Vol. 13(1), pp. 8-14, July-December 2022
DOI: 10.5897/IJSTER2022.0533
Article Number: 697EC0470002
ISSN 2141-6559

http://www.academicjournals.org/IJSTER

ACADEMIC JOURNALS
expand your knowledge

International Journal of Science and Technology Educational Research

Full Length Research Paper

Ear and eye training: Algorithms for the advanced student of audio, image, and video engineering

Michail Chourdakis

Department of Music Studies, University of Athens, Greece.

Received 25 September, 2022; Accepted 18 November, 2022

A lot of research has been taking place when it comes to audio, image, and video processing. Whereas the algorithms involved are very sophisticated, there is still an issue when they need to be taught to students, mainly because they are difficult to comprehend and they require an experienced trainer to guide the students. This article presents an analysis of two of our algorithms designed for ear and eye training, for the student that wishes to immerse himself in multimedia engineering.

Key words: C++, ear training, eye training.

INTRODUCTION

There is always a demand for courses related to audio and video, due to the high popularity of social media and present time high-speed internet availability which makes it possible to exchange large amounts of media. For these advanced lessons to be effective, certain conditions should be present: a good student's background in related fields (math, physics, etc.); high-level source of reading material; access to dedicated software and hardware; and adequately proficient trainer. For the first three conditions above there is already a plethora of possibilities for most budgets. Most students can take courses in basic computer usage; also they can find books and computer applications, capable of many basic and advanced audio, image, and video processing. This article is concerned with the *human trainer* in and how they might be replaced by *computer training*. For example, when equalization in an audio mixing process is discussed (Bobby, 2017), most people can understand the features of a parametric equalizer, but most cannot detect them when they are applied in the sound. When a gaussian blur is to be applied to an image, the effect is instantly visible but when the image is already blurred only a few can detect the parameters used to blur (Gedraite and Hadad, 2011). For the student to adequately understand the mechanisms of the signal processing, a capable trainer must be present. The other fact that stirs our interest is the ability of the student to be trained with his own material (sound sources and images). All existing applications use predefined material known to work in specific situations (for example, for color testing a full-color sea image is the testing material). The idea of our algorithms is to pick from the list of the files that the student will feed them, the best material for each of the training exercise. This describes the research to automate the student's training via a combination of computer algorithms, developed for Windows in C++ and

E-mail: chourdakismichael@gmail.com.

Table 1. Shows the results when the algorithm is applied

Variable	Beginner %	Intermediate %	Advanced %
Human trainer	52	56	62
Algorithm trainer	60	65	81

Source: Author

Table 2. Shows the results when the algorithm is applied

Variable	Beginner %	Intermediate %	Advanced %
Human trainer	32	34	41
Algorithm trainer	36	37	52

Source: Author

HLSL. C++ is the most advanced programming language and the templates feature (Vandevoorde et al., 2017) is particularly useful in our development to aid progress in various-precision models. Windows is the de facto standard for multimedia developing due to the emerging and easy-to-use audio and video libraries, namely, Direct3D and Direct2D (Luna, 2012) for hardware accelerated drawing and ASIO for low-latency audio input/output. HLSL is a C-like language capable of directly addressing the Graphics Processing Unit (Halladay, 2019) which allows us to create fast applications that process images directly into modern graphics cards. (Table 1.) We applied the algorithms discussed in our school advanced image and video classes and we presented statistics with the results of our students' training status, and we licensed our applications to be used from teachers and students in our school.

METHOD

The study applied to a population of 150 high school students in Music School of Alimos in advanced music and image processing classes in 2018 to 2019, 2019 to 2020, 2020 to 2021 and 2021 to 2022. All these students are between 16 and 18 years old, attending the last levels of a normal high school curriculum in music schools. The majority (about 75%) are male students and the vast majority (90%) are about to attend science/engineering-related courses in the university.

This curriculum includes optional courses about advanced sound and multimedia processing. They are separated into three groups, "beginner" (that is, students that have never used any audio/video/ image processing software, about 75%, "intermediate" (students which have some familiarity with such or similar software, about 20%) and "advanced" students that have a good level of familiarity with such apps, about 5% (Table 2).

All algorithms are applied to all students in separated classes and the tables cited describe how their success response is recorded for each of the algorithms described. They are given multiple choice questions within our specialized software and their answers are recorded when a course is applied with or without the algorithm.

