Lifting Transforms on Graphs and Their Application to Video Coding

Eduardo Martínez Enríquez

Department of Signal Theory and Communications
Universidad Carlos III de Madrid

17th of December, 2013

Outline

Motivation

- **Motivation:** In several applications, it is useful to achieve a sparse representation of the signal of interest
  - Standard separable DWT and DCT $\rightarrow$ efficient representation of smooth functions, but...

Motivation

- **Motivation:** In several applications, it is useful to achieve a sparse representation of the signal of interest
  - Standard separable DWT and DCT $\rightarrow$ efficient representation of smooth functions, but...
  - Non-sparse representations of signals with large discontinuities (e.g., contours in images)

Motivation

- **Motivation:** In several applications, it is useful to achieve a sparse representation of the signal of interest
  - Standard separable DWT and DCT $\rightarrow$ efficient representation of smooth functions, but...
  - Non-sparse representations of signals with large discontinuities (e.g., contours in images) $\implies$ interest in *directional transforms*

Motivation

- **Motivation:** In several applications, it is useful to achieve a sparse representation of the signal of interest
  - Standard separable DWT and DCT $\rightarrow$ efficient representation of smooth functions, but...
  - Non-sparse representations of signals with large discontinuities (e.g., contours in images) $\implies$ interest in *directional transforms*
- **Related Work:**
  - Image Processing:
    - [Velisavljevic et al., 2006, Le Pennec and Mallat, 2005, Shen and Ortega, 2008, Fattal, 2009]
  - Video Coding: Lifting in the temporal domain for video coding [Secker and Taubman, 2003, Pesquet-Popescu and Bottreau, 2001]

Motivation

- **Motivation:** In several applications, it is useful to achieve a sparse representation of the signal of interest
  - Standard separable DWT and DCT $\rightarrow$ efficient representation of smooth functions, but...
  - Non-sparse representations of signals with large discontinuities (e.g., contours in images) $\implies$ interest in *directional transforms*
- **Related Work:**
  - Image Processing:
    - [Velisavljevic et al., 2006, Le Pennec and Mallat, 2005, Shen and Ortega, 2008, Fattal, 2009]
  - Video Coding: Lifting in the temporal domain for video coding [Secker and Taubman, 2003, Pesquet-Popescu and Bottreau, 2001]

Nevertheless... these transforms filter in the spatial or in the temporal domain independently

Motivation

Original Motivation

Generalize lifting transforms for video coding

- Spatio-temporal lifting $\rightarrow$ Non separable approach, against common Wavelet-based video coders (t+s)

Motivation

Original Motivation

Generalize lifting transforms for video coding

- Spatio-temporal lifting $\rightarrow$ Non separable approach, against common Wavelet-based video coders (t+s)

more in general...

Motivation

Obtain N-dimensional directional transforms with good compaction ability and other interesting properties

Motivation

Original Motivation

Generalize lifting transforms for video coding

- Spatio-temporal lifting $\rightarrow$ Non separable approach, against common Wavelet-based video coders (t+s)

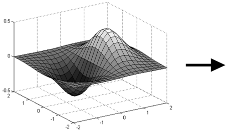
more in general...

Motivation

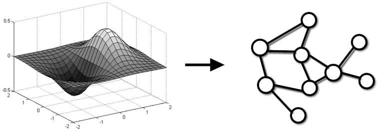
Obtain N-dimensional directional transforms with good compaction ability and other interesting properties

Our new transform is able to filter following **N-dimensional directions** of high correlation... How do we do this?

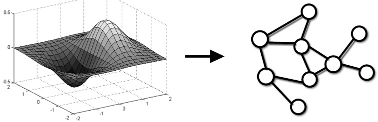
Introduction to the transform



Introduction to the transform

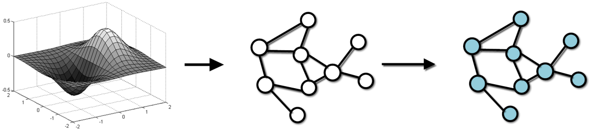


Introduction to the transform



- Describe an N-dimensional signal as a graph
  - Correlated samples should be linked and vice versa
  - Graph captures correlation in N-dimensional domains

Introduction to the transform



- Describe an N-dimensional signal as a graph
  - Correlated samples should be linked and vice versa
  - Graph captures correlation in N-dimensional domains
- Perform a lifting transform on this graph
  - Filtering operations using neighbor nodes $\Longrightarrow$ graph construction defines filtering directions... we can avoid filtering across large discontinuities!!

Objectives

Objectives

- Description of a general framework to construct **N-dimensional directional transforms**
- **Optimization of lifting transforms on graphs**
  - In the sense of energy compaction
- **Video coding application of the proposed transform**

Outline

Lifting Transforms

- Lifting transforms lead to:
  - Multiresolution analysis of a given signal
  - More compact representation of the signal

Lifting Transforms

- Lifting transforms lead to:
  - Multiresolution analysis of a given signal
  - More compact representation of the signal
- Three stages:

Lifting Transforms

- Lifting transforms lead to:
  - Multiresolution analysis of a given signal
  - More compact representation of the signal
- Three stages:
  - **Split** input data in Prediction ($\mathcal{P}$) and Update ($\mathcal{U}$) **disjoint sets**

Lifting Transforms

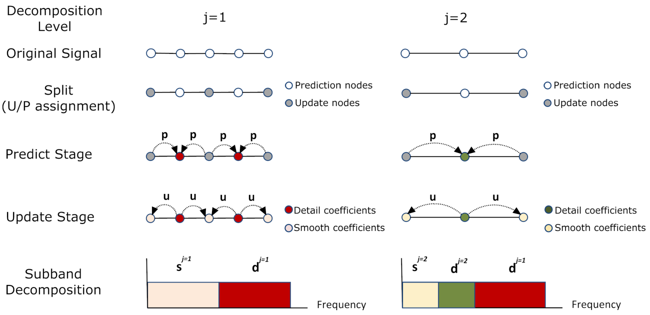
- Lifting transforms lead to:
  - Multiresolution analysis of a given signal
  - More compact representation of the signal
- Three stages:
  - **Split** input data in Prediction ($\mathcal{P}$) and Update ($\mathcal{U}$) **disjoint sets**
  - **Predict** every sample of the $\mathcal{P}$ from the $\mathcal{U}$ set using **p** filters $\rightarrow$ *detail* coefficients

Lifting Transforms

- Lifting transforms lead to:
  - Multiresolution analysis of a given signal
  - More compact representation of the signal
- Three stages:
  - **Split** input data in Prediction ($\mathcal{P}$) and Update ($\mathcal{U}$) **disjoint sets**
  - **Predict** every sample of the $\mathcal{P}$ from the $\mathcal{U}$ set using **p** filters $\rightarrow$ *detail* coefficients
  - **Update** every sample of the $\mathcal{U}$ set using the $\mathcal{P}$ set using **u** filters $\rightarrow$ *smooth* coefficients

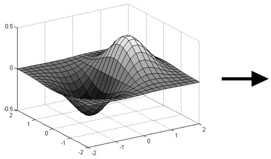
Lifting Transforms

- Lifting transforms lead to:
  - Multiresolution analysis of a given signal
  - More compact representation of the signal
- Three stages:
  - **Split** input data in Prediction ($\mathcal{P}$) and Update ($\mathcal{U}$) **disjoint sets**
  - **Predict** every sample of the $\mathcal{P}$ from the $\mathcal{U}$ set using **p** filters $\rightarrow$ *detail* coefficients
  - **Update** every sample of the $\mathcal{U}$ set using the $\mathcal{P}$ set using **u** filters $\rightarrow$ *smooth* coefficients



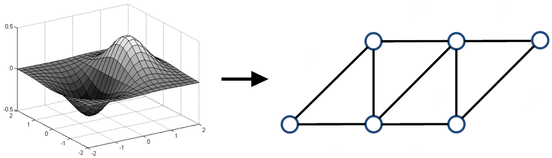
Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?



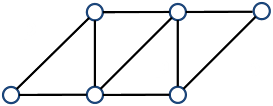
Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?



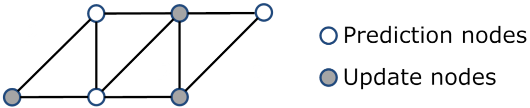
Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?
  - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?



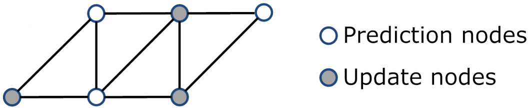
Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
    - How should the graph be constructed to capture the correlation of the signal?
    - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?



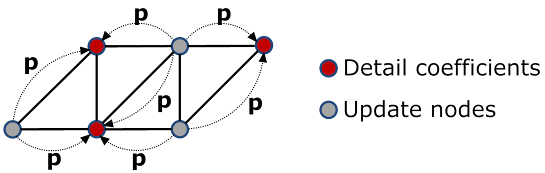
Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?
  - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?
  - How should **u** and **p** filters be designed?



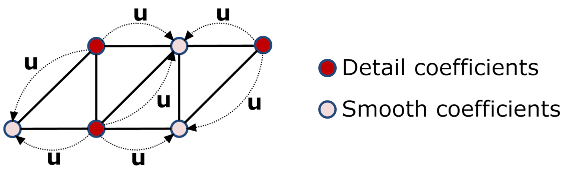
Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
    - How should the graph be constructed to capture the correlation of the signal?
    - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?
    - How should **u** and **p** filters be designed?



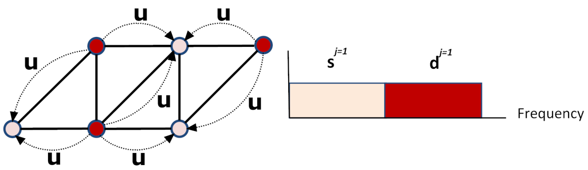
Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?
  - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?
  - How should **u** and **p** filters be designed?



