Motion-Adaptive Transforms based on Vertex-Weighted Graphs

Du Liu and Markus Flierl
KTH Royal Institute of Technology
Stockholm, SE-10044, Sweden
{dul, mflierl}@kth.se

Abstract

Motion information in image sequences connects pixels that are highly correlated. In this paper, we consider vertex-weighted graphs that are formed by motion vector information. The vertex weights are defined by scale factors which are introduced to improve the energy compaction of motion-adaptive transforms. Further, we relate the vertex-weighted graph to a subspace constraint of the transform. Finally, we propose a subspace-constrained transform (SCT) that achieves optimal energy compaction for the given constraint. The subspace constraint is derived from the underlying motion information only and requires no additional information. Experimental results on energy compaction confirm that the motion-adaptive SCT outperforms motion-compensated orthogonal transforms while approaching the theoretical performance of the Karhunen Loeve Transform (KLT) along given motion trajectories.

1 Introduction

Standard hybrid coding techniques utilize the concept of motion-compensated predictive coding. The predicted frames and bi-predicted frames are used to exploit the temporal redundancy of frames to achieve compression efficiency. Image sequences are encoded in a closed-loop fashion. This introduces strong dependencies among coded frames. In the case of transmission with packet loss, this increases the risk of error propagation for sequentially decoded pictures [1]. On the other hand, subband coding schemes are designed in an open-loop fashion where decomposition, quantization and entropy coding operate independently [2]. This approach has the potential to avoid error propagation. Due to the motion in image sequences, motion-adaptive decompositions are required.

Since motion-connected pixels are highly correlated, it is possible to perform linear transforms efficiently [3]. [4] proposes wavelets for signals that are defined on the

vertices of an arbitrarily weighted finite graph. The approach is based on the spectral decomposition of the Laplacian matrix of the graph [5]. One drawback of the method is that the resulting representation is overcomplete since there is no downsampling as in the classic discrete wavelet transform. In [6], a graph-based transform is proposed. A picture is represented as a graph signal, where an image edge cuts the link between two graph nodes. The transform matrix is then generated by the Laplacian matrix of the graph. This method requires additional information on the edge map and the adjacency matrix.

The class of motion-compensated orthogonal transforms (MCOT) is designed to compact signal energy efficiently while strictly maintaining orthogonality for any motion field [7, 8]. The transform can be factored into a sequence of incremental transforms. Scale factors are used in each incremental transform to ensure orthogonality and to achieve good energy compaction. Further, MCOTs can be tailored to required motion accuracies and variable block sizes such that they can be combined for efficient video coding [9].

In contrast to previous work, we consider a transform that does not require additional information on the graph. Our graph is obtained from the motion vector information. The vertex weights are given by scale factors, which are defined according to [7]. As the vertex-weighted graph determines the transform partially, our transform is subspace constrained. In particular, the vertex-weighted graph will define a one-dimensional linear subspace. Our transform is designed to achieve optimal energy compaction, given this constraint.

The paper is organized as follows: Section 2 defines the vertex-weighted graph as obtained from the motion vector information and the scale factors. Section 3 discusses the construction of the subspace-constrained transform. Section 4 presents the experimental results.

2 Motion and Vertex-Weighted Graphs

In the following, we define our vertex-weighted graph. Let $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ be a vector representing n pixels. They are linked by motion vectors. Fig. 1 depicts an example of five pixels linked by motion vectors. The pixels form a graph with a tree

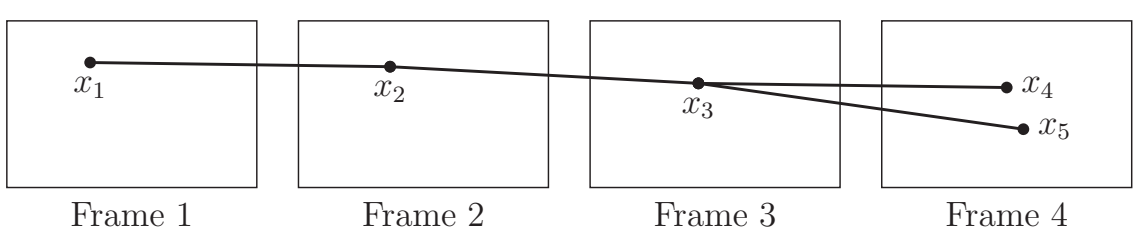

Figure 1: A vertex-weighted graph.

