Induced Markov Chains Combined with Graph Differences for Data Compression

Stefan P. Müller
Institute of Mathematics
Humboldt-Universität zu Berlin
Email: stefan.mueller@math.hu-berlin.de

***Abstract*—Graph-based data occurs in various applications, e.g. finite-element simulations and computer-generated imagery. There are several techniques to compress these data sets with prediction methods and encoding of the residual. The focus of these methods is almost always on the prediction rather than on encoding. The model of induced Markov chains (iMc) is a new strategy to approximate the information content of graph-based data. For this purpose, we define transition probabilities between the occurring values that are dependent on the topology of the graph. The basic idea is to transform a topological relation into a value-based one. The transition probabilities along with an initial distribution can be interpreted as a Markov chain, the so-called iMc. The topology combined with the transition probabilities can be used as side information for an encoder. We combine an iMc encoding scheme with graph differences as a prediction method, since some correlations cannot be entirely removed neither by the prediction method, nor by the iMc by its own.**

I. Introduction

Data compression is nowadays an essential component for the storage and transmission of all kinds of data. One application field is the simulation of car crashes, which began in the early 1980s [1]. It is a central component in the development and quality management of automakers. Due to safety regulations and quality management, a portion of the simulations is stored for some months up to several years. This leads, on the one hand, to a huge number of stored datasets. On the other hand, the simulation results increase in size as the engineers want to improve the accuracy of the simulation results, e.g. by refining the model. The simulation results consist of various data sets like header, initial components like connectivities, and time dependent data like coordinates, velocity, and stresses. The time dependent data will be calculated numerically, and thus it is only precise up to a certain accuracy. This justifies a lossy compression up to a respective precision. Moreover, these data sets are usually defined on elements of a mesh, e.g. nodes, edges, or elements. Hence, we may assume that the correlation of the data can be expressed by the topology of the underlying graph, see also [2]. When we describe a two dimensional shell-element part of a car by a finite element mesh, there generally is a high redundancy in the data. For a good compression rate, it is crucial to eliminate these redundancies. This task can be tackled in two ways. The first one is to find the underlying dependencies in the data. The second one is to predict the data in a way that the resulting variables are independent or at least close to independent. A prediction usually modifies the distribution of a data set. We will investigate a combination of these two approaches, which will lead to the best compression rates in our test case, see Section VIII. For the prediction part, we will use an iterative application of graph differences, which will be specified in Section IV. We exploit the remaining dependencies by considering transition probabilities which are defined on relations between neighbored values, see Section III. We call this technique induced Markov chain (iMc) and will use its statistics as side information for an entropy encoder.

The strategy to use the topology of a general undirected graph for a specialized encoding scheme on node values is otherwise only considered in Markov random fields (MRF). When we consider an entropy encoding scheme with MRFs on a spanning tree for one time step, the time complexity is $\mathcal{O}(n \cdot m^2)$ with n the number of nodes and m the size of the alphabet [3]. Thus, it is mostly used in situations where the size of the alphabet is very small, e.g. black/white images. Regarding iMc, the time complexity can be bounded by $\mathcal{O}(n + m)$, see Section V.

II. Entropy rate of a Markov chain

In this section, we introduce some general definitions and theorems for the entropy calculation of Markov chains.

For an independent and identically distributed (iid) stochastic process X with values in S, the entropy rate can be calculated by

$$H(X) = -\sum_{s \in S} \mathbb{P}[X = s] \log_2 \mathbb{P}[X = s]. \tag{1}$$

In the case of any dependency the given formula would overestimate the information content, see [4]. Hence, the entropy rate will not provide an upper bound for the compression rate

$$K = Ips/H(X), \tag{2}$$

where Ips is the size of a symbol. In the context of crash test simulation results Ips is usually 32 bits.

