

Copyright Lighting & Sound America

April 2025 complete issue: <https://link.lsamedia.com/apr2025>

AI Lighting Programming

By: Brad Schiller



Unless you have been living under a rock for the last few years, you are probably aware of the many uses of Artificial Intelligence, or AI. This technology is changing the way we work with data on many fronts, and it won't be long before it becomes an integral part of your lighting workflow too.

I am not saying that AI bots will soon replace lighting programmers, but there are many possibilities for improvements to our process. Currently, a few software programs promise AI creation of lighting cues, but I believe the real future is in virtual assistants built into lighting consoles. They will work alongside the human programmer to cohesively create and edit the lighting programming in an extremely efficient manner.

onstage. The concept of a virtual assistant is to fix this shortcoming by enabling the console to intelligently assist. You can think of this in the same way predictive text can auto-complete a word or suggest new words to complete a sentence, even correct your spelling or, in our case, the common line syntax. By using historical data, the virtual lighting assistant is able to recognize a pattern in previous tasks and can use this "knowledge" to predict the most likely next action the programmer would want to perform.

As a very basic example, let's say you are programming a sequence and you set fixture one to red in the first cue, fixture two to red in the second cue, and fixture three to red in the third cue. By now, a recognizable pattern should be

Ross Williams' grand idea

Since 1995, UK-based lighting programmer and designer Ross Williams has worked in theatre, concert, television, live events, and film and has been fortunate enough to work alongside some of the world's top lighting designers, lighting directors, and directors of photography. During this time, he has also been involved in many aspects of technology development; recently, he wrote a patent regarding an AI-based virtual lighting assistant agent, or VLA.

Williams' extremely detailed concept lays out a fantastic plan for the future of lighting programming utilizing AI tools. As patents can be rather dry and hard to read, I sat down with him to discuss his ideas.

Brad Schiller: Please describe your concepts of the Virtual Lighting Assistant Agent in simple terms.

Ross Williams: I have long been frustrated that, during all programming, the console I'm sitting behind has no concept of what I'm trying to achieve. Repetitive tasks are just that, and, as a programmer, you are on your own to keep track of everything, perform notation, and wrestle with various mathematical calculations to create the desired looks

formed, and the agent can offer to auto-complete the task based on a set of variables such as all the lights of the same type, in the current selection, or where information is contained in the same preset. This means that, rather than having to send all 250 lights to red one at a time, the assistant will make a recommendation to perform this in one go on your behalf. By using other available data such as XYZ positioning information, the assistant can recognize further patterns, such as the selection of fixtures with seemingly random head numbers, [which] form a circular layout in the real world.

The assistant will also enable the console to be much smarter by remembering what has previously been done in terms of programming and be able to present this information back to the programmer in a much more intuitive manner. Of course, by remembering, I actually mean understanding as opposed to simply recalling. With enough data of any type, the assistant will identify a recognizable pattern that should, in turn, allow for the prediction of the next most likely step in the given sequence, or to recall an entire stage lighting look based on a simple set of input instructions. Of course, it will always be for the programmer to accept or decline any suggestions put forward so that total control remains with the user at all times.

BS: What would the workflow be like for a lighting programmer when using a VLA?

RW: The workflow should be pretty much as it is today, working from the usual interfaces, but with contextual suggestions appearing conveniently for the user to auto-complete ongoing tasks, or to accept any suggested actions based on other environmental or historical data. The assistant would simply need access to the command line of the console on an embedded or externally hosted basis. By entering real-world emotive style instructions, the assistant would be able to translate this into the commands required to execute the desired functions.

BS: Will this just be a console tool to aid the programmer or are there direct benefits for LDs too?

RW: I think everyone will benefit in terms of a more streamlined and efficient workflow, although the concept is primarily aimed at assisting programming and playback of a lighting control system. An LD may find that favorite combinations of presets could be recalled intuitively based on past usage, and a style of more descriptive cue creation that is less “this light at that level” but more “strong moonlight from upstage” would be interpreted in a similar way that AI image generation works today. Clearly, the more data the system has been trained on, the more accurate and useful the recalled results will be.

BS: How does the VLA work regarding telemetry data?

RW: Such data would be used as part of the machine learning process to recognize patterns, cross-reference information, and further help generate contextual suggestions for the user. The assistant would have a pre-programmed understanding of all common external environmental conditional data, including but not limited to date, time, location, climate, temperature, tidal, solar azimuth information, etc., and could also process other industry open-source protocols from external sources such as winch and stage automation controls.

BS: Why did you decide to patent this concept?

RW: Partly as an exercise in learning more about the patent world for myself, but also through a frustration that few others appeared interested in this as a concept or could see a future in this type of workflow. I should add, however, that I have sought advice and help from a few trusted people and would not have reached the point of patent application without their support, enthusiasm, and expertise, especially in the field of machine learning. I am extremely grateful to those who have played a part in this journey so far.

BS: Do you think these ideas will be implemented into consoles soon?

RW: Sadly, no. I think some disruptive change to our established working practices is long overdue. That said, the upcoming generation of console users is growing up in a world of predictive text, Siri, ChatGPT, and AI generative fill, so perhaps this will drive the console manufacturers to wake up and take note. Currently, these concepts feel too “out there” for them to take much notice, but I do believe this will change, and perhaps sooner than we all think.

BS: What other aspects of AI do you predict will affect lighting programming?

RW: Artificial Intelligence is here to stay. We need to embrace it, not fight against it. Whilst it could be possible for some AI system to create a lighting plan or design in the style of a particular designer or programmer, I’m not concerned about this becoming a reality. I already see clients creating their own AI-generated concepts and visuals, but is this any different from the current Google Image-sourced style guides and mood boards we work from today? Someone still needs to turn these concepts into a reality onstage. With enough data and a variety of external references, any complex task could be easily completed based on a recognizable pattern or sequence and processed into a helpful set of results. This is what I see as the future of lighting control programming, aided by but not replaced by AI.

You can read Ross Williams’ VLA patent here:
<https://www.mikewoodconsulting.com/patentdb/pdf/11687760.pdf> 