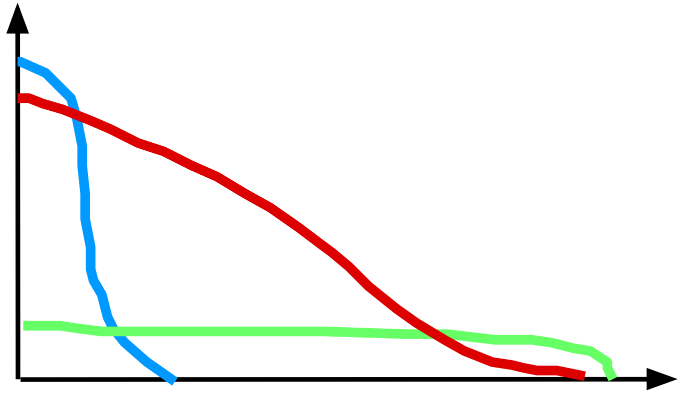
Other Complexity Measurements



Hiroki Sayama
sayama@binghamton.edu

Other complexity measurements

- Algorithmic complexity
- Statistical complexity
- Multiscale complexity

Algorithmic Complexity

Algorithmic complexity

- A.k.a. Kolmogorov complexity, Kolmogorov-Chaitin complexity

- For a given bit string (or any pattern), its algorithmic complexity is the length of the shortest program that produces the bit string deterministically
  - Very intuitive!

Example

27824112462319447142712141073221614
84242121364182912741374122061033101
554662337003501755526 118659
31780890445133666 437
71132566283850 4
38719215810793
44182291127341
32461623081154 3
259764882441226 531
60804020105168

The algorithmic complexity of these digits must *not* be greater than the length of this code! (in Mathematica)

ToString/@NestList[If[OddQ[#],3#+1,#/2]&,27,111]//StringJoin

Meaning of algorithmic complexity

- Similar to pattern recognition and description done by humans
  - You explain how to reconstruct the input, and the length of your explanation will be its algorithmic complexity

- "Logical" data compression
  - Deterministic (not based on probability)
  - Provides the <u>ultimate limit</u> of compression (by definition)

Exercise

- Which of the following strings *do you think* has the smallest or largest algorithmic complexity?

XYXYXYXYXYXYXYXYXYXYXYXYXYX

ABCDEFGHIJKLMNOPQRSTUVWXYZ

JLKFLKLDKFLJLKGFKAGHSLLFHSDHL

QQQQQQQQQQQQQQQQQQQQQ

Properties of algorithmic complexity

- It mildly depends on specific choice of a language, but not too much
  - Difference in algorithmic complexity b/w two languages is bounded by a constant
- Algorithmic complexity of a random sequence is at least the length of the sequence itself
  - Because the program needs to store the sequence itself in it

How to compute it exactly?

- **Simple algorithm:**
  1. Let $n \leftarrow 1$
  2. For all programs of length n:
     - If the output of the program matches the input string, n must be the algorithmic complexity of the string [end]
  3. If no program found, let $n \leftarrow n + 1$
  4. Repeat

- Does it work?

Think about this: Berry's paradox

"The smallest positive integer not definable in fewer than twelve words"

Uncomputability of algorithmic complexity

- We assumed that algorithmic complexity (i.e., minimal number of words needed to define a number) would be computable for all numbers
- But as soon as we assume it, Berry's paradox gives contradiction

  → There is no general way to compute algorithmic complexity

  (Deeply related to the halting problem discussed in next class)

What was wrong?

- Simple algorithm:
  1. Let n ← 1
  2. For all programs of length n:
     - If the output of the program matches the input string, n must be the algorithmic complexity of the string [end]
  3. If no prog
  4. Repeat

