIONING

There is an increasing interest in the application of eigenvalues in combinatorial optimization problems. Spectral partitioning which is based on eigenvectors of Laplace matrix of graph has proved to be one of the most successful heuristic approaches in the design of partition algorithms, in parallel computation, in solving sparse linear systems, etc. (see, e.g. [22], [9]). For additional applications of spectral properties of the graph matrices see [21].

It is easy to show, that 0 is the eigenvalue of  $L$  and  $\mathbf{1} = (1, 1, \dots, 1)^T$  is the corresponding eigenvector. The following theorem is the background of the spectral partitioning (bisection):

**Theorem 4.1.** *Following statements are true:*

- (1)  $L(G)$  is a symmetric matrix;
- (2)  $L(G)\mathbf{1} = 0$ ;
- (3)  $L(G) = QQ^T$ , where  $Q$  is (oriented) incidence matrix of  $G$ ;
- (4) The eigenvalues of  $L(G)$  are nonnegative;
- (5) The number of connected components of  $G$  equal to multiplicity of zero eigenvalue. In particular,  $\lambda_2 \neq 0$  if and only if  $G$  is connected;
- (6)  $\lambda_2(G) = \min\{x^T Lx, \|x\| = 1, x \perp \mathbf{1}\}$ .

In particular, this theorem implies that  $\lambda_1(G) = 0$  is a simple eigenvalue of  $L$  iff the graph  $G$  is connected.

Let's now formulate the (easiest variant of) algorithm of spectral partitioning: Let  $v_2$  be the eigenvector of  $L(G)$  corresponding to  $\lambda_2$ .

- Compute the eigenvector  $v_2$ ;
- for each node  $i$  of  $G$ 
  - if  $v_2(i) < 0$  put node  $i$  in the partition  $N-$
  - else put node  $i$  in the partition  $N+$
  - endif
- endfor

Following theorem shows that above algorithm tends to give connected components  $N-$  and  $N+$ .

Theorem 2. ([9]). Let  $G$  be connected, and  $N-$  and  $N+$  be defined by the above algorithm. Then  $N-$  is connected. If no  $v_2(n) = 0$ ,  $N+$  is also connected.

There are a number of reasons  $\lambda_2$  is called the algebraic connectivity of the graph. Here is another:

Theorem 3 ([9]). Let  $G = (N, E)$  be a graph, and  $G_1 = (N, E_1)$  a subgraph, i.e. with the same nodes and a subset of the edges, so that  $G_1$  is "less connected" than  $G$ . Then  $\lambda_2(L(G_1)) \leq \lambda_2(L(G))$ , i.e. the algebraic connectivity of  $G_1$  is also less than or equal to the algebraic connectivity of  $G$ .

The most common spectral methods use the median of the components of Fiedler's vector to induce a bisection.

### 5. NORMALIZED CUTS PARTITIONING

A graph  $G = (V, E)$  can be partitioned into two disjoint sets,  $A, B, A \cup B = V, A \cap B = \emptyset$ , by simply removing edges, connecting the two parts. The degree of dissimilarity between resulting two pieces can be computed as total weight of the edges that have been removed, that is as the cut:

$$(5.1) \quad cut(A, B) = \sum_{u \in A, v \in B} w(u, v).$$

The optimal bipartitioning of a graph is the partitioning that minimizes above value  $cut(A, B)$ . Although there are exponential number of such partitions, there are the efficient algorithms for solving of this minimization problem. But it is shown that minimum cut criteria favors cutting small sets of isolated nodes of the graph.

To avoid this kind of partitionings, let's introduce the measure of disassociation between two groups - Normalized Cuts (see, e.g. [23]):

$$(5.2) \quad Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)},$$

where  $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$  is the total connection from nodes of  $A$  to the all nodes of the graphs, and  $assoc(B, V)$  is defined similarly.

Based on the above definitions we define a measure for total normalized association within groups for a given partition:

$$(5.3) \quad Nassoc(A, B) = \frac{assoc(A, A)}{assoc(A, V)} + \frac{assoc(B, B)}{assoc(B, V)},$$

where  $assoc(A, A)$  and  $assoc(B, B)$  are the total weights of the edges containing in  $A$  and  $B$  respectively.

It is easy to show that above introduced quantities are related naturally (see [23])

$$Ncut(A, B) = 2 - Nassoc(A, B).$$

The last equation shows, that the above introduced two partition criteria, minimizing the disassociation between the groups and maximizing the association within the groups, are identical.

It is shown that problem of minimizing normalized cuts is NP-complete. To tackle this difficulty the original problem is embedded in real value domain and then an approximate discrete solution is founded efficiently.

Introducing the indicator vector  $x$  such that  $x_i = 1$  if node  $i$  is in  $A$  and  $x_i = -1$  otherwise, the problem of minimizing of  $Ncut(A, B)$  could be rewritten in the equivalent form: find

$$(5.4) \quad \min_x Ncut(x) = \min_y \frac{y^T(D - W)y}{y^T D y}$$

with the conditions  $y(i) \in \{1, -b\}$  and  $y^T D \mathbf{1} = 0$ ;  $b = \sum_{x_i > 0} d_i / \sum_{x_i < 0} d_i$ .

Noting that the objective function is the Rayleigh quotient, the above minimization problem can be solved by solving of the following generalized eigenvalue problem:

$$(5.5) \quad (D - W)y = \lambda D y.$$

However, the minimization problem (5.4) contains two constraints on  $y$ . It is shown, that first of them is automatically satisfied by the solution of generalized eigenproblem. The only reason that it is not necessarily the solution of the original problem is that the second constraint on  $y$  that  $y(i)$  takes on two discrete values is not automatically satisfied. But relaxing this constraint makes optimization problem tractable.

Following the reasons mentioned above, we can make the conclusion, that the effectiveness of normalized cuts method strongly depends on the method of solving eigenproblem.

## 6. MULTILEVEL ALGORITHMS FOR PARTITIONING GRAPHS

First work on using eigenvectors of graphs for partitioning of sparse matrices was [22]. The amount of the work needed for execution of the spectral methods can be reduced if computation of the Fiedler vector is done by using a multilevel algorithm [22]. Multilevel graph partitioning schemes are studied in the several papers, among them are [3], [14], [16].

A multilevel graph bisection algorithm works as follows (formally): consider a weighted graph  $G_0 = (V_0, E)$ , with weights both on vertices and edges. A multilevel graph bisection algorithm consists of the following three phases

- (1) Coarsing phase. The graph  $G_0$  is transformed into a sequence of smaller graphs  $G_1, G_2, \dots, G_m$  such that  $|V_0| > |V_1| > \dots > |V_m|$ .
- (2) Partitioning phase. A partition  $P_m$  of the graph  $G_m = (V_m, E_m)$  is computed that partitions  $V_m$  into two parts, each containing half the vertices of  $G_0$ .
- (3) Uncoarsening Phase. The partition  $P_m$  of  $G_m$  is projected back to  $G_0$  by going through intermediate partitions  $P_{m-1}, P_{m-2}, \dots, P_1, P_0$ .

The Rayleigh quotient iterations (RQI) and SYMMLQ algorithm are used in many works on the partitioning step. The computational results show that spectral partitioning is one of the effective methods. The computational results can be improved if one uses locally optimal preconditioned conjugate gradient method [18] instead of RQI/SYMMLQ.

