A Time Algebra For Editorial Systems

Retrieved from https://www.siggraph.org/learn/instructions-authors on June 5th, 2019

STEPHAN STEINBACH and NICK PORCINO, Pixar Animation Studios

this mentions precision loss, lets call that out later

By treating the spatial dimensions as normed vector spaces subject to manipulation through linear algebra, 3d computer graphics systems typically leverage hierarchical coordinate systems in order to provide better numerical precision, artistic control, and more. This principal of organization is implicit in a modern non-linear editing system, although it is slightly obscured in a metaphor of layered tracks and clips and a variety of heurstics to resolve an image from the tracks. We demonstrate that by applying the same principles of hierarchical affine spaces to the temporal dimension, it is possible to overcome a number of limitations common in the current state of the art non-linear editing systems. Analogously to the application of hierarchical coordinate transformations in 3d computer graphics, hierarchical temporal spaces and a new formulation of frame selection through a combination of temporal predicates and a frame selection algorithm enables these applications to avoid the use of heuristics like "mixdowns" to enable higher fidelity sampling and prevent precision loss in time information that typically occurs through a project life span. We also demonstrate that this new temporal algebra is also applicable to systems where presentation is dynamically recomposed in response to input from an observer or participant, such as in games, interactive media, and theme park attractions.

CCS Concepts: • **Computing methodologies** → **Rendering**; **Ray tracing**.

Additional Key Words and Phrases: ray tracing, global illumination, octrees, quadtrees

ACM Reference Format:
Stephan Steinbach and Nick Porcino. 2017. A Time Algebra For Editorial Systems: Retrieved from https://www.siggraph.org/learn/instructions-authors on June 5th, 2019. *ACM Trans. Graph.* 36, 4, Article 1 (July 2017), 7 pages. https://doi.org/http://dx.doi.org/10.1145/8888888.7777777

1 INTRODUCTION

Professional editing systems are typically built around a central constraint: all video is of the same dimensions and frame rate, and all audio is of the same sampling frequency. If one tries to add media of the wrong sampling rate, a transcode is required to cut the media in. This constraint makes sense in the context of the history of digital non linear editing (NLE) systems.

I think we want to note that it is not only the mathematical roots but also the design of the soft-

The basis for the mathematical roots of non-linear editing begins with its roots in film.

Rolls of film for images, and reels of magnetic tape for sound, were gathered at the end of a shooting day. To register the audio with the images, the audio was transferred to a so called “mag track”. If the source film was 35mm, with four perforations per frame, the audio was transferred to a a 35mm mag track, recorded at the appropriate speed such that each four perforations on the mag track corresponded to 1/24 of a second of sound.

For each take recorded to film and tape, a clapper was recorded; and a frame is marked on the film where the boards of the clapper first visually meet. The audio is then matched to the film so that the peak in the audio recording of the clapper matches the frame where the “sticks” meet. The perforations in the film and the audio track provide the increments forward and back that the mag-track can be synchronized by, and so the visual correspondence of the sound to the picture, on 4-perf 24 frames per second material is accurate to within 1/96 of a second.

The film frames had numbers embossed or otherwise imprinted on the edge of the film. These “edge codes” were the numeric indices of frames on a strip. When the synchronization of the film and mag-track were complete, the film strip’s edge codes were imprinted upon the mag-track. This affordance allowed editors to easily verify that sound and image were in sync during the editing process. These two reels were then set aside and labeled according to an editor’s preference into a collection of raw material, called “takes”, to be used later when the film was assembled.

Every subsequent print of a take would also photographically transfer the edge codes to the new print, so the correspondence between originating materials is in principle never lost.