- In each algorithm, we will describe:
- What the human trainer would do,
What the algorithm does and how it tries to fix potential human trainer errors,

- What would the current training applications do (if any),
- The results after our algorithms were applied to the students at three levels (beginner, intermediate, advanced),
- What would be needed in further research.

RESULTS AND DISCUSSION

Ear training

General

By saying 'ear training' we do not mean the generic (French: 'Dictée') applications that are taught in conservatories, primary schools, elementary music theory and in various applications online. In contrast, we mean the training of comprehension of the core sound elements (frequency, duration, amplitude) to aid the student work in sound engineering (composing, recording, mixing, and mastering) applications. An ear training application must take the following considerations into account: the room in which the sound will be heard and whether earphones will be used or not; and the nature of the sound used to train. The environment used for hearing is significant but, since almost no students own any expensive equipment, our algorithms were designed with medium-range

Table 3. Shows the results when the algorithm is applied

Variable	Beginner %	Intermediate %	Advanced %
Human trainer	71	75	84
Algorithm trainer	75	78	88

Source: Author

Table 4. Shows the results when the algorithm is applied

Variable	Beginner %	Intermediate %	Advanced %
Human trainer	82	89	94
Algorithm trainer	86	92	98

Source: Author

equipment in mind, such as a decent set of active studio monitors or headphones. The type of sound to used for training is the reason why our algorithms functioned correctly. Our algorithms can detect that and exclude such audio sources when EQ testing is running. On the other hand, it pays to train compression in common scenarios that are known to need it, for example the compression of a drum kit in a pop song.

Equalizer

The equalizer is an audio filter that lowers or raises selected frequency ranges in an audio signal (Senior, 2018). The experienced human trainer would, knowing the three basic goals of equalization (Clearness, Boosting, Fit) (Bobby, 2017) that, to train for equalization, a sound source of at least two instruments with clearly separated (Senior, 2018) predominating frequency ranges is needed. Common human errors (Chourdakis, 2022) in sample picking include: picking single-instrument sources only with a limited range, such as a flute, picking sources that contain similarly predominating frequency ranges (like a violin with a viola) which EQ testing is pointless, and picking sources that contain instruments that would never function well in an equalizer pattern (such as picking a dull-sounding electric piano and then trying to make it bright by raising the 5KHz area – there is nothing to raise but noise there) (Bobby, 2017). Existing EQ training courses such as Sound Gym (Sound Gym, 2022) will present the user with a predefined set of sources, carefully selected for the correct task. Our algorithms will aid the user pick the correct sound sources from their *own* sound collection, so, after the algorithm is fed the source songs, then it performs the following steps:

1. FFT scans (Smith, 2007) the source to ensure that only areas that contain a full range of frequencies are presented to the user. This way the above errors are avoided because poor frequency range sources are excluded.
2. Scans for beats (Smith, 2007) to include or exclude parts of the song based on the user's selection of the genre (Stark 2007) for example, pop songs containing full drum kit is tested for higher frequency equalizations.
3. Selectively applies our pitch detection algorithms (Chourdakis, 2007) to have an idea on what sort of instruments are contained in the sound source. For example, vocal music is detected from our algorithms and the equalizer then is biased through the frequency picking area (for example, in female singers, frequencies that boost the 'presence' of the voice in the range of 2500 to 3500 Hz are more likely to be selected for the training).
4. Based on the user's selection of frequency difficulty (1 to 5), limits the testing range of the frequencies. The higher the difficulty, the smaller the range is tested.
5. Based on the user's selection of amplification difficulty (1 to 5), options are presented on how much should the boost (or attenuation) be.

Further research on this subject would be to enable the algorithm to decide the filtering algorithms (Chebyshev, Butterworth etc.) Table 3 and 4 to be used depending on the equalization needs (for example, if tests in a low pass filtering require precise cut off, a Chebyshev I filter would be selected) (Hamming, 1997).