We will list some properties of a Markov chain and its entropy calculation to prove that it is well-defined in the iMc case. Let $X_0, X_1, ...$ be a sequence of random variables with values in S, which is a finite set with m elements $s_1, ..., s_m$, the so-called *state space*. A *discrete Markov chain* M is a sequence of these X_i satisfying the *Markov property* for all $t_0, ..., t_k \in S$:

$$\mathbb{P}[X_{k+1}|X_0 = t_0, ..., X_k = t_k] = \mathbb{P}[X_{k+1}|X_k = t_k], \forall k \in \mathbb{N}.$$

A Markov chain M is called *time homogeneous*, if for all $k \in \mathbb{N}$ and all $s, t \in S$, it holds:

$$\mathbb{P}[X_k = s | X_{k-1} = t] = \mathbb{P}[X_1 = s | X_0 = t].$$

A Markov chain is said to be *irreducible*, if for all combinations of $s, t \in S$, there exists a $k \in \mathbb{N}$ such that

$$\mathbb{P}[X_k = t | X_0 = s] > 0.$$

A Markov chain is called *positive recurrent*, if the mean recurrence time, see [5], of all states $s \in S$ is finite. A Markov chain M is called *mean ergodic* if and only if

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} \mathbb{P}[X_k = s_j | X_0 = s_i] = \mu_j, \ \ \forall i, j \in \{1, ..., m\}.$$

Theorem 2.1 ([6]): An irreducible Markov chain with a finite number of states is positive recurrent.

A *stochastic matrix* is a matrix whose rows are distributions. The *transition probability matrix* $P = (P_{ij})_{i,j=1,...,m}$ with $P_{ij} = \mathbb{P}[X_1 = s_j | X_0 = s_i]$ is a special case of a stochastic matrix. A distribution μ on S is called *invariant* if $\mu P = \mu$.

Theorem 2.2 ([6]): If a Markov chain with a finite number of states is irreducible, it will have an invariant distribution.

Theorem 2.3 ([7]): Let M be an irreducible, positive recurrent Markov chain with invariant distribution μ. Then M is mean ergodic.

The definition of mean ergodicity coincides with the definition of ergodicity in dynamical systems [7].

Theorem 2.4 ([8]): Let M be a mean ergodic, time homogeneous Markov chain with invariant distribution vector μ and transition probability matrix P. The entropy rate for the Markov chain is

$$H(M) = -\sum_{i,j=1}^{m} \mu_i P_{ij} \log_2 P_{ij}. \tag{3}$$

The proof of Theorem 2.4 mainly depends on the chain rule for entropy [4], the finiteness of the state space, and the time homogeneity of the current Markov chain.

Corollary 2.5: The entropy rate of a time homogeneous, irreducible, finite state Markov chain M is determined by Formula (3).

III. Induced Markov chains

In this section, we define the iMc and its entropy rate.

The identification of graph-based data with a Markov chain is based on the calculation of transition probabilities between values of adjacent nodes.

Lemma 3.1 ([5]): The tuple (P, ν) of a stochastic matrix P and a distribution ν defines a finite state space, time homogeneous Markov chain M with transition probabilities given in P and an initial distribution ν.

$$\mathbb{P}[X_0 = s] = \nu_i, \ \mathbb{P}[X_{k+1} = t | X_k = s] = P_{ij}, \ \ \forall s, t \in \mathcal{A}.$$

Let a connected graph $G = (N, E)$ with n nodes, an alphabet $\mathcal{A} = \{a_1, ..., a_m\}$, and a node value vector $v \in \mathcal{A}^n$ be given. We assume that all values of the alphabet $\mathcal{A}$ will be part of the node values vector v.

We introduce the *transition probabilities* based on a graph with node values by the relative frequency of letter pairs as:

$$P_{ij} := \frac{\#\{(k,l) \in E | v_k = a_i \wedge v_l = a_j\}}{\#\{(k,\cdot) \in E | v_k = a_i\}}, \tag{4}$$

for all $i, j \in \{1, ..., m\}$. For an example, see Figure 1. We store the probabilities P_{ij} in the transition probability matrix P. The initial distribution ν is defined by the relative frequency of the node values:

$$\nu_i := \frac{\#\{k \in N | v_k = a_i\}}{n}. \tag{5}$$

We introduce the *induced Markov chain* (iMc) as the Markov chain from Lemma 3.1 with transition probabilities defined by Formula (4) and initial distribution defined by (5).