## 7. HOW GOOD SPECTRAL PARTITIONING IS?

Beside other results, the paper [16] contains the interesting study of behaviour of spectral partitioning on the coarsest level. The results of this paper show, that there are the methods (so-called greedier methods) which not only work faster than spectral methods but also the

quality of partitions have much higher. The authors suggest that this surprising result stems from the Lanczos algorithm's failure to converge in its allotted iterations. This indicates that while spectral methods work well on large graphs whose complexity may foil greedier algorithms, their use on smaller graphs may be overkill (see, e.g. [5]).

Interesting study of the spectral partitioning is in [25], [26]. As the negative, also positive results on the behavior of the spectral partitioning are given in these papers. Namely, it shown that recursive bisection, even when an optimal bisection algorithm is assumed, may produce a  $p$ -way partition that is very far way from the optimal one. From other side, it is shown that:

- (1) for some important classes of graphs that occur in practical applications, such as well-shaped finite-element and finite-difference meshes, recursive spectral bisection "is within a constant factor of the optimal one almost always" ([25]);
- (2) if the balance condition is relaxed so that each block in the  $p$ -way partition is bounded by  $2n/p$ , where  $n$  is the number of vertices of the graph, then modified recursive spectral bisection finds an approximately balanced  $p$ -way partition whose cost is within an  $\mathcal{O}(\log p)$  factor of the cost of the optimal  $p$ -way partition.

### 8. ISOPERIMETRIC PARTITIONING

Let's denote by  $x$  the binary indicator vector ( $x_i = 1$  if the vertex  $v_i \in S$  and  $x_i = 0$  otherwise), where  $S$  is some subset of the set of the vertices. The boundary of set  $S$ ,  $\partial S$  is defined as

$$\partial S = \{e_{ij} | i \in S, j \in \bar{S}\}, \quad \text{and} \quad |\partial S| = \sum_{e_{ij} \in \partial S} w(e_{ij})$$

and the volume as  $Vol(S) = \sum_i d_i$  where sum is taken for all  $v_i \in S$ .

For a given set,  $S$ , we term the ratio of its boundary to its volume the isoperimetric ratio, denoted by  $h(S)$ . The isoperimetric sets for a graph  $G$  are any sets  $S$  and  $\bar{S}$  for which

$$h(S) = h_G = \inf_S \frac{|\partial S|}{Vol(S)},$$

if the condition  $Vol(S) \leq \frac{1}{2}Vol(V)$  holds. Then, if  $L$  is the Laplacian of given graph, we can write:

$$(8.1) \quad |\partial S| = x^T L x,$$

and  $Vol(S) = x^T d$ , where  $d$  is the vector of node degrees. If  $\mathbf{1}$  is the vector of ones, then  $Vol(V) = \mathbf{1}^T d$  and we can replace the above inequality by the constraint

$$(8.2) \quad x^T d = \frac{1}{2} \mathbf{1}^T d.$$

Thus, the isoperimetric constant of the graph  $G$  could be defined as

$$(8.3) \quad h_G = \min_x \frac{x^T L x}{x^T d},$$

subject to (8.2).

Using the Lagrange multiplier method, under additional assumptions above constrained minimization problem can be reduced to solving some linear systems (see, e.g. [12]).

9.  $k$ -MEDIAN AND  $k$ -MEANS ALGORITHMS

Let's formulate the description of the multiway partitioning (clustering) problem (see [2]): given  $m$  points  $\{x^1, x^2, \dots, x^m\}$  in  $n$ -dimensional real space  $R^n$ , and fixed integer  $k$  - number of clusters. Determine  $k$  "centers" in  $R^n$ ,  $\{c^1, c^2, \dots, c^k\}$ , such that the sum of "distances" of each point to a nearest cluster center is minimized. The partitioning problem then is:

$$(9.1) \quad \min_{c^1, c^2, \dots, c^k} \sum_{i=1}^m \min_{l=1, \dots, k} \|x^i - c^l\|,$$

where the norm  $\|\cdot\|$  is some arbitrary norm on  $R^n$ . Note, that objective function in the above problem is, in general, neither convex nor concave. Hence this is difficult minimization problem.

One can reformulate the problem (9.1) as the constrained minimization problem: find

$$(9.2) \quad \min_{c^l, t_{il}} \sum_{i=1}^m \sum_{l=1}^k t_{il} \cdot \|x^i - c^l\|$$

subject to the constraints:

$$\sum_{l=1}^k t_{il} = 1, t_{il} \geq 0, l = 1, \dots, k, i = 1, \dots, m, l = 1, \dots, k.$$

Let's consider the question of solving of constrained minimization problem (9.2) with the 1-norm distance for which (9.2) can be written as the following bilinear problem:

$$(9.3) \quad \min_{c^l, t_{il}} \sum_{i=1}^m \sum_{l=1}^k t_{il} \cdot (e^T d_{il})$$

subject to the constraints:

$$\begin{aligned} -d_{il} &\leq x^i - c^l \leq d_{il}, i = 1, \dots, m, l = 1, \dots, k, \\ \sum_{l=1}^k t_{il} &= 1, t_{il} \geq 0, i = 1, \dots, m, l = 1, \dots, k. \end{aligned}$$

We note that constraints are separated (uncoupled) in the variables  $(c, d)$  and variable  $t$  in the last problem. Hence the uncoupled bilinear programming algorithm can be used. This algorithm alternates between solving a linear program in the variable  $t$  and a linear program in the variables  $(c, d)$ . This approach leads to the following algorithm:

**9.1.  $k$ -median algorithm.** Given  $k$  cluster centers  $c^{1,j}, c^{2,j}, \dots, c^{k,j}$  at iteration  $j$ , compute  $c^{1,j+1}, c^{2,j+1}, \dots, c^{k,j+1}$  by the following two steps:

- (1) Cluster Assignment: For  $i = 1, \dots, m$ , assign  $x^i$  to cluster  $l(i)$  such that  $c^{l(i),j}$  is nearest to  $x^i$  in the 1-norm;
- (2) Cluster Center Update: For  $l = 1, \dots, k$ , set  $c^{l,j+1}$  to be the median of all  $x^i$  assigned to  $c^{l,j}$ .

Stop when  $c^{l,j} = c^{l,j+1}, l = 1, \dots, k$ .

**9.2.  $k$ -mean algorithm.** If we consider the problem (9.2) with the squared 2-norm, we get the well-known  $k$ -mean algorithm: Given  $k$  cluster centers  $c^{1,j}, c^{2,j}, \dots, c^{k,j}$  at iteration  $j$ , compute  $c^{1,j+1}, c^{2,j+1}, \dots, c^{k,j+1}$  by the following two steps:

- (1) Cluster Assignment: For each  $i = 1, \dots, m$ , assign  $x^i$  to cluster  $l(i)$  such that  $c^{l(i),j}$  is nearest to  $x^i$  in the 2-norm;
- (2) Cluster Center Update: For  $l = 1, \dots, k$ , set  $c^{l,j+1}$  to be the mean of all  $x^i$  assigned to  $c^{l,j}$ .

Stop when  $c^{l,j} = c^{l,j+1}, l = 1, \dots, k$ .

If the 2-norm (without squaring) is used in the objective function, the cluster center update subproblem becomes the considerably harder so-called Weber problem.

Determining the number of clusters  $k$  is a hard problem. In most cases one runs several clustering sessions with different values for  $k$  and chooses better results (see, e.g. [2]).

## 10. SPATIOTEMPORAL GROUPING USING NYSTRÖM APPROXIMATION

The purpose of method (see, e.g. [10]) is to reduce the computational requirements of grouping algorithms based on spectral partitioning. In short, the approach exploits the fact that the number of coherent groups in an image sequence is considerably smaller than the number of voxels. The algorithm is based on multiple eigenvector version of Normalized Cuts and on a technique for numerical solution of eigenproblem known as the Nyström method.

