Bulletin of Electrical Engineering and Informatics
Vol. 12, No. 3, June 2023, pp. 1657~1665
ISSN: 2302-9285, DOI: 10.11591/eei.v12i3.4894

Auto-correction of multiple spatial conflicts in multimedia authoring tools

Marvin Chandra Wijaya[1,2], Zulisman Maksom[1], Muhammad Haziq Lim Abdullah[1]
[1]Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
[2]Department of Computer Engineering, Maranatha Christian University, Bandung, Indonesia

Article Info

Article history:

Received Sep 27, 2022
Revised Nov 5, 2022
Accepted Dec 17, 2022

Keywords:

Authoring tools
Auto-correction
Media objects
Multimedia
Verification

ABSTRACT

Multimedia authoring tool serves to facilitate an author to create multimedia presentations. Multimedia authoring tool will convert the multimedia presentation into a document. Mobile devices with different screen sizes that can cause visual media to overlap are called Spatio-temporal conflict. The process in the multimedia authoring tool involves a Spatio-temporal verification process to detect Spatio-temporal conflicts. This study proposes a method for correcting the conflict using a conflict region detection process, Spatio-temporal verification, and auto-correction. The auto-correction method uses two stages of area relocation for vertical conflict and horizontal conflict. The two stages are repeated for the ability to correct up to four conflicting regions. Visual media objects such as overlapping images and videos were successfully separated into non-overlapping media objects. The proposed method succeeded in separating the four media objects that were previously overlapping.

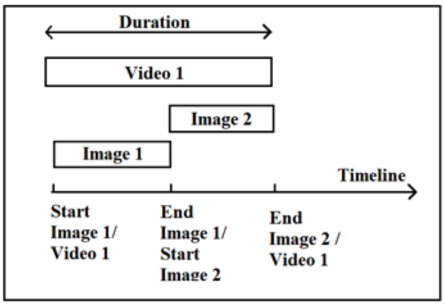
Corresponding Author:

Marvin Chandra Wijaya
Department of Computer Engineering, Maranatha Christian University
Jl. Suria Sumantri 65, Bandung 40164, Indonesia
Email: p031910033@student.utem.edu.my

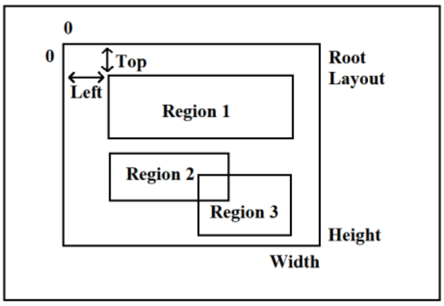
1. INTRODUCTION

Currently, multimedia presentations have been widely used in various fields [1]–[5]. A multimedia authoring tool is useful to make it easier for an author to create a multimedia presentation. A multimedia authoring tool helps an author produce multimedia documents from various media objects inserted by an author [6]. The media objects consist of two kinds: visual media objects and non-visual media objects [7]. Visual media objects such as text, images, animations, videos, and others [8], [9]. Non-visual media objects such as songs, music, sounds, and others [10]. The author uses the functions contained in the multimedia authoring tool to assist in the creation of multimedia documents correctly. The functions contained in the multimedia authoring tool are inserting various media objects, preparing root layouts, preparing regions for visual media sites, cutting copy and pasting on media objects, verifying time computations, verifying spatial-temporal, and others [11]. Two essential parameters in multimedia authoring are timing/temporal, as shown in Figure 1(a), and the position/spatial of each media object, as shown in Figure 1(b). Each media object must have three properties: start time, end time, and duration in a media presentation [12]. These three properties are attached to each media object regardless the media object will be played once or more [13]. A multimedia presentation must have a root layout with a specific width and height.

The root layout is usually divided into several regions of different sizes. A region can have a starting location (top and left) that can be the same or different from other regions. After the visual media objects in the temporal layout are inserted in a certain region, it will become a multimedia presentation [14]. After the process is complete, it will be seen that there is a possibility of an error which is called a spatial-temporal conflict [15]. The conflict can be verified using Spatio-temporal verification and time computation but rarely fixes the error.



(a)



(b)

Figure 1. Two essential multimedia layouts (a) timeline/temporal layout and (b) location/spatial layout

Multimedia authoring tool is an application that began to be developed at the beginning of the 21st century [16]. The study of multimedia authoring tools begins with research on the basic features for managing media objects into a multimedia document [17]. A multimedia authoring tool has three basic processes to produce a multimedia document. The three processes are called the Kernel mechanism: editing, modeling, and verification.