Lemma 3.2: An iMc M induced by a connected graph G is irreducible.

Proof: Let the node values $v \in \mathcal{A}^n$ of the graph G and $s, t \in \mathcal{A}$ be given. W.l.o.g. $E \neq \emptyset$. Let first s and t be different. Because of the minimality of $\mathcal{A}$, there have to be two nodes n_i and n_j with values s and t, respectively. As the graph G is connected, there exists at least one path from n_i to n_j with length k consisting of edges in E. Every edge of this path can be matched to a nonzero probability that connects two values of the state space $\mathcal{A}$. Thus, $\mathbb{P}[X_k = t | X_0 = s] > 0$. For $s = t$, let n_i be a node with value $v_i = s$. There exists an adjacent node n_j. W.l.o.g. let its value be $v_j = t$. Then it holds: $\mathbb{P}[X_2 = s | X_0 = s] \geq \mathbb{P}[X_2 = s | X_1 = t]\mathbb{P}[X_1 = t | X_0 = s]$. □

Corollary 3.3: The entropy rate of an iMc induced by a connected graph G is well defined and can be calculated by Equation (3).

As we do not want to compress a Markov chain with infinite many time steps but a graph with n nodes, we use the relative frequencies of the values ν instead of the stationary distribution μ in Formula (5).

IV. Iterative graph differences and their iMc

In this section, we introduce graph differences as a prediction method for graph-based data. Furthermore we investigate how these differences can be applied iteratively and state some properties of the resulting graphs and their iMc.

Let a graph G, an alphabet $\mathcal{A}$, and a node value vector v as in Section III be given. We match every edge the absolute difference of the starting node and the ending node:

$$u = |v_{\text{end}} - v_{\text{start}}| \in \mathcal{A}^{(1)}, \tag{6}$$

with a new alphabet $\mathcal{A}^{(1)}$ that contains all edge values.

We construct the *line graph*, see [9], $G^{(1)} = (N^{(1)}, E^{(1)})$, whereby every edge $e_i \in E$ is assigned to a node $n_i^{(1)} \in N^{(1)}$, and the node value vector consists of the edge values of the graph G. Two nodes of $N^{(1)}$ are adjacent, if and only if their corresponding edges are incident in G. If two edges $e_i, e_j \in E$ are connected in G, the corresponding nodes $n_i^{(1)}, n_j^{(1)}$ are

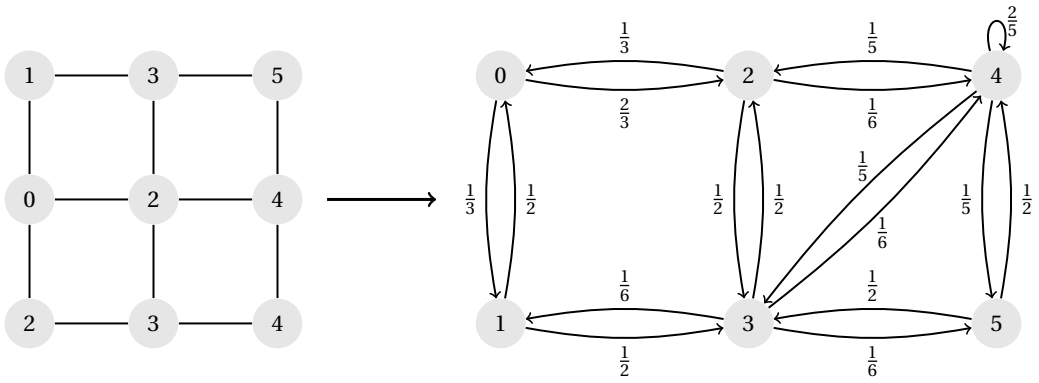

Fig. 1. Mesh with node values and the induced Markov chain with transition probabilities calculated by Formula (4).