Compression

Testing a compressor is more complex because two main parameters are required (Senior, 2018): *threshold* and *ratio*. The inexperienced human trainer would probably: pick sources that they are already compressed, pick sources that do not need that much (if at all) compression, such as plain classic piano, or test high compression ratios where only lower ones have a meaning (such as testing vocal sources). Our algorithm is fed the source

Table 5. Shows the results when the algorithm is applied

Results/students	Beginner %	Intermediate %	Advanced %
2D human trainer	82	86	94
2D algorithm trainer	90	92	99
3D human trainer	52	55	60
3D algorithm trainer	58	62	64

Source: Author

songs and performs the following steps:

1. Scans the amplification of the source with the aid of an audio fingerprint (Jang et al., 2009) algorithm to ensure that the areas containing similar instruments have an almost full amplitude range (that is, they are *not* already compressed). Many sounds found on YouTube or those that are generated from electronic instruments are already compressed; these are discarded from the list.
2. Songs containing beats (Smith, 2007) are more aggressively scanned for full amplitude range because, especially in mastered mixes, drumkits are already compressed. Generally, no-beat songs and vocal songs are preferred. Existing solutions provide their own beat recordings which are uncompressed.
3. Depending on the user selected difficult level, presents tests either with only the ratio (with a fixed threshold value), or with only the threshold (with a fixed ratio value) or with both attributes being variable.
4. Songs that contain vocals are mainly tested for *threshold* because ratio is usually low in these compression modes; Songs that contain drums are mainly tested for *ratio* because threshold is usually fixed in these modes.

The compression test is a difficult process for the students. It requires continuous exercising on many music genres and a good pair of monitors. Further research in this algorithm would be to enable the manipulation of the attack and the release parameters in order to train the ear on how fast (or slow) the compressor is applied or erased.

Time variance

It is expected from a mastered audio piece to have some level of rhythm quantization, especially for the beats. However, the student ought to have the capability to detect the *need* for quantization and therefore, to become trained to detect small variances in the rhythm. This time the experienced human trainer will almost always pick the correct sources, because they are already manually trained in detecting rhythm inaccuracies. Our algorithm, therefore, will only pick songs with a continuous beat pattern, helping the student exclude sources without it. The algorithm will apply the beat detection and will also reject songs that are already quantized (Katz, 2015), while the rest of the songs will be presented to the user with a ratio selection (depending on the difficulty). The ratio varies between 0.5 (half time) and 2.0 (double time). This time the results are finer for our students because time stretching is relative (that is, they compare acoustically the unstretched sound to the stretched sound). A better version of the algorithm would be required to separate the beats from the rest of the sound for the student to focus on the rhythm.

Pitch variance

In contrast with the previous time stretching formula, pitch altering *does* have meaning in vocal songs where the pitch is likely to fault. It is true that in instruments without fingerboards (like the violin) pitch testing also applies, but our algorithms focus only on vocal songs which are far more common recordings between students than strings instruments. The drawback in this training course is that one must pick specific sources; Ready songs found online are not likely to have pitch faults. Therefore, one class of students must be able to produce their own vocal recordings and then another class of students shall be trained to detect pitch faults in them.

Therefore, our algorithm detects the vocal sounds and then intentionally alters the pitch randomly in order to create instability; applies both the generic pitch detection (Chourdakis, 2007) and specific pitch detection algorithms related to special genres. For example, in Byzantine Music the notes are not fully semitones (100 cents) but can be as small as 16 cents (Chourdakis, 2012); Table 5 and 6 presents the user with pitch shift options, depending on the type of music scale used (Semitone based or Byzantine based etc.). Further research in this field would be able to detect pitch faults in existing (preferably bare) vocal sources, something that would need adequate pitch recognition and source separation.

Panning detection

The final application of our ear training algorithm helps

Table 6. Differences.

Variable	Ear	Images	Video
Realtime	Yes	No	Yes
Hardware – dependent	Always	Rare	Rare
Transmission – dependent	No	Yes	Yes
Compression – dependent	Low	Low	High

Source: Author

Table 7. Shows the results when the algorithm is applied

Variable	Beginner %	Intermediate %	Advanced %
Human trainer	81	84	86
Algorithm trainer	82	87	88

Source: Author

Table 8. Shows the results when the algorithm is applied

Variable	Beginner %	Intermediate %	Advanced %
Human trainer	31	36	40
Algorithm trainer	32	38	41

