Visual effects such as stage lighting or fog machines are widely used in live music performances to enhance the emotion and mood of the music played. Such schemes are designed to visually immerse the audience in the feeling of the song. Video displays such as TV screens or video projectors are now standard facilities in small to large size venues and recent trends in art tend to design computer programs to automatically allow interactions between the music and the visual effects.

We propose here to create a portable affordable system that will help automatically generate in real-time a visual artistic rendering of the music being played live in a small or medium venue without the undesireable budget constraints that face many working artists. As an alternative to lightings, we illustrate our multi-stream music analysis by creating a real-time OpenGL animation that reacts to events in the music piece.

The musical content of each instrument is well separated, since, in well engineered performances the sound pressure level of a particular instrument contributes greater than the spill from the other instruments at its corresponding microphone. Thus, no processing for source separation is required for the different instruments. The temporal and spectral characteristics of these signals can then be analysed simultaneously to generate enhanced visual effects.

For each audio stream, a Fast Fourier Transform (FFT) is computed every 23.2ms (or 1024 samples) as follow:

\[ X(n\Delta, f) = \int_{n\Delta}^{n\Delta+\Delta} x(t) \cdot \exp(-2\pi f t) \, dt \]  \hspace{1cm} (1)

Using a adapted passband filter, each instrument is separated from any possible spils coming from other sources, and information such as the band energy of the instrument (or the voice) is recorded:

\[ A(n\Delta) = \int_{f_\text{low}}^{f_\text{high}} |X(n\Delta, f)| \, df \]  \hspace{1cm} (2)

where \( f_\text{low}, f_\text{high} \) define the frequency band of the instrument. Without too much additional computational cost, the median frequency or the mean of the spectrum is also computed as follow:

\[ \overline{f}(n\Delta) = \frac{\int_{f_\text{low}}^{f_\text{high}} |X(n\Delta, f)| \cdot f \, df}{\int_{f_\text{low}}^{f_\text{high}} |X(n\Delta, f)| \, df} \]  \hspace{1cm} (3)

These are the measures used in our system. Other informative features such as pitch can also be computed if the computation time remains low for the hardware used.

A simulation of a stage complete with lighting effects is rendered on screen using the OpenGL graphics library. Various graphics methods are used in the render (3D modelling, texture mapping and tesselated objects).

The visualisation of the information extracted from the music played is done in three ways:

1. Concentrated spot lights to light up currently active musicians.
2. Ambient lighting to reflect the overall mood of the song. Brighter colors illuminate the stage when songs occupy higher frequency bands and have faster tempos to reflect the more exited performance.
3. Physical movement of the musicians on stage. The faster their arms and body move, the bigger their part in the overall mix.

The system runs on a standard laptop (Model: HP NN9420) with Intel Core Duo CPU at 166GHz, 512mb of RAM memory and a graphics card ATI Radeon X1600 256MB. The music analysis and the animation (25 frames per second) are created in real-time.

Top: the singer and one guitarist are not playing and are in the dark. Bottom: The guitarist is playing a solo, and using the median frequency \( \overline{f} \), the colour of the red lights varies from blue (low values of \( \overline{f} \)) to red (high values of \( \overline{f} \)).