There is no general way to predict the output of a program

Having said that…

- For a randomly behaving infinite sequence (e.g., Markov information source), its algorithmic complexity per event converges almost always to its entropy (entropy rate)

$$\bar{K}(X) \sim \bar{H}(X)$$

  - Because the "algorithm" is simply a compressed recording of the sequence

Problems in Information and Algorithmic Complexities

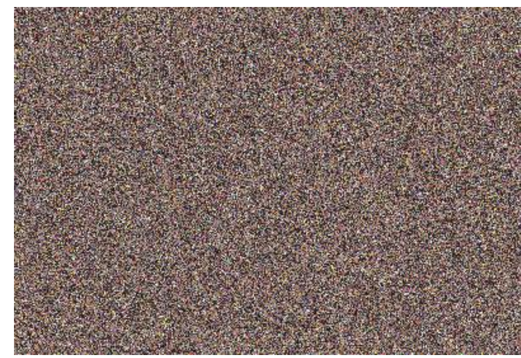
Similarities of information and algorithmic complexities

- They measure how much information would be needed to fully specify the system's state in every single detail
  - Ordered → regular, low information entropy, short algorithm, low uncertainty, more specificity
  - Disordered → random, high information entropy, long algorithm, high uncertainty, less specificity

Example: How complex are they?

Ordered
Regular

Disordered
Random



Both information and algorithmic complexities tell you that <u>this one is most complex!!</u>

Exercise

- Imagine the following three systems:
  1. A cup of quality durum wheat flour (semolina) imported from Italy
  2. Nicely cooked spaghetti made of 1, with some meat sauce and cheese added
  3. "Soup" made by putting 2 into a blender
- Which one is most/least complex?
- Which one do you *personally* think most complex?

Description vs. uncertainty

- Structured→ more organizations, more features to describe, longer narrative, therefore naturally "more complex"
- Disordered → more uncertainty, more information needed, therefore mathematically "more complex"

- How to resolve this paradox?

What's missing?

- Statistical view
- Multiscale view

Statistical Complexity

What's missing? (1)

Statistical view

Many microscopic states of a system are essentially identical to each other from a statistical (macroscopic) viewpoint

Let's aggregate them!

Statistical complexity

- Crutchfield & Young (1989)
- Considers all the microscopic states (past histories) that have the same probability distribution of future states to be identical
- Can avoid trivial "random = complex" conclusion

How it works

- Define "causal states" using the following equivalence relation $x \sim x'$:

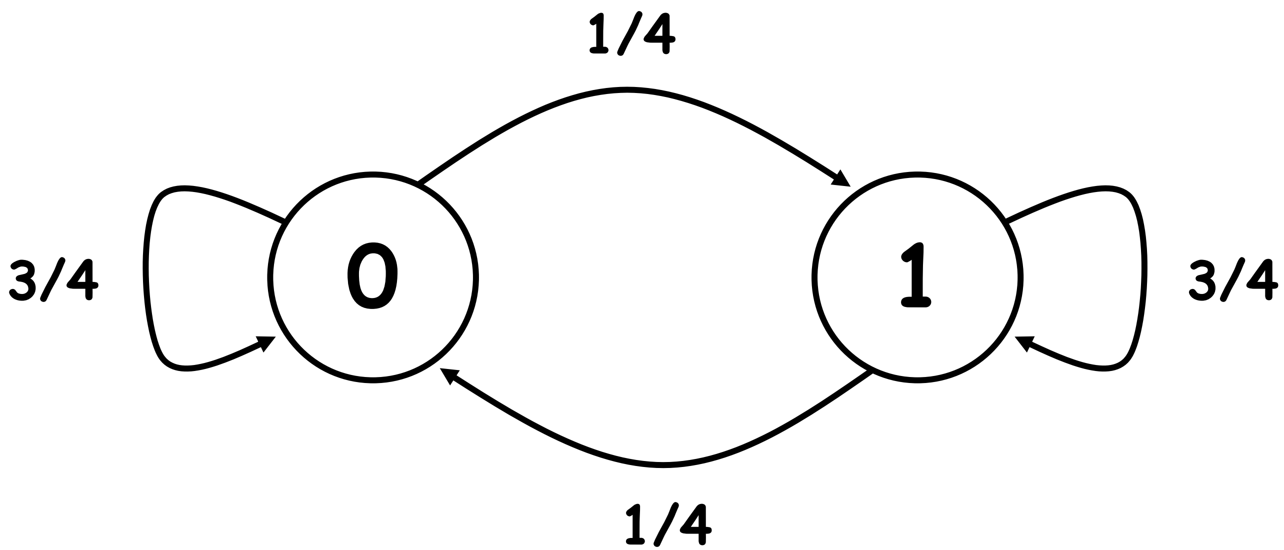
$$P(x_{future}|x_{past}) = P(x_{future}|x'_{past}) \quad \forall\ x_{future}$$

x_{past}, x'_{past} and x_{future} can be finite or infinite histories, separated at a time point t