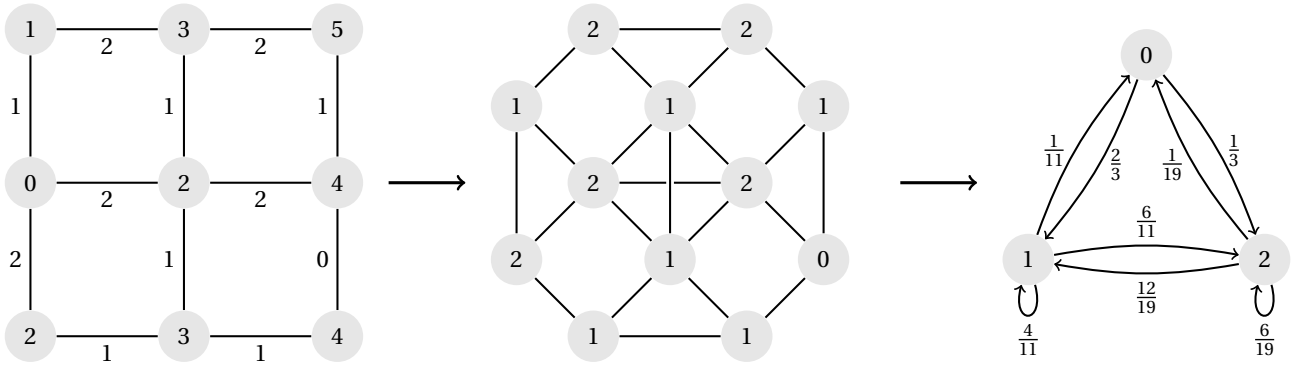

Fig. 2. Mesh with edge and node values, the corresponding line graph and its induced Markov chain. The edge values in the left graph are calculated by Formula (6). The transition probabilities are determined by Equation (4).

connected in $G^{(1)}$. For the example of Figure 1, we get a graph as in Figure 2. The line graph has usually more edges than the underlying graph

$$\#E^{(1)} = \sum_{\langle n_i, n_j \rangle \in E} (\deg(n_i) - 1 + \deg(n_j) - 1). \quad (7)$$

As the number of neighbors in shell element graphs is on average about four, the number of iterative applications of graph differences is limited, due to the size of the line graphs. We will now combine the graph differences with the iMc. For this purpose, we have to ensure that a mean ergodic iMc for G results in a new mean ergodic iMc for $G^{(1)}$ after an application of the graph differences.

Lemma 4.1 ([9]): The line graph of an irreducible graph is irreducible.

Due to Lemma 4.1 and Corollary 3.3, the entropy rate of the iMc for a line graph is also well-defined by Equation (3), if it is well-defined for the underlying graph.

In the construction of the line graph, we omit the signs of edge values. Without the sign, the data can generally not be reconstructed. Hence, in an application, it has to be saved as side information. The following Lemma helps us to quantify the amount of information of the signs.

Lemma 4.2: The entropy rate of an iid process X with a finite number of states $a_1, ..., a_m \in \mathcal{A}$ and a symmetric distribution with mean 0 can be expressed by

$$H(X) = H(|X|) + \frac{\#\text{Nonzero Entries}}{\#\text{Entries}}. \quad (8)$$

PROOF: Let $a_1, ..., a_m \in \mathcal{A}$ be the states with probabilities $p_1, ..., p_m$ and $\sum_{i=1}^{m} p_i = 1$. For m even, set $k = \frac{m}{2}$, $q = 0$, and $0 \log_2 0 = 0$. For m odd, set $k = \frac{m-1}{2}$ and $q = p_{k+1}$.

$$\begin{aligned} H(X) &= -\sum_{i=1}^{m} p_i \log_2 p_i \\ &= -\sum_{i=1}^{k} 2p_i \log_2 2p_i - q \log_2 q + (1-q) \\ &= H(|X|) + \frac{\#\text{Nonzero Entries}}{\#\text{Entries}} \end{aligned}$$

□

V. TIME COMPLEXITY OF IMC GENERATION

In this section, we estimate the time complexity of the iMc statistics calculation.

