Dan Feldman, University of Haifa

It is the first time that we discuss the **coresets**, a powerful technique which enables the use of smaller sets of data instead of larger ones without compromising the quality of the output. To learn more about it, we decided to interview one of the most important researchers in the field: **Dr. Dan Feldman**, who after three years at the **Computer Science and Artificial Intelligence Lab of the MIT (CSAIL)**, is now Director of the **Robotics and Big Data Lab** and Senior Lecturer at the **Computer Science Department of the University of Haifa**. He was kind enough to tell us more.

Computer Vision News: *What are coresets and why are they so important? Why are they so powerful? Why should we use them?*

Dan Feldman: The main idea is that in computer science we usually have a problem that someone suggests, let's say clustering. Different solutions will be suggested to solve this problem. We usually want to develop a better algorithm over time with an improved running time, memory or space.

With coresets, the philosophy is different. Instead of trying to suggest another algorithm, we want to prove that we can reduce the data, so that running existing algorithms on the reduced "small data" will provably give approximated result, as running them on the original "Big data". We can usually reduce the data, not by half, but by order of magnitude: for example, from n to log(n). This is done, not by designing a new algorithm for solving the problem, but by just running the new algorithm and existing algorithm on small compressions.

Unlike other compression techniques like zip or mp4, coreset is data reduction and not just compression of the input in the sense that it's problem-dependent. A point may be important for one problem, but not important for another problem.

We keep seeing papers on optimization for new problems, but we still have general techniques such as linear and quadratic programming. We have Singular Value Decomposition (SVD) and Principal Component Analysis (PCA). We have the derivatives. We have general techniques on how to optimize functions if they have specific properties. These days we also try to find more general techniques for coreset constructions.

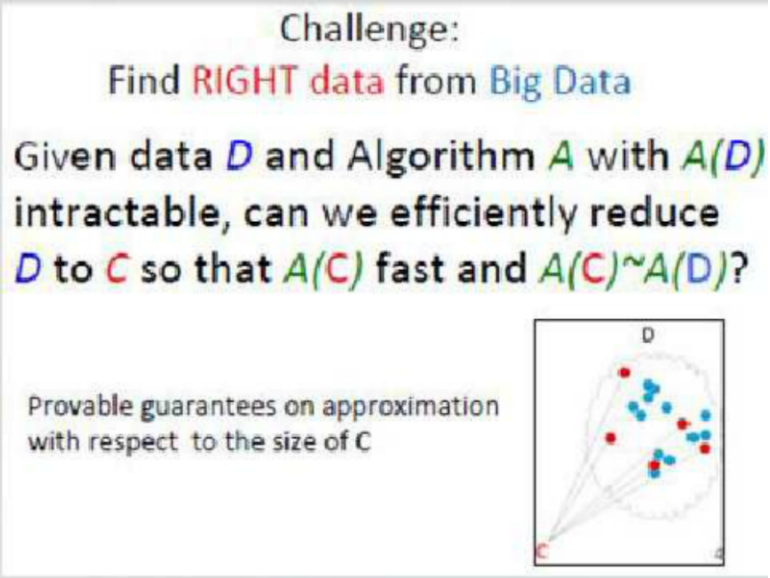
"Unlike other compression techniques like zip or mp4, coreset is data reduction and not just compression of the input"

CVN: *Is it because coresets are so powerful that you chose them as your main technique?*

Feldman: Yes - There are many social, academic and industrial reasons why we use coresets these days. If you know how to optimize a program and use the original data with the small data, of course you get some errors. Surprisingly, **the results on the coresets are usually better than the original data**. If you give me data to find the optimal solution, I can do better by moving some of the data. If we just have strange heuristics that give you some number without any proof of why it's good or bad, usually these heuristics only find local minima. The coresets remove a lot of noise, thus the local minima are much smaller and better. In some sense, data reduction removes most of the noise. That's how we get better results compared to running the algorithm on the original coreset. It's very good for business: we still use all of your expert knowledge. But also academically, I started using the coresets for theoretical problems. These days, I'm using the coresets for drones, image processing, computer vision robotics and EEG.

CVN: *Which kind of algorithm works better and on which kind of problems?*

Feldman: For many problems we can prove that a small coreset does not exist: removing one input point would yield a very bad approximation. In this discussion we assume that a small coreset exists, but we need to find it and prove its guarantees for every new problem. As I said, we try to find general solutions for problems that don't satisfy specific requirements. This is a kind of optimization. **We don't expect anyone to find one single technique to optimize all the problems in the world.**



"I hope to bridge the gap between theory and practice using these coresets"

CVN: In what way is the coreset that you use different from the coreset for k-means used by Sariel Har-Peled?