The Nyström method is a technique for finding numerical approximations to eigenfunctions of the following problem:

$$\int_a^b K(x, y)\phi(y)dy = \lambda\phi(x).$$

Using some quadrature rule at the set of evenly spaced on  $[a, b]$  (without loss of generality,  $a = 0, b = 1$ ) points  $\xi_1, \xi_2, \dots, \xi_n$  one gets the approximation to this equation

$$(10.1) \quad \frac{(b-a)}{n} \sum_{j=1}^n K(x, \xi_j)\hat{\phi}(\xi_j) = \lambda\hat{\phi}(x)$$

where  $\hat{\phi}(x)$  is an approximation to true  $\phi(x)$ . Substituting in this equation  $x = \xi_i$  one gets the matrix eigenvalue problem  $K\hat{\Phi} = n\hat{\Phi}\Lambda$  with Gram matrix  $K = [K_{ij}], K_{ij} = K(y_i, y_j)$  and matrix containing as a columns the approximate eigenvectors with corresponding eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$ .

The main step of the Nyström method is substituting back to the equation (10.1) these approximate eigenpairs and getting Nyström extension of eigenvectors

$$(10.2) \quad \hat{\phi}_i(x) = \frac{1}{n\lambda_i} \sum_{j=1}^n K(x, \xi_j)\hat{\phi}_i(\xi_j)$$

The preceding analysis suggests that it should be possible to find approximate eigenvectors of a large Gram matrix by solving a much smaller eigenproblem using only a subset of the entries and employing Nyström extension to fill in the rest.

Newertheless, the method requires  $\mathcal{O}(n^3)$  operations, where  $n$  is the number of samples employed in approximation. Therefore here is the opportunity to reduce the computational work

using locally optimal block preconditioned conjugate gradient method [18] for calculating the eigenvectors.

## 11. USED IMAGES, CODES AND METHODS

The results presented in this paper are related to the picture from European Southern Observatory (ESO) web-site "www.eso.org" and comes with permission for reproduction provided the ESO is credited. The photograph is of the "Very Large Telescope at Paranal" in the Atacama desert region in Chile. The size of the initial image is  $2746 \times 3000 \times 3$ , that is presented in a `MATLAB` code as 197712000 double array. That is, we used the picture with 8,238,000 pixels.

The original code is made by Leo Grady and its locations are <http://eslab.bu.edu/software/graphanalysis/graphanalysis.html> and <http://eslab.bu.edu/software/graphanalysis/graphanalysisDEMOS.html>. We modified several `.m`-files to run the code for the sequence of the images of required sizes.

In order to study the behavior of several spectral characteristics of the matrices related to the graphs of images the partitioning was executed for different sizes of images by 2 methods - spectral (Fiedler) partitioning and NCuts partitioning. For both of given methods are considered also two different cases - one of them defines the graph geometry by the 4-point pattern, and second one - by the 8-point pattern. The images of different sizes were obtained by resizing of the original image using the `MATLAB` command `imresize`, namely by dividing of the  $X$  and  $Y$  sizes of original image by the sequence of some numbers. These sequences may be obtained in different ways, that depend on our purposes. In this paper the case is considered, when this sequence is power sequence (the geometric progression). The first term `nBegin`, last term `nLast` and the number of images `nPictures`, we need to partition, are defined by ourself. By these initial data common quotient of the progression `resizemult` is defined.

## 12. COMPUTATIONAL RESULTS

The computational experience showed, that for the `nBegin` = 4 (the size of laplacian matrix is  $514500 \times 514500$ ) `MATLAB` code working on GODZILLA computer makes the spectral and NCuts segmentation for 4-point connection using 1.9GB and 2.05GB of memory respectively. For 8-point connection spectral partitioning also works (using the memory 2.43GB) but the NCuts goes out of memory.

The results of using the computer memory for some sizes of matrices are given in Table 1.

For `nBegin` = 5, `nLast` = 512, `nPictures` = 50, the code works about 75 minutes and calculates all the needed quantities.

The pictures given in this paper are related to the case when `nBegin` = 8, `nLast` = 128, `nPictures` = 20. This choice gives the matrices of sizes from  $483 \times 483$  to  $128625 \times 128625$ . This problem tooks about 15 minutes of the CPU time on GODZILLA. Other results, that is the dependences of the condition numbers in subspaces, of the time for smallest eigenvalues, of few small eigenvalues on the sizes of matrices, are obtained for the following initial data: `nBegin` = 5, `nLast` = 128, `nPictures` = 50.

We studied the behavior of the condition numbers of the laplacian and normalized laplacian (NCuts) matrices  $L$  and  $L2$  respectively for the initial values `scaleIso` = `scaleNcuts` = 100. Condition number in our case means the ratio of the largest and smallest positive eigenvalues (the condition number in the subspaces, orthogonal to first eigenvector), because the smallest

Nr.	Size of Matrix	Spectral-4	NCUT-4	Spectral-8	NCUT-8
1	25,232	100 MB	180 MB	145 MB	150 MB
2	48,530	170 MB	255 MB	217 MB	240 MB
3	101,565	360 MB	420 MB	453 MB	480 MB
4	128,625	450 MB	520 MB	560 MB	605 MB
5	329,400	1.17 GB	1.28 GB	1.5 GB	1.6 GB
6	514,500	1.9 GB	2.05 GB	2.43 GB	Out of Memory

TABLE 1. Computer Memory Usage

eigenvalues of both matrices are zero. The eigenvalues were calculated using the MATLAB function `eigs`. The related results in the graphical view are given below (see Figure 1).

We studied also the CPU time needed for calculation of the smallest and largest eigenvalues of matrices considered in the project. The graphics showing the dependence of the mentioned time on the sizes of the matrices are given below. These times were calculated by two different ways - using `tic`, `toc` and `cputime` functions. The results given below (see Figure 2) are obtained by the function `cputime`.

Figure 3 and Figure 4 show the distribution of 10 small eigenvalues of the matrices  $L$  and  $L2$ , and the results of bisections are given on the Figures 5-10.

## REFERENCES

- [1] S. T. Barnard, H. D. Simon, *A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems*, Proceedings of 6th SIAM Conf. Parallel Processing for Scientific Computing, 1993, pp. 711–718.
- [2] P. S. Bradley, U. M. Fayyad and O. L. Mangassarian, *Data mining: Overview and optimization opportunities*, INFORMS Journal on Computing, 11:217-238, 1999.
- [3] T. Bui and C. Jones, *A heuristic for reducing fill in sparse matrix factorization*, Proceedings of 6th SIAM Conf. Parallel Processing for Scientific Computing, 1993, pp. 445–452.
- [4] F. Cao, J. R. Gilbert and Sh.-H. Teng, *Partitioning meshes with lines and planes.*, Technical Report CSL-96-01, Xerox Palo Alto Research Center, January, 1996.
- [5] B. L. Chamberlain, *Graph partitioning algorithms for distributing workloads of parallel computations*, url = "<http://citeseer.nj.nec.com/chamberlain98graph.html>", 1998.
- [6] C. Farhat, *A simple and efficient automatic FEM domain decomposer*, Computers and Structures, 28(1988), 579–602.
- [7] C. M. Fiduccia and R. M. Mattheyses, *A linear-time heuristic for improving network partitions*, In: 19th Design Automation Conference, pp. 175-181. IEEE Computer Society Press, June, 1982.
- [8] M. Fiedler, *Algebraic connectivity of graphs*, Czech. Math. J. 23(1973), 298-305.
- [9] M. Fiedler, *A Property of Eigenvectors of Nonnegative Symmetric Matrices and its Application to Graph Theory*, Czech. Math. J. 25(1975), 619-637.
- [10] C. Fowlkes, S. Belongie and J. Malik, *Efficient Spatiotemporal Grouping Using the Nyström Method*, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2001.
- [11] A. George and J. W. H. Liu, *Computer solution of large sparse positive definite systems*, Prentice-Hall, 1981.
- [12] L. Grady and E. L. Schwartz, *Isoperimetric graph partitioning for data clustering and image segmentation*, url = "<http://eslab.bu.edu/publications/tech-reports/2003/grady2003isoperimetric-TR.pdf>", 2003.
- [13] M. T. Heath and P. Raghavan, *A cartesian parallel nested dissection algorithm*, SIAM Journal on Matrix Analysis and Applications. 16(1995), 235–253.

