SCC.NRG.AI4ME

Self-optimising distributed encoding networks

Thomas Swarbrick
PhD Candidate t.swarbrick@lancaster.ac.uk

What is an object in object based media?

- Customisation of Characteristic, but the meta characteristics remain constant.
- Large customisable overlap, but not computationally feasible on the client-side.

What is an object in object based media?

- Customisation of Characteristic, but the meta characteristics remain constant.
- Large customisable overlap, but not computationally feasible on the client-side.

Traditional Approach

- Changeable T-shirt colour
- Geographically Specific Weather Map

What is an object in object based media?

- Customisation of Characteristic, but the meta characteristics remain constant.
- Large customisable overlap, but not computationally feasible on the client-side.

Traditional Approach

- Changeable T-shirt colour
- Geographically Specific Weather Map

But..

What is an object in object based media?

- Customisation of Characteristic, but the meta characteristics remain constant.
- Large customisable overlap, but not computationally feasible on the client-side.

Traditional Approach

- Changeable T-shirt colour
- Geographically Specific Weather Map

But..

- Fixed in scope
- Susceptible to compute cost constraints.

What is an object in object based media?

- Customisation of Characteristic, but the meta characteristics remain constant.
- Large customisable overlap, but not computationally feasible on the client-side.

Traditional Approach

- Changeable T-shirt colour
- Geographically Specific Weather Map

But..

- Fixed in scope
- Susceptible to compute cost constraints.

XO

- Bring the Object Based Media (OBM) principle lower down the tool-chain.
- Most Compression Algorithms, make use of the 2D DCT II/III as part of their compression/analysis (Think MPEG).

Bring the Object Based Media (OBM) principle lower down the tool-chain.
Most Compression Algorithms, make use of the 2D DCT II/III as part of their compression/analysis (Think MPEG).

Theorem

let position be time-series like such that.

$$A = \{S_0^A \cdots S_n^A\}, B = \{S_0^B \cdots S_m^B\}$$

where for a given object S_x assume.

$$S_x \in A, S_x \in B$$

$$A = \{S_x|\Sigma_1\}, B = \{S_x|\Sigma_2\}$$

such that.

$$\Sigma_1 \not\subset B, \Sigma_2 \not\subset A$$

Bring the Object Based Media (OBM) principle lower down the tool-chain.
Most Compression Algorithms, make use of the 2D DCT II/III as part of their compression/analysis (Think MPEG).

Theorem

let position be time-series like such that.

$$A = \{S_0^A \cdots S_n^A\}, B = \{S_0^B \cdots S_m^B\}$$

where for a given object S_x assume.

$$S_x \in A, S_x \in B$$

$$A = \{S_x | \Sigma_1\}, B = \{S_x | \Sigma_2\}$$

such that.

$$\Sigma_1 \not\subset B, \Sigma_2 \not\subset A$$

But how do you identify S_X? No idea.

Theorem

Assume S_x is identified, and we remove signal S_x from the sets.

$$A \neq \sigma(B) + \Sigma_1, B \neq \sigma(A) + \Sigma_2$$

i.e. A and B are no longer correlated signals.

Theorem

Assume S_x is identified, and we remove signal S_x from the sets.

$$A \neq \sigma(B) + \Sigma_1, B \neq \sigma(A) + \Sigma_2$$

i.e. A and B are no longer correlated signals. So take some other set C.

$$C = \{S_0^C \cdots S_p^C\}$$

assume S_y is such that,

Theorem

Assume S_x is identified, and we remove signal S_x from the sets.

$$A \neq \sigma(B) + \Sigma_1, B \neq \sigma(A) + \Sigma_2$$

i.e. A and B are no longer correlated signals. So take some other set C.

$$C = \{S_0^C \cdots S_p^C\}$$

assume S_y is such that,

$$S_y \in B, S_y \in C, S_y \notin A, S_y \notin S_x$$

Theorem

Assume S_x is identified, and we remove signal S_x from the sets.

$$A \neq \sigma(B) + \Sigma_1, B \neq \sigma(A) + \Sigma_2$$

i.e. A and B are no longer correlated signals. So take some other set C.

$$C = \{S_0^C \cdots S_p^C\}$$

assume S_y is such that,

$$S_y \in B, S_y \in C, S_y \notin A, S_y \notin S_x$$

$$B = \Sigma_2 = \{S_y | \Sigma_3\}, C = \{S_y | \Sigma_4\}$$

Theorem

Assume S_x is identified, and we remove signal S_x from the sets.

$$A \neq \sigma(B) + \Sigma_1, B \neq \sigma(A) + \Sigma_2$$

i.e. A and B are no longer correlated signals. So take some other set C.