Source: Author

the student to detect audio placement, that is, where the sound comes from. For this, a multiple 5.1 or 7.1 monitor set is also useful if available (3D testing). The human trainer now must manually adjust the pan to the mono or stereo sources that are likely to be available. Our algorithm does this automatically: in monophonic signals the Semiautonomous Audio Panning System (Gonzalez and Reiss, 2010) is applied to separate them to stereo signals; and in multi-channel signals, their amplification is adjusted to simulate surround movement if a 3D testing is needed, or they are converted to stereo if a 2D testing is needed. When doing a 2-D pan testing, the algorithm presents a level to the student (from -1 to 1) asking him to place the sound. This obviously requires headphones or a well-aligned placing of the monitors. When doing 3-D pan testing (Amin, 2017), the user uses our 3D designer to place a floating orb in a virtual room, depending on the placement of the sound. Further research in this algorithm would include the ability to detect how much a stereo recording is already separated and the ability to decide where to expand or shrink the pan. In 3D the ability to use known transformations to manipulate the sound (for example, rotate the entire source by 90 degrees) would allow the student to be trained with multiple-choice questions (instead of manually having to place the floating orb).

Miscellaneous training

There are a lot of other audio algorithms that can be used for training such as reverb detection, modulation detection, noise detection, vibrato detection. We are trying to enhance our algorithms as much as possible to cover an even broader section of audio training.

Eye training

General

Eye training is different from ear training. Table 6 to 9 summarizes the differences. Our algorithms do not consider hardware or compression situations. Unless the image is HDR, or the video is >= 4K in resolution (an unlikely situation for schools as of 2022) everyone has adequate hardware to view and image or a video in full quality. Related to compression, perfect quality is assumed by our algorithms because video streaming processors such as H.265 and VP90 (Sullivan et al., 2012) heavily interfere with various video features such as colors. Another thing to consider is the actual ability of the computer to train the student. Whereas, in the sound field, the computer could relatively easily pick optimal

Table 9. Shows the results when the algorithm is applied

Variable	Beginner %	Intermediate %	Advanced %
Human trainer	51	54	59
Algorithm trainer	52	57	62

Source: Author

parameters for the student training, this is not necessarily true for the eye training. For example, in the fields of image segmentation (Venetsanopoulos and Plataniotis, 2000) and motion detection (Gedraite and Hadad, 2011) the human eye is far more efficient than any algorithm discovered until now; the same occurs in various other fields of image analysis. Therefore, we have a limited number of tests available, and these tests do not significantly enhance the abilities of our students as much as the audio training.

Colour tests

The student must carefully study the features of RGB and HSL coloring in images. A human trainer here would not be able to pick accurate images for training since the levels of the RGB would not be easily visible for selection, the human eye is more sensitive to luminance changes than to hue changes (that's the basis of the YUV formats). Our algorithm finds the histogram of the images selected by the user and only picks images with a full histogram (that is, images that have a high range of exposure in all colors, both dark and bright). Images in grayscale (or images with low saturation generally) are excluded as well. Then, it performs various color tests: color hue rotation (helps the student to detect the hue of the color); saturation rotation (helps the student to detect the intensity of the colour); luminance rotation (helps the student detect the power of the light); and color matching between different images (Venetsanopoulos and Plataniotis, 2000) to allow the student to give a ratio of difference between images. A better version of the algorithm would pick only images with large areas of similar hue values (like a sky) in order to help the student work with the hue rotation only in real-life images.

Filtering

Like audio filtering which is usually an application of the Z-Transform in infinite filters (Hamming, 1997), image and video filtering is usually an application of a 3x3 or 5x5 convolution kernel (Shapiro and Stockman, 2001) that results in image sharpening or image blurring. This time the teacher would pick the correct images for testing because kernel filters like blur or sharpening are easy to detect. What our algorithm will do here is to perform a sharpening on all images, followed by a focus-unblur algorithm (Yuzhikov.com, 2022) to only select images that are not already defocused or blurred or heavily sharpened. Then questions the user based on the difficulty level on the power level of either a blur, or a sharpening. The test is difficult. Our students have difficulties in detecting the kernel parameters even when the images are picked with the above parameters. A better version of the algorithm would only select images with clearly defined shapes (rectangles, triangles etc.) so blurring would be easier to detect.

Lighting

Apart from color and filtering, there are other various effects to be tested in an image such as: brightness/ exposure, opacity, chroma keying and highlights/ shadows. Essentially these are applications of the light (Evening, 2018); Brightness/Exposure/Opacity change the light to the entire image, whereas the highlights, shadows and the chroma key change specific portions of the light. The algorithm will, depending on the user selection of difficulty, pick images that have a full range of light or opacity value (that is, 0 to 255) and with both dark and bright elements for the 'hard' level and pick images that have a low range of light (most of the image is bright or dark) for the 'easy' level. Because the human trainer's eye is sensitive to the luminosity of the images, our last algorithm does not enhance much the training process.