Mechanical film editing systems, such as the Moviola, allowed an editor to physically cut printed takes, and join them together in a new order with gum, into a “cut” of a film. When two takes are meant to overlap via an effect such as a cross dissolve, a long diagonal cut, known as a “straight cut”, was made in the two segments and then the two pieces of the film would be joined along the diagonal cut. The process proceeds, with many different elements, such as visual effects, being prepared, possibly independently, and slowly worked into the “final cut”. Throughout this process, the take names, and edge codes, ensure that the source materials can be found, and utilized by the optical laboratory that combines all of the materials into an “answer print” which hopefully represents a final form of the film. Once satisfactory, a “release print” for distribution to theaters was prepared from the “answer print.”

not only a fixed sampling rate but an entire fixed coordinate system for both pieces

Throughout all of this meticulous and laborious process, the source materials are maintained in synchronization by virtue of a fixed sampling rate for all sources and products, and edge codes that dictate an exact correspondence between media. An edit decision list is a recipe to produce an answer print given a library of sources, edge code ranges, and effect types.

The introduction of video tape and the non-linear editing process was a revolution. Tapes were prepared from source materials, with time codes in the video serving the same role as film’s edge

Authors’ address: Stephan Steinbach, steinbach@pixar.com; Nick Porcino, nporcino@pixar.com, Pixar Animation Studios.

0730-0301/2017/7-ART1 $15.00
https://doi.org/http://dx.doi.org/10.1145/8888888.7777777

codes. The edit decision list no longer directed a series of photographic composites to produce a new strip of film, but rather queued portions of video tape to play from possibly several machines to another recording machine. EditDroid introduced the next level of sophistication in 1984, which was to replace video tapes with source material on laserdisc. Instead of waiting for tape to rewind, or play forward in real time, sources could be randomly accessed, and represented in rough form through computer interfaces. Throughout the 1990s the sophistication of playback and recording systems improved until the editing process converged on what might be considered an emulation of the earlier film editing process.

ref

This organic but meticulous evolution gave rise to the form of the modern NLE, which at its very heart, represents a bin of film strips, layered in rows that represent the order of an optical composite, indexed by fixed sample rate frame codes, and registered and synchronized to those frame codes by modern day sprocket holes.

despite all of the advances in playback and encodings, still represents its document model by way of traditional, analogue workflow

If this was the end of the story, there would be no need for further work. The classic format of an EDL would be sufficient to describe the composition of a film from source materials in a reliable way, irrespective of the non-linear editing system in use. As is easily anticipated, life is in fact not so easy. The relentless primacy of the 24 frames per second rate is an uneasy fit over modern technology. Film can be shot at speeds over 100 frames per second; video at rates that are a multiple of 30 or 25. North American broadcast video sources may be at NTSC rates, which are rational rates with a divisor of 1001; the well known rate of 29.97 is 30 * 1000 / 1001. Audio might be record at 44.1kHz, 48kHz or other rates, and the modern NLE allows the elements to be composed in time with an arbitrary degree of precision.

Contemporary editing systems resolve these rates by imposing workflow restrictions. For example, in Avid Media Composer, each track requires that media be of the same rate, and furthermore that there is an "Edit Rate" for the entire document, which describes the temporal resolution to which media is snapped when performing operations on co-incident in time media.

In order to mix media of different rates, editors are required to transcode media (or "render" it in the language of the application) so that it is embedded in the conforming rate. This creates a generational data loss – as is well established by sampling theory, while some rates are trivial to transform into others,

To carry editorial operations into the language of mathematics, an editor reduces and composes a large amount of source material for observation, in order that it may be presented as single resolved samples of that media arranged along a single monotonically increasing number line in playback time. Editors, however, do not work directly on that number line.

my sense is that somewhere in here is where we want to do the survey and talk about

It is technically possible to arrange the raw source material along such a number line, but it must first be trimmed and composed relative to other material. In practice editing systems present the working documents as complex hierarchical working structures that feature nesting, compositing, referencing and many other features familiar to users of digital content creation (DCC) systems. This paper aims to mathematically express the operations that editors use to transform their complex hierarchical working structure into that single number line result by specifying that time is a one dimensional normed vector space. This extends to systems with interactivity because human experience is perceptually linear over time; the final experience is still perceived as a sequence of monotonic moments. As such, media can also include data recorded during the performance or playback such as button presses on a joystick or the state of train cars in an amusement park ride.