The editing process is the initial process in making multimedia presentations that are very dependent on the user interface on multimedia authoring tools. The editing process is the initial process in making multimedia presentations that are very dependent on the user interface on multimedia authoring tools [18]. Ease in the editing process is the first thing that an author pays attention to, various ways have been attempted and researched to increase the ease of making multimedia presentations [19]. One of them is the use of what you see is what you get (WYSIWYG), a must in a multimedia authoring tool [20]. The system in making multimedia presentations is important in facilitating interaction with the author, such as using a collaborative system in a multimedia presentation. A correct and more advanced control needs to be implemented in a multimedia authoring tool [21].

Expressivity is the ease of inserting media objects and processing the media objects. Reusing a media object for use in a multimedia presentation is also a feature of expressivity [22]. The important thing in the multimedia authoring tool is the existence of temporal formatting. The presence of spatial layout and temporal layout in the user interface is a must. Temporal formatting can be used as a pre-verification for the initial checking of spatial-temporal conflicts. The verification can use a system called time computation, which is applied to the temporal layout of a multimedia authoring tool [23]. The formatting process on the temporal layout is a continuous consistent check that works in the background when the author is still editing the editing process [24].

Verification results must be informed to the author using the user interface contained in the multimedia authoring tool. The information provided must be complete enough, such as the type of error, location, and others. Information to users can be placed in the message zone [2]. The information provided is as detailed as possible, the information is given in several tabs. This can make it easier for an author to improve multimedia presentations [25]. The spatial representation referred to in the study entitled "Adaptation Spatio-temporal et hypermedia de documents multimedia" has eight conditions, as shown in Figure 2. Fatma *et al*. [25], the spatial layout was verified using the Spatio-temporal inconsistencies verification algorithm (SIVA). Spatio-temporal can be used to check conflicts between regions [26].

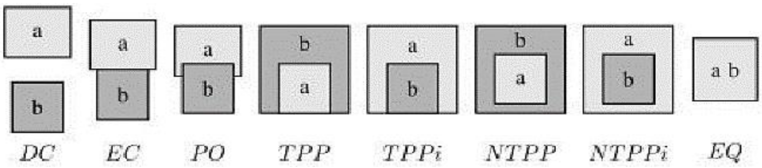


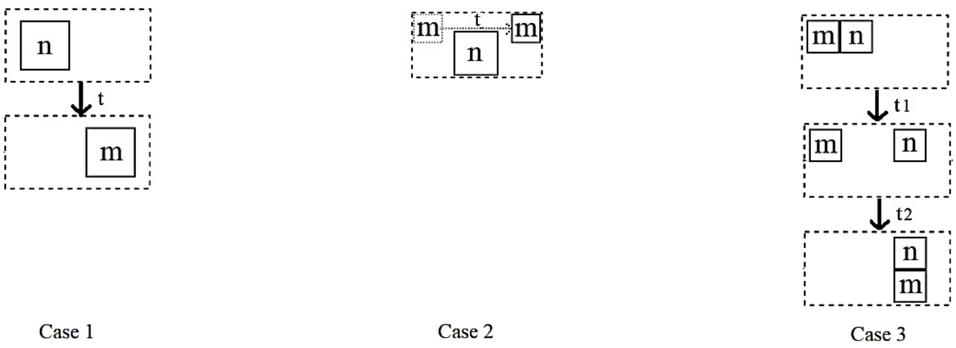
Figure 2. Spatial representation by sébastien laborie

$$Temp\ verification = ((start\ n\ \geq start\ m) \wedge (start\ n \leq (start\ m + dur\ m))) \vee ((start\ m\ \geq start\ n\ ) \wedge (start\ m \leq (start\ n + dur\ n))) \quad (1)$$

$$Spatial\ verification = (left\ m \leq (left\ n + height\ n)) \wedge ((top\ n - width\ n) \leq (top\ m - width\ m)) \wedge ((top\ m - width\ m)\ \leq top\ n) \quad (2)$$

According to Santos *et al.* [27], conducted Spatio-temporal validation research on multimedia documents to find and correct errors before multimedia document deployment. The study groups the possible Spatio-temporal conflict into three cases. The first case is if media object n is presented before media object m, as shown in Figure 3(a).

In the second case, it occurs if the media object n has a fixed position and the media object m has a moving position, as shown in Figure 3(b). The spatial-temporal layout validation must check whether there is overlap at a certain time. Case 3 is a more complex case than case 2, as shown in Figure 3(c). In the third case, there are two shifts in the media object; at t1 there is a shift in the object media n and at t2 there is a shift in the object media m. In this study, the size and location of overlapping media objects were changed at certain time conditions. This system checks every region that is active at a certain time. These regions will be validated whether there is overlapping, if there is overlapping, scaling and position shifting will be carried out to avoid overlapping, as shown in Figure 4.