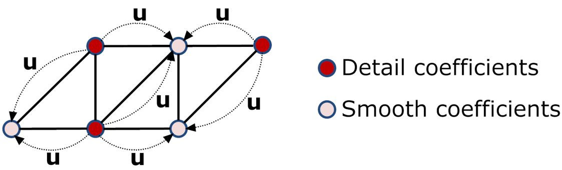
Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?
  - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?
  - How should **u** and **p** filters be designed?



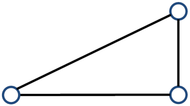
Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?
  - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?
  - How should **u** and **p** filters be designed?
  - How should the graph be constructed at decomposition levels $j > 1$?



Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?
  - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?
  - How should **u** and **p** filters be designed?
  - How should the graph be constructed at decomposition levels $j > 1$?



Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?
  - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?
  - How should **u** and **p** filters be designed?
  - How should the graph be constructed at decomposition levels $j > 1$?

Lifting Transforms on Graphs

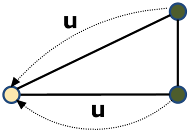
- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?
  - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?
  - How should **u** and **p** filters be designed?
  - How should the graph be constructed at decomposition levels $j > 1$?

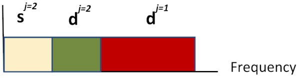
Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?
  - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?
  - How should **u** and **p** filters be designed?
  - How should the graph be constructed at decomposition levels $j > 1$?

Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?
  - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?
  - How should **u** and **p** filters be designed?
  - How should the graph be constructed at decomposition levels $j > 1$?





Lifting Transforms on Graphs

- Given a graph, non assumptions about it structure... **different number of neighbors for each node** $\rightarrow$ Some questions and peculiarities arise!
  - How should the graph be constructed to capture the correlation of the signal?
  - How should the $\mathcal{U}/\mathcal{P}$ assignment be performed?
  - How should **u** and **p** filters be designed?
  - How should the graph be constructed at decomposition levels $j > 1$?
- Let us try to answer these questions!!

Outline

Graph Construction

Graph-Based Representation of a Signal

- In our case...
  - Samples $\longrightarrow$ Nodes
  - Links $\longrightarrow$ Capture correlation between nodes
- Several graph representations. **Challenge: how to link nodes to accurately capture correlation?**

Graph Weighting

- Weights the links between nodes
- **Affects different processes of the transform!!**

Graph Construction

Graph-Based Representation of a Signal

- In our case...
  - Samples $\longrightarrow$ Nodes
  - Links $\longrightarrow$ Capture correlation between nodes
- Several graph representations. **Challenge: how to link nodes to accurately capture correlation?**

Graph Weighting

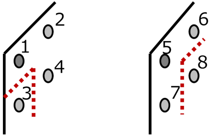
- Weights the links between nodes
- **Affects different processes of the transform!!**

Directional Interpretation

- Graph representation defines filtering directions
- Weighting gives more or less importance to the defined directions

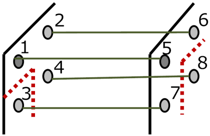
Example: Graph Construction From a Video Signal

- **Example** of the description of a video sequence as a weighted graph of connected pixels
  - Link pixels:



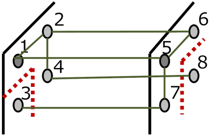
Example: Graph Construction From a Video Signal

- **Example** of the description of a video sequence as a weighted graph of connected pixels
  - Link pixels: Temporal domain



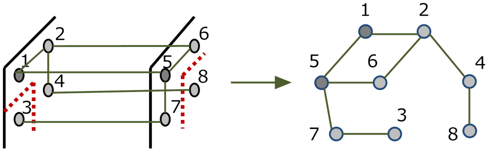
Example: Graph Construction From a Video Signal

- **Example** of the description of a video sequence as a weighted graph of connected pixels
  - Link pixels: Temporal domain, Spatial domain



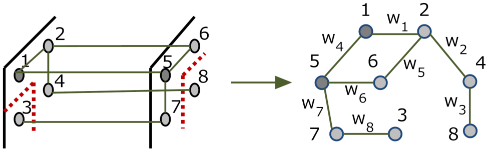
Example: Graph Construction From a Video Signal

- **Example** of the description of a video sequence as a weighted graph of connected pixels
  - Link pixels: Temporal domain, Spatial domain $\rightarrow$ Graph



Example: Graph Construction From a Video Signal

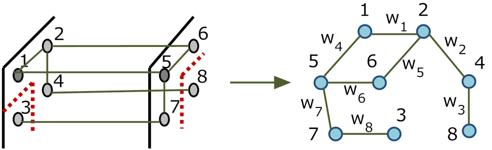
- **Example** of the description of a video sequence as a weighted graph of connected pixels
  - Link pixels: Temporal domain, Spatial domain $\rightarrow$ Graph



  - Weight links: Expected correlation between pixels intensity (reliability)

Example: Graph Construction From a Video Signal

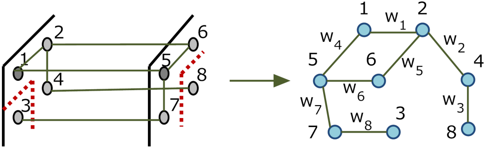
- **Example** of the description of a video sequence as a weighted graph of connected pixels
  - Link pixels: Temporal domain, Spatial domain → Graph



  - Weight links: Expected correlation between pixels intensity (reliability)
- Apply lifting transform on this graph

Example: Graph Construction From a Video Signal

- **Example** of the description of a video sequence as a weighted graph of connected pixels
  - Link pixels: Temporal domain, Spatial domain $\rightarrow$ Graph



  - Weight links: Expected correlation between pixels intensity (reliability)
- Apply lifting transform on this graph

How to find the weights?

Graph Weighting

- Correlations between nodes depends on:
  - Nature of links between them (e.g., spatial or temporal in video).
  - Specific graph representation used

Graph Weighting

- Correlations between nodes depends on:
  - Nature of links between them (e.g., spatial or temporal in video).
  - Specific graph representation used

- Two different approaches to weight the graph:

Graph Weighting

- Correlations between nodes depends on:
  - Nature of links between them (e.g., spatial or temporal in video).
  - Specific graph representation used

- Two different approaches to weight the graph:
  - *Fixed weights*: As a function of the a-priori knowledge of the signal and its graph representation

Graph Weighting

- Correlations between nodes depends on:
  - Nature of links between them (e.g., spatial or temporal in video).
  - Specific graph representation used

- Two different approaches to weight the graph:
  - *Fixed weights*: As a function of the a-priori knowledge of the signal and its graph representation

    but... local correlation depends on the underlying signal (e.g., changes with the video content) $\Longrightarrow$ weights would change as well

Graph Weighting

- Correlations between nodes depends on:
  - Nature of links between them (e.g., spatial or temporal in video).
  - Specific graph representation used

- Two different approaches to weight the graph:
  - *Fixed weights*: As a function of the a-priori knowledge of the signal and its graph representation

    but... local correlation depends on the underlying signal (e.g., changes with the video content) $\Longrightarrow$ weights would change as well
  - *Optimal weights*: Optimizing the weights as a function of the signal at hand
    - Optimal weights can be computed for any subgraph

Graph Weighting

- Correlations between nodes depends on:
  - Nature of links between them (e.g., spatial or temporal in video).
  - Specific graph representation used

- Two different approaches to weight the graph:
  - *Fixed weights*: As a function of the a-priori knowledge of the signal and its graph representation

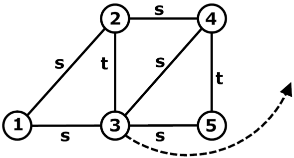
    but... local correlation depends on the underlying signal (e.g., changes with the video content) $\Longrightarrow$ weights would change as well
  - *Optimal weights*: Optimizing the weights as a function of the signal at hand
    - Optimal weights can be computed for any subgraph
- How to optimize the weights?

Optimal Weighting for a Video Representation

Given a graph representation of a video signal...

Optimal Weighting for a Video Representation

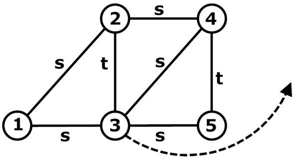
Given a graph representation of a video signal...



$$\bar{x}_3^s = \frac{1}{3}\left(x_1 + x_4 + x_5\right)$$
$$\bar{x}_3^t = x_2$$
$$\hat{x}_3 = w_s \bar{x}_3^s + w_t \bar{x}_3^t$$

Optimal Weighting for a Video Representation

Given a graph representation of a video signal...



$$\bar{x}_3^s = \frac{1}{3}(x_1 + x_4 + x_5)$$
$$\bar{x}_3^t = x_2$$
$$\hat{x}_3 = w_s \bar{x}_3^s + w_t \bar{x}_3^t$$

Definition (Weighted predictor)

- Mean value of the **spatial** neighbors of node *i*:

$$\bar{x}_i^s = \frac{1}{|\mathcal{N}_i|^S} \sum_{j \in \mathcal{N}_i^S} x_j. \tag{1}$$

  - $\mathcal{N}_i^S$ is the set of spatial neighbors of *i*

- Linear predictor of *i*:

$$\hat{x}_i = w_s \bar{x}_i^s + w_t \bar{x}_i^t. \tag{2}$$

Optimal Weighting for a Video Representation

Problem (Optimal weighting problem)

- *Given a graph representation $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: Find the optimal graph weights (w_s and w_t) that minimize the quadratic prediction error (**assuming one-hop prediction filters defined in previous slide**) over all the nodes $i \in \mathcal{V}$:*

$$\min_{w_s, w_t} \sum_{i \in \mathcal{V}} \left(x_i - w_s \bar{x}_i^s - w_t \bar{x}_i^t \right)^2 .$$

Optimal Weighting for a Video Representation

Problem (Optimal weighting problem)