The presentation of media involves a series of coordinate transformations from the coordinate spaces inherent to the media, through coordinate spaces present in the working document, to the final perceptual coordinate space in which the media is observed. By carefully examining the transformations we develop an algebra that consistently and completely describes how to compute information about both the working document and its presentation at any given moment.

Editors consider a number of questions during their work, such as these:

- "What frame of media is playing during the nth frame in the top timeline?"
- "Which frames need to be rendered overnight for today's cut?"
- "Given a clip in a video track, which segments of audio correspond to it?"

These questions appear simple on the surface but hide a fundamental complexity that stems from the way editorial systems have evolved over the history of cinematic and television editing. The variety of sampled media, compositing and timing operations, and display systems have evolved continuously as technology and creative expression have advanced. As non-linear editing systems have evolved along with these advances, the inevitable result is that many heuristics and ad hoc exceptions have been introduced along the way, obscuring the operations that actually occur in those systems.

An illustration of this complexity is the composition of media sampled at different rates, passed through timing effects and trims. Even a simple document with one video track and one audio track will have an audio sampling rate at tens to hundreds of kilohertz, and the video sampling rate will be in the tens of hertz; neither will be a fundamental multiple of the other. Editors may layer compositing and timing effects on these tracks, and will have the expectation that the audio and video remain in sync.

OpenTimelineIO [Ope 2020] is an open source library meant to enable interchange between a variety of editorial systems. During its implementation it became painfully obvious that there was a lack of a rigorous definition of the operation of these systems and their operations. A survey of the literature reveals many of the necessary components to solve these issues, but no formal description shows unifying those components into a coherent framework. A survey of existing editorial software revealed a variety of opinionated implementations of various interpretations and many ad hoc accommodations for weakly defined practices and workflow preferences. There is clearly a dearth of literature describing how to engineer systems that approach these problems in mathematically formal ways. This lack of a recognized and commonly shared formalism makes it difficult to robustly implement systems for editorial operations that function consistently, and without ambiguity, between editorial systems. This prompted an investigation and derivation of a formalism to rigorously anchor editorial operations in the realm of algebra.

2 FOUNDATIONS

According to Euclid, a point is an indivisible location “which has no part” - no width, length, or breadth. Two points and the span between them describe a line, and relations such as the triangle inequality teach how to measure space. A Euclidean geometrical space is a normed vector space, which enables an algebraic set of possible operations on this metric space.

Useful algebras are not restricted to spatial transformations. Porter & Duff [Porter and Duff 1984] demonstrate an algebra on pixels that allows the principled transformation of one image into another through a series of mathematically reducible operations. This insight on the use of an algebra to describe visual composition inspires the idea of an algebra for temporal operations. We develop an equivalence of time as a normed vector space enabling metric operations on time analogous to those operations in Euclidean space. The operations familiar from three dimensional graphics, such as the transformations between hierarchies of coordinate spaces are also applicable in the domain of time. Additionally, by introducing the idea of a sampling function as a discrete entity in editorial systems, we can reason about operations that are performed on samples, as distinct from operations that are performed on time.

We present a time coordinate matrix for affine transformation, encoding an offset and a scale, allowing time coordinate concatenation analogously to how spatial coordinate spaces are composed in three dimensional computer graphics. We also present a framework for encoding any transformation so that a media fragment can be hierarchically transformed through a series of spatial and temporal operations invariantly with respect to observation to yield a single observed perceptual moment.

This follows the derivations for homogeneous coordinate transformation systems for geometric spaces in computer graphics as described by Roberts [Roberts 1963].

Furthermore, Allen’s Interval Algebra [Allen 1983] works out the possible predicates in this space, and introducing those predicates here allows us to define a time algebra for editing operations.