- [14] B. Hendrickson and R. Leland, *A multilevel algorithm for partitioning graphs*, Tech. rep. SAND93-1301, Sandia National Laboratories, Albuquerque, NM, 1993.
- [15] B. Hendrickson and R. Leland, *An improved spectral graph partitioning algorithm for mapping parallel computations*, SIAM Journal on Scientific Computing, 16(1995), 452-469.
- [16] G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM Journal on Scientific Computing. 20(1998), 359-392.
- [17] B. W. Kernighan and S. Lin, *An efficient heuristic procedure for partitioning graphs*, The Bell System Technical Journal, 49(1970), 291-307.
- [18] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: locally optimal block preconditioned conjugate gradient method*, SIAM Journal on Scientific Computing, 23(2001), 517-541.
- [19] G. L. Miller, Sh.-H. Teng, W. Thurston, and S. A. Vavasis, *Automatic mesh partitioning*, In: A. George, J. R. Gilbert and J. W. H. Liu, editors, Graph Theory and Sparse Matrix Computation, Vol. 56 of IMA volumes in mathematics and its applications, pp. 57-84. Springer-Verlag, 1993.
- [20] G. L. Miller, Sh.-H. Teng, W. Thurston, and S. A. Vavasis, *Geometric separation for finite element meshes*, SIAM Journal on Scientific Computing, 19(1998), 364-386
- [21] B. Mohar and S. Poljak, *Eigenvalues in combinatorial optimization*, In: "Combinatorial and Graph-Theoretical Problems in Linear Algebra", R. A. Brualdi, S. Friedland, V. Klee, Eds., IMA Volumes in Mathematics and Its Applications. Vol. 50, Springer-Verlag, 1993, pp. 107-151.
- [22] A. Pothen, H. D. Simon and K.-P. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM Journal on Matrix Analysis and Applications. 11(1990), 430-452.
- [23] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(2000), 888-905.
- [24] H. D. Simon, *Partitioning of unstructured problems for parallel processing*, Computing Systems in Engineering, 2(2-3)(1991), 135-148.
- [25] H. D. Simon and S.-H. Teng, *How good is recursive bisection?*, SIAM Journal on Scientific Computing. 18(1997), 1436-1445.
- [26] D. A. Spielman and S.-H. Teng, *Spectral partitioning works: Planar graphs and finite element meshes*, Technical Report UCB CSD-96-898.