(a) (b) (c)

Figure 3. Spatio-temporal validation cases (a) case 1, (b) case 2, and (c) case 3

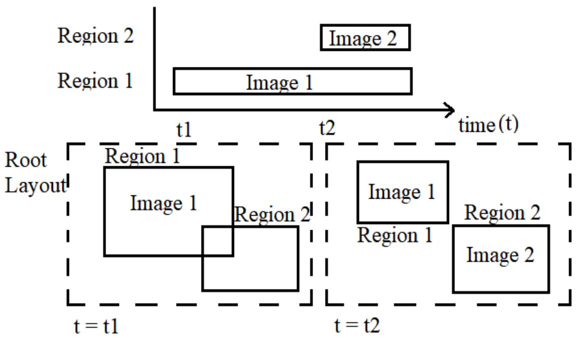


Figure 4. Spatio-temporal conflict corrections

Bouyakoub and Belkhir [28], in 2012 conducted a study to find a solution to the Spatio-temporal problem of conflict in SMIL builder V2. In this study, the spatial conflicts consisted of 81 types of spatial conflicts that could occur. Experiments were carried out in this study by providing overlapping regions. Region 1 and region 2 are overlapping areas, while region 3 does not overlap at all. First, the system verifies overlapping regions, in the experiment it will detect region 1 and region 2 overlapping. Second, the system determines that region 2 is the problem and overrides other regions. Third, the system will assign an area for region 1 and region 3 to calculate the empty area needed for region 2. Fourth, if there is an empty area for an overlapping region, then region 2 will be assigned to one part of the area that is still vacant, as shown in Figure 5.

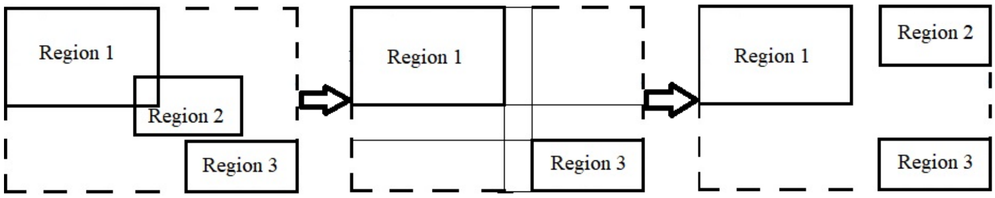


Figure 5. Overlapping corrections

2. METHOD

The process proposed in this study is divided into several stages: overlapping detection, Spatio-temporal verification, and auto-correction, as shown in Figure 6. The first process in spatial auto-correction is initiated by detecting overlapping regions. Algorithm 1 is an algorithm to detect overlapping regions.

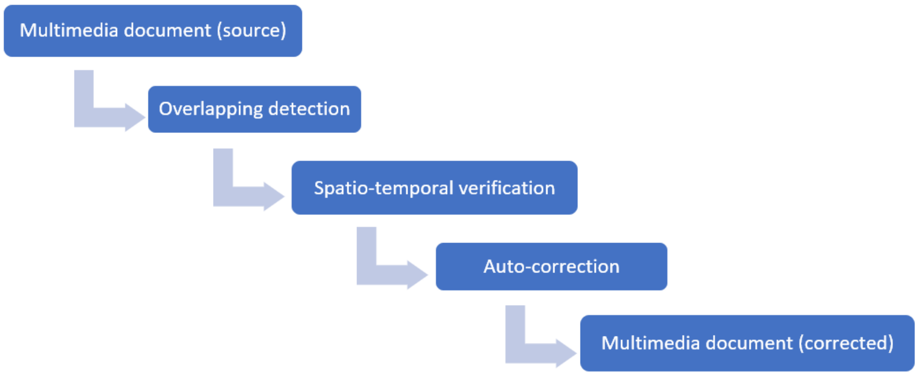


Figure 6. Auto-correction method

```
Algorithm 1 Multiple Overlapping Detection
n ← number of regions
k ← 0
for(i=1; i≤n ; i++)
    For(j=1; j<n; j++)
        if(((region(j,left) > region(i,left)) && (region(j,left) <
        region(i,left)+region(i,width))) || ((region(j,left) + region(j,width) >
        region(i,left)) && (region(j,left)+region(j,width) < region(i,left)
        +region(i,width)))
        &&
        (region(j,top) > region(i,top)) && region(j,top) <
        region(i,top)+region(i,height))) || ((region(j,top)+ region(j,height) >
        region(i,top)) && (region(j,top)+region(j,height) < region(i,top)
        +region(i,height))))
            k++
            overlap(k,1) ← i
            overlap(k,2) ← j
        end if
    end for
end for
```

