Topological Computer Vision
new area within Topological Data Analysis

Vitaliy Kurlin, http://kurlin.org
Microsoft Research Cambridge
and Durham University, UK

The secondment at Microsoft
is supported by the EPSRC
Impact Acceleration Account
Computer Vision problems

Problem (briefly): image in $\rightarrow$ description out.

Input: a matrix of greyscale (or colour) values.

http://kos.informatik.uni-osnabrueck.de/3Dscans

Questions: how many objects, what are their boundaries, how do they relate to each other?
State-of-the-art recognition

*ImageNet*: 14M images, 21K classes: cars etc.

All images are labelled by 25K humans, how?

Deep neural net: 60M features are optimised on all the pairs \{image→label\} for several weeks.

Smallest average error 6.6% on new images.
Failures of deep learning

No theory behind: brute-force black box, also we crucially need lots of correctly labelled data.

10× difference between ‘dogs’ is in the middle. Where is a ‘noisy’ dog, how is it misclassified?

What is the last image with 99.6% confidence?
Where are self-driving cars?

**Answer**: wait until the error drops to 0.01%.

**Google** team report at CVPR in June 2015:

**Small problem**: 200 decisions per second, any wrong decision can lead to a fatal accident.

**Big problem**: get a *stable-under-noise* output.
TDA = topological data analysis quantifies persistent topological structures analysing unorganised data across all scales.


In 2008: TDA-based start-up Ayasdi, Stanford.

**Goal**: also use *machine learning* and *statistics*. 
What are data in TDA?

Input: a cloud of points with pairwise distances without any scale, ≠ neighbours, noise bound.

2D cloud: edge pixels in an image, a noisy scan.

High-dim cloud: a vector of features, histogram.
Learning the shape of data

PCA works for *linear approximations*. What if the data are around a non-linear structure?

All past skeletonisation algorithms are *iterative*, use *parameters* (scale or weights of criteria).

**Problem:** given only a sample $C$ of a graph $G$, compute a skeleton *provably approximating* $G$ (the same homotopy, geometrically close to $G$).
Comparison: PCA vs TDA

For a dense sample $C$ of a narrow ellipse, PCA outputs a straight line $L$ approximating $C$.

TDA-based clustering Mapper (Ayasdi) outputs an ellipse-like graph $G$ depending on a scale.

Project $C$ to $L$ covered by overlapping bins.

Is it possible to get $G$ without using a scale?
From a cloud $C$ to a filtration

**Def:** the $\alpha$-offset of a cloud $C$ in a space $M$ is the union of balls $C^\alpha = \bigcup_{p \in C} B(p; \alpha)$ of a radius $\alpha$.

Filtration $C^0 \subset \cdots \subset C^\alpha \subset \cdots$ in a metric space.
Output of TDA: all life spans

The evolution of all holes is summarised by bars $[\text{birth}, \text{death}]$ in the barcode or by dots $(\text{birth}, \text{death})$ in the persistence diagram.
Stability of persistence

**Th** (Cohen-Steiner, Edelsbrunner, Harer, 2007)

If a data cloud $C$ is perturbed by $\varepsilon$ (in the $\varepsilon$-offset $C^\varepsilon$), the persistence diagram is perturbed by $\varepsilon$, namely there is an $\varepsilon$-matching of all dots in PDs.
Computer Graphics application

**Problem**: complete all closed contours or paint all regions that they enclose (a *segmentation*).
Counting holes in noisy clouds

The graph $G$ has $H_1$ of rank 36, hence any $\varepsilon$-sample $C$ of $G$ will probably have 36 holes.

VK, CVPR 2014: guarantees for true # holes.
Using stability of persistence

We can find the *widest diagonal gap* separating 36 points from the rest of persistence diagram.
Harder than counting cycles

Initial regions ↔ red dots in PD (too many).

We should merge 36 regions of high persistence with all remaining regions of lower persistence.
Merging initial regions

We output exactly 36 *highly persistent* regions.

Parameterless skeletonisation

Homologically Persistent Skeleton $\text{HoPeS}(C)$ is the first *universal structure* on a cloud $C$ that optimally captures all 1D persistence on $C$. 

