Optimization of Sound Spatialization Resource Management through Clustering

Jens Herder
Spatial Media Group
University of Aizu
Fukushima-ken 965-8580
Japan
voice: [+81](242)37-2537; fax: [+81](242)37-2549
e·mail: herder@u-aizu.ac.jp
www: http://www.u-aizu.ac.jp/~herder

Abstract

Level-of-detail is a concept well-known in computer graphics to reduce the number of rendered polygons. Depending on the distance to the subject (viewer), the objects' representation is changed. A similar concept is the clustering of sound sources for sound spatialization. Clusters can be used to hierarchically organize mixels and to optimize the use of resources, by grouping multiple sources together into a single representative source. Such a clustering process should minimize the error of position allocation of elements, perceived as angle and distance, and also differences between velocity relative to the sink (i.e., Doppler shift). Objects with similar direction of motion and speed (relative to sink) in the same acoustic resolution cone and with similar distance to a sink can be grouped together.

Keywords: Sound spatialization, resource management, audio rendering, clustering, and human perception

1 Introduction

Human interface resources can be classified using different taxonomies. One such organization classifies according to resources provided by the computer. Another is to classify by the ability of the user to perceive information [OH93].

It is not necessary to compute (i.e., use system resources) display data which the user cannot perceive because of occlusion, masking, or low level. Rendering devices have limited capabilities. A sound spatialization backend (e.g., Acoustetron) can render (i.e., put monaural sound sources into three-dimensional space) only a limited number of mixels (e.g., eight channels) simultaneously. The responsibility of the resource manager [HC97a] is to determine the set of computable renderable data $R_{\text{computation}}$, which is a subset of both $R_{\text{displayable}}$ and $R_{\text{perceivable}}$.

$$R_{\text{computation}} \subseteq R_{\text{displayable}} \cap R_{\text{perceivable}}$$

$R_{\text{displayable}}$ is the set of resources which are available for the display. $R_{\text{perceivable}}$ is the set of resources which the user can perceive. At a given point in time, $R_{\text{computation}}$ is optimal if there is no larger set which fulfills the requirements. This is not necessarily the best solution over a long period in time because allocated resources cannot be freed immediately, so that in a dynamically changing scene, a non-optimal solution for a short time period could give, on average, a better impression.

A sound spatialization resource manager [HC97b] controls sound resources and optimizes fidelity (presence) under given conditions.

The basic idea of clustering [Her99a] is illustrated in Figure 1. Consider the cluster in the upper left corner. The flat ellipsoid surrounding a sound source represents the radiation pattern.

The external vector denotes direction of motion and speed of the object. Imagine two cars on a road, chasing each other but not close to an observer. Both move away from the sound sink in the middle of the drawing. Similarly, the sources clustered in the upper right corner are not moving (imagine a group of people talking at a distance), and can be easily represented as a single source which mixes the signals of all sources in the cluster. The other sound sources cannot be clustered because they have different motion direction or do not fit into a single resolution cone (i.e., direction would be perceived differently).

The required information regarding velocity and moving direction is obtained via object monitoring. The sources in the lower part of the Figure 1 cannot be clustered because of different motion direction (i.e., different Doppler shift).

A visualization of the clustering algoritm introduced in this paper can be found in [Her99b].

2 Clustering algorithm

The sound resource allocation algorithm described in [HC97b] can be extended and improved by introducing sound source clustering. Figure 2 shows how clustering is included into the algorithm. The previous algorithm is used for calculating the set of audible sources, but does not evaluate the priorities before clustering takes place. After clustering, priorities can be used to determine the set of active source for audio rendering.

Clustering Algorithm 1 is presented in pseudocode. A sound source is added to a cluster if the perceptual error between representative (i.e., virtual) sound source and all sound sources in the cluster is smaller than an experimentally determined threshold (e.g., using data obtained by [MM90]). Error can be calculated for direction, distance, and Doppler shift. A cluster is valid for only one sink. Figure 3 shows an example in which two sound sources are clustered together and represented by a representative sound source. The sound sources are within the resolution cone of the representative sound source. The resolution cone shape varies depending on azimuth and elevation. The ellipses axes of the cones denote error in azimuth and elevation. A generalizing of the data suggests that the error in azimuth to the front is small, growing larger to the sides, while error in elevation decreases at the sides. The back has much higher error than the front.