The algorithm can detect all overlaps that may occur in each region in a root layout. The overlapping formula in the algorithm uses a formula to detect a collision between two objects. When several regions overlap, it does not mean that the region needs to be relocated. The combination of temporal and spatial information is called Spatio-temporal, determining the actual conflict between regions [26]. The next process is to perform Spatio-temporal verification, which serves to find out whether overlapping regions display media objects at the same time. If it turns out that there are no media objects that are displayed simultaneously in the overlapping region, then the region does not need to be relocated. However, if media objects are displayed simultaneously in overlapping regions, one of those regions needs to be relocated. Algorithm 2 is an algorithm to verify conflict two media objects in different regions.

Algorithm 2: Spatio-temporal verification

```
m ← number of media objects
for(i=1; i≤m; i++)
    media_object(i,end,region) ← media_object(i,begin,region) +
    media_object(i,duration,region)
end for
for(i=1;i≤m; i++)
     for(j=i;j≤m-1; j++)
        if(media_object(i,begin,region) > (media_object(j,begin,region)
            t ← media_object(i,begin,region)
            media_object(i,begin,region) ← media_object(j,begin,region)
            media_object(j,begin,region) ← t
        end if
     end for
end for
c ← 0
for(i=1; i≤n; i++)
    x ← overlap(k,1)
    y ← overlap(k,2)
    if ((media_objek(i,end,x) > media_objek(i,begin,y) && media_objek(i,end,x) <
    media_objek(i,end,y)) || (media_objek(i,end,y) > media_objek(i,begin,x) &&
    media_objek(i,end,y) < media_objek(i,end,x)))
        c++
        conflict(c,1) ← x
        conflict(c,2) ← y
    end if
end for
```

Every occurrence of a spatial conflict that has been detected needs to be categorized into several possibilities. Spatial conflicts are divided into several categories as follows. The first category is the horizontal conflict between several regions, as shown in Figure 7(a). The second category is the occurrence of vertical conflicts, as shown in Figure 7(b). The process for correcting conflicts by relocating regions can be processed using Algorithm 3.

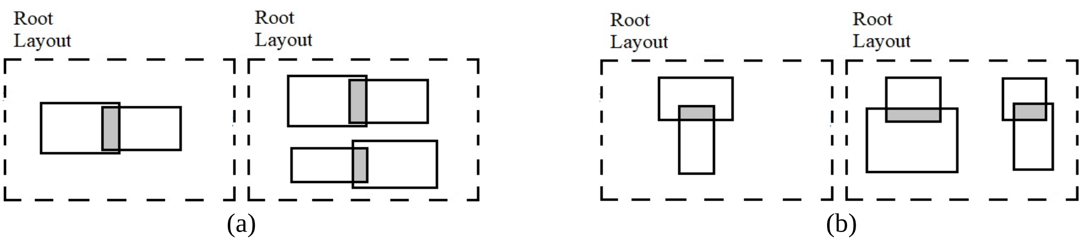


Figure 7. Two types of spatial conflicts (a) horizontal conflicts and (b) vertical conflicts

