Responsive Subtitles

by **University of Salford MANCHESTER**

Using a responsive subtitle approach [1] allows the subtitles to be dynamically re-blocked while a video is playing to meet the requirements of both the consumer and the display.

Features

Advantages

• The subtitles are rendered to fit the available space.

• Changing the font size can change the number of characters displayed but reading speed remains the same

• Adapts to new display types such as mobile phones or head mounted displays.

• The subtitles can avoid obscuring other features, such as graphics or faces, or other accessibility features such as the sign language below:

How it is works

• The responsive subtitles library runs in the browser, using traditional subtitle files, so no additional work is required from the broadcaster.

• The subtitles are atomized into single words preserving the styles and timings for each component.

• Heuristics provide rules, such as a changing speaker or break in the flow of dialogue to reconstruct the subtitles.

• A best-fit algorithm optimizes the distribution of words.

Personalisation

• The content producers provide a traditional subtitle file and a container location for the subtitles.

• The consumer can then change the subtitle size to meet their needs.

• As the subtitles are rendered locally, this also gives the scope to manipulate the rendering style, such as the font, colours or provide alternative rendering positions, such as speech bubble aligned with the speakers.
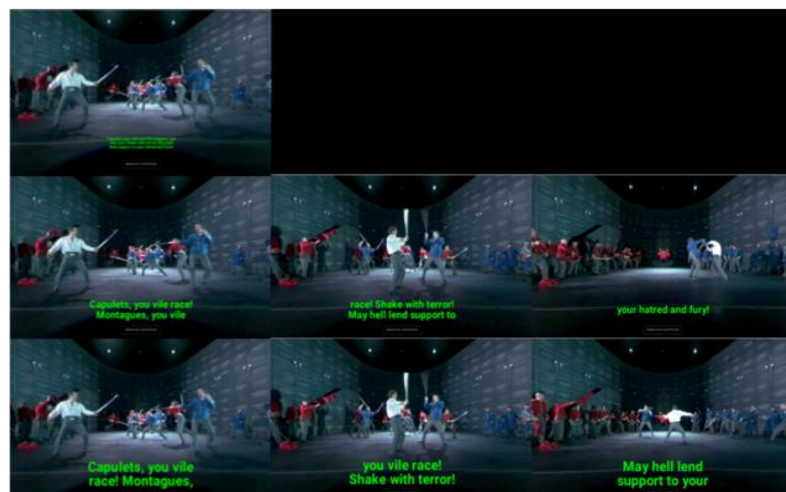
Integration with IMSC.js

ResponsiveSubs.js

• Our library provides an extension to IMSC.js (a JavaScript library for rendering IMSC1 Text and Image Profile documents to HTML5).

• IMSC documents can be loaded into a TT-object. The responsive subtitle library restructures TT-objects based on line character width and line count

• The integration with IMSC.js allows many existing web applications to be adapted to use responsive subtitles with ease.

• The library can also be used with nonlinear VR applications where IMSC subtitles are being used, such ar VR and computer games.

The ResponsiveSubs.js structure is shown below:

(a) IMSC.js converts the TTML document into a TT Object.

(b) The responsive library atomises the TT object into words, preserving an interpolated time and style for each word.

(c) The words are reconstructed into phrases split by a pause in the dialogue or a change of speaker.

(d) The phrases are subdivided using a best-fit algorithm to meet the line length requirements.

(e) a new TT object is generated with IMSC.js.
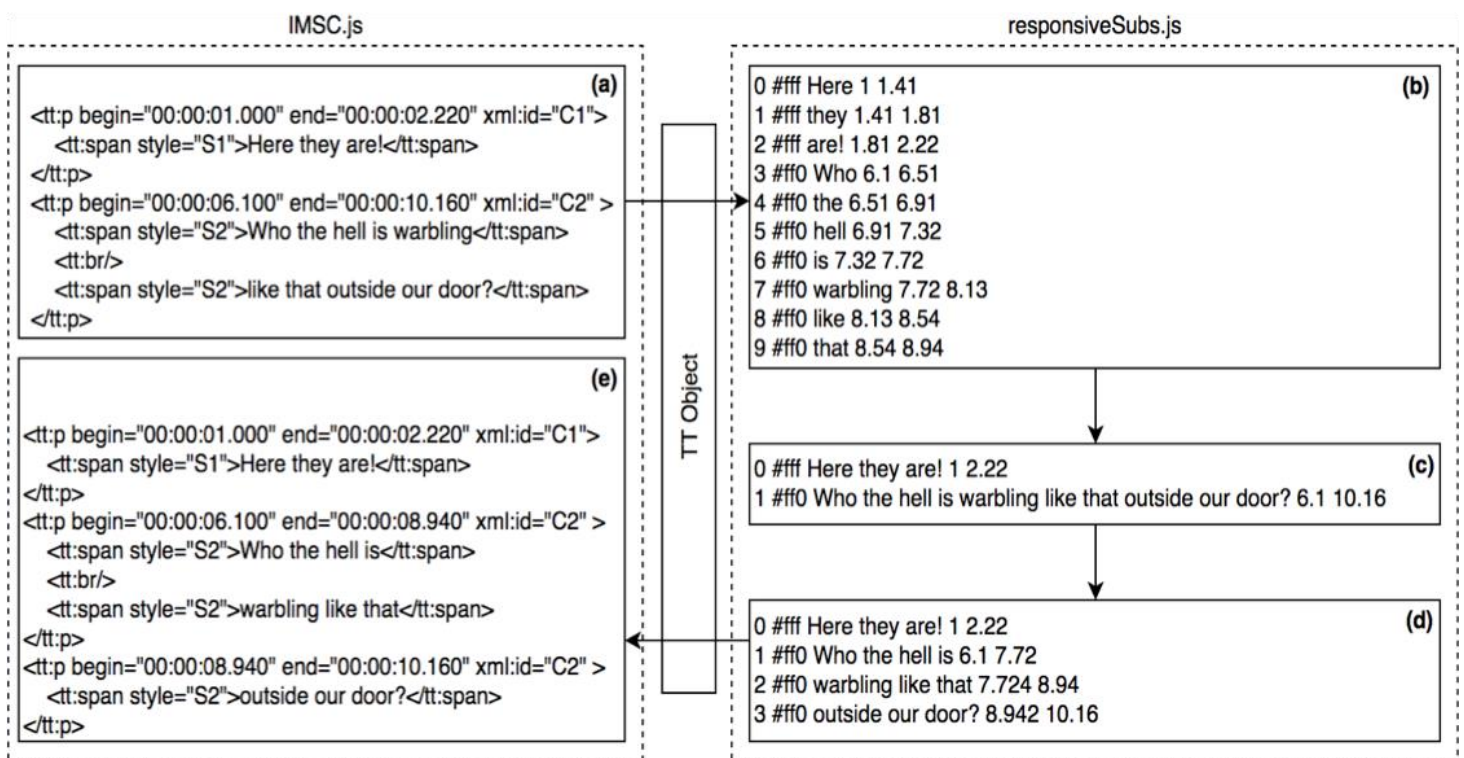
[1] Hughes, CJ , Armstrong, M, Jones, R and Crabb, M 2015, Responsive design for personalised subtitles, in: The 12th Web for All Conference, 18-20 May 2015