3 (

Non-linear Editing Systems)

PLACEHOLDER

Avid Media Composer workflows involve rate converted media at import time to conform them to the project settings (http://resources.avid.com/SupportFiles/attach/Media[...]2020.6.pdf, p.200).$Audioisei$

Media Composer requires that the framerate be set at project creation time to match media and delivery requirements (p33). The framerate of sequences within the project cannot be changed after initial creation.

Media Composer conforms clips to the project settings using a so-called timewarp adapter. One adapter may be applied per clip. The adapter is controlled by a retiming curve, which may have discontinuities in it. The transformation’s origin is at an anchor frame, which designates a point in the clip that corresponds to a frame in the project.

After Effects has similar concepts, calling the anchor frame, the "hold in place" frame. Media Composer does not alter the clip in place, but After Effects does (TODO verify). (http://resources.avid.com/SupportFiles
$renderstheclipaccordingtooneofthesealgorithms$:

Duplicated Field • Both Fields • Interpolated Field and VTR-Style (approximately the same) • Blended Interpolated and Blended VTR (approximately the same) • FluidMotion Draft • FluidMotion

Field interpolation refers to interlaced video frames. FluidMotion refers to an image process algorithm that synthesizes new frames by analyzing the contents of neighboring frames. Other resampling options are available, including "strobing" which simply holds a frame for some duration and does not apply interpolation to the image. VTR algorithms take alternating scan lines from different frames and blend them in a manner meant to emulate tape based video frame interpolation.

4 DOMAIN

The creation and presentation of media involves a complex set of operations between different temporal domains, as shown in Figure 1. This set of operations encapsulates the temporal transformations media undergoes, from wall clock time where actors do their work, to various internal temporal spaces, and back to the space where the system can be viewed by observers.

Diegetic time is the time in which a story occurs, it is the time the characters experience. Diegetic time may also be called action time, or story time. It is the time footsteps are heard synchronized with the image of a character walking, or the time in a simulation for 3d computer graphics. Diegetic time is hierarchical, the piece of music played by an orchestra filmed for a movie clip has its own diegetic time, itself embedded in the film clip’s action time

A scene is transformed from exogenous time to diegetic time via a capture process to produce media.

Media itself is indexed, for example by frames. A sampling function exists to convert action time into an index to retrieve a particular frame.

Editorial time is liminal, it is the time outside of time. An editor has simultaneous access to any point in diegetic time, and can refer back to world time with regards to source media. Scene graphs as used in computer graphics also exist in liminal editorial time, as the scenegraph may represent any time, without specifying a present time.

System time is the endogenous time inherent to the system. For the purposes of our algebra, two important endogenous system times are the sampling time, and the presentation time.

Sampling and rendering at the system time provide a view into the liminal domain.

Projection and Display take elements from system time back into world time where a viewer may observe them.

There are two interactive cycles in this system. Input devices may control a non-linear editor, or an engine, such as a game engine, which can produce new liminal data for sampling and rendering back into system time, and then into world time. The second cycle may render an element modified in system time all the way back into story time for re-composition into editorial time.

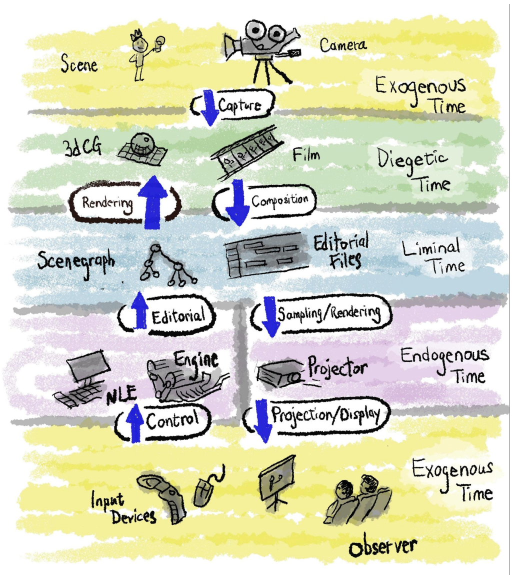

Fig. 1. A temporal cosmology

In order to construct an algebra that describes these transformations we must first define the elements that can be operated upon.