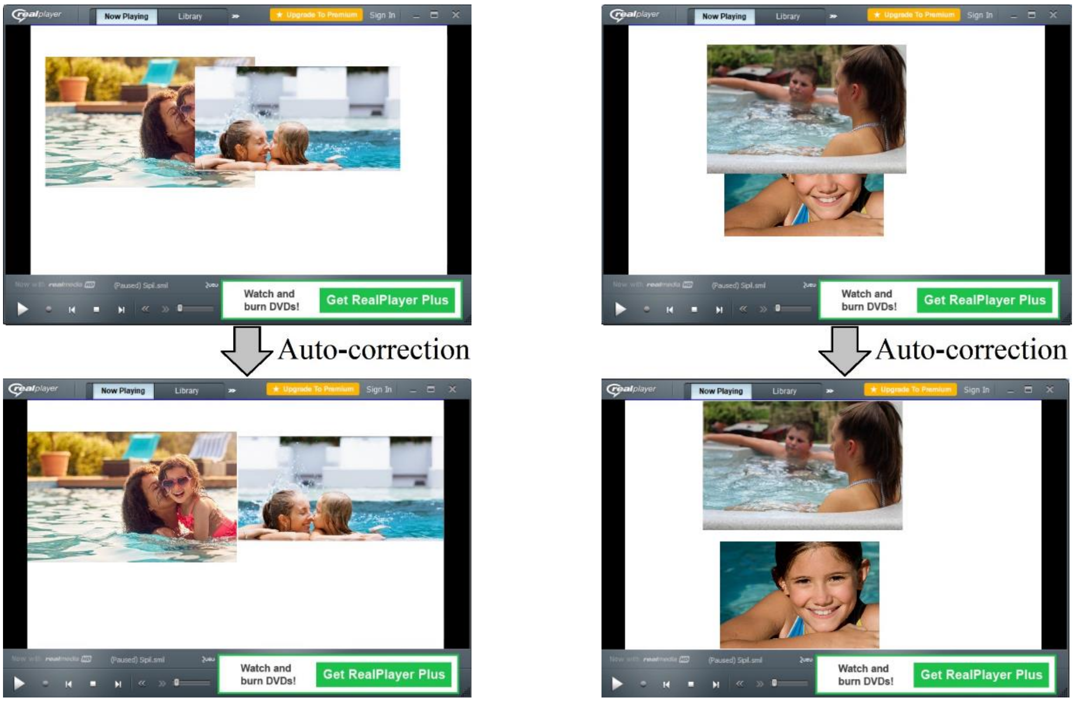
Algorithm 3: Auto-correction

```
c ← number of conflicts
For(a=1; i≤n; i++)
   i ← conflict(a,1)
   j ← conflict(a,2)
   if(((region(j,left) > region(i,left)) && (region(j,left) <
   region(i,left)+region(i,width))) || ((region(j,left) + region(j,width) >
   region(i,left)) && (region(j,left)+region(j,width) < region(i,left)
   +region(i,width))) = true
   &&
   (region(j,top) > region(i,top)) && region(j,top) <
   region(i,top)+region(i,height))) || ((region(j,top)+ region(j,height) >
   region(i,top)) && (region(j,top)+region(j,height) < region(i,top)
   +region(i,height))) = false
       if((region(i,left) < (region(j,left))
          region(i,left) ← 0
          region(j,left) ← root(width)-region(j,width)
       else
          region(j,left) ← 0
```

```
10:          region(i,left) ← root(width)-region(i,width)
          end if
       end if
11:    if(((region(j,left) > region(i,left)) && (region(j,left) <
       region(i,left)+region(i,width))) || ((region(j,left) + region(j,width) >
       region(i,left)) && (region(j,left)+region(j,width) < region(i,left)
       +region(i,width))) = false
       &&
       (region(j,top) > region(i,top)) && region(j,top) <
       region(i,top)+region(i,height))) || ((region(j,top)+ region(j,height) >
       region(i,top)) && (region(j,top)+region(j,height) < region(i,top)
       +region(i,height))) = true
12:       if((region(i,top) < (region(i,top))
13:          region(i,top) ← 0
14:          region(j,top) ← root(height)-region(j,height)
          else
15:          region(j,height) ← 0
16:          region(i,height) ← root(height)-region(i,height)
          end if
       end if
    end for
```