Let the maximal degree of a node in G be denoted by d, the number of nodes by n, the number of time steps by r, and the number of quantum steps and the size of the alphabet by m, respectively. The time complexity for the calculation of the transition probabilities is $\mathcal{O}((d+d\cdot r)\cdot n+m)$, see Table I. For

TABLE I
TIME COMPLEXITY OF THE IMC STATISTIC CALCULATION.

Calculation of	Time complexity
Adjacency matrix	$\mathcal{O}(d \cdot n)$
Quantized node values	$\mathcal{O}(r \cdot n)$
Initial probabilities	$\mathcal{O}(m + r \cdot n)$
Transition probabilities	$\mathcal{O}(m + d \cdot n \cdot r)$
$\sum$	$\mathcal{O}((d + d \cdot r) \cdot n + m)$

a part with shell elements, the average number of neighbors is approximately four. The maximal degree of a node d is usually be bounded by ten. Thus, d can be neglected.
The time complexity of an encoding scheme with iMc does not depend on a product of the number of nodes and the number of quantum steps like for MRFs [3]. Therefore iMc encoding is applicable in the case of a large number of quantum steps.

VI. APPLICATION OF IMC IN AN ENCODER

The modeling of graph-based data as an iMc can be used in an arithmetic encoder that uses the connectivities and statistics as side information. A possible implementation is to establish the graph, then define a tree, and walk from the root to the leafs. We store the root value without encoding. For non-root elements of the tree, we have the information of its predecessor, and therefore, we can apply the transition probabilities. This leads to an encoding scheme whereby the applied distribution from one node to another will generally change. Arithmetic encoders can handle this situation [10].
There are several possibilities to find a tree of a graph, like breadth-first search and depth-first search [11]. If we use a deterministic strategy, we will not have to store the structure of the tree. In [12], it was shown that in some cases, it can

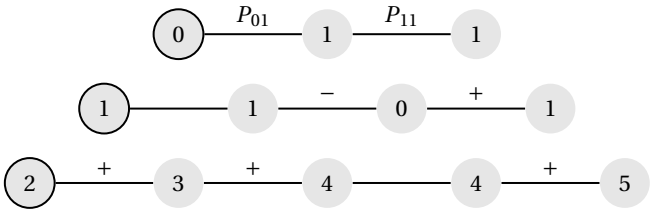

Fig. 3. Snippet of a tree for the example from Figure 1 with two graph differences. The bold-framed values, the signs, and the two transition probabilities have to be provided to an iMc encoder as side information.

be useful to store the tree. In these cases, we can build a minimal spanning tree on the graph with edge values equal to $-\log_2(P_{ij})$, if a_i is the value of the starting node and a_j of the ending node. When we employ graph differences, we additionally have to save the sign of the difference between the value of a node and its predecessor. The usually increasing number of edges in the difference graph (see Equation (7)) does not increase the amount of information when encoding the differences, as we do not have to visit all nodes of the new graph, but $n-1$ nodes. For the prediction, we need a spanning tree of the underlying graph G and the nodes $n_i^{(1)} \in N^{(1)}$ of the line graph that correspond to edges of the spanning tree. A possible realization of iMc with two graph differences in an encoder is pictured in Figure 3.