- Construct a TPM among the detected "causal state" classes
- Calculate entropy of asymptotic causal states obtained from the TPM

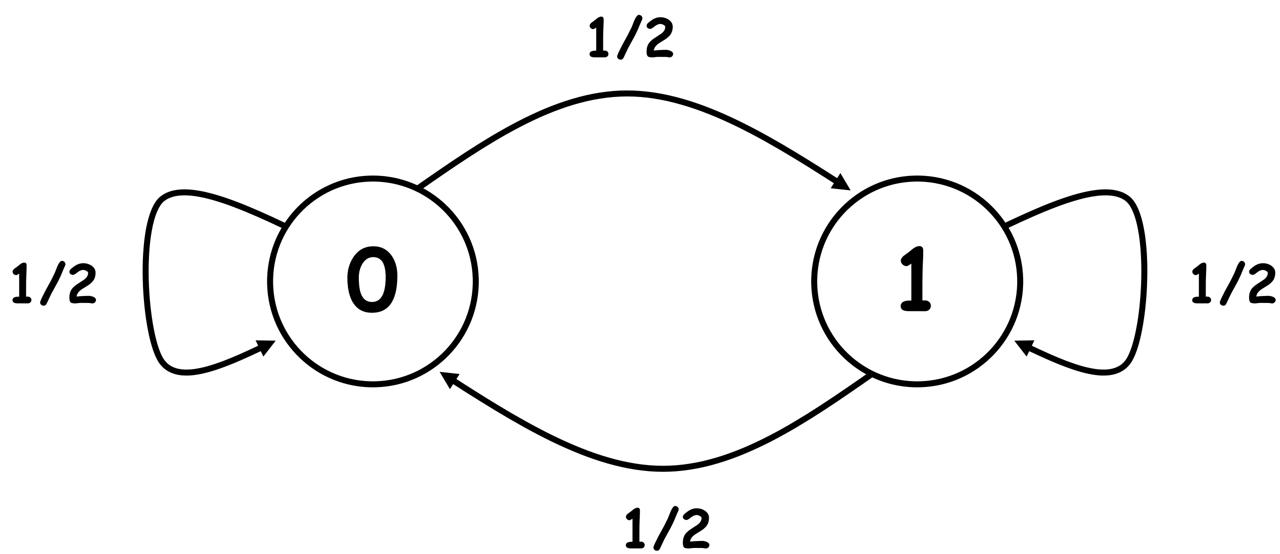
Exercise

- Calculate the entropy rate and the statistical complexity of the following Markov information source:



Exercise

- Calculate the entropy rate and the statistical complexity of a completely random bit string:



Exercise

- Calculate the entropy rate and the statistical complexity of the following Markov information source
- The next state will be:
  - 1 if the previous three states were "000"
  - 0 if the previous three states were "111"
  - Randomly chosen from {0, 1} otherwise

Relationships with information and algorithmic complexities

- Statistical complexity measures…
- Information complexity in asymptotic probability distribution of aggregated causal states
- Algorithmic complexity of a program that can statistically generate an ensemble of patterns that belong to the class to which the input belongs