3. RESULTS AND DISCUSSION

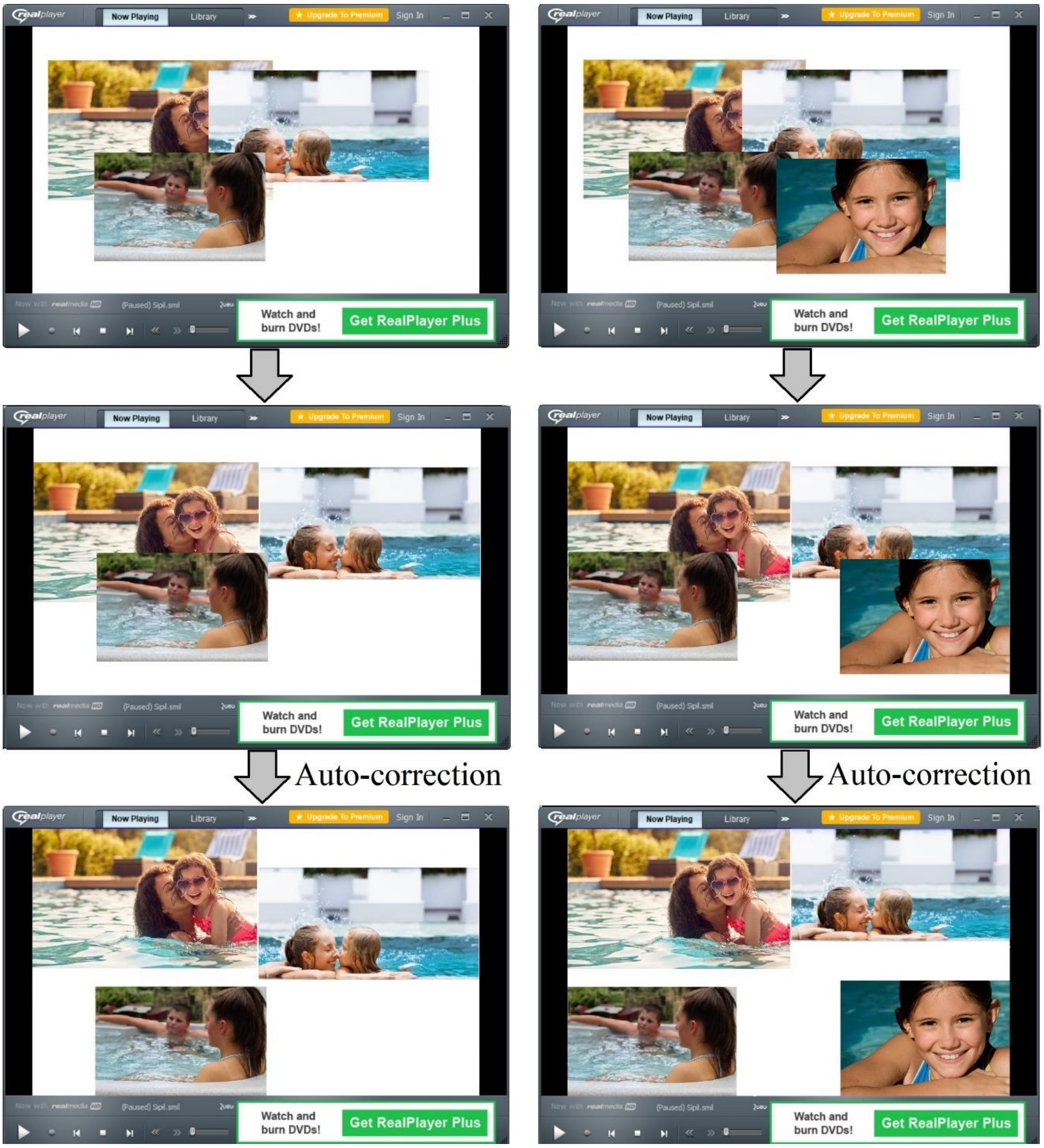
Experiments in this study were carried out with several kinds of data sets. In the first data set, two regions overlap horizontally, as shown in Figure 8(a). The two images overlap; after the auto-correction process is carried out, the correction results are obtained by relocating the two images so they no longer overlap. Figure 8(b) shows auto-correction on two vertically overlapping images.



(a) (b)

Figure 8. Two types of correction (a) horizontal overlap corrections and (b) vertical overlap corrections

Figure 9(a) shows the auto-correction of three overlapping images. The correction algorithm will perform two correction stages: first is to process a horizontal correction and then a vertical correction. Figure 9(b) shows four overlapping images in a multimedia presentation. The auto-correction process is also carried out in two stages, the first stage is to make horizontal corrections. At this stage, the two pairs of images are separated horizontally but still overlap vertically. In the second stage, the relocation is processed vertically, so that the whole image does not overlap anymore.



(a) (b)

Figure 9. Multiple spatial overlap corrections (a) three regions and (b) four regions

This study has succeeded in providing novelty in the auto-correction of spatial conflicts by correcting four conflicting regions in multimedia authoring. In previous studies, it has been able to correct when there is a spatial-temporal conflict, but only in two or three regions. Table 1 compares the success of correcting the spatial conflicts in this study with existing studies.

Table 1. Comparison with previous studies

Studies	Conflicting areas		
	Two regions	Three regions	Four regions
The proposed study	√	√	√
[27]	√		
[25]	√		
[28]	√	√	

4. CONCLUSION

In a multimedia presentation, often, an author makes an error in creating a region area that causes the overlapping of several media objects. A novel method in this study is to auto-correct multiple spatial conflicts using several stages of the process: overlapping detection, Spatio-temporal verification, and auto-correction. This study conducted several experiments with two spatial overlaps, three spatial overlaps, and four spatial overlaps. The algorithm proposed in this study automatically corrected up to four spatial conflicts. Correcting multiple spatial conflicts can be successful using two relocation stages: horizontal relocation and vertical relocation. This study has limitations in solving cases if too many pairs of overlapping media objects cause the inability to be completed perfectly. Improvements that could be investigated in the future include separating the spatial-temporal overlapping into different time domains.

ACKNOWLEDGEMENTS

This research was supported by Maranatha Christian University Bandung Indonesia and Universiti Teknikal Malaysia Melaka, Malaysia.