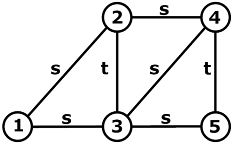
- *Given a graph representation $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: Find the optimal graph weights (w_s and w_t) that minimize the quadratic prediction error (**assuming one-hop prediction filters defined in previous slide**) over all the nodes $i \in \mathcal{V}$:*

$$\min_{w_s, w_t} \sum_{i \in \mathcal{V}} \left(x_i - w_s \bar{x}_i^s - w_t \bar{x}_i^t \right)^2 .$$

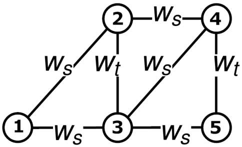
Optimal weights solution

$$\mathbf{w}^* = \begin{bmatrix} w_s^* \\ w_t^* \end{bmatrix} = \begin{bmatrix} \sum_{i \in \mathcal{V}} \bar{x}_i^s \bar{x}_i^s & \sum_{i \in \mathcal{V}} \bar{x}_i^s \bar{x}_i^t \\ \sum_{i \in \mathcal{V}} \bar{x}_i^t \bar{x}_i^s & \sum_{i \in \mathcal{V}} \bar{x}_i^t \bar{x}_i^t \end{bmatrix}^{-1} \cdot \sum_{i \in \mathcal{V}} x_i \begin{bmatrix} \bar{x}_i^s \\ \bar{x}_i^t \end{bmatrix} \tag{3}$$

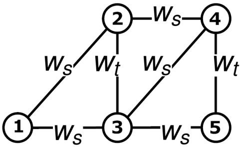
Optimal Weighting for a Video Representation



Optimal Weighting for a Video Representation

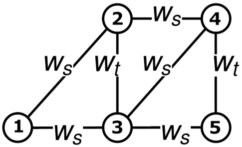


Optimal Weighting for a Video Representation



- At this point... weighted graph, useful for different signal processing operations

Optimal Weighting for a Video Representation



- At this point... weighted graph, useful for different signal processing operations
- We focus on lifting $\implies$ next step: $\mathcal{U}/\mathcal{P}$ assignment

$\mathcal{U}/\mathcal{P}$ Assignment

- $\mathcal{U}/\mathcal{P}$ Assignment on graphs:
  - Great freedom to select the $\mathcal{U}/\mathcal{P}$ assignment , but...

$\mathcal{U}/\mathcal{P}$ Assignment

- $\mathcal{U}/\mathcal{P}$ Assignment on graphs:
  - Great freedom to select the $\mathcal{U}/\mathcal{P}$ assignment , but...
  - different $\mathcal{U}/\mathcal{P}$ assignment $\implies$ different invertible transforms, with different performance!!

$\mathcal{U}/\mathcal{P}$ Assignment

- $\mathcal{U}/\mathcal{P}$ Assignment on graphs:
  - Great freedom to select the $\mathcal{U}/\mathcal{P}$ assignment , but...
  - different $\mathcal{U}/\mathcal{P}$ assignment $\implies$ different invertible transforms, with different performance!!
- Our goal is energy compaction. Two different approaches...

$\mathcal{U}/\mathcal{P}$ Assignment

- $\mathcal{U}/\mathcal{P}$ Assignment on graphs:
  - Great freedom to select the $\mathcal{U}/\mathcal{P}$ assignment , but...
  - different $\mathcal{U}/\mathcal{P}$ assignment $\implies$ different invertible transforms, with different performance!!
- Our goal is energy compaction. Two different approaches...
  - Graph-based:
    - Given a **weighted graph**, find bipartitions without assumptions about the signal

$\mathcal{U}/\mathcal{P}$ Assignment

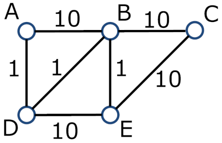
- $\mathcal{U}/\mathcal{P}$ Assignment on graphs:
  - Great freedom to select the $\mathcal{U}/\mathcal{P}$ assignment , but...
  - different $\mathcal{U}/\mathcal{P}$ assignment $\implies$ different invertible transforms, with different performance!!
- Our goal is energy compaction. Two different approaches...
  - Graph-based:
    - Given a **weighted graph**, find bipartitions without assumptions about the signal
  - Signal model-based:
    - Assume a signal model and a predictor
    - Minimizes the expected value of the quadratic prediction error

Proposed Graph-based $\mathcal{U}/\mathcal{P}$ Assignment

- Assume a given weighted graph, where weights *w* of the links represent a similarity measure between nodes

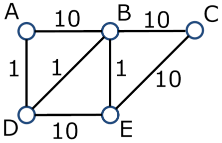
Proposed Graph-based $\mathcal{U}/\mathcal{P}$ Assignment

- Assume a given weighted graph, where weights *w* of the links represent a similarity measure between nodes



Proposed Graph-based $\mathcal{U}/\mathcal{P}$ Assignment

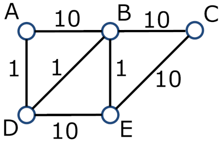
- Assume a given weighted graph, where weights *w* of the links represent a similarity measure between nodes



- Intuitively, a $\mathcal{P}$ node is better predicted if it has a higher number of similar (with high *w* value) $\mathcal{U}$ neighbors...

Proposed Graph-based $\mathcal{U}/\mathcal{P}$ Assignment

- Assume a given weighted graph, where weights *w* of the links represent a similarity measure between nodes



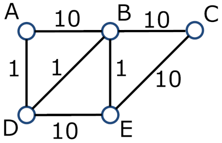
- Intuitively, a $\mathcal{P}$ node is better predicted if it has a higher number of similar (with high *w* value) $\mathcal{U}$ neighbors...

Proposed Criteria

Maximize the reliability with which update nodes can predict prediction neighbors

Proposed Graph-based $\mathcal{U}/\mathcal{P}$ Assignment

- Assume a given weighted graph, where weights *w* of the links represent a similarity measure between nodes



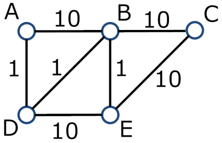
- Intuitively, a $\mathcal{P}$ node is better predicted if it has a higher number of similar (with high *w* value) $\mathcal{U}$ neighbors...

Proposed Criteria

Maximize the reliability with which update nodes can predict prediction neighbors $\Longrightarrow$ Maximize the sum of the weights (W) of the links between $\mathcal{P}$ and $\mathcal{U}$ sets

Proposed Graph-based $\mathcal{U}/\mathcal{P}$ Assignment

- Assume a given weighted graph, where weights *w* of the links represent a similarity measure between nodes



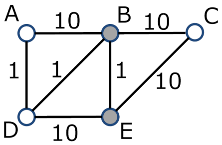
- Intuitively, a $\mathcal{P}$ node is better predicted if it has a higher number of similar (with high *w* value) $\mathcal{U}$ neighbors...

Proposed Criteria

Maximize the reliability with which update nodes can predict prediction neighbors $\Longrightarrow$ Maximize the sum of the weights (W) of the links between $\mathcal{P}$ and $\mathcal{U}$ sets $\Longrightarrow$ **Weighted Maximum Cut** (WMC) problem

Proposed Graph-based $\mathcal{U}/\mathcal{P}$ Assignment

- Assume a given weighted graph, where weights *w* of the links represent a similarity measure between nodes



- Intuitively, a $\mathcal{P}$ node is better predicted if it has a higher number of similar (with high *w* value) $\mathcal{U}$ neighbors...

Proposed Criteria

Maximize the reliability with which update nodes can predict prediction neighbors $\Longrightarrow$ Maximize the sum of the weights (W) of the links between $\mathcal{P}$ and $\mathcal{U}$ sets $\Longrightarrow$ **Weighted Maximum Cut** (WMC) problem

Signal Model-Based $\mathcal{U}/\mathcal{P}$ Assignment. Problem Formulation.

Notation

- x_i: data value associated with node i in the graph
- $\hat{x}_i$: predictor for each node $i \in \mathcal{P}$

Signal Model-Based $\mathcal{U}/\mathcal{P}$ Assignment. Problem Formulation.

Notation

- x_i: data value associated with node i in the graph
- $\hat{x}_i$: predictor for each node $i \in \mathcal{P}$

Problem ($\mathcal{U}/\mathcal{P}$ Assignment problem formulation)

Given a model for x_i and a predictor $\hat{x}_i$, find the $\mathcal{U}/\mathcal{P}$ assignment that minimizes E_{tot} ***for a given number of $\mathcal{P}$ nodes,*** *$|\mathcal{P}|$:*

$$\min_{\mathcal{U}/\mathcal{P}} E_{tot} = \min_{\mathcal{U}/\mathcal{P}} \sum_{i \in \mathcal{P}} \mathbb{E}\{(x_i - \hat{x}_i)^2\}. \tag{4}$$

Signal Model-Based $\mathcal{U}/\mathcal{P}$ Assignment. Problem Formulation.

Notation

- x_i: data value associated with node i in the graph
- $\hat{x}_i$: predictor for each node $i \in \mathcal{P}$

Problem ($\mathcal{U}/\mathcal{P}$ Assignment problem formulation)

Given a model for x_i and a predictor $\hat{x}_i$, find the $\mathcal{U}/\mathcal{P}$ assignment that minimizes E_{tot} ***for a given number of $\mathcal{P}$ nodes, $|\mathcal{P}|$:***

$$\min_{\mathcal{U}/\mathcal{P}} E_{tot} = \min_{\mathcal{U}/\mathcal{P}} \sum_{i \in \mathcal{P}} \mathbb{E}\{(x_i - \hat{x}_i)^2\}. \tag{4}$$

- Minimize (4) practical if some constraint in $|\mathcal{P}|$

Signal Model-Based $\mathcal{U}/\mathcal{P}$ Assignment. Problem Formulation.