Multiscale Complexity

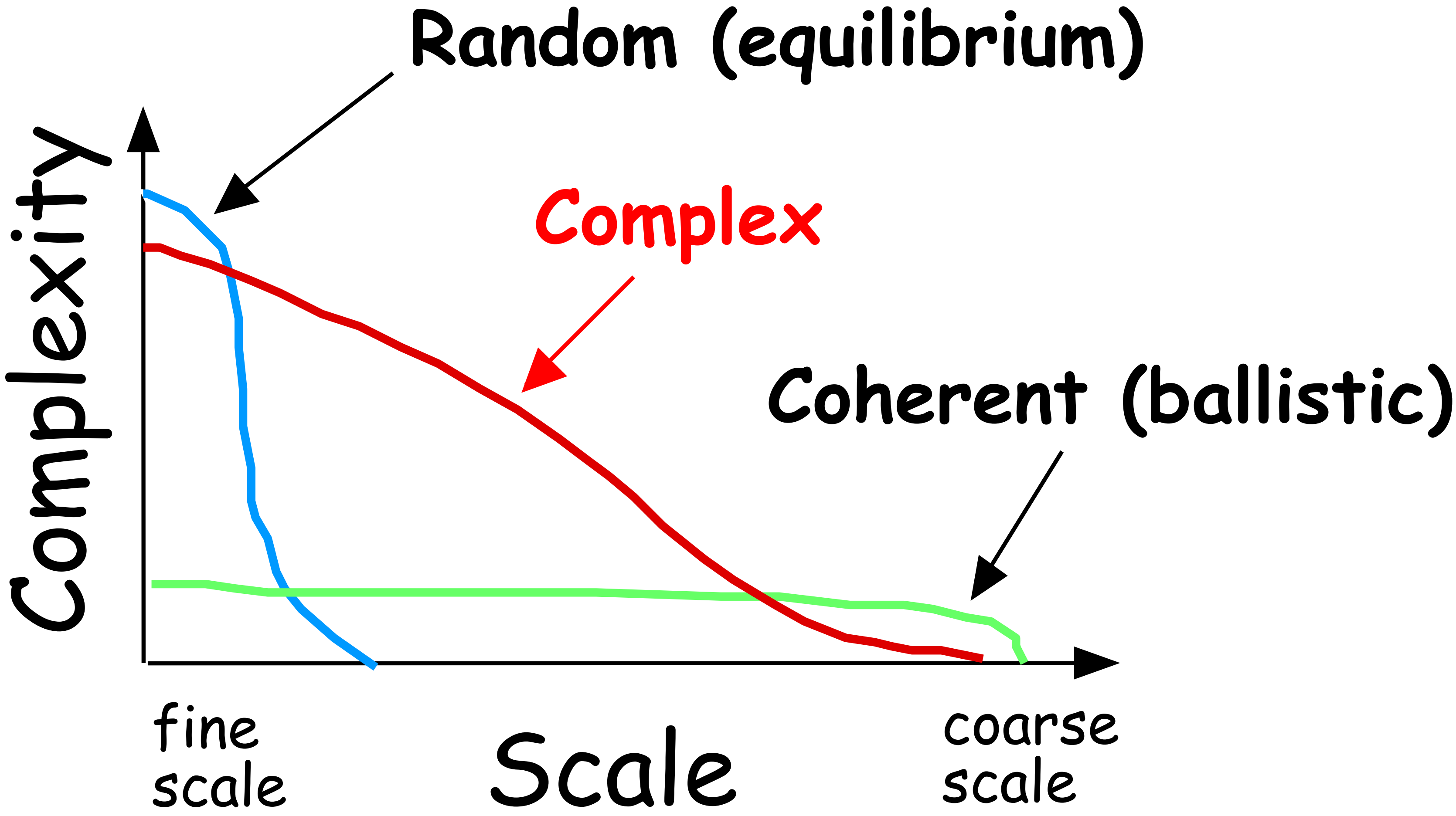
What's missing? (2)

Multiscale view

A system's state can be characterized at different spatial/temporal scales, ranging from micro to macro

Let's measure them at multiple scales!

Complexity profile (Bar-Yam 1997)