VII. STRATEGY FOR THE GENERATION OF RESULTS

In this section, we briefly introduce our strategy to generate the results listed in Section VIII.
We investigate the iMc in a way that is independent of the special choice of either an entropy encoder or a spanning tree. Thereby, the distinctions of iid and iMc distributions will be considered, although the entropy encoder shall not be fixed. Therefore, we list the theoretical best compression rate, see Equations (2) and (3), for the iMc on an average tree. We compare these results with those of the iid case, see Equations (1) and (2).
We compress the statistics losslessly by exploiting that the probabilities are rational numbers. For this purpose, we split the probabilities into one array of numerators and one of denominators. For the structure of the transition probability matrix, we use the Compressed Sparse Row format. All integer arrays are predicted and encoded with the zlib [13] afterwards. When we apply graph differences, we add the percentage amount of non-zero entries per difference to the entropy rate, see Equation (8), for both the iid case and the iMc case. For certain spanning trees, it is possible that the entropy rate of the sign array is smaller than the percentage of non-zero values.

VIII. RESULTS

In this section, we list and discuss the results of applying the iMc for data on graphs and on their line graphs as a distribution for an entropy encoder. We compare the results with those regarding the assumption that the data on graphs and their line graphs is independent and identically distributed. We investigate the x coordinate of the biggest part in a variation of the Dodge Neon model [14] with a refined mesh. It is a underbody part with cardan tunnel, has a shell element mesh with 78869 nodes and an x coordinate range of $[-4016, -1900]$. The model was simulated with Pam-Crash [15] where 25 time steps were recorded.
The practical results are listed in Table II and III. The outcomes of iid and iMc encoding differ in two aspects: the size of the encoded data and the size of the statistics as side information. For the iMc case, we have to store the transition probability matrix P for the statistics. In the iid case, the relative frequencies of the values have to be stored as side information. The differences in the size of side information depend on the precision, and whether the data was predicted or not.
The entropy rate is calculated by the Formulas (1) and (3) with the initial distribution, which is equal to the relative frequencies in the iid case. As already mentioned in Section II the entropy rate of iid modeled data overestimates the information content, if there are any dependencies. The iMc can reveal the existence of these dependencies even after two graph differences. Thus, the entropy in the iMc case is always smaller than the entropy for the iid model, see columns "Entropy rate" in Table II and III. Therefore, the size of the

TABLE II
SIZE OF COMPRESSED STATISTICS, ENTROPY AND COMPRESSION RATES FOR THE FIRST TIME STEP OF THE x COORDINATE OF THE NEON PART.

Absolute precision	Number of differences	iMc			iid			Ratio
		Entropy rate	Size of statistics	Compression rate	Entropy rate	Size of statistics	Compression rate	
0.01	0	2.53	479267	0.63	15.64	4779	1.98	0.32
	1	7.62	165073	1.31	9.62	3290	3.22	0.41
	2	8.32	297348	0.83	9.90	3204	3.13	0.27
0.1	0	3.62	205689	1.31	13.53	4072	2.30	0.57
	1	5.71	9898	4.77	6.35	428	5.00	0.95
	2	6.20	11852	4.32	7.20	413	4.42	0.98
1.0	0	2.98	24480	5.86	10.47	1042	3.02	1.94
	1	3.24	644	9.69	3.39	93	9.42	1.03
	2	4.02	693	7.82	4.36	93	7.32	1.07

TABLE III
SIZE OF COMPRESSED STATISTICS, ENTROPY AND COMPRESSION RATES FOR SECOND TO 25TH TIME STEP OF THE x COORDINATE FOR THE NEON PART. ALL TIME STEPS ARE PREDICTED BY ITS PREDECESSOR.