(1) Time Points, which are unique temporal coordinates
(2) Time Intervals, which have a time point, and a duration
(3) Samplings, which are sets of time points
(4) Media, which are elements of composition, indexable by a time point

Given these elements, we may define several classes of operations:

(1) Operations over time. Functions that take time as an argument and return some new time. Given some time t and function F: $t' = F(t)$
(2) Operations over intervals of time. Functions that take time intervals and return a new interval. Given a time coordinate c, duration d, a time interval c, d and function F: $c, d' = F(c, d)$
(3) Sampling Functions. Functions that take time intervals and return a sampling. Given a time interval i and some function S: $s = S(i)$
(4) Operations on Samplings, or resampling. Functions that take time samples and return a new time sampling. Given one or more time samplings $s1...s_n$, and a function F: $s' = F(s1...s_n)$
(5) Media Functions. functions that a time sampling and media as arguments, return media. Given a time sampling s, some media $m1...m_n$, and a function M: $m' = M(s, m1 \ldots m_n)$
(6) Operations on media. Functions that take media and return media exist, but are out of the scope of this paper.

Using these operations it is possible work all the way back from the exogeneous time of an observer to the exogeneous time when a scene was captured, to select appropriate samples from media in order to render a scene for presentation, to provide a means to translate input from a human interface device to modifications of a scenegraph in a game engine, and many other operations besides.

5 ALGEBRA

5.1 Normed Vector Space

We define a time continuum as a normed vector space in which time may be measured, as in Euclidean geometric space.

5.2 Time Point

A time point is a location on the time continuum with no extent.

5.3 Time Coordinate System

Within this normed vector space we can define a coordinate system with a unit basis. The algebra recognizes some special values inherent to an open ended one dimensional number line. These special values are positive infinity, negative infinity, and not a time.

5.4 Time Coordinate System Transformation

A time coordinate system, tcs, is transformable to any other time coordinate system via function composition. As an example, if there is a function named f that transforms one space to another, and a second function g that also transforms one space to another in a different way, then the functions can be composed into a new function F by composing $F = g(f(tcs))$

There exists a subset of transformation functions that can be represented by affine transformation matrices. We define the affine transformations on time coordinates as scaling and offset.

This transformation can be represented with a homogeneous coordinate transformation matrix of the form:

$$H = \begin{bmatrix} scale & offset \\ 0 & 1 \end{bmatrix} \tag{1}$$

The use of this matrix means that a time coordinate can be transformed from an enclosing space to the local space, and that given the inverse of the matrix a local time coordinate can be transformed into the enclosing space. These matrices can be concatenated so that a series of operations can be expressed in a single composed final matrix, while still retaining the property of invertibility. They can also be decomposed and recomposed as necessary.

$$H = \begin{bmatrix} scale & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & translation \\ 0 & 1 \end{bmatrix} \tag{2}$$

$$t' = \begin{bmatrix} t & 1 \end{bmatrix} H \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3}$$

$$HH^{-1} = Identity \tag{4}$$

There is a special value I, which when multiplied by a time yields the same time; this is the multiplicative identity. Similarly, there is a special value I_o, which when added to a time yields the same time; this is the additive identity.

There are other transformations that are non-affine. They take a tcs and yield a tcs, but these non-affine transformations cannot be concatenated. If a hierarchy of these algebraic operations is composed into a graph, this consideration becomes important as the non-affine operations impose limits on possible equivalent permutations and invertibility of the graph and its subgraphs.

A useful property of an algebra in a normed vector space is that distribution, association, and commutation can be used to simplify and meaningfully rewrite expressions.