Feldman: It's not! Over the years, we have developed coresets for different problems and we keep improving and redefining them so that we can solve the problems.

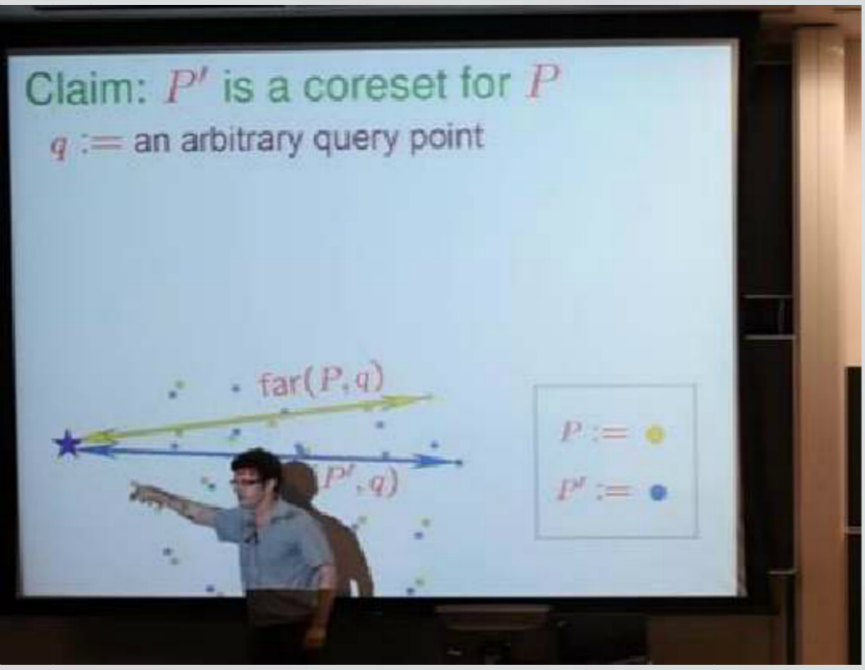
With every problem, we have a long line of research. In computer vision, there are numerous problems that I believe we can solve using coresets. Researchers from theoretical computer science are not interested in or (in the more common case) not familiar with this kind of problems.

Coresets are a new paradigm. It is more a state of mind than an exact mathematical definition because the exact definition changes from paper to paper. Actually, the number of definitions is very similar to the number of papers on coresets.

CVN: *I understand that coresets are less useful when you have too many layers or too many parameters.*

Feldman: Right - If you have n points and you're looking for n parameters, like in the Traveling Salesman Problem for example, you probably can't use coresets again unless you assume something about the input. Every polynomial time algorithm is a coreset in the sense that you have an exponential number of solutions, but you only search a small number. You still guarantee that the solution will be there. You compress the solution space. In some way, every efficient algorithm uses this kind of approach for compressing data.

The other related fields which we are trying to connect with coresets are compress sensing, sketching, all the "sufficient statistics" and property testing. Unlike with coresets, with property testing you're not allowed to look at all of the data; the challenge is the same: **to solve the problem with small data**. However, they have a much harder constraint because they cannot read all of the data, but only some of it. In coreset, we usually assume that we can scan all of the data. That's how we can solve more problems than property testing, for example.



"We don't expect anyone to find one single technique to optimize all the problems in the world"

CVN: *If our readers want to try to experiment with coresets themselves, is there any place where they can find tutorials, courses or examples?*

Feldman: We upload new coresets on my lab's website. We hope to publish a library within the next few weeks, but you can also get it online or send me an email. We'll try to have one library for everything.

CVN: *What surprised you from working on the coresets?*

Feldman: Recently we had a big surprise. We now examine coresets for two fields using differential privacy. The idea is to do machine learning, while preserving the privacy and anonymity of the users, so that we extract statistics from the data without revealing information about the individuals.

Surprisingly, our STOC'11 paper that we now implement shows a formal connection that says that if you can compress the data, it also means that you can have a private version; we refer to it as private coresets of the data, or a sanitized database. It means that you can add small noise to data so that the statistics will be preserved and the k-means will still

be the same. Yet you cannot reveal anything about the individuals from the k-means. There shouldn't be any connection between compressing the data and adding this small noise to preserve their privacy, but there is such a connection.