structure. x_1 is the root which will become the temporal lowband pixel. x_2 to x_5 will become the temporal high band pixels. x_1 to x_5 will be processed together by one

transform. The weight of each vertex in this graph will be associated with an energy value.

Any orthonormal transform matrix $T = [t_1, t_2, \ldots, t_n]$ maps the vector $\mathbf{x}$ of pixel intensities according to

$$\mathbf{y} = T^T \mathbf{x}, \tag{2.1}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$ is the output of the transform and t_k with $k = 1, \ldots, n$ are the column vectors in T.

We reuse the concept of scale factors from [7] and let $c_1, c_2, \ldots, c_n$ be the scale factors of the pixels $x_1, x_2, \ldots, x_n$, respectively, i.e., $x_k = c_k x'_k$, $\forall k = 1, 2, \ldots, n$. Each scale factor is related to the count how often the corresponding pixel is used as reference. The weights of the vertices are then $c_1, c_2, \ldots, c_n$ and the vector of scale factors is $\mathbf{c} = [c_1, c_2, \ldots, c_n]^T$. Similar to [7], we assume that these n pixels are connected by ideal motion such that $x'_1 = x'_2 = \cdots = x'_n$. Then, the covariance matrix of $\mathbf{x}$ is

$$L = \mathbf{E}[\mathbf{x}\mathbf{x}^T] = \sigma^2_{x'_1} \mathbf{c}\mathbf{c}^T. \tag{2.2}$$

We are looking for a transform matrix T such that it decorrelates $\mathbf{x}$

$$\Lambda = T^T L T, \tag{2.3}$$

where Λ is a diagonal matrix. Since L is singular with rank 1, Λ is only non-zero in the first diagonal element. Hence, matrix T compacts all the energy into the first coefficient and $\mathbf{x}$ is processed by T as follows:

$$\begin{bmatrix} \sqrt{\sum_{k=1}^n c_k^2} \cdot x'_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = T^T \begin{bmatrix} c_1 x'_1 \\ c_2 x'_1 \\ \vdots \\ c_n x'_1 \end{bmatrix}, \tag{2.4}$$

that is,

$$\sqrt{\mathbf{c}^T \mathbf{c}} = t_1^T \mathbf{c} \tag{2.5}$$

$$0 = t_k^T \mathbf{c}, \forall k = 2, 3, \ldots, n. \tag{2.6}$$

The energy is conserved since $\mathbf{c}^T \mathbf{c} = \sum_{k=1}^n c_k^2$.

Geometrically, (2.5) and (2.6) can be explained as a projection of $\mathbf{c}$ onto an n-dimensional orthogonal space $\mathbb{R}^n$ that is defined by the n orthonormal vectors $t_1, t_2, \ldots, t_n$. To obtain $\sqrt{\mathbf{c}^T \mathbf{c}}$ in the first projection, the first column t_1 in T should be collinear with $\mathbf{c}$, such that the projection of $\mathbf{c}$ onto t_1 captures the total energy. Since T is orthogonal, there is no energy left to be projected on $t_2, \ldots, t_n$. Therefore, the first vector of the transform matrix is determined by $\mathbf{c}$, which is our subspace constraint,

$$t_1 = \frac{\mathbf{c}}{\|\mathbf{c}\|_2} = \frac{1}{\sqrt{\mathbf{c}^T \mathbf{c}}} \mathbf{c}. \tag{2.7}$$

If $\mathbf{x}$ is not due to ideal motion, i.e., $x_1, x_2, \ldots, x_n$ are affected by noise, we will not obtain perfect energy compaction into one coefficient. However, we keep the subspace constraint $\mathbf{c}$ as it reflects the vertex-weighted graph.

