Lifting Media Fragment URIs to the next level

Thomas Kurz[1] and Harald Kosch[2]

[1] Salzburg Research, Salzburg, Austria,
thomas.kurz@salzburgresearch.at,
www.salzburgresearch.at
[2] University of Passau, Passau, Germany,
www.dimis.fmi.uni-passau.de

Abstract. The Media Fragment URI specification was released in 2012 and has been taken up by research and industry to some extend. Nevertheless the impact is weak in comparison to other W3C recommendations. Missing features, under-specified parts and a weak integration into common standards could be a key issues for that. In this paper we describe possible extensions that strengthen the specification in this points in order to make the Media Fragment URIs attractive for a broader community.

Keywords: Linked Media, Media Fragment URI, Media

1 Introduction

With the dawn of the Social Web Multimedia content and its semantic description came into focus in the past years, reflected in many efforts like W3C recommendations for a Media Resources Ontology [2] and Media Fragment URIs [9], which are taken up by research [8] and industry [5], [7]. Currently the Fragment URI specification is limited regarding expressiveness and thus not able to cover complex use cases. In the following we describe the current state of the Media Fragment URI specification, outline an approach that aims to overcome these limitations by spatial-temporal extensions and compare it to related recommendations. Additionally we sketch the integration into Cascading Style Sheets (CSS) to weave Media Fragments deeper into the infrastructure of the Web.

2 Media Fragment URIs 1.0

The Media Fragment URI 1.0 specification "provides for a media-format independent, standard means of addressing Media Fragments on the Web using Uniform Resource Identifiers (URI)". It utilizes the fragment identifiers specified in RFC 3986 [3] to add specific semantics for media segment identification. [9]. As URI fragments are handled client side by definition, the working group additionally recommends standard name-value pairs (analogous to the fragment syntax and semantic), which can be used within URI queries and thus can be

handled directly on the media server. For Media Fragments this is important in particular as any preprocessing like e.g. spatial-temporal cropping can drastically decrease the size of file that has to be transmitted to the client.

The working group describes four fragment dimensions, namely temporal, spatial, track and id. The temporal dimension is specified as "an interval with a begin time and an end time" and can be given in Normal Play Time (npt), SMPTE timecodes and as real-world clock time. E.g. `video.mp4#t=1,10` defines a temporal fragment of video.mp4 that starts at second 1 and ends at second 10.

The spatial dimension "selects an area of pixels from visual media streams." In the current version only the identification of rectangular regions is supported. The units that are considered for spatial clipping are pixel and percentage. The dimensions of *track* and *id* is specified very weak regarding semantics but can be used to identify a specific media layer and/or a specific predefined, named section of the media item. On this paper we focus on extensions for spatial-temporal fragments. Fragment dimensions can be combined in several ways to describe e.g. spatial-temporal fragments, like outlined in Figure 1.

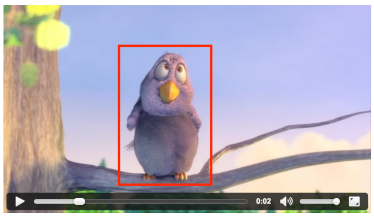

Fig. 1. Spatial-temporal fragment: `video.org#t=1,3&xywh=100,35,80,120`

Using Media Fragment URI standard for media section identification convinces of the easy of use and its seamless integration into well known Web infrastructure. Nevertheless the limitations often cause problems:

Inprecise spatial fragments: Spatial regions often cannot be sufficiently specified with rectangles. This fact may cause problems in calculating relations between fragments, e.g. if bounding boxes of spatial objects overlap, whereby the objects itself don't.

Lacking support for moving objects: Spatial regions in videos rarely stay on the same position during longer temporal sections (e.g. actors moving around within a scene). To sufficiently describe such scenarios many short spatial-temporal fragments have to be used, which leads to a big overhead in data transfer and recombination.

Missing styling support for fragment display: While common web component styling is well supported via Cascading Style Sheets (CSS), this support is currently lacking for Media Fragments. This limitation causes a major programming overhead for projects and raises the barrier for using the Media Fragment URI standard in productive web projects.

3 Media Fragment URI Extensions

In the following we present how Media Fragment URIs can be extended to various directions. A demo based on a basic implementation is available on `http://tkurz.github.io/media-fragment-uris-ideas/`.