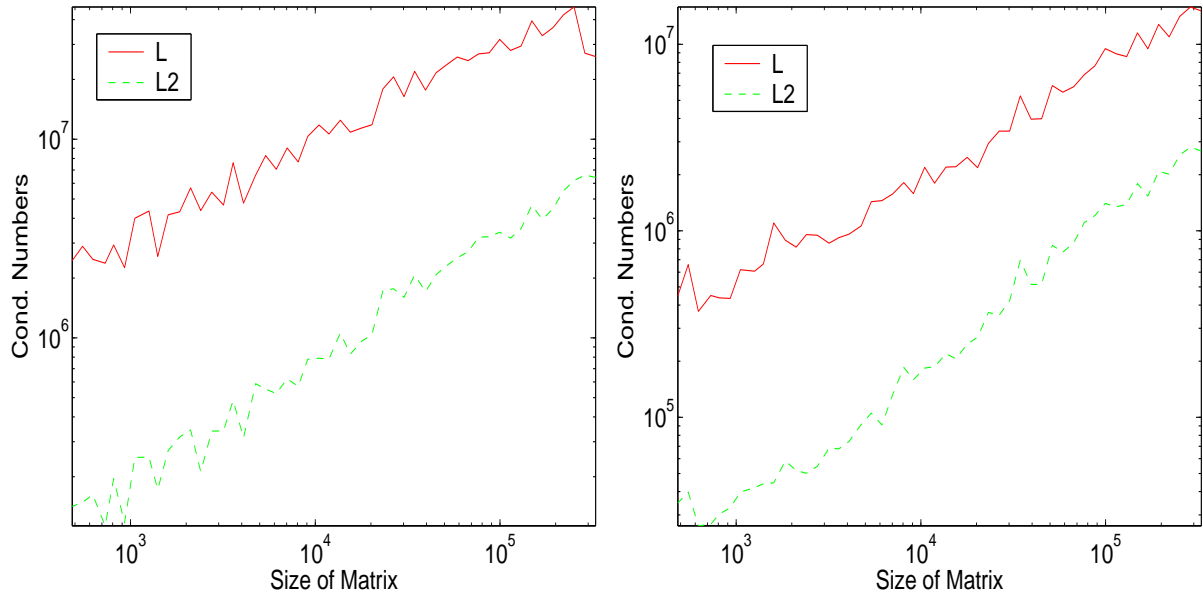


FIGURE 1. Condition Numbers, 4-Point - left, 8-Point - right

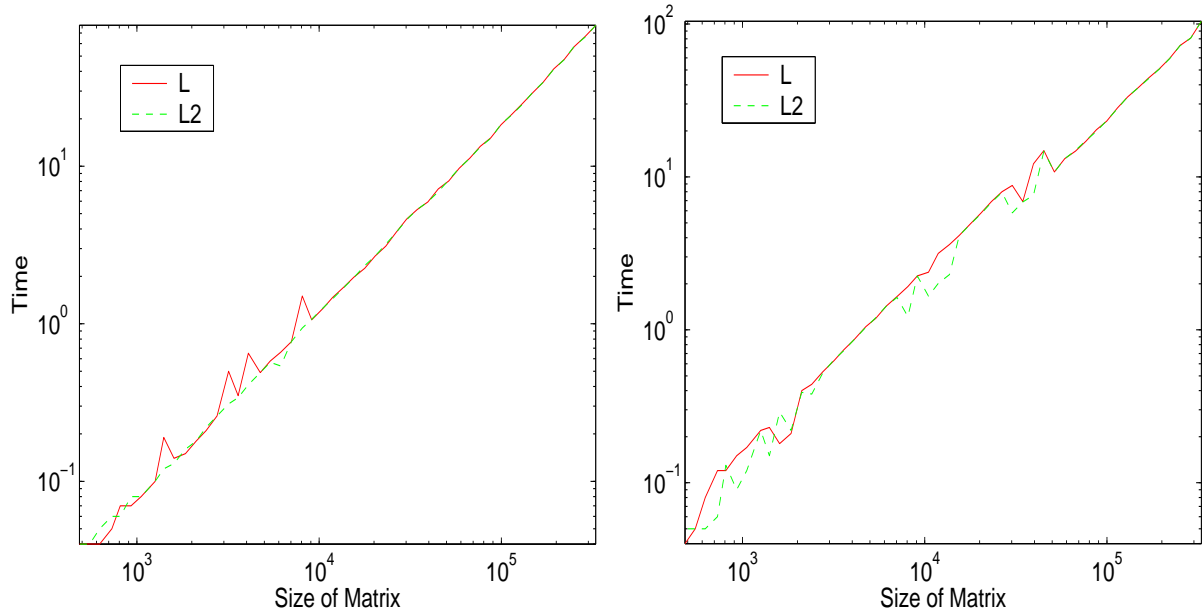


FIGURE 2. Time for Smallest Eigenvalues, 4-Point - left, 8-Point - right

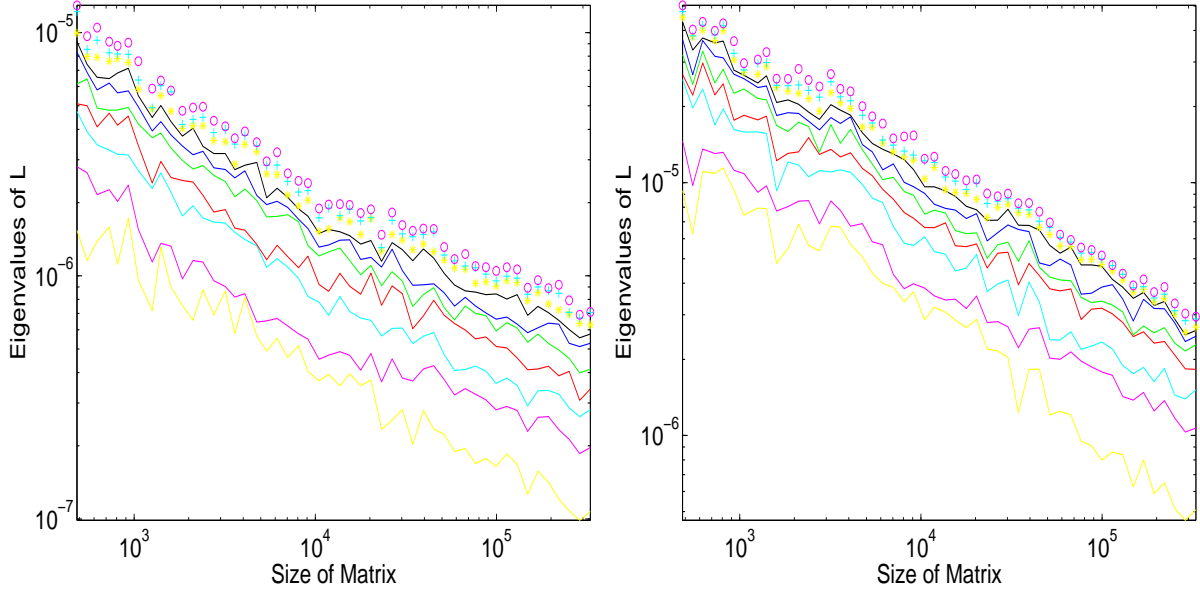


FIGURE 3. 10 Small Eigenvalues of L, 4-Point - left, 8-Point - right

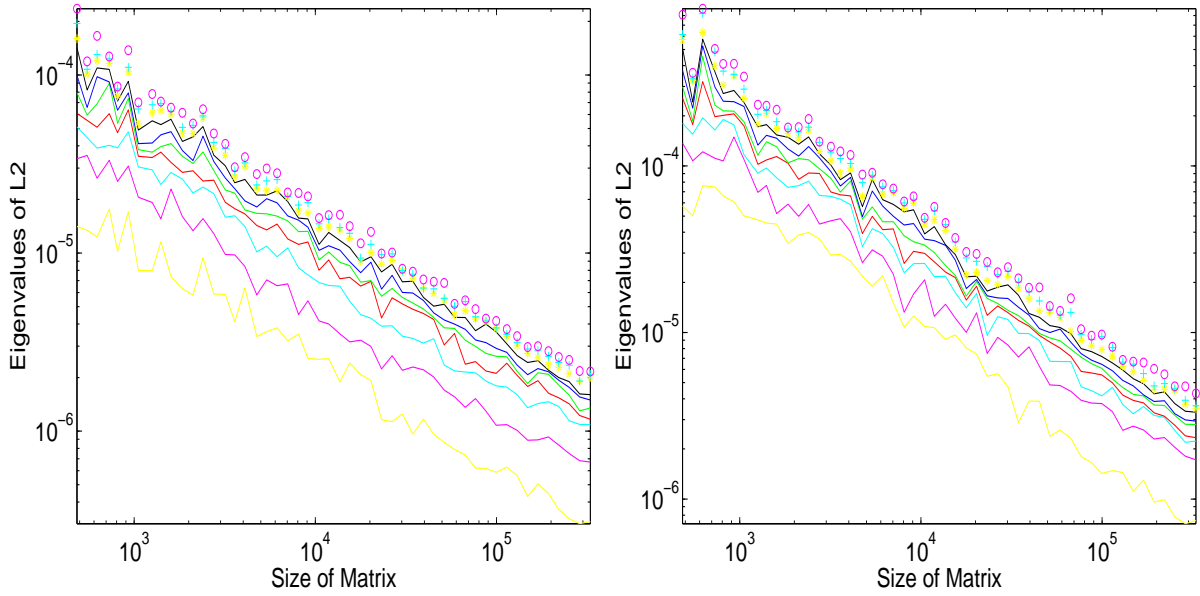


FIGURE 4. 10 Small Eigenvalues of L2, 4-Point - left, 8-Point - right

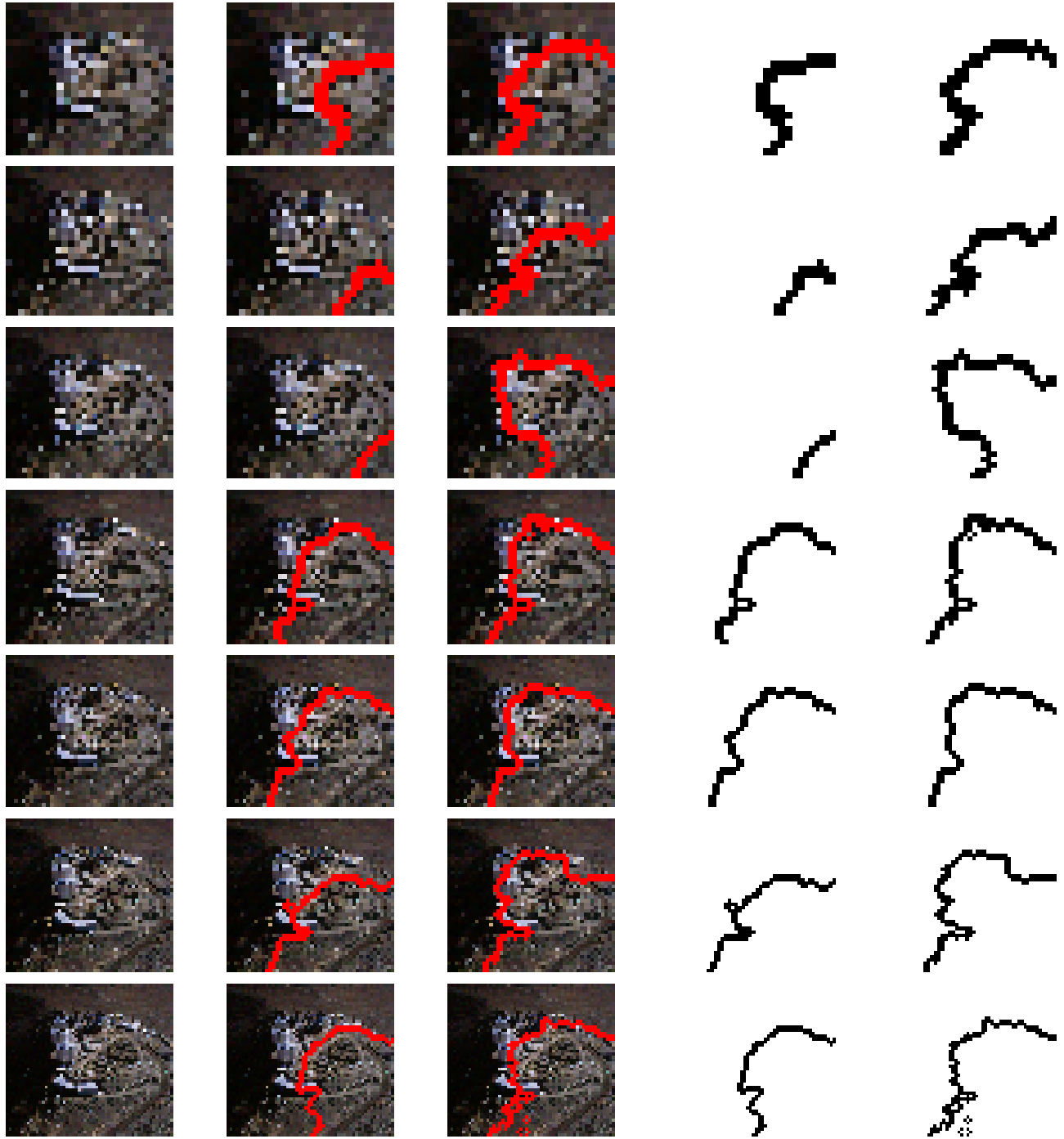