3 Subspace-Constrained Transform (SCT)

As t_1 is determined, our goal is to construct the remaining vectors of the orthonormal basis that can achieve optimal energy compaction, given the constraint. Now, assume the n-dimensional orthonormal basis $B = [b_1, b_2, \ldots, b_n]$ with $b_1 = t_1$. Such a basis exists, as we can define the $(n-1)$-dimensional subspace $[b_2, \ldots, b_n]$ by, e.g., the Gram-Schmidt orthonormalization. Then our target basis $t_2, \ldots, t_n$ is simply a rotation of $[b_2, \ldots, b_n]$. Now, our goal is to find this rotation such that the remaining energy is optimally compacted.

3.1 Representation of Orthogonal Subspace

Let the rotation matrix be

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}. \tag{3.1}$$

Then, $t_k = b_2 a_{2k} + b_3 a_{3k} + \cdots + b_n a_{nk}$ for $k = 2, \ldots, n$, where each b_j is a $n \times 1$ vector and each a_{jk} is a scalar. t_k lies in the $(n-1)$-dimensional subspace. The first element in A is set to one with zeros on all the other positions in the first row and the first column. With that, we always preserve the first vector $t_1 = b_1$. We have

$$T = BA, \tag{3.2}$$

that is,

$$\begin{bmatrix} t_1 & t_2 & \cdots & t_n \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2} & \cdots & a_{nn} \end{bmatrix}. \tag{3.3}$$

Now, we define the rotation part of A as A', where A' is an $(n-1) \times (n-1)$ matrix.

$$A' = \begin{bmatrix} a'_2 & a'_3 & \cdots & a'_n \end{bmatrix} = \begin{bmatrix} a_{22} & a_{23} & \cdots & a_{2n} \\ a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}, \tag{3.4}$$

where a'_k is the k-th column of A' with dimension $(n-1) \times 1$. A' rotates the basis $b_2, \ldots, b_n$ to obtain $t_2, \ldots, t_n$. A can be expressed as

$$A = \begin{bmatrix} 1 & 0 \\ 0 & A' \end{bmatrix}. \tag{3.5}$$

Let $B' = [b_2, b_3, \ldots, b_n]$ be the basis that will be rotated. Then $B = [b_1, B']$. Since $b_2 \perp b_3 \perp \cdots \perp b_n$, we have

$$B'^T B' = I_{(n-1)\times(n-1)}, \tag{3.6}$$

which is an $(n-1) \times (n-1)$ identity matrix. Thus,

$$T = \begin{bmatrix} b_1 & B' \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & A' \end{bmatrix} = \begin{bmatrix} t_1 & B'A' \end{bmatrix} = \begin{bmatrix} t_1 & B'a'_2 & B'a'_3 & \cdots & B'a'_n \end{bmatrix}. \tag{3.7}$$

Each $B'a'_k$ has a dimension of $n \times 1$, which is the same as t_k. The covariance based on B is given by

$$M = B^T L B = \begin{bmatrix} b_1^T L b_1 & b_1^T L b_2 & \cdots & b_1^T L b_n \\ b_2^T L b_1 & b_2^T L b_2 & \cdots & b_2^T L b_n \\ \vdots & \vdots & \ddots & \vdots \\ b_n^T L b_1 & b_n^T L b_2 & \cdots & b_n^T L b_n \end{bmatrix}. \tag{3.8}$$

Let M' be the dimension-reduced covariance of M with dimension $(n-1) \times (n-1)$,

$$M' = B'^T L B' = \begin{bmatrix} b_2^T L b_2 & b_2^T L b_3 & \cdots & b_2^T L b_n \\ b_3^T L b_2 & b_3^T L b_3 & \cdots & b_3^T L b_n \\ \vdots & \vdots & \ddots & \vdots \\ b_n^T L b_2 & b_n^T L b_3 & \cdots & b_n^T L b_n \end{bmatrix}. \tag{3.9}$$

M' is the covariance based on B'. M' is symmetric since $b_i^T L b_j = (b_i^T L b_j)^T = b_j^T L b_i$, $\forall i, j \in 2, 3, \ldots, n$.