![Diagram of sparse cloud C and its homologically persistent skeleton HoPeS(C)]
MST and 0D persistence of \( \{C^\alpha\} \)

For a cloud \( C \) in any metric space, \( \text{MST}(C) \) is an *optimal structure on* \( C \) that encodes all 0D persistence for the filtration of the \( \alpha \)-offsets \( C^\alpha \).

\( \text{HoPeS}(C) \) extends this idea to 1D persistence.
$\text{HoPeS}(C) = \text{MST}(C) \cup \text{critical edges}$

**Def:** each critical edge gives birth to a class for $\text{birth} \leq \alpha < \text{death}$ in 1D persistence of $\{C^\alpha\}$.

$\text{HoPeS}(C)$ is a rotation-and-scale invariant structure on $C$, encodes all 1D persistence.
Reduced skeleton $\text{HoPeS}(C; \alpha)$

**Def:** for any scale $\alpha > 0$ to get $\text{HoPeS}(C; \alpha)$ from $\text{HoPeS}(C)$, remove all edges with a length $|e| > 2\alpha$ and critical edges $e$ with $\text{death}(e) \leq \alpha$. 
Optimality of $\text{HoPeS}(C; \alpha)$

Th (VK’15). $\text{HoPeS}(C; \alpha)$ for any scale $\alpha$ has the minimum length among all graphs $G \subset C^\alpha$ with the same homology $H_0, H_1$ as $C^\alpha$, so $\text{HoPeS}(C)$ ‘captures’ homology of the cloud $C$ at all scales.
Recognising visual markers

Shop barcodes are not readable by humans.

We can make *visual markers* like Egyptian hieroglyphs *readable* by *humans and robots*.

VK, CAIP’15: Computer Analysis of Images and Patterns
Global stability of HoPeS′(C)

Input: only a cloud C without any parameters.

Output: HoPeS′(C), time $O(n \log n)$ for $C \subset \mathbb{R}^2$.

Cor (VK): derived skeleton HoPeS′(C) stays in a small offset under perturbations of a cloud C.

HoPeS(C) is extended to any finite metric space C and to any filtration of complexes on C.

Key obstacle in low level vision

Cameras now produce huge images $\geq 16$Mp.

Fastest algorithms run in $O(#\text{pixels})$ time, but most can’t run in real time.

Can we *smartly* reduce the number of pixels without losing the original quality?
Over-segmentation problem

Split a large image into a smaller number of superpixels. Past work: a superpixel is a union of square pixels at a given resolution, often disconnected irregular shapes with holes.

If instead of $n = 1M$ pixels we produce 10K (1%) good superpixels, time $O(n^2)$ will be $10^4$ faster.
Resolution independence

Our approach: a mesh of convex polygons minimising the exact reconstruction error =

$$\sum_{\text{pixels}} (\text{Intensity} - \sum \text{Area}(\text{Polygon} \cap \text{Pixel}) \text{Colour})^2.$$ 

The key advantages are

- an objective evaluation
- subpixel straight edges
- nice convex superpixels.
Comparison with SLIC superpixels

Left: clustering-based superpixels by D. Harvey.

Right: our resolution-independent superpixels.
A noisy cloud of edge points is the input $C$ for producing an initial skeleton.

Some light blue edges seem invisible to humans.
Cloud → skeleton → superpixels

**Pipeline** for TDA in a superpixel segmentation.

**Left**: edge points detected at subpixel resolution.

**Middle**: a simplified initial skeleton of a cloud $C$.

**Right**: convex resolution-independent superpixels.
Original vs reconstructed

Which image is original, which is reconstructed?

What’s the ratio of superpixels to original pixels?
Summary: TDA for Vision

Microsoft contributes 15K GBP for a postdoc in the new area of Topological Computer Vision.

- guaranteed stable-under-noise algorithms
- time $O(n \log n)$ for any point cloud $C \subset \mathbb{R}^2$
- HoPeS($C$) is a 1D persistent structure giving a provably correct reconstruction of a graph

Future impact case: embed our optimised superpixel algorithm into Microsoft products.