Absolute precision	Number of differences	iMc			iid			Ratio
		Entropy rate	Size of statistics	Compression rate	Entropy rate	Size of statistics	Compression rate	
0.01	0	2.28	304320	8.97	11.18	926	2.86	3.13
	1	1.39	36944	20.65	1.75	147	18.25	1.13
	2	0.77	23838	36.71	0.99	124	32.25	1.14
0.1	0	0.69	24943	40.28	7.91	422	4.04	9.96
	1	0.44	2282	71.47	0.57	68	56.28	1.27
	2	0.18	1628	168.65	0.25	51	127.54	1.32
1.0	0	0.14	1559	224.95	4.97	139	6.44	34.92
	1	0.11	267	301.43	0.12	31	260.51	1.16
	2	0.03	213	1188.76	0.04	28	786.40	1.51

entropy encoded data in the iid case is bigger than in the iMc case. If we consider only one time step, the differences in the sizes of the side information is too big to gain an improvement compared to iid encoding. When we consider time dependent data that is predicted in time, and we save only one transition probability matrix for 24 time steps, then the iMc encoding achieves significant better compression rates. Furthermore, as a result of the high amount of zeros in the case of time predicted data, it is optimal to apply two graph differences in contrast to one graph difference in the case of one time step.

IX. CONCLUSION

We investigated data compression of time and spatial dependent data defined on a graph. Even after two graph-differences the data is not yet fully decorrelated. This can be exploited by an encoder that uses the connectivities and the iMc statistics as side information. Especially if the number of different values can be reduced by time differences or a coarse quantization, this leads to significantly better compression rates. The iMc can be combined with a connectivity compression method like the TG coder [2] or the Cut Border Machine [16]. One big advantage of the iMc is, that it can be employed on general graphs and is not limited to a certain regularity of the mesh. Another advantage is that the iMc can easily be combined with multiple graph differences, although this can become computationally expensive. A possible strategy to avoid the growth of the line graphs, is to specify a certain spanning tree or a set of spanning trees as the underlying graph.

ACKNOWLEDGMENT

The author would like to thank Rudolph Lorentz, Matthew Reyes, Caren Tischendorf, and Lennart Jansen for many productive discussions and the Berlin Mathematical School for conference travel support.

REFERENCES

[1] E. Haug, *Engineering safety analysis via destructive numerical experiments*, EUROMECH 121, Polish Academy of Sciences, Engineering Transactions 29(1), p. 3949, 1981.

[2] C. Touma, C. Gotsman, *Triangle mesh compression*, Proc. Graphics Interface '98, pp. 26-34, 1998.

[3] M. Reyes, *Cutset Based Processing and Compression of Markov Random Fields*, Ph.D. dissertation, University of Michigan, 2010.

[4] T. Cover, J. Thomas, *Elements of information theory*, Wiley-Interscience, New York, NY, 1991.

[5] D. Stroock, *An Introduction to Markov Processes*, Springer, Graduate texts in mathematics, Berlin, Germany, 2005.

[6] P. Hoel, S. Port, C. Stone, *Introduction to stochastic processes*, Houghton Mifflin, Boston, 1972.

[7] A. Klenke, *Probability Theory*, Springer, London, England, 2008.

[8] L. Ekroot, T. Cover, *The Entropy of Markov Trajectories*, IEEE Transactions on information theory, vol. 39, no. 4, 1993.

[9] D. Cvetkovic, P. Rowlinson, S. Simic, *Spectral Generalizations of Line Graphs*, Cambridge University Press, Cambridge, England, 2004.

[10] D. Salomon, *Data Compression*, Springer, London, England, 2007.

[11] T. Cormen, C. Leiserson, R. Rivest, C. Stein, *Introduction to algorithms*, The MIT Press, Cambridge, Massachusetts, 2001.

[12] M. Rettenmeier, *Data Compression for Fluid Dynamics on Irregular Grids*, Logos Verlag Berlin, Germany, 2012.

[13] G. Roelofs, J. Gailly, M. Adler *zlib 1.2.8*, http://www.zlib.net, 2013.

[14] National Crash Analysis Center, *Dodge Neon, Detailed model (272,485 elements)*, http://www.ncac.gwu.edu/vml/models.html, 2006.

[15] ESI Group, *PAM-Crash*, http://virtualperformance.esi-group.com/applications-structural-crash, 2014.

[16] S. Gumhold, W. Straßer, *Real time compression for triangle mesh connectivity*, SIGGRAPH '98, pp.133-140, 1998.