FIGURE 5. The Results of Partitioning for 4-Point Pattern, Pictures 1-7. Original Image - Spectral Partit. - NCUT Partit.- Spectral Line - NCUT Line  
 Sizes of images (pixels): 483, 648, 868, 1188, 1599, 2112, 2856

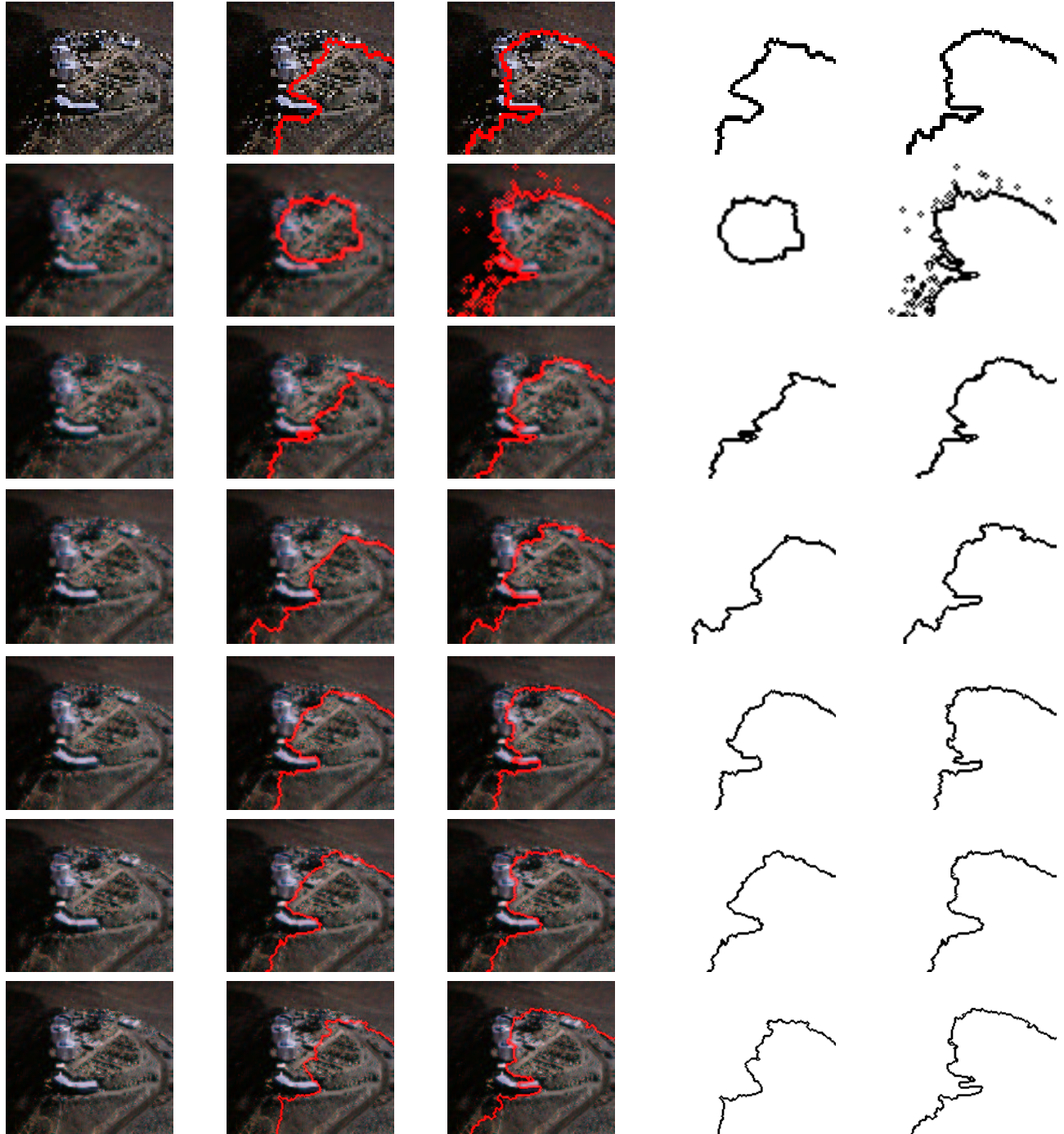


FIGURE 6. The Results of Partitioning for 4-Point Pattern, Pictures 8-14.  
 Original Image - Spectral Partit. - NCUT Partit.- Spectral Line - NCUT Line.  
 Sizes of images (pixels): 3835, 5100, 6873, 9200, 12296, 16605, 22152