"Coresets are a new paradigm. It is more a state of mind than an exact mathematical definition"

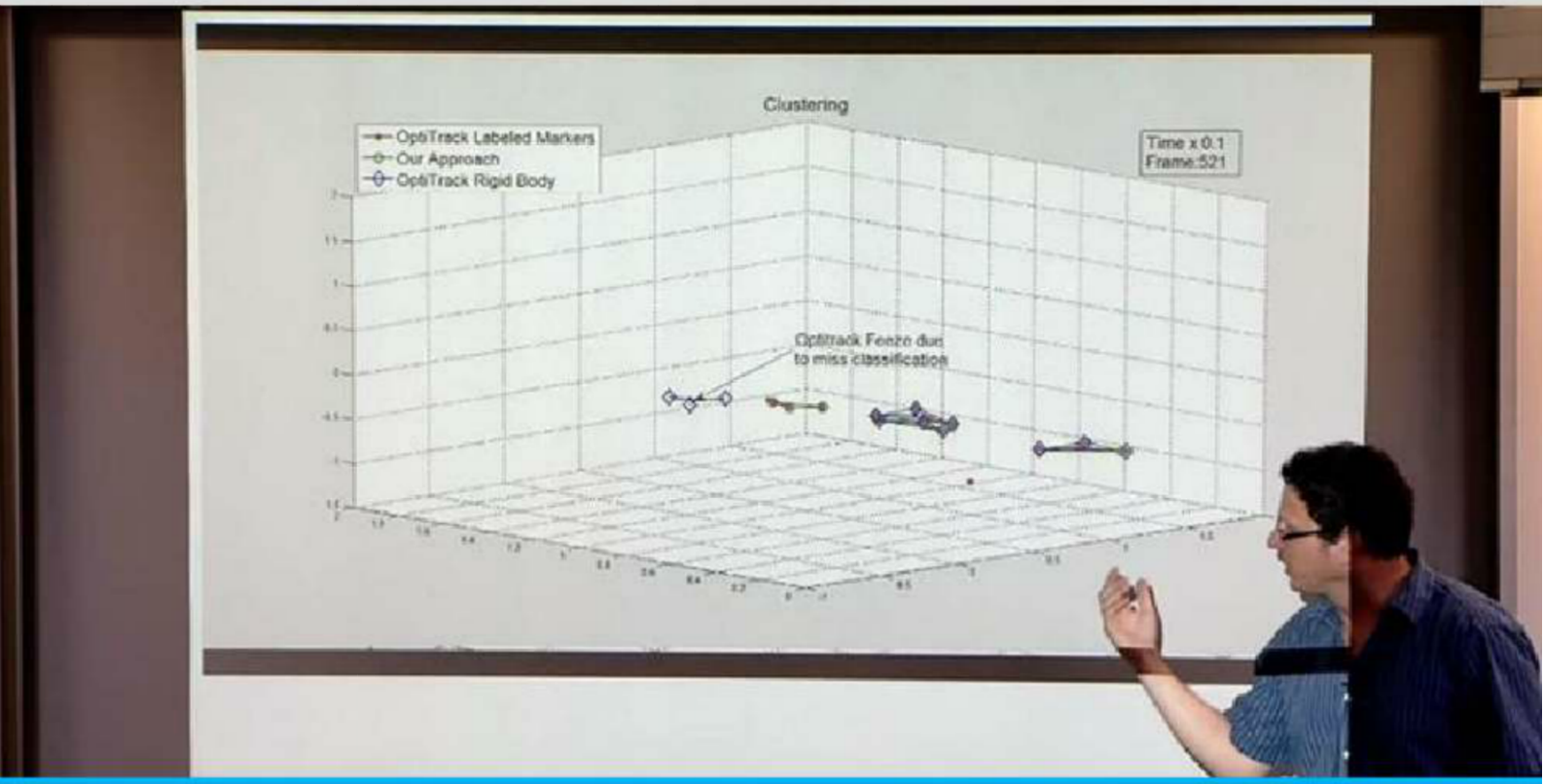
CVN: *It sounds like we have a distinctive coreset in our mind.*

Feldman: Kind of... We always need to decide what is important and what is not. Think at how we manage our time, what we see, what we hear and so on. The idea of compression and compressing the right thing is problem-dependent. I am sure that we do it all the time for all the senses that we have. Some people do it better than others.

CVN: What is the biggest breakthrough that you have seen?

Feldman: The main breakthrough in this field is the fact that **we now have a general framework coreset for any problem**. Unfortunately, I think the coresets are still buried in theoretical computer science conferences. Engineers are not using them because there is very little code out there.

I hope that in coming years we will have more implementations so that people who cannot read my papers can still run the coresets and evaluate them for their business or for their research. We want to **bring the coreset to the people** and bring the research into the industry.



This image and the previous one were taken on June 28 at the Simons Institute for the Theory of Computing: Core-sets for Real-Time Tracking using Caratheodory Theorem, with Applications to Drones. The video of the conference is here.