5.5 Time Interval

A Time Interval is a tuple comprised of a pair of time points over the time continuum in which a signal exists, $t0$ and $t1$, and the clusivity for each time point. We write this as: $(t0, t1)$, $(t0, t1]$, $[t0, t1)$, or $[t0, t1]$ to signal the clusivity of the points.

The algebra also recognizes the special intervals of:

- Always: ranging from negative infinity to positive infinity inclusive on both ends
- Never: exclusive on both ends with an identical start and end points.

Time intervals can be uniformly offset by adding the same value to each endpoint. They can be scaled about a pivot by negatively offsetting the interval by the offset, scaling the end point, and then offsetting positively by the pivot value. These operations do not alter the clusivity of the points.

Time Intervals can furthermore be collected into sets. Set operations on time intervals yield new sets.

Finally, for predicates on time intervals, we refer to Allen's interval algebra [Allen 1983]. The predicates on two intervals are:

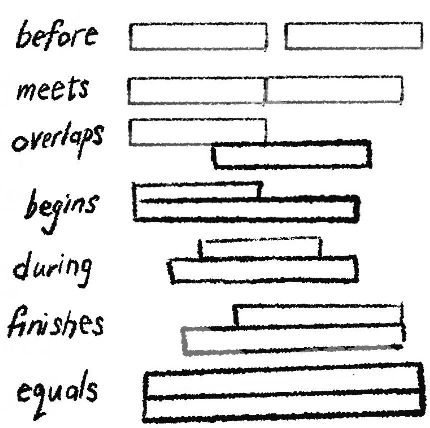

Fig. 2. Allen's Interval Algebra

As in Allen, complex expressions can be composed from these elementary predicates.

We extend the interval algebra to consider the clusivity of points. For example, in the meets relation, if the end of one interval and the beginning of the other are both inclusive, and the values of the adjoining time coordinates are equal, then the intervals do not meet, rather, they overlap. If however either or both of the adjoining time coordinates are exclusive and the adjoining time coordinates are equal then the intervals do meet, and do not overlap. Similar reasoning with clusivity applies to all the predicates.

5.6 Time Sampling

Certain media sources operate in the continuous domain, such as simulations and evaluation curves. These sources must interoperate in an editorial system with other sources that exist only discretely in time.

In practical scenarios, media is discretely sampled in time; a camera observes a scene in a series of peeks gated by a shutter. Following Euclid's observation that spatial intervals bound points, so time intervals bound time points. This equivalence allows working with clips of sampled media in the interval algebra, as if the media were continuous. A time sampling function couples the continuous and discrete time domains.

The formal separation of time sampling from the time interval and the continuous time domain enables us to reason about the treatment of discrete samples of the media, after the encompassing transformations of scaling and translation have been dealt with. The notion of separating operations that manipulate the time domain from operations that manipulate sample times allows precise reasoning about the placement in time of individual elements through every level of the time coordinate system hierarchy.

5.7 Cyclical Sampling Functions

Cyclical and regular sampling functions are a special subclass of sampling functions used to transform a continuous time coordinate into a frame rate based set of sample times. A common example of a regular sampling function would be a function that requests time samples at the 24hz rate used for most cinema presentations.

The most common kind of sampling function for a cyclical sampling rate is a rounding operation. Some rounding strategies include:

- Round time down to the preceding frame time
- Round time up to the next frame time
- Synthesize a new sample at a requested time by blending neighboring samples

This algebra does not dictate a strategy for rounding, rather it provides system designers with a language for describing the function they choose for transformations from a continuous time coordinate into a sample index.

The common practice of using modulus to encode cyclical coordinates follows the definition of a homogeneous projection of the time. If there is a scale in the homogeneous coordinates transform matrix of a cyclic sampling function, the projection would be a division by the scale such that the cycled portion maps canonically over the domain $(0, 1)$. This is useful for performing cyclical transformations like framerate pull downs without resorting to ad hoc approximations or algorithms, and for other operations such as calculating intersections between syncopated patterns.

It is important to note that the sampling operation is by definition non-affine, and non-invertible. It is also likely to be lossy.