Shape Extension

Currently Media Fragment URI's spatial dimension is limited to rectangular shapes (*xywh*). An extension to basic geometric shapes, like circles, ellipses, etc. would allow a more fine-grained fragment description. In [1] we recommended, inspired by SVG Basic Shape specification in [4] four shapes in addition or substitution to *xywh*:

Rectangle: `rect=x,y,w,h[,rx,ry]`
The integers denote x, y, width, height and (optionally) the x and y radius (rx and ry) of the ellipse used to round off the corners of the rectangle respectively.

Circle: `circle=x,y,r`
The integers denote x and y as the center of the circle and r as the radius.

Ellipse: `ellipse=cx,cy,rx,ry`
The integers denote cx, cy (the center of the ellipse) and rx, ry (the radius of the ellipse).

Polygon: `polygon=x1,y2*(,xn,yn)`
The value is composed by 2*n comma-separated integers (with $n \in N$). The integers denote x1, y1 as starting point and xn, yn as points on the polyline that borders the polygon; the polygon is closed.

The value is an optional format `pixel:` or `percent:`, the defaulting format is pixel. We give an example for an ellipse fragment in Figure 2, all the other shapes work accordingly.

Transformation Extension

Even with this shape extensions the identification of spatial fragments is limited. Additionally, with regard to further extensions for example animations, a proper representation of shape transformation and translation is lacking. We aim

to overcome this limitations by introducing four shape transformations:

Translate: `translate=x[,y]` The integers denote x for horizontal and y (optionally) for vertical translation.

Scale: `scale=x[,y]` The integers denote x for horizontal and y (optionally) for vertical scale.

Rotate: `rotate=a[,x,y]` The integers denote a as rotation angle and x,y as center of rotation. The default center is denoted by the center of the bounding box of the region to rotate.

Skew: `skew=x[,y]` The integers denote x for horizontal and y (optionally) for vertical skew.

Transformations in Media Fragment URIs are only considered if one and only one shape is defined. Transformations can be stacked. If a transformation type occurs more than once, only the first value is considered. Like for shapes, the value has an optional format `pixel:` or `percent:`, whereby the defaulting format is pixel. Figure 3 shows a transformed shape.

Animated Transformation Extension

The static shapes and transformations mainly focus on still images. But spatial fragments often needs to transform over time e.g. for videos or interactive charts. We introduce animated transformations as temporal extension to the static in order to satisfy this need.

Animated Translate: `aTranslate=d1,x1[,y1]*[;dn,xn[,yn]]`
The value is an optional format pixel: or percent: (defaulting to pixel) plus a semicolon-separated list of comma-separated numbers. The first number of each number set (d.) is defined as duration and may be defined in percent (for videos) or milliseconds (for images). The other numbers represent the translation as specified.

Animated Scale: `aScale=d1,x1[,y1]*[;dn,xn[,yn]]`
Analogous to animated translate.

Animated Rotate: `aRotate=d1,r1[,x1,y1]*[;dn,rn[,xn,yn]]`
Analogous to animated translate.

Animated Skew: `aSkew=d1,x1[,y1]*[;dn,xn[,yn]]`
Analogous to animated translate.

Animated Transformations in Media Fragment URIs are only considered if one and only one shape is defined. Animated transformations can be stacked. If an

Fig. 2. Shape Extension: `image.jpg#ellipse=percent:50,52,15,22`

Fig. 3. Transformation Extension: `image.jpg#rect=230,100,80,55&rotate=25`

animated transformation type occurs more than once, only the *first* value is considered. Figure 4 shows how a spatial fragment is animated over time in both scale and translation. In this case there is no transformation until 45% of the temporal fragment (3.5 seconds overall), in the next 10% of time the shape translates to south-west and scales to 70%. During the remaining time there is no transformation.

Fig. 4. Animated Transformation Extension: `video.mp4#ellipse=330,100,50,80&aTranslate=0.45,0,0;0.1,-50,50&aScale=0.45,1;0.1,0.7&t=0.5,4`

The current standard as well as the proposed extension still lack a proper formal description, which makes it hard to apply set operations like intersection, union, etc. to Media Fragments. In [6] we already worked out such a model, which can be a basis for further specification.

4 Related approaches

On `http://github.com/tomayac/dynamic-media-fragments` the author describes, how spatial Media Fragments *xywh* can be extended to temporal dynamics by stringing together quadruples, whereby each one identifies a rectangular shape. The shapes are equally distributed in time (represented by a temporal fragment or the whole video play time). The approach is aligned with CSS transitions and such fits smoothly into current browser animation implementations. To extend the approach from equal to fixed distribution, the author suggested to extend the quadruples to a micro syntax representing the time in percentage. Another interesting approach is described on `https://github.com/oaubert/mediafragment-prototype`. The author introduces a new fragment parameter *shape*, which represents the spatial dimension and utilizes SVG path definition as values. The main difference to our approach is the fact that shapes are not first class entities (defined by a name-value pair) but are values of one spatial dimension descriptor. For the temporal dynamic the author introduces a trajectory parameter with an SVG path value, which makes the defined shape follow the given path within a given temporal fragment. The author also suggest to extend both the shape and the trajectory values to basic SVG shapes.