REFERENCES

[1] Obaida M. Al-Hazaimeh, A. A. Abu-Ein, M. M. Al-Nawashi, and N. Y. Gharaibeh, "Chaotic based multimedia encryption: a survey for network and internet security," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 4, pp. 2151–2159, 2022, doi: 10.11591/eei.v11i4.3520.

[2] S. Bouyakoub and A. Belkhir, "SMIL builder: An incremental authoring tool for SMIL documents," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 7, no. 1, pp. 1–30, 2011, doi: 10.1145/1870121.1870123.

[3] S. Kusairi, C. H. W. Pangestu, N. Hidayat, and A. Suryadi, "Kinematic Test in Multimedia Formats: A Preliminary Validation," *TEM Journal*, vol. 10, no. 4, pp. 1789–1795, 2021, doi: 10.18421/TEM104-41.

[4] V. Monita, I. D. Irawati, and R. Tulloh, "Comparison of routing protocol performance on multimedia services on software defined network," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1612–1619, 2020, doi: 10.11591/eei.v9i4.2389.

[5] A. R. Idrais, I. Aljarrah, and O. Al-Khaleel, "A spatial image compression algorithm based on run length encoding," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2607–2616, 2021, doi: 10.11591/eei.v10i5.2563.

[6] M. C. Wijaya, Z. Maksom, and M. H. L. Abdullah, "Novel Framework for Translation Algorithms in Multimedia Authoring Tools," *IAENG International Journal of Computer Science*, vol. 49, no. 3, pp. 628–636, 2022.

[7] D. C. A. Bulterman, P. Cesar, E. Munson, and M. Da Graça Pimentel, "Multimedia authoring and annotation," *Multimedia Tools and Applications*, vol. 70, no. 2, pp. 1161–1165, 2014, doi: 10.1007/s11042-014-1963-9.

[8] M. R. Bhuiyan *et al.*, "Hajj pilgrimage video analytics using CNN," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2598–2606, 2021, doi: 10.11591/eei.v10i5.2361.

[9] M. C. Wijaya, Z. Maksom, and M. H. L. Abdullah, "Speedup Multimedia Authoring Process: Incrementality of Multimedia Authoring Verification," *Journal of Engineering Science and Technology Review*, vol. 15, no. 3, pp. 170–176, 2022, doi: 10.25103/jestr.153.19.

[10] M. C. Wijaya, Z. Maksom, and M. H. L. Abdullah, "A Brief of Review : Multimedia Authoring Tool Attributes," *Ingénierie des Systèmes d'Information*, vol. 26, no. 1, pp. 1–11, 2021, doi: 10.18280/isi.260101.

[11] C. Chen, Y. Wang, K. Yi, T. Wang, and H. Xiang, "Semantic analysis of action with spatio-temporal features based on object detection," *Engineering Letters*, vol. 28, no. 2, pp. 616–623, 2020.

[12] O. Gaggi and A. Bossi, "Analysis and verification of SMIL documents," *Multimedia Systems*, vol. 17, no. 6, pp. 487–506, 2011, doi: 10.1007/s00530-011-0233-1.

[13] J. dos Santos, C. Braga, and D. C. Muchaluat-Saade, "An Executable Semantics for a Multimedia Authoring Language," in *Formal Methods: Foundations and Applications. SBMF 2013. Lecture Notes in Computer Science*, 2013, vol. 8195, pp. 67–82, doi: 10.1007/978-3-642-41071-0.

[14] M. C. Wijaya, Z. Maksom, M. Haziq, and L. Abdullah, "A Brief Review : Multimedia Authoring Modeling," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 13, no. 1, pp. 39–48, 2022.

[15] M. C. Wijaya, Z. Maksom, and M. H. L. Abdullah, "Two Verification Phases in Multimedia Authoring Modeling," *Journal of information and communication convergence engineering*, vol. 19, no. 1, pp. 42–47, 2021, doi: 10.6109/jicce.2021.19.1.42.

[16] M. Jourdan, C. Roisin, L. Tardif, and L. Villard, "Authoring SMIL documents by direct manipulations during presentation," *World Wide Web*, vol. 2, no. 4, pp. 179–190, 1999, doi: 10.1023/A:1019256924296.

[17] M. Jourdan, "A formal semantics of SMIL: A web standard to describe multimedia documents," *Computer Standards and Interfaces*, vol. 23, no. 5, pp. 439–455, 2001, doi: 10.1016/S0920-5489(01)00091-5.