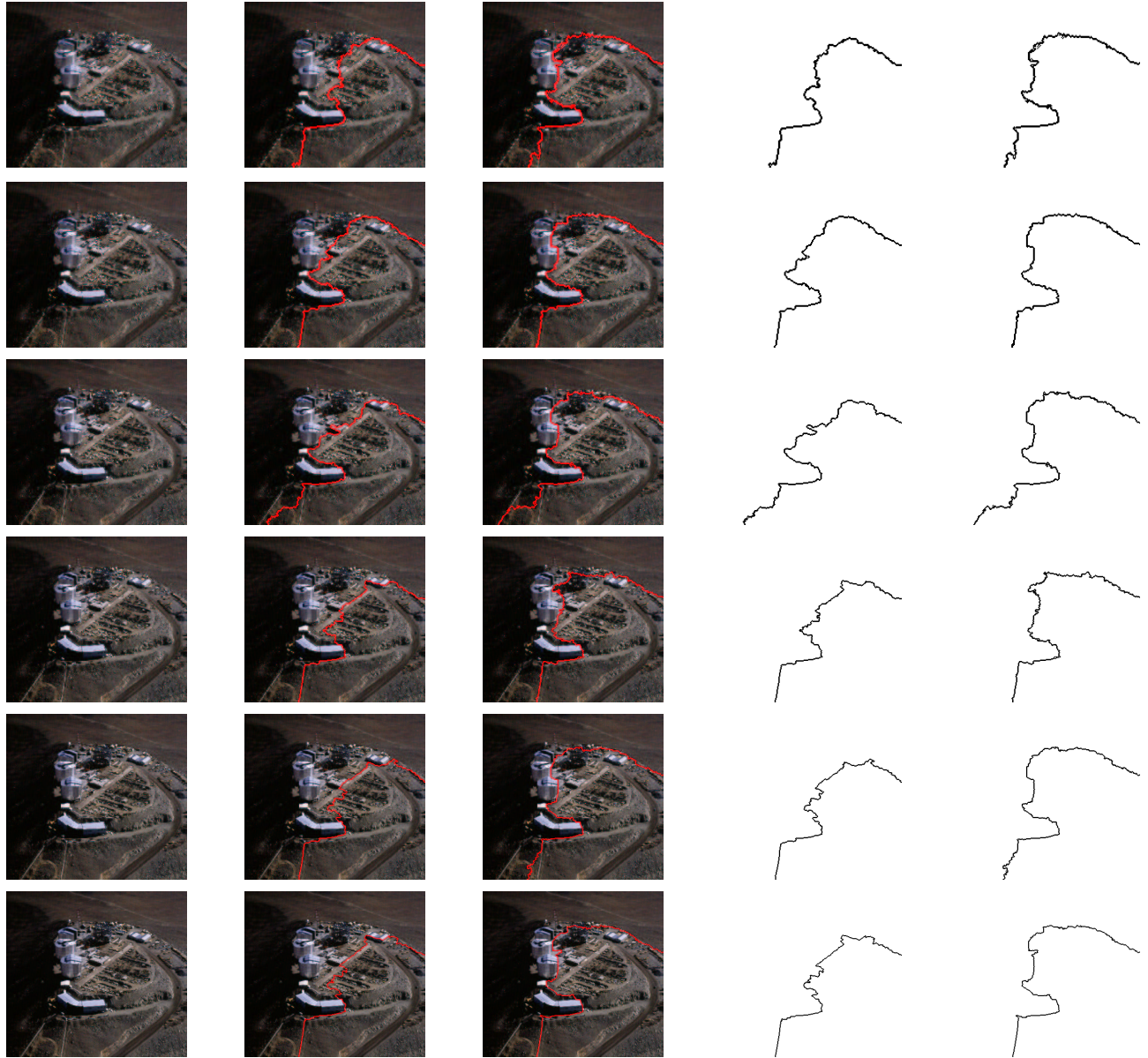


FIGURE 7. The Results of Partitioning for 4-Point Pattern, Pictures 15-20. Original Image - Spectral Partit. - NCUT Partit.- Spectral Line - NCUT Line. Sizes of images (pixels): 29700, 39919, 53482, 71680, 95904, 128625

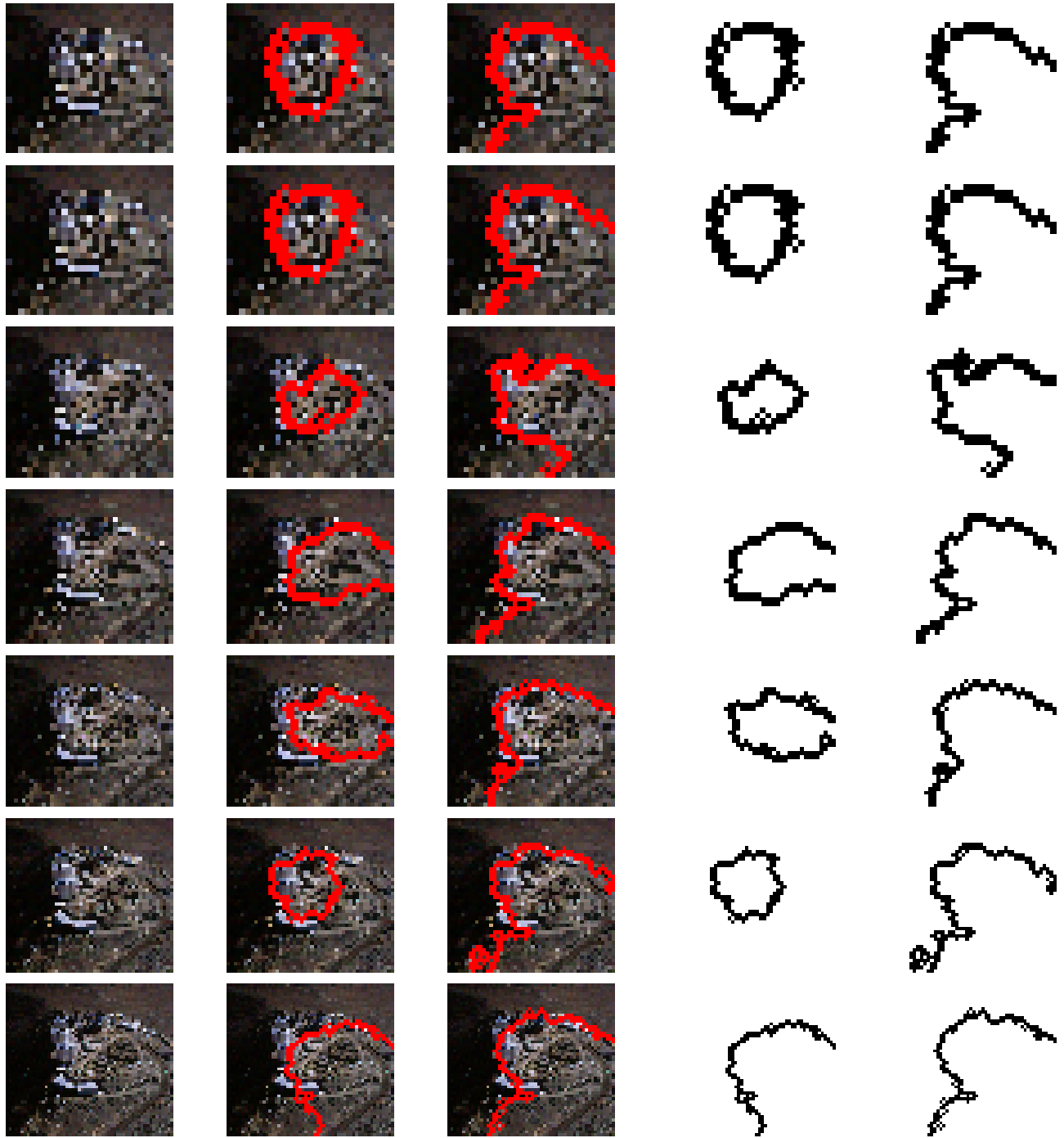


FIGURE 8. The Results of Partitioning for 8-Point Pattern, Pictures 1-7.  
 Original Image - Spectral Partit. - NCUT Partit.- Spectral Line - NCUT Line.  
 Sizes of images (pixels): 483, 648, 868, 1188, 1599, 2112, 2856