3.2 Optimal Energy Compaction

Now, we look for the vector t_k that maximizes the energy concentration in the $(n-1)$-dimensional subspace. As t_k is normalized and orthogonal to $t_1, t_2, \ldots, t_{k-1}$, our constrained energy compaction problem is:

$$\min_{t_k} \quad -t_k^T L t_k, \qquad k = 2, \ldots, n, \tag{3.10}$$

$$\text{s.t.} \quad t_k^T t_k = 1, \tag{3.11}$$

$$t_k^T t_j = 0, \qquad j = 1, \ldots, k-1, \tag{3.12}$$

$$t_1 = \frac{\mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{c}}}. \tag{3.13}$$

We rewrite the above constrained problem in unconstrained form [10] and define the cost function

$$J_k = -t_k^T L t_k + \lambda_k (t_k^T t_k - 1) + \sum_{j=1}^{k-1} \nu_{kj} t_k^T t_j, \tag{3.14}$$

where λ_k and ν_{kj} are positive Lagrange multipliers. Our goal is to minimize the cost function of the unconstrained problem.

The vectors $t_2, t_3, \ldots, t_n$ will be constructed successively. First, we consider the cost function for t_2,

$$J_2 = -t_2^T L t_2 + \lambda_2(t_2^T t_2 - 1) + \nu_{21} t_2^T t_1, \tag{3.15}$$

where λ_2 and ν_{21} are positive Lagrange multipliers. We have

$$t_2^T L t_2 = (B' a_2')^T L (B' a_2') = a_2'^T B'^T L B' a_2' = a_2'^T M' a_2' \tag{3.16}$$
$$t_2^T t_2 = (B' a_2')^T (B' a_2') = a_2'^T B'^T B' a_2' = a_2'^T a_2' \tag{3.17}$$
$$t_2^T t_1 = (B' a_2')^T b_1 = a_2'^T B'^T b_1 = 0. \tag{3.18}$$

From (3.16) we see that the energy contribution of M' in a_2' is the same as the contribution of L in t_2. (3.17) states that the normalization of t_2 is the same as that of a_2', i.e., when $a_2'^T a_2' = 1$, we obtain $t_2^T t_2 = 1$. (3.18) implies that the way we construct t_2 guarantees that t_2 is orthogonal to t_1 since $B' \perp b_1$. Thus, we can eliminate the last part of the cost function. (3.15) can be rewritten as

$$J_2 = -a_2'^T M' a_2' + \lambda_2(a_2'^T a_2' - 1). \tag{3.19}$$

To minimize J_2, we differentiate J_2 with respect to a_2', that is,

$$\frac{\partial J_2}{\partial a_2'} = -2M' a_2' + 2\lambda_2 a_2' = 0 \tag{3.20}$$
$$M' a_2' = \lambda_2 a_2'. \tag{3.21}$$

Thus, to minimize J_2, a_2' needs to be an eigenvector of M' with corresponding eigenvalue λ_2.

Note, the direct differentiation of J_2 with respect to t_2 in (3.15) is not helpful. t_2 is an n-dimensional vector and the underlying problem gives only $n-1$ independent equations.

Next, we construct t_3 such that t_3 maximizes the concentration of the remaining energy. Since t_3 needs to be orthogonal to t_1 and t_2, the cost function for t_3 is

$$J_3 = -t_3^T L t_3 + \lambda_3(t_3^T t_3 - 1) + \nu_{31} t_3^T t_1 + \nu_{32} t_3^T t_2, \tag{3.22}$$

where λ_3, ν_{31}, and ν_{32} are positive Lagrange multipliers. We have

$$t_3^T L t_3 = a_3'^T M' a_3' \tag{3.23}$$
$$t_3^T t_3 = a_3'^T a_3' \tag{3.24}$$
$$t_3^T t_1 = 0 \tag{3.25}$$
$$t_3^T t_2 = (B' a_3')^T (B' a_2') = a_3'^T a_2'. \tag{3.26}$$