5.8 Use of Sampling Rates in Continuous Domain Coordinates

The distinction between the discrete sample domain and the continuous time domain helps to disambiguate a common notation in editorial systems, which refers even to continuous time in a cyclical sampled notation. For example, depending on the implementation, "frame 94 in 24hz" can refer to:

- The continuous interval starting at continuous time 94*24s and lasting until (but not including) 95*24s
- The time point exactly at 94*24s
- The samples associated with the above continuous coordinate or interval

Our hope is that by providing distinct language for referring to time points, intervals and samples the intent of the system can be unambiguously described.

5.9 Sampling Function Operations

To illustrate how the sampling function interacts with editorial operations, we present a simple example. Given media composed of six samples which play in order, each for 1/24th of a second, having a total diegetic duration of 1/4 s. There are several operations that might be meant when someone says "speed up this media":

- Show all samples for 1/48 of a second, the endogenous time is now 1/8 of a second.
- Show every second sample for 1/24 of a second. The endogenous time is again, 1/8 of a second. This is a modification of the sampling function itself, not a modification of the endogenous time.
- Synthesize new samples by interpolating between existing samples, each sample (old and new) is played for 1/48th of a second. The endogenous time is the same as the diegetic time of 1/4 seconds. This is also a modification of the sampling function and not an operation on the endogenous time.

5.10 Sampling Function for Continuous Signals

For continuous signals, the sampling function is trivial because a continuous signal always exists wherever it is observed.

6 EXAMPLES

To illustrate the application of this algebra to a simple but useful real world use case, we propose the example of a viewer program in which two media clips play one after the other with no overlap or gap between them.

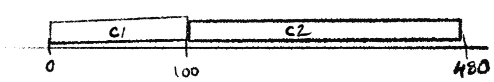

Fig. 3. Example 1

6.1 Named Coordinate Systems

Given clip C1 which has duration 100 and clip C2 which has duration 380, we can define some practical working coordinate systems and transformations:

- World Time: this is the time of the entire track, running from 0 to the total duration of C1 and C2, 480
- Normalized World Time: this is the normalized form of World Time, which maps World Time to a $[0, 1)$ interval.
- Local Time: the coordinate system defined within each of C1 and C2, which is defined by the interval $[0, 100)$ and $[0, 380)$, respectively. The local-to-world transformation for C1 is the identity operation and for C2 it is a translation by the duration of C1, 100.
- Normalized Local Time: maps the intervals of Local time to a $[0, 1)$ interval.
- Media Time: because the clip may be itself performing a clipping operation of some kind on its referenced media, there is an additional coordinate system that is the intrinsic recorded (diegetic) coordinate system of the media.

These coordinate systems all take place in the liminal space.

6.2 Determining the Current Media Sample

If we play this timeline, and pause at a particular frame, we can use the algebra to determine which frame of the clip that is playing is currently visible.

Given time t_w in world time, and a clip C1 of duration D1 and clip C2 of duration D2, the time in the clip's local coordinate space, t_l, can be computed as:

$$t_w(t_l) = \begin{cases} t_{l_{c1}} = t_w - t_{start_{c1}}, & \text{if } t_w <= D1 \\ t_{l_{c2}} = t_w - D1 - t_{start_{c2}}, & \text{if } t_w > D1 \end{cases} \tag{5}$$

Therefore, given a media sampling function, $S(t) = floor(t/24)$, by algebraic expansion of equation 5 we can compute the samples that correspond to t_w, $S(t_w)$:

$$S(t_w) = floor(t_w(t_l)) \tag{6}$$

$$S(t_w) = \begin{cases} floor((t_w - t_{start_{c1}})/24), & \text{if } t_w <= D1 \\ floor((t_w - D1 - t_{start_{c2}})/24), & \text{if } t_w > D1 \end{cases} \tag{7}$$