Notation

- x_i: data value associated with node i in the graph
- $\hat{x}_i$: predictor for each node $i \in \mathcal{P}$

Problem ($\mathcal{U}/\mathcal{P}$ Assignment problem formulation)

Given a model for x_i and a predictor $\hat{x}_i$, find the $\mathcal{U}/\mathcal{P}$ assignment that minimizes E_{tot} ***for a given number of $\mathcal{P}$ nodes, $|\mathcal{P}|$:***

$$\min_{\mathcal{U}/\mathcal{P}} E_{tot} = \min_{\mathcal{U}/\mathcal{P}} \sum_{i \in \mathcal{P}} \mathbb{E}\{(x_i - \hat{x}_i)^2\}. \quad (4)$$

- Minimize (4) practical if some constraint in $|\mathcal{P}|$
- Goal:
  - Find expression of E_{tot}

Signal Model-Based $\mathcal{U}/\mathcal{P}$ Assignment. Problem Formulation.

Notation

- x_i: data value associated with node i in the graph
- $\hat{x}_i$: predictor for each node $i \in \mathcal{P}$

Problem ($\mathcal{U}/\mathcal{P}$ Assignment problem formulation)

Given a model for x_i and a predictor $\hat{x}_i$, find the $\mathcal{U}/\mathcal{P}$ assignment that minimizes E_{tot} ***for a given number of $\mathcal{P}$ nodes,*** *$|\mathcal{P}|$:*

$$\min_{\mathcal{U}/\mathcal{P}} E_{tot} = \min_{\mathcal{U}/\mathcal{P}} \sum_{i \in \mathcal{P}} \mathbb{E}\{(x_i - \hat{x}_i)^2\}. \quad (4)$$

- Minimize (4) practical if some constraint in $|\mathcal{P}|$
- Goal:
  - Find expression of $E_{\text{tot}} \Longrightarrow$ Minimize it with a greedy algorithm

Signal Model-Based $\mathcal{U}/\mathcal{P}$ Assignment. Problem Formulation.

Notation

- x_i: data value associated with node i in the graph
- $\hat{x}_i$: predictor for each node $i \in \mathcal{P}$

Problem ($\mathcal{U}/\mathcal{P}$ Assignment problem formulation)

Given a model for x_i and a predictor $\hat{x}_i$, find the $\mathcal{U}/\mathcal{P}$ assignment that minimizes E_{tot} ***for a given number of $\mathcal{P}$ nodes,*** *$|\mathcal{P}|$:*

$$\min_{\mathcal{U}/\mathcal{P}} E_{tot} = \min_{\mathcal{U}/\mathcal{P}} \sum_{i \in \mathcal{P}} \mathbb{E}\{(x_i - \hat{x}_i)^2\}. \tag{4}$$

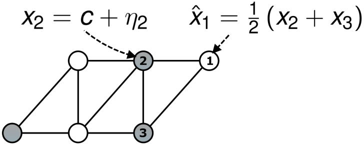
- Minimize (4) practical if some constraint in $|\mathcal{P}|$
- Goal:
  - Find expression of $E_{\text{tot}} \Longrightarrow$ Minimize it with a greedy algorithm
- We discuss the simplest data generation model...

Definition of the Noisy Model (NM)

Given an **unweighted graph**...

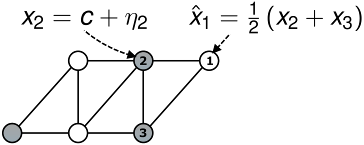
Definition of the Noisy Model (NM)

Given an **unweighted graph**...



Definition of the Noisy Model (NM)

Given an **unweighted graph**...



Definition (NM)

Let x_i be a noisy version of some constant c,

$$x_i = c + \eta_i, \tag{5}$$

η_i: independent noise variables with zero mean and variance v

Definition (Unweighted Predictor)

For each node $i \in \mathcal{P}$, consider the predictor

$$\hat{x}_i = \frac{1}{m_i} \sum_{j \in \mathcal{N}_i \cap \mathcal{U}} x_j, \tag{6}$$

- $\mathcal{N}_i$: set of neighbors of node i
- m_i: number of $\mathcal{U}$ neighbors of i ($|\mathcal{N}_i \cap \mathcal{U}|$)

Noisy Model (NM)

Proposition (*Noisy Model Prediction Error*)

Let x_i and $\hat{x}_i$ satisfy previous definitions. The total prediction error over all nodes $i \in \mathcal{P}$ is given by

$$E_{totNM} = \sum_{i \in \mathcal{P}} \mathbb{E}\{(x_i - \hat{x}_i)^2\} = v\left(|\mathcal{P}| + \sum_{i \in \mathcal{P}} \frac{1}{m_i}\right). \tag{7}$$

Noisy Model (NM)

Proposition (*Noisy Model Prediction Error*)

Let x_i and $\hat{x}_i$ satisfy previous definitions. The total prediction error over all nodes $i \in \mathcal{P}$ is given by

$$E_{totNM} = \sum_{i \in \mathcal{P}} \mathbb{E}\{(x_i - \hat{x}_i)^2\} = v\left(|\mathcal{P}| + \sum_{i \in \mathcal{P}} \frac{1}{m_i}\right). \tag{7}$$

- E_{totNM} increases with the variance of the nodes v

Noisy Model (NM)

Proposition (*Noisy Model Prediction Error*)

Let x_i and $\hat{x}_i$ satisfy previous definitions. The total prediction error over all nodes $i \in \mathcal{P}$ is given by

$$E_{totNM} = \sum_{i \in \mathcal{P}} \mathbb{E}\{(x_i - \hat{x}_i)^2\} = v \left(|\mathcal{P}| + \sum_{i \in \mathcal{P}} \frac{1}{m_i} \right). \quad (7)$$

- E_{totNM} increases with the variance of the nodes v
- E_{totNM} decreases with the number of $\mathcal{U}$ neighbors of $\mathcal{P}$ nodes

Noisy Model (NM)

Proposition (*Noisy Model Prediction Error*)

Let x_i and $\hat{x}_i$ satisfy previous definitions. The total prediction error over all nodes $i \in \mathcal{P}$ is given by

$$E_{totNM} = \sum_{i \in \mathcal{P}} \mathbb{E}\{(x_i - \hat{x}_i)^2\} = v \left(|\mathcal{P}| + \sum_{i \in \mathcal{P}} \frac{1}{m_i} \right). \tag{7}$$

- E_{totNM} increases with the variance of the nodes v
- E_{totNM} decreases with the number of $\mathcal{U}$ neighbors of $\mathcal{P}$ nodes
- **Question**: Given a $|\mathcal{P}|$ and a number of links between $\mathcal{P}$ and $\mathcal{U}$ sets (Cut), $W = \sum_{i \in \mathcal{P}} m_i$... which is the optimal $\mathcal{U}/\mathcal{P}$ assignment?

Noisy Model (NM)

Proposition (*Noisy Model Prediction Error*)

Let x_i and $\hat{x}_i$ satisfy previous definitions. The total prediction error over all nodes $i \in \mathcal{P}$ is given by

$$E_{totNM} = \sum_{i \in \mathcal{P}} \mathbb{E}\{(x_i - \hat{x}_i)^2\} = v \left(|\mathcal{P}| + \sum_{i \in \mathcal{P}} \frac{1}{m_i} \right). \quad (7)$$

- E_{totNM} increases with the variance of the nodes v
- E_{totNM} decreases with the number of $\mathcal{U}$ neighbors of $\mathcal{P}$ nodes
- **Question**: Given a $|\mathcal{P}|$ and a number of links between $\mathcal{P}$ and $\mathcal{U}$ sets (Cut), $W = \sum_{i \in \mathcal{P}} m_i$... which is the optimal $\mathcal{U}/\mathcal{P}$ assignment?

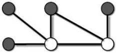
Corollary (*Optimal Criteria for Fixed $|\mathcal{P}|$ and W*)

For a fixed $|\mathcal{P}|$ and W, the optimal criteria to minimize E_{totNM} over these $\mathcal{P}$ nodes is that **all of them have the same number of $\mathcal{U}$ neighbor nodes**

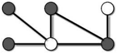
Let see an example...

Noisy Model (NM)

- NM Vs. MC:



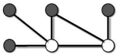
W=5 $$E_{\mathrm{tot}} = v\left(|\mathcal{P}| + \sum_{i\in\mathcal{P}}^{|\mathcal{P}|} \frac{1}{m_i}\right) = v\left(2 + \frac{1}{3} + \frac{1}{2}\right) = v\left(2 + \frac{5}{6}\right).$$



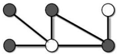
W=5 $$E_{\mathrm{tot}} = v\left(|\mathcal{P}| + \sum_{i\in\mathcal{P}}^{|\mathcal{P}|} \frac{1}{m_i}\right) = v\left(2 + \frac{1}{4} + 1\right) = v\left(2 + \frac{5}{4}\right).$$

Noisy Model (NM)

- NM Vs. MC:



W=5 $E_{\text{tot}} = v\left(|\mathcal{P}| + \sum_{i\in\mathcal{P}}^{|\mathcal{P}|} \frac{1}{m_i}\right) = v\left(2+\frac{1}{3}+\frac{1}{2}\right) = v\left(2+\frac{5}{6}\right).$

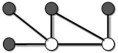


W=5 $E_{\text{tot}} = v\left(|\mathcal{P}| + \sum_{i\in\mathcal{P}}^{|\mathcal{P}|} \frac{1}{m_i}\right) = v\left(2+\frac{1}{4}+1\right) = v\left(2+\frac{5}{4}\right).$

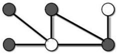
- Example with real data
  - Sequence *Mobile*
  - We design a greedy algorithm $\rightarrow$ each iteration minimizes E_{totNM}
  - Define $E_{\text{av-meas}} = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}(x_i - \hat{x}_i)^2 = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}(d_i)^2$

Noisy Model (NM)

- NM Vs. MC:

