

Technical tips: Presentation

1. Aims and format of the presentation

The presentation is one of the three elements that you have to deliver to complete the course and get a grade (together with the final report and source code). Designing and holding an *efficient* and *effective* presentation is quite challenging. To help you with this, several tips are given below.

First, some details on the format of the presentation:

- Goal: The presentation aims to show two things: (1) how you **designed** your retrieval solution, and (2) how your solution **works**. Keep these elements in mind when constructing the presentation!
- Duration: 10..15 minutes (depending on the final number of groups that will register)
- Group: The presentation is for the entire team, not a single person. As such, all team members must participate at some point.
- Format: The best solution is to use a mix of slides and a demo.

Recall that your presentation should be efficient and effective. This means:

- *Efficient*: Say what you want concisely, shortening/compressing all collateral non-essential information
- *Effective*: Focus on the main points you want to convey, and say these clearly

2. How to structure your presentation

(1) Slides: They show *how* you designed your solution

Slides form the core element in your presentation, together with the (live) demo. Designing a compact, informative, easy-to-follow, slide-set is key to a good presentation and reduces the chance of many/complex questions afterwards. To do this, consider the following tips:

- Use a **top-down** structure: Start showing the end-to-end pipeline of your solution, e.g. in an image. Then, go to each step and detail this one. For each step, present its input (what it reads), its output (what it creates), detail how you designed it, and show evidence that it works as expected.
- Focus on design **decisions** and **results**. Shorten/skip obvious things. For instance, you do not have to show the PCA alignment formulas or the EMD formulas – just refer to the names of these algorithms. More generally, condense/skip all irrelevant implementation details (or refer to them in footnotes).
- Be **specific**: List/discuss all the design decisions which are particular to your implementation. In particular, list parameter values, e.g. the number of sample points and bins for the histogram descriptors, weights used for the distance function, algorithms used for the distance function (L2, EMD, KL, etc), implementation used for spatial search (ANN, something else). These help your audience getting a pretty good idea what is 'behind the hood' of your solution.
- Be **fair** and **balanced**: Show both examples of cases where your solution worked very well but also examples where it worked less well (e.g., good/poor results of querying).

- **Explain** things: If you make a non-trivial statement, e.g. “our system is very fast” or “querying certain shapes works very poorly”, try to explain *why* you think this happens. You don’t have to provide a formal proof, but you have to give an intuition why that happens, e.g., “because our entire pipeline is linear in the number of total shape vertices processed / we parallelize the most expensive step X” or “there are outlier shapes whose features differ very much from those of all other shapes in the database”.
- Be **visual**: Use short sentences; replace long terms by notations (where possible); use preferably images to explain things.
- Be **result rich**. For example, the normalization steps are best shown by snapshots of a shape before/after a normalization step; or histograms of the relevant shape characteristic before/after normalization. The distance function is best shown by a distance matrix covering all shapes in your database. The querying step is best illustrated by a matrix of snapshots, one query per row, leftmost column is query object, other columns are results, sorted by distance.
- A good tip for the presentation **length**: For an N-minute presentation, you typically use no more than N slides (title slide, end slide, and transition slides not counted). Do not aim for more. Even professional presenters have difficulty squeezing more slides per minute.

(2) Demo: It shows *what* you accomplished in the end

The demo of your end-to-end MR system is the final ‘proof’ that you successfully understood the course’s material and were able to implement a working solution. Organizing a good, compact, and convincing live demo presentation is however very challenging. For this, consider the next tips:

- **Prepare, rehearse, and test** the demo in advance! Probably the most common error of demos during presentations is that one assumes “they will work as usual”. This typically does not happen (*presentation effect*). So, make sure your demo really runs smoothly, test this several times, and make the demo session as smooth/simple as possible, so as to avoid problems.
- **Demos** take time. Do not try to show all aspects you built in your tool, no matter how proud you are about certain features (e.g., ten different rendering modes, lots of nifty menu entries, many different input formats, etc). Focus on the essential functionality of the tool, mention the other functionality, but do not attempt to show all of it. Recall this is a demo, not a tool manual or tutorial.
- A good way to save time during a demo is to create **presets**: These can be special GUI entries, small scripts, or combinations of the two, which execute some end-to-end function of the tool. You can fine-tune these presets to use whatever input data, parameters, presentation mode, etc of your tool as you desire. During the demo, you then have only to invoke (call) a preset. This not only saves time (setting N parameters to the right values costs time...) but also reduces risks that you do something wrong in the hurry of the presentation.
- A good tip to save time is to start demoing the **key features** of your tool. Once these are done and if you see you have more time, you can go to show the less important features.

3. Online presentation aspects

- **Rehearse** and **time**: Make absolutely sure that your presentation fits the allotted timeframe. This is extremely important, as the moderator (chair) will warn you when your time is nearly up (3..2..1 minutes left) and will cut your short when you are out of time. If you go over time, there will be less..no time for questions, which will adversely affect your presentation grade.
- **Live demos** are the best but are also risky (many things can go wrong so your great demo doesn't work as intended). To reduce these risks: (1) Do not change/tweak the code after rehearsing the presentation. (2) Pre-record a video of the tool demo or, if you want to be fully safe, the entire presentation + tool demo. Then, if something goes wrong during the live presentation, you can always switch to showing the video. NB: Pre-recording a video has the added benefit that it forces you to time your presentation.
- **Test** your entire presentation in MS Teams (log in as presenter, start a meeting, invite a colleague to join as audience). Test all aspects: voice, video playback, slideset showing, demo (screen sharing), switching between various presenters. A good tip is to record your presentation in Teams. Then, play it back to see how clear the audio and video were.
- Your entire **team** needs to be online during the presentation. It is up to you how and when you 'switch' between various speakers, just make sure that everybody will have a chance to speak, either during the presentation or during the question-answering.
- The **session chair** will introduce the speakers following the pre-announced schedule; collect any questions from the audience (placed via chat) and likely prepare some own questions; maintain time and warn..stop you when you are getting overtime; at the end of the presentation, select a few questions to ask, and follow-up on your answers when needed.