Schematic examples

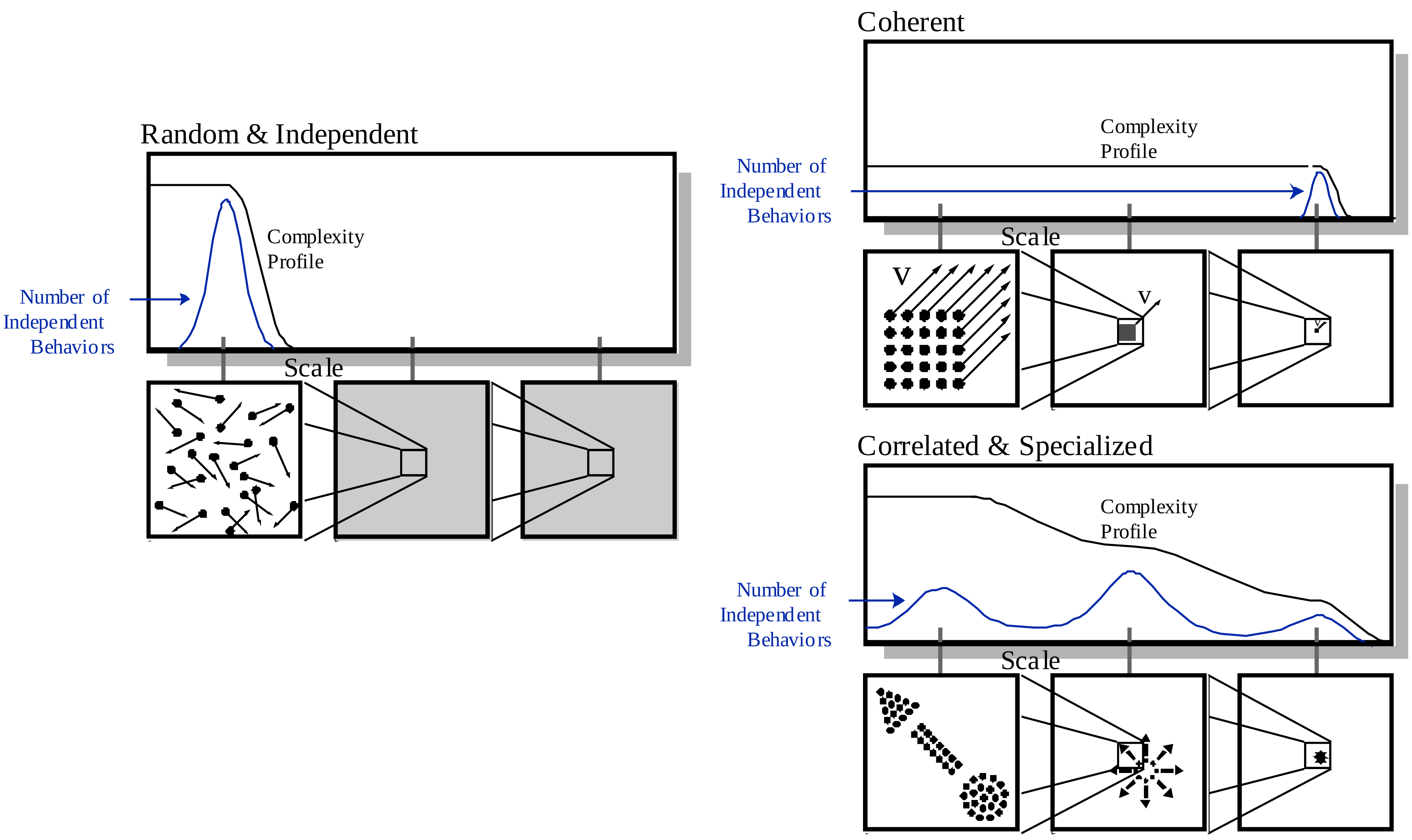
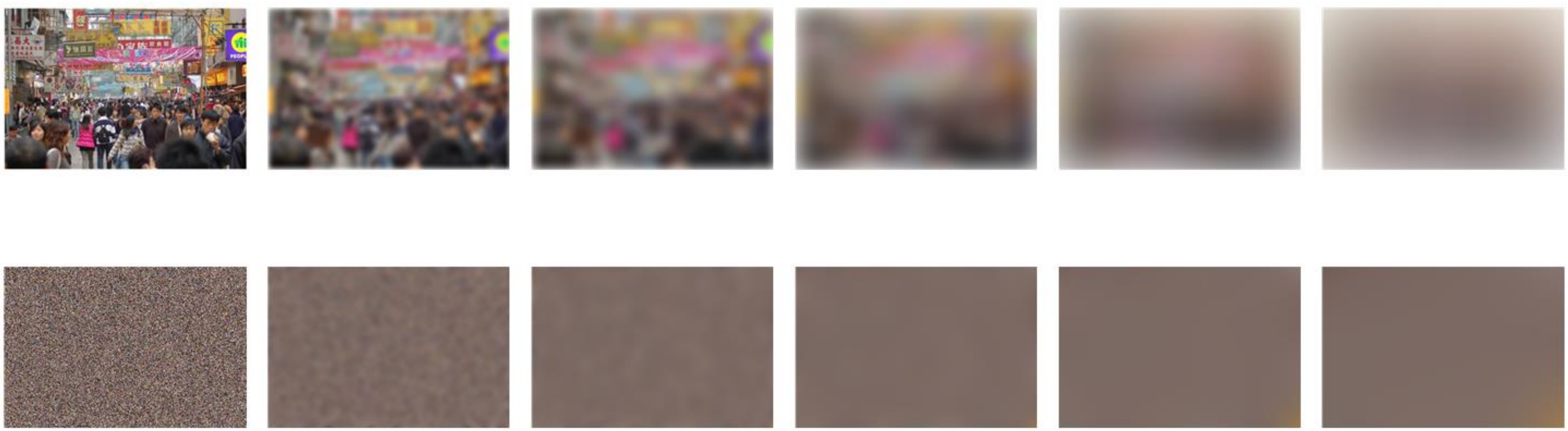
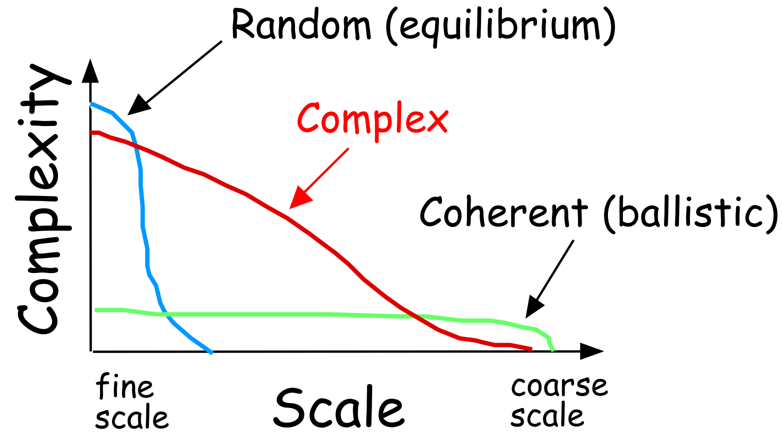