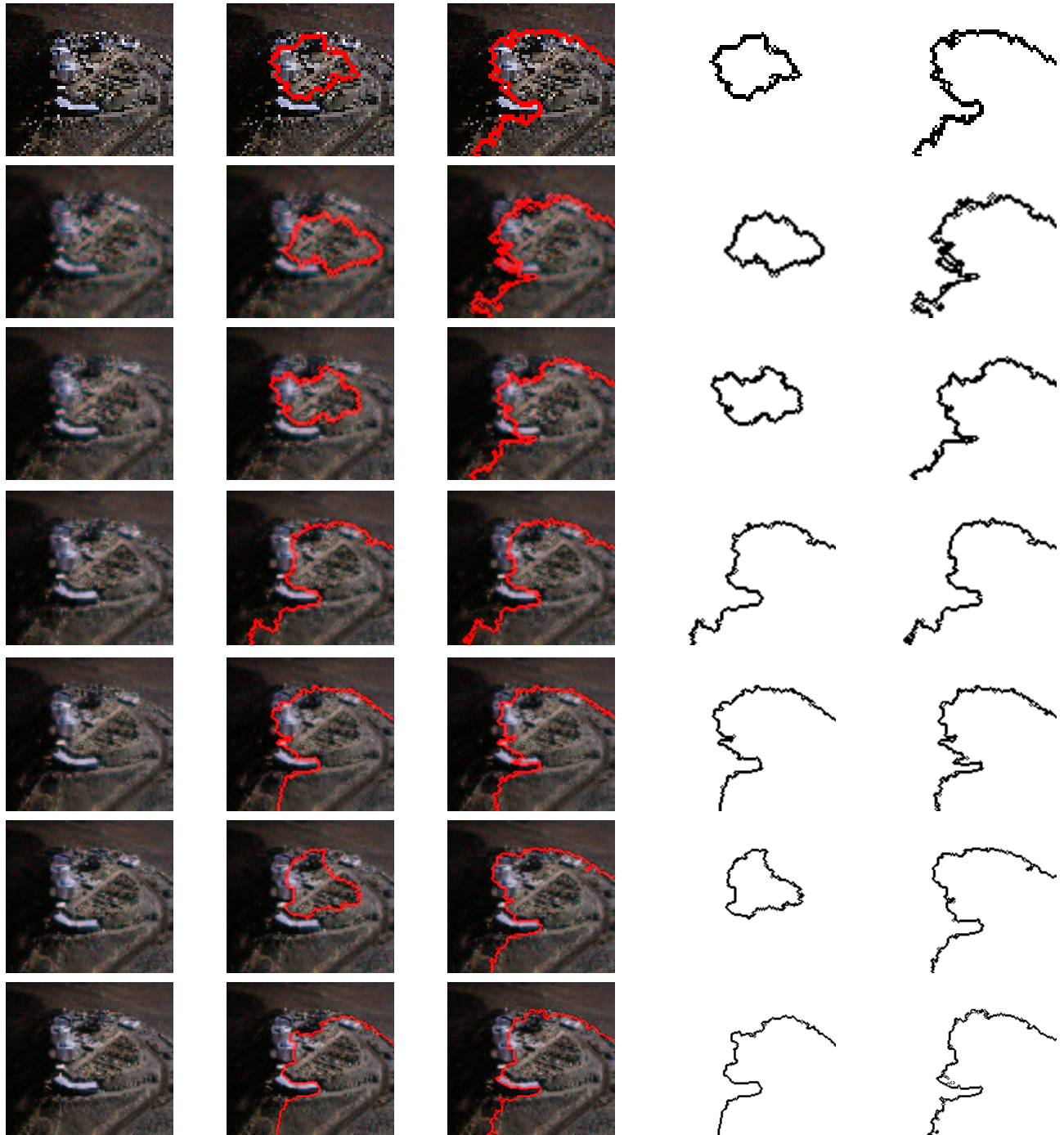


FIGURE 9. The Results of Partitioning for 8-Point Pattern, Pictures 8-14.  
 Original Image - Spectral Partit. - NCUT Partit.- Spectral Line - NCUT Line.  
 Sizes of images (pixels): 3835, 5100, 6873, 9200, 12296, 16605, 22152

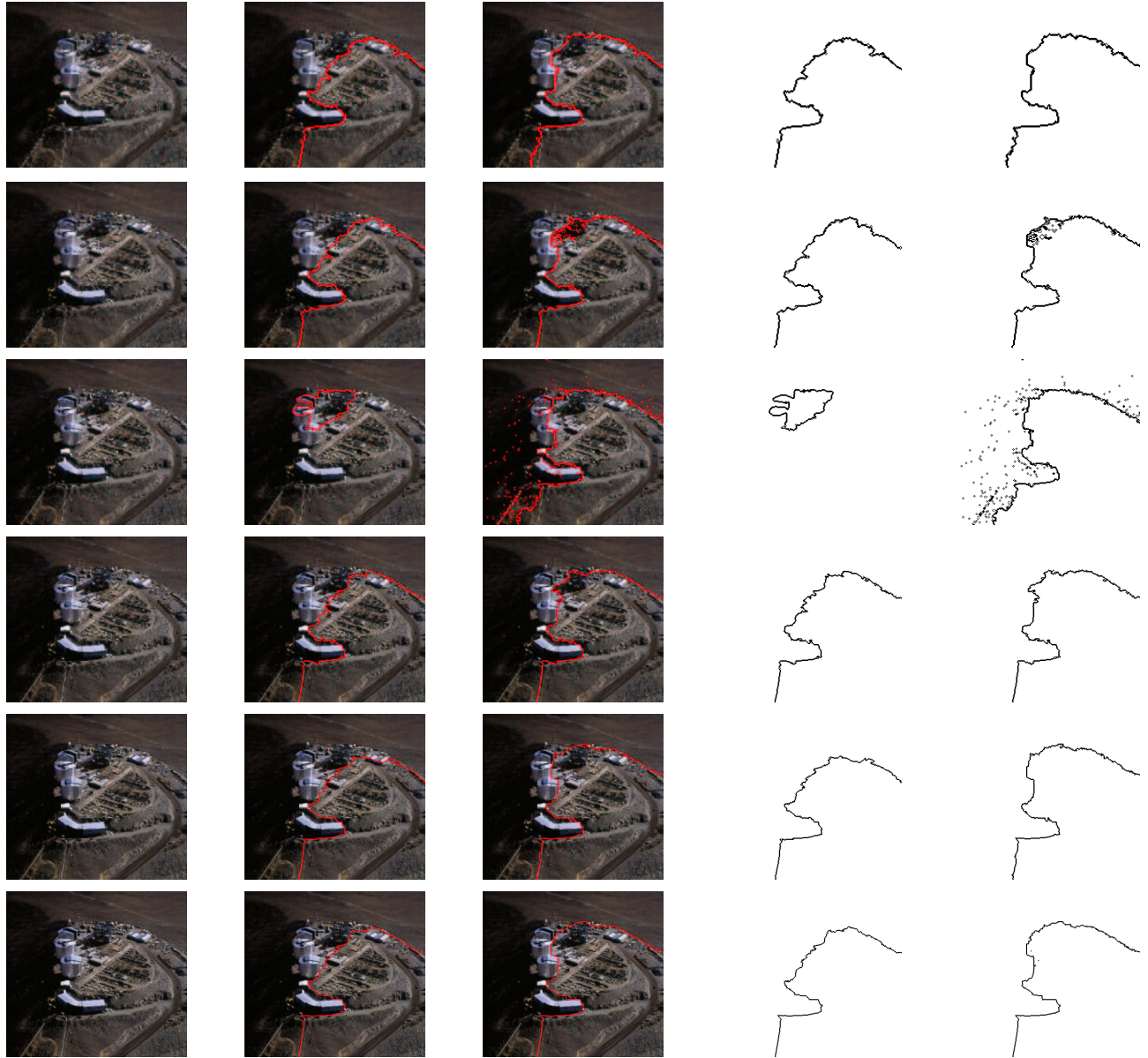


FIGURE 10. The Results of Partitioning for 8-Point Pattern, Pictures 15-20.  
 Original Image - Spectral Partit. - NCUT Partit.- Spectral Line - NCUT Line.  
 Sizes of images (pixels): 29700, 39919, 53482, 71680, 95904, 128625