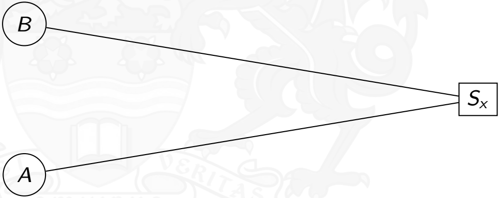
$$C = \{S_0^C \cdots S_p^C\}$$

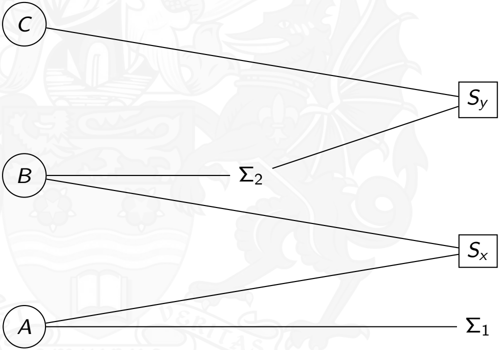
assume S_y is such that,

$$S_y \in B, S_y \in C, S_y \notin A, S_y \notin S_x$$

$$B = \Sigma_2 = \{S_y|\Sigma_3\}, C = \{S_y|\Sigma_4\}$$

Figuring out if S_y is not in A or S_x? No Idea.