Image examples



Relationships with other concepts



Higher complexity on finer scale

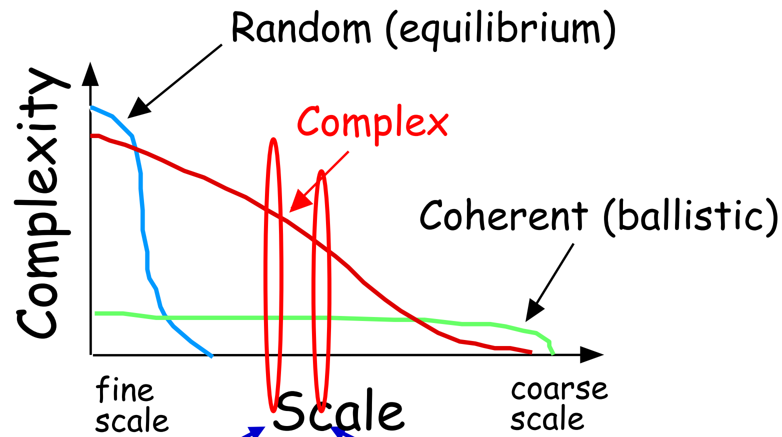
Higher complexity on larger scale

Independence
Randomness

Coherence
Correlation
Cooperation
Interdependence

Interdependencies increase relevant scale

Emergence and scale



Finer scale description
(all the) Trees

Larger scale description
Forest

Emergence (relationships)

Exercise

- Obtain an image file
- Blur the image using several different blurring radii (r)
- Calculate "complexity" of each blurred image (using differential entropy etc.)
- Plot the measured complexity over r

- How does the result depend on the content of the original image?

Summary

- Information and algorithmic complexities both characterize random patterns as "most complex"
- Statistical complexity aggregates statistically identical states
- Multiscale complexity considers complexity profile over scales
- They both can capture the "natural complexity" we see in the real world