As t_3 shall be orthogonal to t_2, a_3' needs to be orthogonal to a_2'. As a_2' is an eigenvector of M' and M' is a symmetric matrix, we assume that a_3' is also an eigenvector of M', thus, $a_3' \perp a_2'$. This assumption will be verified later. The cost function for t_3 is

$$J_3 = -a_3'^T M' a_3' + \lambda_3(a_3'^T a_3' - 1). \tag{3.27}$$

Differentiation with respect to a'_3 yields

$$\frac{\partial J_3}{\partial a'_3} = -2M'a'_3 + 2\lambda_3 a'_3 = 0, \tag{3.28}$$

which gives the linear eigenproblem

$$M'a'_3 = \lambda_3 a'_3. \tag{3.29}$$

That is, a'_3 is an eigenvector of M' that is orthogonal to a'_2. At the same time, this eigenvector minimizes above cost function.

We continue the process for $t_4, t_5, \ldots, t_n$ and obtain $a'_4, a'_5, \ldots, a'_n$ as eigenvectors of M'. Thus, A' is a matrix of eigenvectors of M'. Thus, $A'^T A' = A' A'^T = I_{(n-1)\times(n-1)}$. Note that M' is obtained by B', and hence, A' is dependent on B'. For any chosen orthonormal basis B', we can calculate its corresponding A'. We are free to choose the basis B'. In any case, we obtain our transform matrix T according to (3.7) which always satisfies $T^T T = T T^T = I_{n\times n}$. The first vector t_1 reflects the vertex-weighted graph and the transform matrix T achieves optimal energy compaction for the given subspace constraint.

4 Experimental Results

In our experiments, we evaluate the energy compaction for the QCIF sequences *Foreman* and *City*. We compare the motion-compensated orthogonal transform [7], SCT, and KLT for given graphs. The basis vectors t_k, $k = 2, 3, \ldots, n$, for SCT and the basis vectors t_k, $k = 1, 2, \ldots, n$, for the KLT are signal dependent. For a practical coding scheme, they need to be signaled to the decoder. At this point, we look at the energy compaction only. Practical coding schemes will be considered in future work.

To estimate the covariance matrix, we consider samples that use the same vertex-weighted graph, i.e., samples with the same motion vectors and the same scale factors. If the number of samples is not sufficient for a reliable estimate, the covariance matrix is not determined and we ignore those samples. In the experiments, the smallest number of samples is set to thirty and graphs with less than nine connections are considered for evaluation. The motion-compensated orthogonal transform does not need to estimate the covariance matrix. However, we use only the samples that provide a reliable estimate for SCT and KLT to ensure a fair comparison.

For each instance of the graph, the transform determines the co-located subband coefficients. Considering the example in Fig. 1, the output coefficients $y_1, \ldots, y_5$ will be at the same positions as $x_1, \ldots, x_5$, respectively. The number of frames that are used to construct the graphs is set to four. At this point, a larger frame number affects the reliability of the estimated covariance matrix negatively. The size of a group of pictures (GOP) is chosen to be sixteen. A hierarchical decomposition is performed on each GOP. In the experiments, a GOP of sixteen frames is separated into four sets and each set contains four successive frames. Each set outputs one lowband and three highbands. Thus, the first decomposition level produces four temporal lowbands to

be forwarded to the second decomposition level. If a different GOP size is chosen, the number of frames included in the graphs and the number of decomposition levels can be rearranged. The graphs are defined by 16×16 block motion with a search range of ± 32.