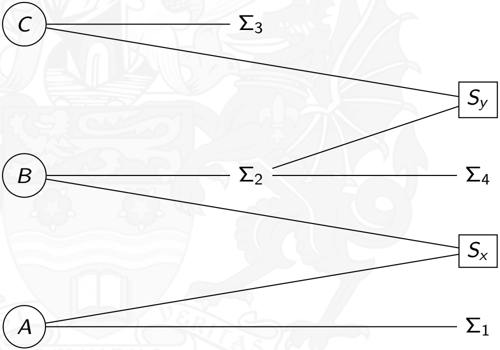

B
A
S_x

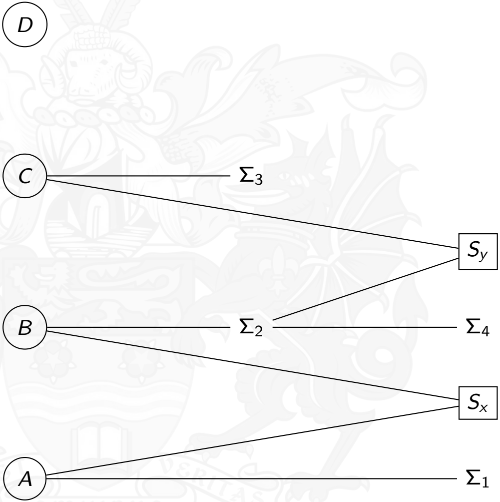


C
B
A
S_y
S_x
Σ_2
Σ_1

High Level Overview

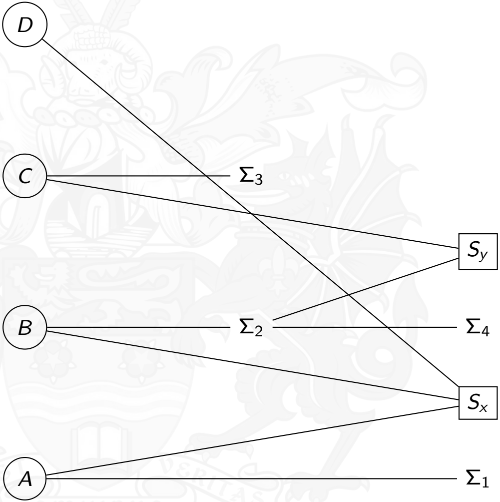
Self-Optimising Objects

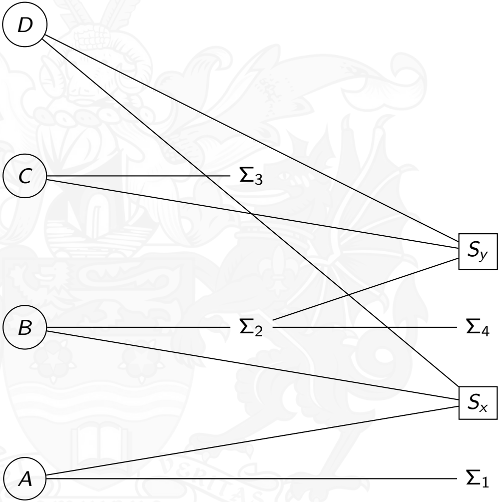


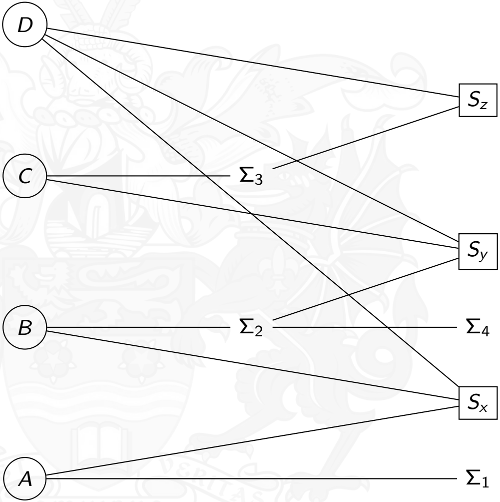


D
C
Σ_3
S_y
B
Σ_2
Σ_4
S_x
A
Σ_1

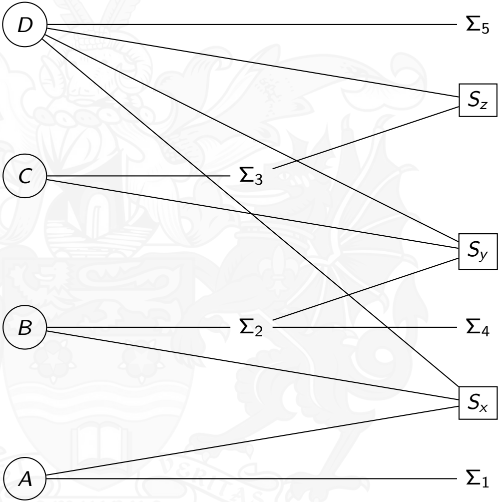
High Level Overview
Self-Optimising Objects