[18] V. S. Sari, E. Purwaningsih, Winarto, and N. A. Pramono, "Development of the Interactive Multimedia Software 'Inquiry Play-Room' as an Electronic Learning Resource for Rotation and Equilibrium Topic," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 7, pp. 81–93, 2021, doi: 10.3991/ijim.v15i07.21561.

[19] B. Meixner, K. Matusik, C. Grill, and H. Kosch, "Towards an easy to use authoring tool for interactive non-linear video," *Multimedia Tools and Applications*, vol. 70, no. 2, pp. 1251–1276, 2014, doi: 10.1007/s11042-012-1218-6.

[20] C. Yang, C.-K. Chu, and Y.-C. Wang, "Extension of Timeline-based Editing for Non-deterministic," *Journal Of Information Science and Engineering*, vol. 24, no. 5, pp. 1377–1395, 2008, doi: 10.1688/JISE.2008.24.5.6.

[21] M. Y. Sung and D. H. Lee, "A collaborative authoring system for multimedia presentation," *IEEE International Conference on Communications*, vol. 3, no. c, pp. 1396–1400, 2004.

[22] M. J. Antonacci, D. C. Muchaluat-Saade, R. F. Rodrigues, and L. F. G. Soares, "Improving the expressiveness of XML-based hypermedia authoring languages," in *Multimedia Modeling: Modeling Multimedia Information and Systems, MMM 2000*, 2020, pp. 71–88.

[23] A. Bossi and O. Gaggi, "Enriching SMIL with assertions for temporal validation," *Proceedings of the ACM International Multimedia Conference and Exhibition*, no. January 2007, pp. 107–116, 2007, doi: 10.1145/1291233.1291256.

[24] L. F. G. Soares, R. F. Rodrigues, and D. C. Muchaluat Saade, “Modeling, authoring and formatting hypermedia documents in the HyperProp system,” *Multimedia Systems*, vol. 8, no. 2, pp. 118–134, 2000, doi: 10.1007/s005300050155.
[25] F. Z. Mekahlia, A. Ghomari, S. Yazid, and D. Djenouri, “Temporal and Spatial Coherence Verification in SMIL Documents with Hoare Logic and Disjunctive Constraints: A Hybrid Formal Method,” *Journal of Integrated Design and Process Science*, vol. 20, no. 3, pp. 39–70, 2017, doi: 10.3233/jid-2016-0020.
[26] B. Huang, S. Yi, and W. T. Chan, “Spatio-temporal information integration in XML,” *Future Generation Computer Systems*, vol. 20, no. 7, pp. 1157–1170, 2004, doi: 10.1016/j.future.2003.11.005.
[27] J. A. F. Dos Santos, C. Braga, C. Débora, C. Roisin, and N. Layaïda, “Spatio-temporal validation of multimedia documents,” in *Proceedings of the 2015 ACM Symposium on Document Engineering*, 2015, no. September, pp. 133–142, doi: 10.1145/2682571.2797060.
[28] S. Bouyakoub and A. Belkhir, “A Spatio-Temporal Authoring Tool for Multimedia SMIL Documents,” *International Journal of Multimedia Technologies*, vol. 2, no. 3, pp. 83–88, 2012.

BIOGRAPHIES OF AUTHORS

Marvin Chandra Wijaya received a BS in Computer and Electrical Engineering from Maranatha Christian University, Indonesia in 1996, an MBA from Parahyangan Catholic University, Indonesia, in 1999 and MS in Computer Engineering from Institut Teknologi Bandung, Indonesia, in 2002. Since 1996, he has been working as a lecturer at Maranatha Christian University, Indonesia. His research interests include multimedia, artificial intelligence, and embedded system. He can be contacted at email: marvin.cw@eng.maranatha.edu.

Zulisman Maksom is a Director for the Centre for Instructional Resources and Technology, Universiti Teknikal Malaysia Melaka. He graduated with a BA (Hons) in Computer Graphic Design from Wanganui School of Design, New Zealand, and received MA in Design and Manufacture from De Montfort University, UK. He also has a Doctor in Design from the Swinburne University of Technology, Australia. His research interests are in UI and UX, communication design, instructional design, graphic design technology and multimedia, pedagogical technology, and learning technology. He is also active in industry partnerships with various consultation projects, such as experiential learning and online course development. He can be contacted at email: zulisman@utem.edu.my.

Muhammad Haziq Lim Abdullah is a senior lecturer and researcher under a subgroup Specialists in Special Needs Awareness and Research (SPEAR), Human-Centered Computing-Information System Lab (HCC-ISL), Centre of Advanced Computing, Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka. His research interests are human-computer interaction, co-design, assistive technology, serious games, and augmented reality. He can be contacted at email: haziq@utem.edu.my.