Tables 1 and 2 show the relative energy in the temporal subband frames for the first set of four frames on the first decomposition level for *Foreman* and *City*, respectively. On the first decomposition level, the scale factors are all ones as no pixel has been processed before. Thus, the first basis vector t_1 is a normalized all-one vector. Tables 3 and 4 show the relative energy in the temporal subband frames on the second decomposition level for *Foreman* and *City*, respectively. Note, SCT provides the same lowband energy as MCOT since both share the same basis vector t_1. For higher basis vectors t_k, $k > 1$, SCT closes the gap to the optimal Karhunen Loeve Transform (KLT). Tables 3 and 4 show that the second largest energy component of SCT is much greater than that of MCOT. This is due to the improved energy compaction of SCT. Note, the subspace constraint is uniquely determined by the underlying motion information. No extra information needs to be transmitted to recover the subspace constraint itself. This is not the case for the KLT.

Table 1: Relative energy in the temporal subband frames of the first four frames on the first decomposition level for *Foreman*.

	Lowband	Highband 1	Highband 2	Highband 3
MCOT	99.36%	0.12%	0.42%	0.10%
SCT	99.36%	0.54%	0.08%	0.02%
KLT	99.42%	0.49%	0.07%	0.02%

Table 2: Relative energy in the temporal subband frames of the first four frames on the first decomposition level for *City*.

	Lowband	Highband 1	Highband 2	Highband 3
MCOT	93.99%	1.54%	3.13%	1.34%
SCT	93.99%	4.08%	1.40%	0.53%
KLT	94.18%	3.95%	1.35%	0.52%

Table 3: Relative energy in the temporal subband frames on the second decomposition level for *Foreman*.

	Lowband	Highband 1	Highband 2	Highband 3
MCOT	98.32%	0.37%	0.72%	0.59%
SCT	98.32%	1.29%	0.30%	0.09%
KLT	98.35%	1.30%	0.28%	0.07%

Table 4: Relative energy in the temporal subband frames on the second decomposition level for *City*.

	Lowband	Highband 1	Highband 2	Highband 3
MCOT	90.12%	2.95%	4.27%	2.66%
SCT	90.12%	7.47%	1.86%	0.55%
KLT	91.22%	6.50%	1.77%	0.51%

5 Conclusions

This paper presents a class of motion-adaptive transforms that is based on vertex-weighted graphs. Each graph is determined by motion vector information where the weights of the vertices are defined by scale factors. The vertex-weighted graph determines uniquely the first basis vector of the linear transform. This first vector defines a subspace that constrains the energy compaction of our transform. Our SCT is constructed step-by-step from given graphs, where each graph can be defined by temporal, spatial, or spatio-temporal connections. SCT achieves optimal energy compaction, given our subspace constraint. Moreover, if there exist additional constraints, our SCT is able to incorporate these while achieving optimal energy compaction, given these constraints.

6 Acknowledgments

This work has been supported in part by the Swedish Research Council under the grant 2011-5841.

References

[1] G. Sullivan and T. Wiegand, "Video compression – from concepts to the H.264/AVC standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, 2005.

[2] V. K. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 9–21, 2001.

[3] E. Martínez-Enríquez, F. Díaz-de-María, and A. Ortega, "Video encoder based on lifting transforms on graphs," in *Proc. of the IEEE International Conference on Image Processing*, Sep. 2011, pp. 3509–3512.

[4] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

[5] F. R. K. Chung, *Spectral Graph Theory*, ser. CMBS Regional Conference Series in Mathematics, No. 92. American Mathematical Society, 1997.

[6] W.-S. Kim, S. K. Narang, and A. Ortega, "Graph based transforms for depth video coding," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2012, pp. 813–816.

[7] M. Flierl and B. Girod, "A motion-compensated orthogonal transform with energy-concentration constraint," in *Proc. of the IEEE International Workshop on Multimedia Signal Processing*, Oct. 2006, pp. 391–394.

[8] ——, "Half-pel accurate motion-compensated orthogonal video transforms," in *Proc. of the IEEE Data Compression Conference*, Mar. 2007, pp. 13–22.

[9] D. Liu and M. Flierl, "Video coding with adaptive motion-compensated orthogonal transforms," in *Proc. of the Picture Coding Symposium*, May 2012, pp. 293–296.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.