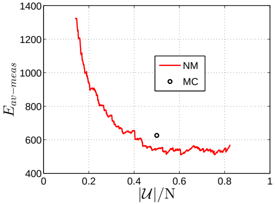


W=5 $\quad E_{\text{tot}} = v\left(|\mathcal{P}| + \sum_{i\in\mathcal{P}}^{|\mathcal{P}|} \frac{1}{m_i}\right) = v\left(2 + \frac{1}{3} + \frac{1}{2}\right) = v\left(2 + \frac{5}{6}\right).$



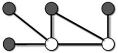
W=5 $\quad E_{\text{tot}} = v\left(|\mathcal{P}| + \sum_{i\in\mathcal{P}}^{|\mathcal{P}|} \frac{1}{m_i}\right) = v\left(2 + \frac{1}{4} + 1\right) = v\left(2 + \frac{5}{4}\right).$

- Example with real data
  - Sequence *Mobile*
  - We design a greedy algorithm $\rightarrow$ each iteration minimizes E_{totNM}
  - Define $E_{\text{av-meas}} = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}(x_i - \hat{x}_i)^2 = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}(d_i)^2$

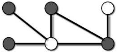


Noisy Model (NM)

- NM Vs. MC:

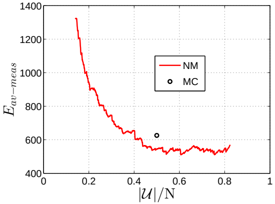


W=5 $$E_{\text{tot}} = v\left(|\mathcal{P}| + \sum_{i\in\mathcal{P}}^{|\mathcal{P}|} \frac{1}{m_i}\right) = v\left(2+\frac{1}{3}+\frac{1}{2}\right) = v\left(2+\frac{5}{6}\right).$$



W=5 $$E_{\text{tot}} = v\left(|\mathcal{P}| + \sum_{i\in\mathcal{P}}^{|\mathcal{P}|} \frac{1}{m_i}\right) = v\left(2+\frac{1}{4}+1\right) = v\left(2+\frac{5}{4}\right).$$

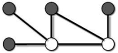
- Example with real data
  - Sequence *Mobile*
  - We design a greedy algorithm $\rightarrow$ each iteration minimizes E_{totNM}
  - Define $E_{\text{av-meas}} = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}(x_i - \hat{x}_i)^2 = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}(d_i)^2$



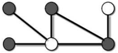
- $E_{\text{av-meas}}$ decreases as $|\mathcal{U}|/N$ increases

Noisy Model (NM)

- NM Vs. MC:

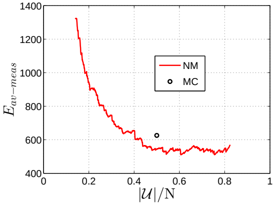


W=5 $\quad E_{\text{tot}} = v\left(|\mathcal{P}| + \sum_{i\in\mathcal{P}}^{|\mathcal{P}|} \frac{1}{m_i}\right) = v\left(2 + \frac{1}{3} + \frac{1}{2}\right) = v\left(2 + \frac{5}{6}\right).$



W=5 $\quad E_{\text{tot}} = v\left(|\mathcal{P}| + \sum_{i\in\mathcal{P}}^{|\mathcal{P}|} \frac{1}{m_i}\right) = v\left(2 + \frac{1}{4} + 1\right) = v\left(2 + \frac{5}{4}\right).$

- Example with real data
  - Sequence *Mobile*
  - We design a greedy algorithm $\rightarrow$ each iteration minimizes E_{totNM}
  - Define $E_{\text{av-meas}} = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}(x_i - \hat{x}_i)^2 = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}}(d_i)^2$



- $E_{\text{av-meas}}$ decreases as $|\mathcal{U}|/N$ increases
- $E_{\text{av-meas}}$ NM lower than $E_{\text{av-meas}}$ MC... more balanced assignment

Remarks about the Spatio-Temporal Model (STM)

- STM model...

[1] Under some reasonable simplifications

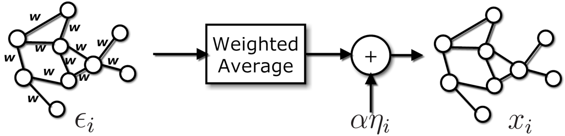
Remarks about the Spatio-Temporal Model (STM)

- STM model...
  - **Spatial, temporal neighbors $\implies$ differentiated correlations**

[1] Under some reasonable simplifications

Remarks about the Spatio-Temporal Model (STM)

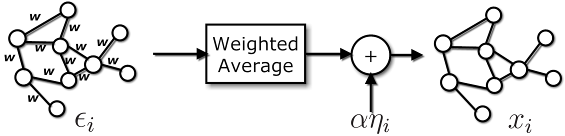
- STM model...
  - **Spatial, temporal neighbors $\Longrightarrow$ differentiated correlations**
  - Given a **weighted graph** of random noise, x_i is generated as:



[1] Under some reasonable simplifications

Remarks about the Spatio-Temporal Model (STM)

- STM model...
  - **Spatial, temporal neighbors $\Longrightarrow$ differentiated correlations**
  - Given a **weighted graph** of random noise, x_i is generated as:

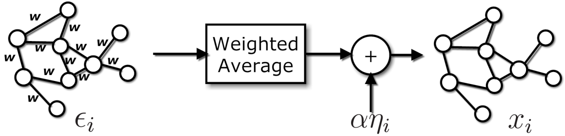


- Predictor: Weighted predictor

[1] Under some reasonable simplifications

Remarks about the Spatio-Temporal Model (STM)

- STM model...
  - **Spatial, temporal neighbors $\Longrightarrow$ differentiated correlations**
  - Given a **weighted graph** of random noise, x_i is generated as:

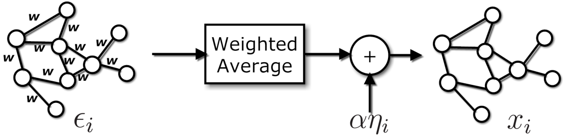


- Predictor: Weighted predictor
- Some intuitions about the prediction error in the STM...

[1] Under some reasonable simplifications

Remarks about the Spatio-Temporal Model (STM)

- STM model...
  - **Spatial, temporal neighbors $\Longrightarrow$ differentiated correlations**
  - Given a **weighted graph** of random noise, x_i is generated as:

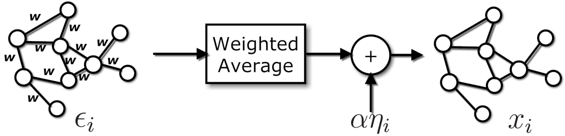


- Predictor: Weighted predictor
- Some intuitions about the prediction error in the STM...
  - The factors in the E_{totST} are weighted by w_s and w_t

[1] Under some reasonable simplifications

Remarks about the Spatio-Temporal Model (STM)

- STM model...
  - **Spatial, temporal neighbors $\Longrightarrow$ differentiated correlations**
  - Given a **weighted graph** of random noise, x_i is generated as:

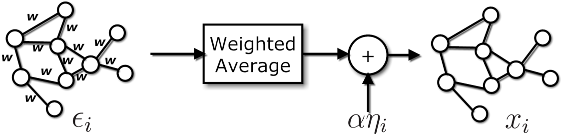


- Predictor: Weighted predictor
- Some intuitions about the prediction error in the STM...
  - The factors in the E_{totST} are weighted by w_s and w_t
  - In order to minimize E_{totST}[1] for a given number of links between $\mathcal{U}$ and $\mathcal{P}$ sets
    - **Every node should have the same proportion of temporal and spatial update neighbors**

[1] Under some reasonable simplifications

Remarks about the Spatio-Temporal Model (STM)

- STM model...
  - **Spatial, temporal neighbors $\implies$ differentiated correlations**
  - Given a **weighted graph** of random noise, x_i is generated as:

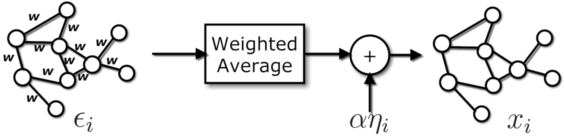


- Predictor: Weighted predictor
- Some intuitions about the prediction error in the STM...
  - The factors in the E_{totST} are weighted by w_s and w_t
  - In order to minimize E_{totST}[1] for a given number of links between $\mathcal{U}$ and $\mathcal{P}$ sets
    - **Every node should have the same proportion of temporal and spatial update neighbors**
    - The right proportion depends on w_t and w_s (the higher w_t, the higher proportion of temporal update neighbors)

[1] Under some reasonable simplifications

Remarks about the Spatio-Temporal Model (STM)

- STM model...
  - **Spatial, temporal neighbors $\Longrightarrow$ differentiated correlations**
  - Given a **weighted graph** of random noise, x_i is generated as:



- Predictor: Weighted predictor
- Some intuitions about the prediction error in the STM...
  - The factors in the E_{totST} are weighted by w_s and w_t
  - In order to minimize E_{totST}[1] for a given number of links between $\mathcal{U}$ and $\mathcal{P}$ sets
    - **Every node should have the same proportion of temporal and spatial update neighbors**
    - The right proportion depends on w_t and w_s (the higher w_t, the higher proportion of temporal update neighbors)
  - STM outperforms WMC in terms of $E_{\text{av-meas}}$

[1] Under some reasonable simplifications

Prediction Filter design

Goal Design prediction filter **in the context of lifting on graphs**

Prediction Filter design

Goal Design prediction filter **in the context of lifting on graphs**

Observations
- Not possible to calculate weights for each relative location
- The proper choice of **p** depends on how data is correlated across nodes

Prediction Filter design

Goal Design prediction filter **in the context of lifting on graphs**

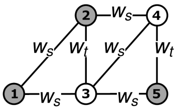
Observations

- Not possible to calculate weights for each relative location
- The proper choice of **p** depends on how data is correlated across nodes

Proposed Solution Filter design based on the weights of the graph!

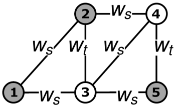
Prediction Filter design

Example Given a weighted graph and an $\mathcal{U}/\mathcal{P}$...



