This image-quality tool analyses a sequence of screenshots collected by a user to generate three metrics: a frame-count, smoothness, and an image quality metric. The user needs to provide the screenshots in a single directory, and a text file that lists the screenshots in the order in which they were collected. The text file MUST reside in the same directory as the screenshots and must be named fileList.txt. So, in short, the user provides two items:

1.) A directory with all the screenshots
2.) A file named fileList.txt that lists the screenshots in the order in which they were collected.

This ordering is critical to get valid values for the three metrics. The file, fileList.txt must live in the same directory as the screenshots. The tool, which runs in a container, uses the screenshots and the ordering as listed in fileList.txt to generate values for the three metrics. The container is run using the script, genUE.bash.

Description:

genUE.bash starts a docker container, optionally with the flag --gpus=all. On container startup, the screenshots in the data directory are processed to compute frame-count, smoothness, and image-quality. The scripts the compute these metrics use as many cores as are available to exploit maximum parallelism.

Parameters:

[-v]: sets verbose Mode.
[-g]: starts docker container with flag --gpus=all. Use only if VM has GPUs attached.
-t <full-path to test data directory>: Full-path to directory with Screenshots
-s <full-path to results directory>: Full-path to directory into which results will be appended in csv format in file Res.csv

Typical use Sequence:

1.) Create an Ubuntu 20.04 VM
2.) Pull down container, vmmarkml-docker-local.artifactory.eng.vmware.com/hsivaraman/imgq03
3.) Tag container as imgq03:latest. Example command "sudo docker tag vmmarkml-docker-local.artifactory.eng.vmware.com/hsivaraman/imgq03 imgq03:latest"
4.) Copy or mount screenshots to the VM.
5.) Run the script genUE.bash with the parameters listed above. Parameters "-t" and "-s" are required.
Example: ./genUE.bash -t `pwd`/testData -s `pwd`/Results. In this example the screenshots and the file named, fileList.txt live in the `pwd`/testData directory. The directory `pwd`/Results should exist and on completion, the results will be appended to `pwd`/Results/Res.csv

Estimated run-time: On an 18 core server, the processing rate is about 10 fps

Results:

The results are appended to a CSV file, Res.csv in the Results directory specified using the -s flag to the script, genUE.bash. The results consist of three numbers:

1.) Frame-count: The total number of frames. This is the count of number of distinct frames which are not blurred version of following frames. This value divided by the total time over which the screenshots were collected is the FPS. The user needs to do the division to compute the FPS.
2.) Smoothness: This is a number that ranges between 0.0 and 1.0, with 1.0 being the best score.
3.) Image Quality: This is a number that ranges from 0.0 to 1.0 with 1.0 being the best score.

While these three numbers can be used as is, they make sense only when compared to the values from a reference system, e.g. a laptop running the same workload locally. To understand this, consider the following example. The workload running on a local laptop might have an FPS of 26. The same workload on the VDI system of interest might have an FPS of 23. Since these two values are somewhat close, it makes sense to compare the remaining metrics of smoothness and image-quality. If the smoothness on the local laptop is 0.9, and that on the VDI system is about 0.75, we can conclude that the VDI system has a smoothness comparable to that of a local laptop, or we could say that the VDI system is (0.75/0.9) *100 percent smooth. Similarly, we could compare the image-quality values from the VDI system and the local laptop to get a percentage image-quality measure for the VDI system. Consider a second example where the VDI system has scores, expressed as a triplet of values, (FPS, Smoothness, Image-Quality) of (1.0, 1.0, 0.95). This means the VDI system displays at the rate of almost 0 FPS. This is ridiculously low compared to say a local laptop with scores of (26, 0.9, 0.75), and at this point given that the VDI is hardly changing the displayed frame, it is guaranteed to have perfect smoothness, and possibly perfect image-quality, because the image never really changes. So, when we try to interpret the scores, we need scores from a reference system, AND we need the FPS to be close, typically within about 75% of that on the reference system. IF the FPS on the SUT is less than 75% of that on the reference system, the remaining two values can be misleading to use or compare.