5 Styling Media Fragments

To make Media Fragment URIs more attractive for Web designers and developers, an integration into common Web standards is essential. In this section we sketch how CSS can be adopted to Media Fragments for both spatial and temporal dimension. In our approach the fragment can be seen as an element, which

Fig. 5. Media Fragment `image.jpg#circle=175,55,40` styled with CSS extension

is contained within a layer on top the original media item (image or video). To access this element we introduce a pseudo selector `::fragment`. To allow access to the layer itself without the fragment (e.g. in order to set the opacity of the media item and not influence the styling of the fragment), we define a second pseudo selector `::fragment-outer`). In the example, we set the opacity of this outer fragment and a red border to the fragment itself:

```
img {
    background-color:white;
}
img::fragment {
    border: 1 px solid red;
}
img:fragment-outer {
    opacity: 0.5;
}
```

Figure 5 shows the result of the styling. A special case is the clipping of an fragment, which means that the fragment becomes the first class entity regarding display. This can be solved by setting the *display* attribute of the media item to *none* while keeping the fragment shown with *display: block*. The pseudo selector also allows to access and alter the fragment even programmatically e.g with Javascript. As the fragment is handled like a common HTML element it is possible to add sub-element like e.g. spans with description text.

6 Conclusion

In this paper we prosed extensions of the current Media Fragment URI specification regarding additional shapes, transformation and dynamic. Additionally we sketched an extension of the CSS standard to properly add style information to Media Fragments. The whole work is still in an early phase but can be seen as a starting point for discussion to present the opportunities of Media Fragment URIs to a broader community and trigger a process to lift the standard to a next level.

7 Acknowledgments

The paper is mainly inspired by the discussions at the WWW conference in 2015[3]. This paper is developed within MICO, a research project partially funded by the European Commission 7th FP (grant agreement no: 610480).

References

1. Patrick Aichroth, Johanna Björklund, Kai Schlegel, Thomas Kurz, and Thomas Köllmer. Specifications and Models for Cross-Media Extraction, Metadata Publishing, Querying and Recommendations - Final Version. Technical report, Media in Context - MICO, December 2015.
2. Werner Bailer, Chris Poppe, WonSuk Lee, Martin Höffernig, and Florian Stegmaier. Metadata API for media resources 1.0. W3C recommendation, W3C, March 2014. http://www.w3.org/TR/2014/REC-mediaont-api-1.0-20140313/.
3. T. Berners-Lee, R. Fielding, and R. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986, Network Working Group, January 2005.
4. Jon Ferraiolo. Scalable vector graphics (SVG) 1.0 specification. W3C recommendation, W3C, September 2001. http://www.w3.org/TR/2001/REC-SVG-20010904.
5. Thomas Kurz, Georg Güntner, Violeta Damjanovic, Sebastian Schaffert, and Manuel Fernandez. Semantic enhancement for media asset management systems. *Multimedia Tools and Applications*, pages 1–27, 2012. 10.1007/s11042-012-1197-7.
6. Thomas Kurz, Kai Schlegel, and Harald Kosch. Enabling access to Linked Media with SPARQL-MM. In *Proceedings of the 24nd international conference on World Wide Web (WWW2015) companion (LIME15)*, 2015.
7. Lyndon J. B. Nixon, Matthias Bauer, Cristian Bara, Thomas Kurz, and John Pereira. Connectme: Semantic tools for enriching online video with web content. In Steffen Lohmann and Tassilo Pellegrini, editors, *I-SEMANTICS (Posters & Demos)*, volume 932 of *CEUR Workshop Proceedings*, pages 55–62. CEUR-WS.org, 2012.
8. Robert Sanderson, Paolo Ciccarese, and Benjamin Young. Web annotation data model. Working draft, W3C Working Draft, October 2015.
9. Raphaël Troncy, Davy Van Deursen, Erik Mannens, and Silvia Pfeiffer. Media Fragments URI 1.0 (basic). W3C recommendation, W3C, September 2012.

[3] `https://lists.w3.org/Archives/Public/public-media-fragment/2015May/0003.html`