Prediction Filter design

Example Given a weighted graph and an $\mathcal{U}/\mathcal{P}$...

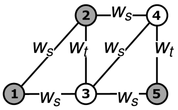


- Prediction filter for node $3 \in \mathcal{P}$:

$$\hat{x}_3 = \frac{w_s}{2}(x_1 + x_5) + w_t x_2 \implies \mathbf{p}_3 = \left[\frac{w_s}{2}, w_t\ , \frac{w_s}{2}\right] \tag{8}$$

Prediction Filter design

Example Given a weighted graph and an $\mathcal{U}/\mathcal{P}$...



- Prediction filter for node $3 \in \mathcal{P}$:

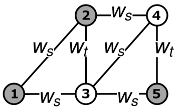
$$\hat{x}_3 = \frac{w_s}{2}(x_1 + x_5) + w_t x_2 \implies \mathbf{p}_3 = \left[\frac{w_s}{2}, w_t, \frac{w_s}{2}\right] \tag{8}$$

Observations

- Similar predictors to the ones used for weighting the graph!! $\implies$ **Near-optimal solution in the sense of minimizing detail coefficient energy**

Prediction Filter design

Example Given a weighted graph and an $\mathcal{U}/\mathcal{P}$...



- Prediction filter for node $3 \in \mathcal{P}$:

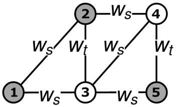
$$\hat{x}_3 = \frac{w_s}{2}(x_1 + x_5) + w_t x_2 \implies \mathbf{p}_3 = \left[\frac{w_s}{2}, w_t, \frac{w_s}{2}\right] \quad (8)$$

Observations

- Similar predictors to the ones used for weighting the graph!! $\implies$ **Near-optimal solution in the sense of minimizing detail coefficient energy**
  - Main difference $\rightarrow$ now we have the $\mathcal{U}/\mathcal{P}$ assignment...
  - Optimal weights could be recalculated, but lead to similar values

Prediction Filter design

Example Given a weighted graph and an $\mathcal{U}/\mathcal{P}$...



- Prediction filter for node $3 \in \mathcal{P}$:

$$\hat{x}_3 = \frac{w_s}{2}(x_1 + x_5) + w_t x_2 \implies \mathbf{p}_3 = \left[\frac{w_s}{2}, w_t, \frac{w_s}{2}\right] \tag{8}$$

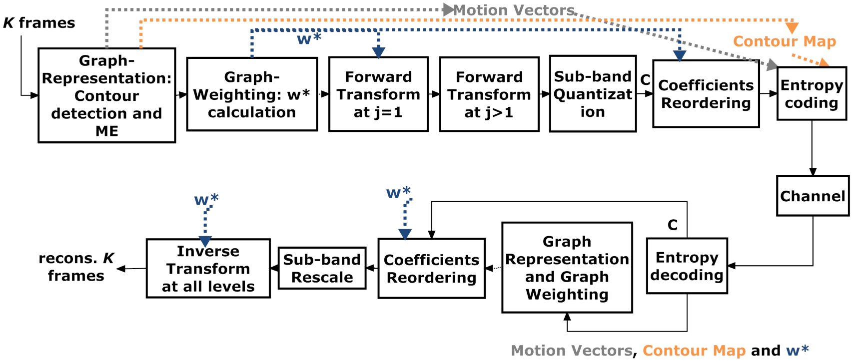
Observations

- Similar predictors to the ones used for weighting the graph!! $\implies$ **Near-optimal solution in the sense of minimizing detail coefficient energy**
  - Main difference $\rightarrow$ now we have the $\mathcal{U}/\mathcal{P}$ assignment...
  - Optimal weights could be recalculated, but lead to similar values
- Predictors used in the STM!!

Outline

Encoder and Decoder Data flow

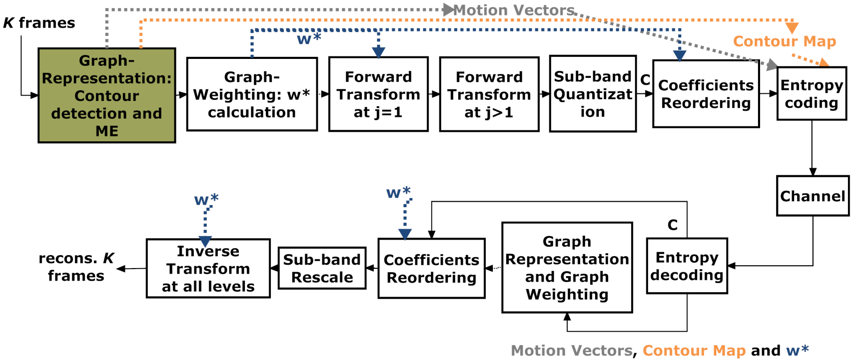
- Apply the transform to video coding leading to a 3-dimensional directional transform
- Proposed coding scheme:



Side information: Motion Vectors, Edge map, $\mathbf{w}^*$.

Encoder and Decoder Data flow

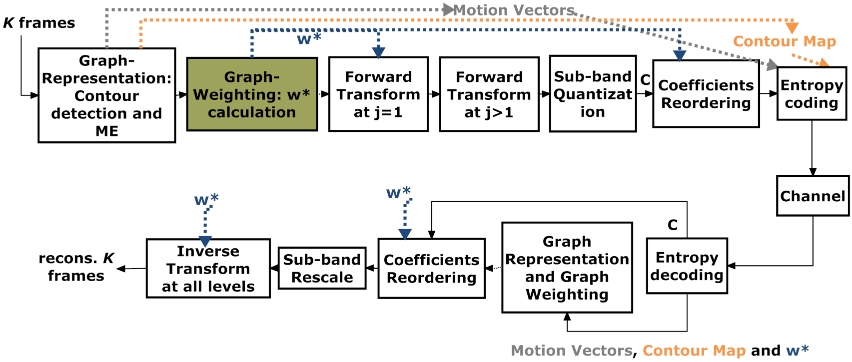
- Apply the transform to video coding leading to a 3-dimensional directional transform
- Proposed coding scheme:



Side information: Motion Vectors, Edge map, $\mathbf{w}^*$.

Encoder and Decoder Data flow

- Apply the transform to video coding leading to a 3-dimensional directional transform
- Proposed coding scheme:



Side information: Motion Vectors, Edge map, $\mathbf{w}^*$.

Fixed Weighting Vs. Optimal Weighting

- Fixed weighting: $w_t = 10$, $w_s = 2$
- Optimal weighting: Calculated in a **frame-by-frame** basis

Fixed Weighting Vs. Optimal Weighting

- Fixed weighting: $w_t = 10$, $w_s = 2$
- Optimal weighting: Calculated in a **frame-by-frame** basis
- Results
  - Comparison in terms of $E_{\text{av-meas},j=1}$:

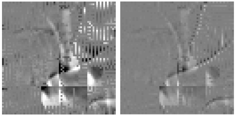
	Carphone	***Mobile***	***Airshow* (scene cut)**	***Football* (fast motion)**
$E_{\textbf{av-meas},j=1}$ **Fixed weights**	14	44	34	408
$E_{\textbf{av-meas},j=1}$ $\mathbf{w}^*$	12	37	17	240

Fixed Weighting Vs. Optimal Weighting

- Fixed weighting: $w_t = 10$, $w_s = 2$
- Optimal weighting: Calculated in a **frame-by-frame** basis
- Results
  - Comparison in terms of $E_{\text{av-meas},j=1}$:

	Carphone	***Mobile***	***Airshow* (scene cut)**	***Football* (fast motion)**
$E_{\textbf{av-meas},j=1}$ **Fixed weights**	14	44	34	408
$E_{\textbf{av-meas},j=1}$ $\mathbf{w}^*$	12	37	17	240

  - Detail coefficient values:



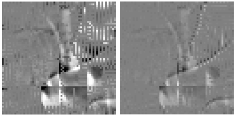
Airshow

Fixed Weighting Vs. Optimal Weighting

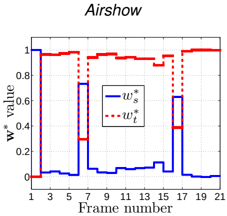
- Fixed weighting: $w_t = 10$, $w_s = 2$
- Optimal weighting: Calculated in a **frame-by-frame** basis
- Results
  - Comparison in terms of $E_{\text{av-meas},j=1}$:

	Carphone	***Mobile***	***Airshow* (scene cut)**	***Football* (fast motion)**
$E_{\textbf{av-meas},j=1}$ **Fixed weights**	14	44	34	408
$E_{\textbf{av-meas},j=1}$ $\mathbf{w}^*$	12	37	17	240

  - Detail coefficient values:

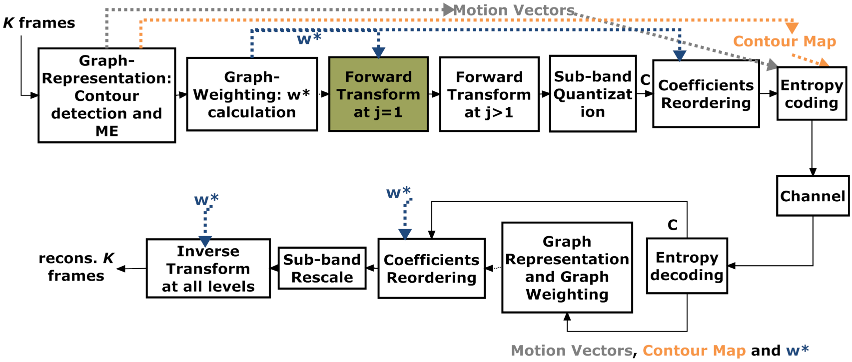


  - $\mathbf{w}^*$ evolution:



Encoder and Decoder Data flow

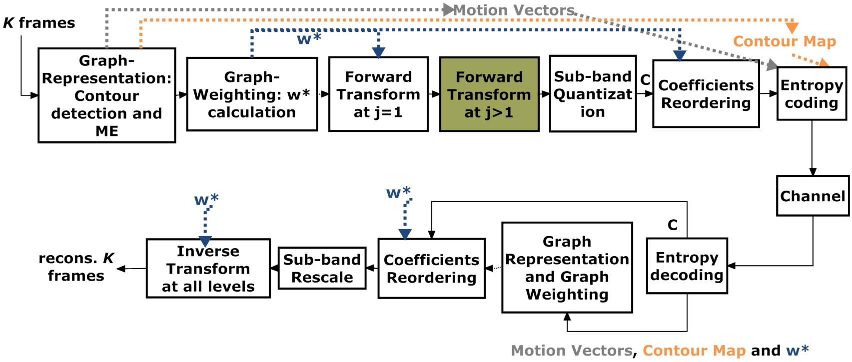
- Apply the transform to video coding leading to a 3-dimensional directional transform
- Proposed coding scheme:



Side information: Motion Vectors, Edge map, $\mathbf{w}^*$.