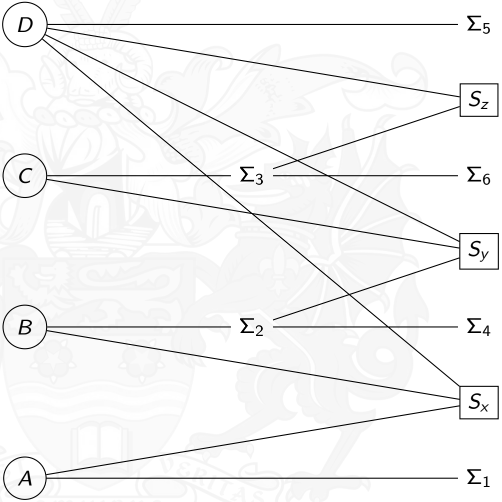

D
C
Σ_3
S_y
B
Σ_2
Σ_4
S_x
A
Σ_1



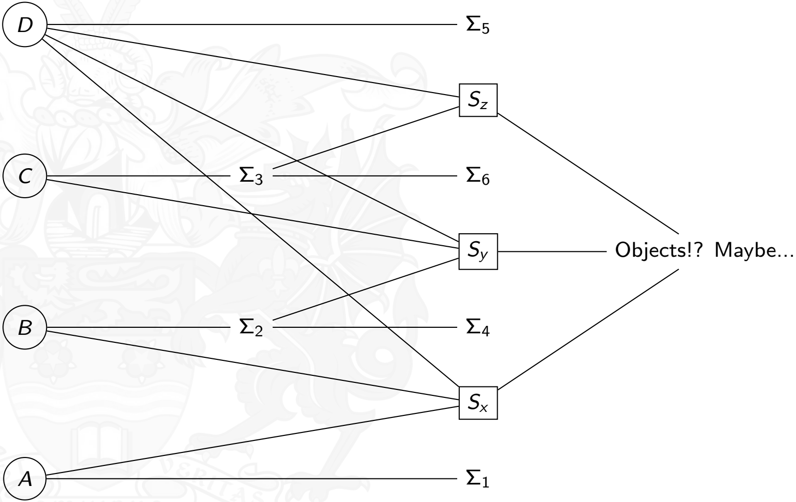
D
S_z
C
Σ_3
S_y
B
Σ_2
Σ_4
S_x
A
Σ_1

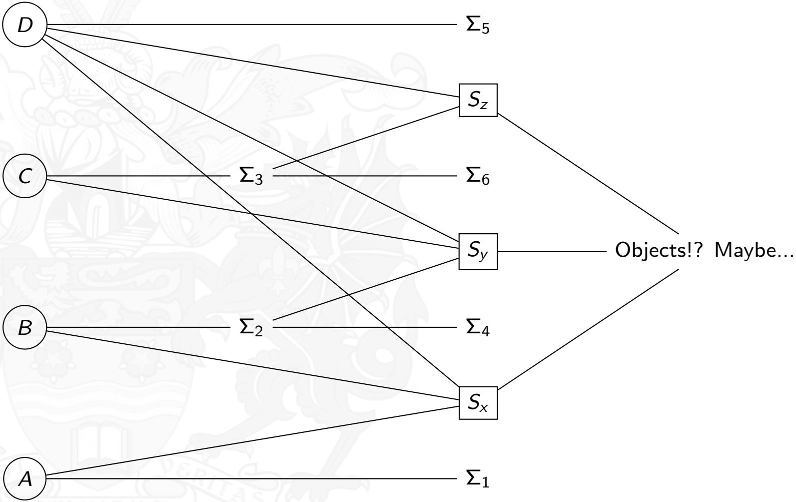


D
Σ_5
S_z
C
Σ_3
S_y
B
Σ_2
Σ_4
S_x
A
Σ_1



D
Σ_5
S_z
C
Σ_3
Σ_6
S_y
B
Σ_2
Σ_4
S_x
A
Σ_1


D
C
B
A
Σ_5
S_z
Σ_3
Σ_6
S_y
Σ_2
Σ_4
S_x
Σ_1
Objects!? Maybe...



How do you optimise the ordering for encoder objects? No idea.

Toy Example
Alice, Bob, Charlie, Dave

Client	Servicing Nodes
A	Σ_1, S_x
B	Σ_4, S_x, S_y
C	Σ_6, S_y, S_z
D	Σ_5, S_x, S_y, S_z

Client	Servicing Nodes
A	Σ_1, S_x
B	Σ_4, S_x, S_y
C	Σ_6, S_y, S_z
D	Σ_5, S_x, S_y, S_z

Theorem

assume that if the set is an itemised set of objects.

$$\Sigma_n \to \emptyset$$

Discard Σ_n as remainder Over large number of nodes.

$$A = S_x, B = S_x, S_y \cdots$$

Client	Servicing Nodes
A	Σ_1, S_x
B	Σ_4, S_x, S_y
C	Σ_6, S_y, S_z
D	Σ_5, S_x, S_y, S_z

Theorem

assume that if the set is an itemised set of objects.

$$\Sigma_n \to \emptyset$$

Discard Σ_n as remainder Over large number of nodes.

$$A = S_x, B = S_x, S_y \cdots$$

How to Assess if Σ_n is just artefact or truly unique? No idea.

Similar Encoding properties/objects would be dynamically grouped, and cached to be distributed across a network.

Distributing Nodes
How does sub-division help us?

Similar Encoding properties/objects would be dynamically grouped, and cached to be distributed across a network.

Example

Add a new Client E. Existing parts of E that are already cached closer to the client fetch faster / optimisation to identify similar signals from a lower-quality/partially computed version.

Client	Servicing Nodes
A	Σ_1, S_x
B	Σ_4, S_x, S_y
C	Σ_6, S_y, S_z
D	Σ_5, S_x, S_y, S_z
E	Σ_7, S_z, S_y

Similar Encoding properties/objects would be dynamically grouped, and cached to be distributed across a network.

Example

Add a new Client E. Existing parts of E that are already cached closer to the client fetch faster / optimisation to identify similar signals from a lower-quality/partially computed version.

Client	Servicing Nodes
A	Σ_1, S_x
B	Σ_4, S_x, S_y
C	Σ_6, S_y, S_z
D	Σ_5, S_x, S_y, S_z
E	Σ_7, S_z, S_y

But how to identify S_z in E without full render? No idea.

Assuming that both signal A and B are correlated if they contain the same object with a different single change.

- Cross-Correlation?
- Steerable Pyramids?

Current Idea...

Network Distribution

How do you distribute this over a network?

Encoding/Decoding Weights

How do you identify ways to optimise self consumption/propagation