6.3 Clipping Media

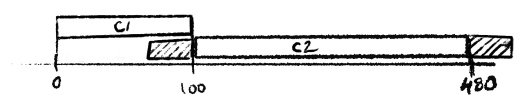

Fig. 4. Example 2

The previous example is extended to include a clipping operation that C2 performs on its media space. In this case, the clip space is defined by the interval $[0, 380)$, while the media space is defined by the interval $[0, 480)$ with the transformation media-to-clip being:

The media has an intrinsic, diegetic coordinate system which is used by the sampling function to index and reference samples of the media. The media also has an exogenous coordinate system which relates the media to some external coordinate system, such as the real world time when the media was produced.

In this case, the liminal coordinate system is expressed in the normalized space of the timeline, the origin to some end point). To transform a cursor from the normalized diegetic space back to the liminal space of the timeline, we construct the matrix:

$$H = \begin{bmatrix} s1 & -in \\ 0 & 1 \end{bmatrix} \tag{8}$$

Where s is the un-normalized trimmed duration of C2.
Expanded algebraically:

$$t_w = t_w + In \tag{9}$$

The rest of the calculation is other equations are then the same as the previous example.

6.4 Overlapping Media

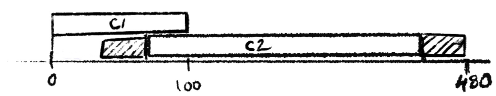

Fig. 5. Example 3

Given two overlapping clips, C1 and C2, the samples to be passed to a media compositing function, such as an "over" operation, can be computed by transforming the t_w defined in world space into the clip space for each clip and passing the clip space coordinate to the respective sampling function for each clip. For an exploration of these compositing functions we refer the reader to Porter and Duff [Porter and Duff 1984].

The media function $M(s)$ is passed a sampling:

$$M(s) = M(S(t_w)) \tag{10}$$

The sampling is defined as:

$$S(t_w) = \begin{cases} floor((t_w + In)/24), & \text{if } t_w <= D1 \\ floor((t_w - In)/24), & \text{if } t_w > D1 \&\& t_w < out \end{cases} \tag{11}$$

7 APPLICATIONS

Beyond the simple examples presented in the previous section, this algebra can be applied to a number of adjacent problems that link editorial data with other kinds of data in digital content creation, including:

- A DCC that mixes the notion of shots and clips with more traditional 3d DCC system ideas like animation clips and constraints
- Translating data between different editorial packages, using an intermediate representation
- Integrating show timings with sensor input in theme park attractions
- Pipeline integration software that helps artists track data across sequences
- Describing existing systems more precisely, so that users understand in which coordinate system data is intended to be input and how that data is transformed.

8 CONCLUSIONS

We presented nomenclature for describing the different temporal contexts of editing operations, elements and operations on time, time intervals, sampling, and media, and developed an algebra and predicate for reasoning about the composition of media in and across temporal contexts. These tools can be used to specify and design robust and inspectable editorial systems, and provide a mathematical language to describe existing systems. The open source project OpenTimelineIO implements the transformation matrices directly in its opentime module, and uses the ideas from this paper in its documentation to describe its functionality.

Future work will explore concepts of intermediate representation, temporal constraint solving, and the exploration of temporal operators.

We expect further developments and advancements in temporal reasoning to contribute to both streamlining and optimizing editorial workflows, robust interchange between tools, and to open up powerful new expressive tools for working with time.

REFERENCES

2020. *OpenTimelineIO - Open Source API and interchange format for editorial timeline information.* Retrieved Jan 22, 2020 from http://opentimeline.io

James F. Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Commun. ACM* 26, 11 (Nov. 1983), 832–843. https://doi.org/10.1145/182.358434

Thomas Porter and Tom Duff. 1984. Compositing Digital Images. *SIGGRAPH Comput. Graph.* 18, 3 (Jan. 1984), 253–259. https://doi.org/10.1145/964965.808606

Lawrence Roberts. 1963. *Machine Perception of Three-Dimensional Solids.*