Encoder and Decoder Data flow

- Apply the transform to video coding leading to a 3-dimensional directional transform
- Proposed coding scheme:



Side information: Motion Vectors, Edge map, $\mathbf{w}^*$.

Extending the Transform to Multiple Levels of Decomposition

Goal Carry out a Multiresolution analysis

Extending the Transform to Multiple Levels of Decomposition

Goal Carry out a Multiresolution analysis $\Longrightarrow$ extend the transform to multiple levels of decomposition

Extending the Transform to Multiple Levels of Decomposition

Goal Carry out a Multiresolution analysis $\implies$ extend the transform to multiple levels of decomposition

Proposed Solution

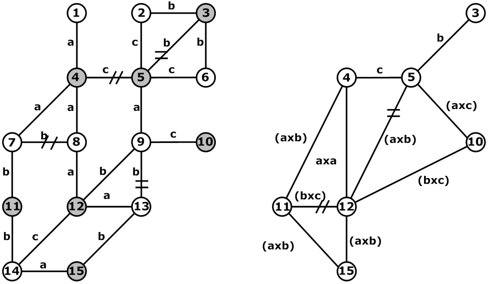
- **Graph construction** at level j from level $j-1$:
  - Connect $\mathcal{U}$ nodes that are directly connected or at two-hop

Extending the Transform to Multiple Levels of Decomposition

Goal Carry out a Multiresolution analysis $\Longrightarrow$ extend the transform to multiple levels of decomposition

Proposed Solution

- **Graph construction** at level j from level $j-1$:
  - Connect $\mathcal{U}$ nodes that are directly connected or at two-hop



(j-1) level

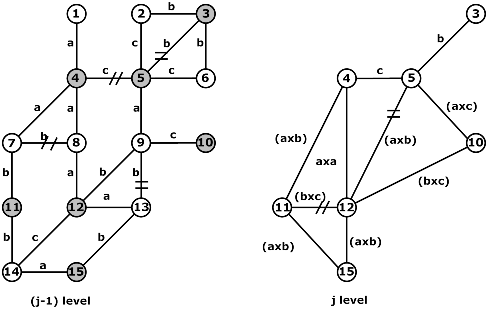
j level

Extending the Transform to Multiple Levels of Decomposition

Goal Carry out a Multiresolution analysis $\Longrightarrow$ extend the transform to multiple levels of decomposition

Proposed Solution

- **Graph construction** at level j from level $j-1$:
  - Connect $\mathcal{U}$ nodes that are directly connected or at two-hop
- **Graph weighting** at level j from level $j-1$:
  - Product of the weights in the path between connected nodes at level $j-1$

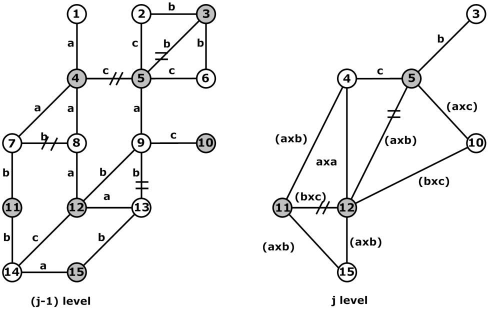


Extending the Transform to Multiple Levels of Decomposition

Goal Carry out a Multiresolution analysis $\Longrightarrow$ extend the transform to multiple levels of decomposition

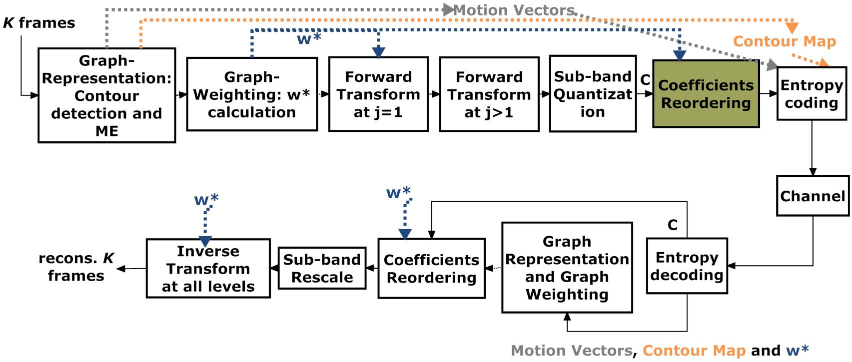
Proposed Solution

- **Graph construction** at level j from level $j-1$:
  - Connect $\mathcal{U}$ nodes that are directly connected or at two-hop
- **Graph weighting** at level j from level $j-1$:
  - Product of the weights in the path between connected nodes at level $j-1$



Encoder and Decoder Data flow

- Apply the transform to video coding leading to a 3-dimensional directional transform
- Proposed coding scheme:



Side information: Motion Vectors, Edge map, $\mathbf{w}^*$.

Coefficient Reordering

Goal Reorder the coefficients generated by our graph-based transform in an efficient way

- e.g., Zig-zag scanning order in DCT based encoders.
- e.g., Hierarchical trees in wavelet-based encoders [Shapiro, 1992, Kim and Pearlman, 1997]

Coefficient Reordering

Goal Reorder the coefficients generated by our graph-based transform in an efficient way

- e.g., Zig-zag scanning order in DCT based encoders.
- e.g., Hierarchical trees in wavelet-based encoders [Shapiro, 1992, Kim and Pearlman, 1997]

Proposed Solution Reorder the coefficients using graph information:

1. Inter-subband reordering
2. Intra-subband reordering

Inter-Subband Reordering

Observation Energy in the middle-high frequency subbands will be low

Inter-Subband Reordering

Observation Energy in the middle-high frequency subbands will be low

Proposed reordering Group coefficients that belong to the same subband $\rightarrow$ Long strings of zero coefficients after quantization

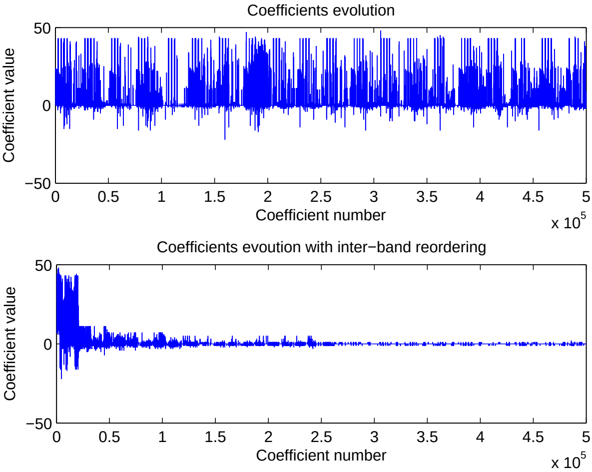
The coefficients will be sorted as:

$$\mathbf{coeffs}_{inter} = \left[\mathbf{s}^{j=N}, \mathbf{d}^{j=N}, \mathbf{d}^{j=N-1}, \ldots, \mathbf{d}^{j=1}\right].$$

$$- \xrightarrow{\textbf{Frequency}} +$$

- $\mathbf{s}^{j=N}$: smooth coefficients at level of decomposition $j = N$ (the lowest frequency subband)
- $\mathbf{d}^{j}$: detail coefficients at generic level of decomposition j

Inter-Subband Reordering Example



20 frames of the sequence *Carphone*

Top: original coefficients. Bottom: reordered coefficients

Intra-Subband Reordering

Observations

- The graph is known at the encoder and the decoder
- Weights provide an estimation of the reliability with which one predict node is predicted from update neighbors

Intra-Subband Reordering

Observations

- The graph is known at the encoder and the decoder
- Weights provide an estimation of the reliability with which one predict node is predicted from update neighbors

Assumption Detail coefficients will be smaller if they have been predicted from more “reliable” neighbors

Intra-Subband Reordering

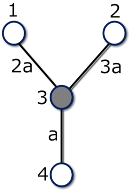
Observations

- The graph is known at the encoder and the decoder
- Weights provide an estimation of the reliability with which one predict node is predicted from update neighbors

Assumption Detail coefficients will be smaller if they have been predicted from more “reliable” neighbors

Proposed reordering Reorder the coefficients **in each subband** as a function of the reliability of their links

Intra-Subband Reordering Example



- Detail coefficients (the white nodes) of a generic subband j, $\mathbf{d}^j = [1, 2, 4]$
- Reliability values: $\mathbf{r}^j = [2a, 3a, a]$

Intra-subband reordered coefficients $\mathbf{d}^j_{intra} = [4, 1, 2]$.