Other effects

There are many other effects (Ke et al., 2005) related to image and video processing such as:

1. Rotation, video stabilization and generic 2D and 3D transforms: We will not consider computer-based training in these because the human eye can more easily detect e.g., an image rotation than the computer.
2. Advanced coloring effects: color temperature, image vibrance, chromatic aberration. These are essentially combinations of color and lightning. We will consider them in the future.
3. Advanced filtering like bilateral and high pass. We may consider them in the future; they are difficult for our students to understand.

4. Distortion effects like shake, zoom, fisheye and warp: The human eye can more easily detect them than the computer.

As new processing algorithms become available, efficient, and reasonably successful, we will consider them to enhance our eye training algorithms.

Conclusion

Our aim is to train our students not only to be able to use the effects discussed here, but to also know when and if they need them, and to which extent. The application of these algorithms in our school in scholar seasons 2020 to 2021 and 2021 to 2022 has aided them to be more efficiently trained especially in sound; In video and image engineering, more work needs to be done, considering Computer Vision (Ke et al., 2005) frameworks such as Open CV and DLib used by our applications. In the future, we hope to be able to expand our training algorithms to fully cover the training needs of audio, image, and video processing.

CONFLICT OF INTERESTS

The author has not declared any conflict of interests.

REFERENCES

Amin M (2017). Radar for indoor monitoring: Detection, classification, and assessment. CRC Press.
Bobby O (2017). The Mixing Engineer's Handbook, Media Group, 2017.
Chourdakis M (2007). Time Domain Pitch Recognition.4th International Conference of Sound and Music Computing, Lefkada.
Chourdakis M (2012). Byzantine teaching through computer analysis. 1st Symposium of Byzantine Music Students.
Chourdakis M (2022), Intelligent Algorithms to support teaching. Journal of Modern Education Review pp. 391-397.
Evening M (2018). Adobe Photoshop CC for Photographers. Routledge, 2018.
Gedraite E, Hadad M (2011). Investigation on the effect of a Gaussian Blur in image filtering and segmentation. Proceedings ELMAR-2011 (pp. 393-396). IEEE.
Gonzalez E, Reiss J (2010). A real-time semiautonomous audio panning system for music mixing. EURASIP Journal on Advances in Signal Processing pp. 1-10.
Halladay K (2019). Practical Shader Development: Vertex and Fragment Shaders for Game Developers. A Press.
Hamming R (1997). Digital Filters, Courier Corporation.
Jang D, Yoo CD, Lee S, Sunwoong K, Kalker T (2009). Pairwise boosted audio fingerprint. IEEE Transactions on Information Forensics and Security 4(4):995-1004.
Katz B (2015). Mastering Audio the art and the Science. Focal Press.
Ke Y, Hoiem D, Sukthankar R (2005). Computer vision for music identification. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) 1:597-604.
Luna F (2012). Introduction to 3D game programming with DirectX 11. Mercury Learning and Information.
Senior M (2018). Mixing Secrets for the Small Studio, Routledge.
Shapiro L, Stockman G (2001). Computer Vision, Prentice Hall..
Smith S (2007). The DSP Guide, California Technical Publishing, 2007.
SoundGym (2022). [Online]. Available: https://www.soundgym.co/playground/eq.
Stark A, Plumbley M, Davies M (2007). "Audio effects for real-time performance using beat tracking. In Audio Engineering Society Convention 122. Audio Engineering Society.
Sullivan G, Ohm J, Han W, Wiegand T (2012). Overview of the high efficiency video coding (HEVC) standard. IEEE Transactions on Circuits and Systems for Video Technology 22(12):1649-1668.
Vandevoorde D, Josuttis N, Gregor D (2017). C++ Templates: The Complete Guide, Addison-Wesley Professional, 2017.
Venetsanopoulos A, Plataniotis K (2000). Color Image Processing and Applications. Springer Science & Business Media.
Yuzhikov (2022). Yuzhikov.com, 2022. [Online]. Available: http://yuzhikov.com/articles/BlurredImagesRestoration1.htm.