$- \xrightarrow{\textbf{Reliability}} +$

- Coefficients energy is lower as the reliability increases

Reordering Results

Test Conditions

- 20 Frames, 2 QCIF sequences
- Comparison: without reordering the coefficients, employing inter-reordering, and inter and intra reordering
- Rate is obtained with an arithmetic coder

Results

	Without reordering	**Inter reordering**	**Inter and Intra reordering**
***Foreman* (32.9 dB)**	503 Kbps	404 Kbps	350 Kbps
***Carphone* (36 dB)**	502 Kbps	425 Kbps	371 Kbps

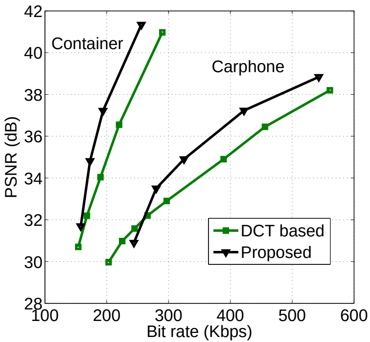
Experimental Results. Test Conditions

- **Comparison against a motion-compensated DCT video encoder**

	DCT-based	**Proposed**
Coefficients quantization	Uniform dead-zone	Subband dependent
Coefficients scanning	Zigzag scanning order	Inter and Intra reordering
Symbols	Run-length encoding	Run-length encoding
Entropy coding	Arithmetic coder	Arithmetic coder
Block size ME	16×16	16×16

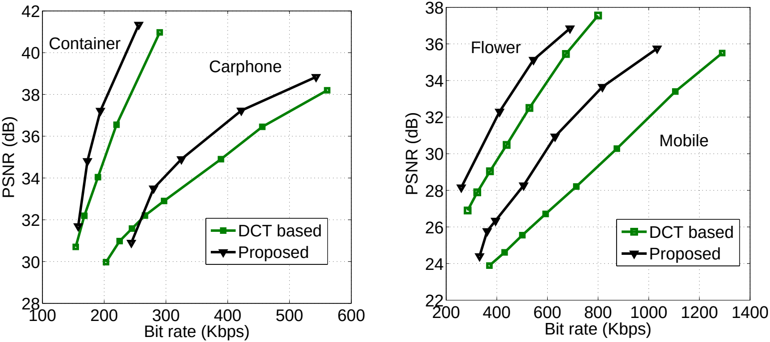
- **Other parameters**:
  - Proposed:
    - Five levels of decomposition
  - DCT-based:
    - 8×8 DCT
- **Side information:**
  - Proposed:
    - MVs differentially encoded with respect to a predicted MV
    - Contour maps $\Longrightarrow$ encoded with JBIG
    - Optimal weights, once every frame. $9 bits/frame$
  - DCT-based:
    - MVs differentially encoded with respect to a predicted MV

Experimental Results



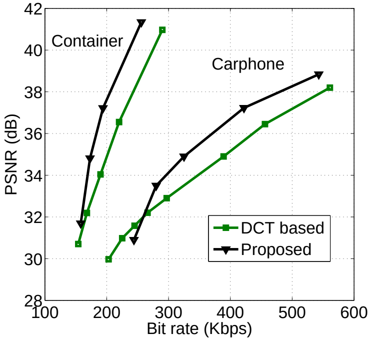
- Proposed method outperforms the DCT based approach in medium to high qualities

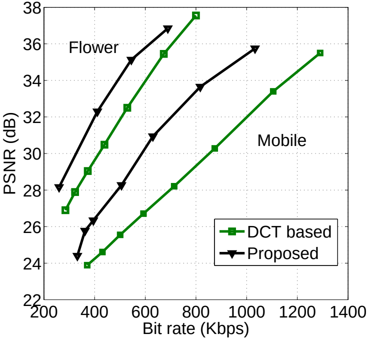
Experimental Results



- Proposed method outperforms the DCT based approach in medium to high qualities
- 4 dB better in *Mobile*, around 1-2 dB better in *Container*, *Carphone* and *Flower*

Experimental Results





- Proposed method outperforms the DCT based approach in medium to high qualities
- 4 dB better in *Mobile*, around 1-2 dB better in *Container*, *Carphone* and *Flower*
- Contour maps side information of about 10*Kbps*; weights side information insignificant

Other Contributions in Two Minutes

- **Low Complexity Approach**
  - **Problem:** The $\mathcal{U}/\mathcal{P}$ complexity increases rapidly with the number of nodes N.

Other Contributions in Two Minutes

- **Low Complexity Approach**
  - **Problem:** The $\mathcal{U}/\mathcal{P}$ complexity increases rapidly with the number of nodes N.
  - **Proposed solution:** Operate in a distributed manner
    - Obtain the $\mathcal{U}/\mathcal{P}$ assignment in blocks of size B
    - Transfer this information to neighboring blocks

Other Contributions in Two Minutes

- **Low Complexity Approach**
  - **Problem:** The $\mathcal{U}/\mathcal{P}$ complexity increases rapidly with the number of nodes N.
  - **Proposed solution:** Operate in a distributed manner
    - Obtain the $\mathcal{U}/\mathcal{P}$ assignment in blocks of size B
    - Transfer this information to neighboring blocks
  - **Results:**
    - Complexity reduction of around 200 times
    - Negligible loss of performance

Other Contributions in Two Minutes

- **Low Complexity Approach**
  - **Problem:** The $\mathcal{U}/\mathcal{P}$ complexity increases rapidly with the number of nodes N.
  - **Proposed solution:** Operate in a distributed manner
    - Obtain the $\mathcal{U}/\mathcal{P}$ assignment in blocks of size B
    - Transfer this information to neighboring blocks
  - **Results:**
    - Complexity reduction of around 200 times
    - Negligible loss of performance
- **Rate Distortion Optimization**
  - **Problem:** $\mathcal{U}/\mathcal{P}$ strategies **for a given** $|\mathcal{P}|$ **nodes**, but... How to choose $|\mathcal{P}|$?

Other Contributions in Two Minutes

- **Low Complexity Approach**
  - **Problem:** The $\mathcal{U}/\mathcal{P}$ complexity increases rapidly with the number of nodes N.
  - **Proposed solution:** Operate in a distributed manner
    - Obtain the $\mathcal{U}/\mathcal{P}$ assignment in blocks of size B
    - Transfer this information to neighboring blocks
  - **Results:**
    - Complexity reduction of around 200 times
    - Negligible loss of performance
- **Rate Distortion Optimization**
  - **Problem:** $\mathcal{U}/\mathcal{P}$ strategies **for a given** $|\mathcal{P}|$ **nodes**, but... How to choose $|\mathcal{P}|$?
  - **Proposed solution:** Rate Distortion Optimization of the graph: chooses the $|\mathcal{P}|$ that minimizes distortion under a given rate restriction

Other Contributions in Two Minutes

- **Low Complexity Approach**
  - **Problem:** The $\mathcal{U}/\mathcal{P}$ complexity increases rapidly with the number of nodes N.
  - **Proposed solution:** Operate in a distributed manner
    - Obtain the $\mathcal{U}/\mathcal{P}$ assignment in blocks of size B
    - Transfer this information to neighboring blocks
  - **Results:**
    - Complexity reduction of around 200 times
    - Negligible loss of performance
- **Rate Distortion Optimization**
  - **Problem:** $\mathcal{U}/\mathcal{P}$ strategies **for a given** $|\mathcal{P}|$ **nodes**, but... How to choose $|\mathcal{P}|$?
  - **Proposed solution:** Rate Distortion Optimization of the graph: chooses the $|\mathcal{P}|$ that minimizes distortion under a given rate restriction
  - **Results:** Promising results as comparing with the WMC!!! (around 1-2 dBs better in some sequences)

Outline

Conclusions

- Framework for the construction of N-dimensional directional transforms

Conclusions

- Framework for the construction of N-dimensional directional transforms
- Optimization of lifting transforms on graphs in order to obtain a compact representation of the graph signal

Conclusions

- Framework for the construction of N-dimensional directional transforms
- Optimization of lifting transforms on graphs in order to obtain a compact representation of the graph signal
- Application to video coding $\Longrightarrow$ Spatial and temporal correlation jointly exploted $\Longrightarrow$ Good results!!

Future Work

- Investigate other models in the $\mathcal{U}/\mathcal{P}$ assignment, as Markov random fields

Future Work

- Investigate other models in the $\mathcal{U}/\mathcal{P}$ assignment, as Markov random fields
- Results of the RDO promising $\implies$ design a practical RDO process

Future Work

- Investigate other models in the $\mathcal{U}/\mathcal{P}$ assignment, as Markov random fields
- Results of the RDO promising $\Longrightarrow$ design a practical RDO process
- Other applications: multichannel-audio coding, image and video denoising, biomedical signals compact representation

Thank you!

References I

Fattal, R. (2009).
Edge-avoiding wavelets and their applications.
In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pages 1–10, New York, NY, USA. ACM.

Kim, B.-J. and Pearlman, W. A. (1997).
An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT).
In *Data Compression Conference, 1997. DCC '97. Proceedings*, pages 251 –260.

Le Pennec, E. and Mallat, S. (2005).
Sparse geometric image representations with bandelets.
Image Processing, IEEE Transactions on, 14(4):423 –438.

Pesquet-Popescu, B. and Bottreau, V. (2001).
Three-dimensional lifting schemes for motion compensated video compression.
In *ICASSP '01: Proceedings of the Acoustics, Speech, and Signal Processing, 2001. on IEEE International Conference*, pages 1793–1796, Washington, DC, USA.

Secker, A. and Taubman, D. (2003).
Lifting-based invertible motion adaptive transform (limat) framework for highly scalable video compression.
Image Processing, IEEE Transactions on, 12(12):1530 – 1542.

Shapiro, J. M. (1992).
An embedded wavelet hierarchical image coder.
In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 4, pages 657 –660 vol.4.

Shen, G. and Ortega, A. (2008).
Compact image representation using wavelet lifting along arbitrary trees.
In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2808 –2811.

References II

Velisavljevic, V., Beferull-Lozano, B., Vetterli, M., and Dragotti, P. L. (2006).
Directionlets: anisotropic multidirectional representation with separable filtering.
Image Processing, IEEE Transactions on, 15(7):1916 –1933.