Algorithm 1 Clustering algorithm for sound sources

```
for each sink in sinks do
  workSet ⇐ sources of sink
  while workSet is not empty do
    add source from workSet to repSourceSet
    remove source from workSet
    for each source in workSet do
      repSource ⇐ rep(repSourceSet + source)
      if    inLimits(sink,repSource,
                     repSourceSet + source)
      then
        add source to repSourceSet
        remove source from workSet
      end if
    end for
    add rep(repSourceSet) to repSources including mixing data
  end while
end for
```

Clustering Algorithm 1 converges quickly, because in the while loop, the `workSet` is reduced in the worst case at least by one source. (The number of steps for the while loop is $\sum_{i=1}^{n} n-i = 1/2n(n+1)$.) The complexity is $O(n^2 * m)$, where n is the number of sound sources and m is the number of sound sinks. The algorithm is not optimal in the sense that there might be another clustering configuration which has fewer clusters. An optimal algorithm would calculate all possible configurations and would choose that configuration with the fewest clusters. minimize perceptual errors as well as number of clusters

The `rep` function returns an aggregate sound source for a set of sound sources. The position of that representative source can be the centroid (mean position) of the set.

$$\texttt{rep}(sources) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} source_i \qquad (1)$$

The boolean `inLimits` function returns `true` if the sources are within the spatial not perceivable limits (i.e., localization errors).

$$\texttt{inLimits}(sink, rep, sources) =$$

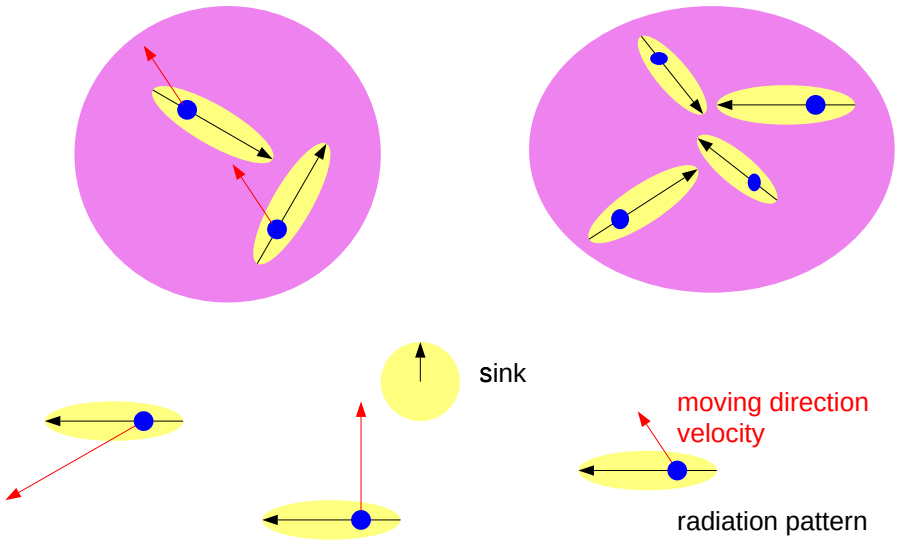

Figure 1: Clustering of sources in resolution cone with similar moving direction and speed: the cluster in the left upper corner shows two cars chasing each other in the distance in direction away from the sink; the cluster in the right upper corner represents a stationary group of people talking; the other sound sources cannot be clustered because of different motion direction, or because they do not fit into one resolution cone

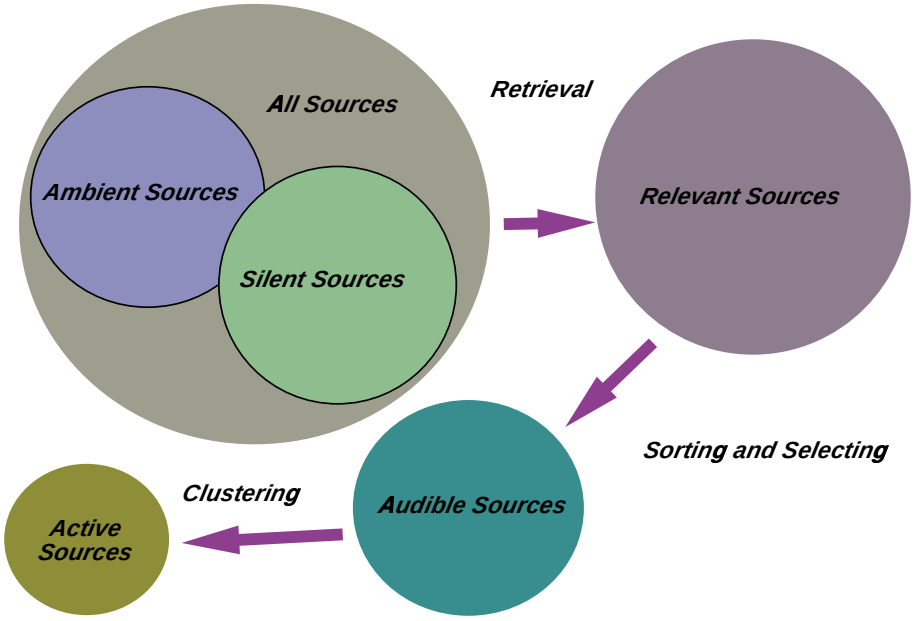

Figure 2: Clustering reduces the number of required spatialization channels; in a first retrieval step ambient sources and sources which are not audible either by intensity or the sink is not in the audible range are removed; in second step sources are sorted by priority; finally the clustering algorithm presented in this papers tries to minimize this set; in a final step sources which now might contain representative source of clusters are selected and send to the spatialization backend

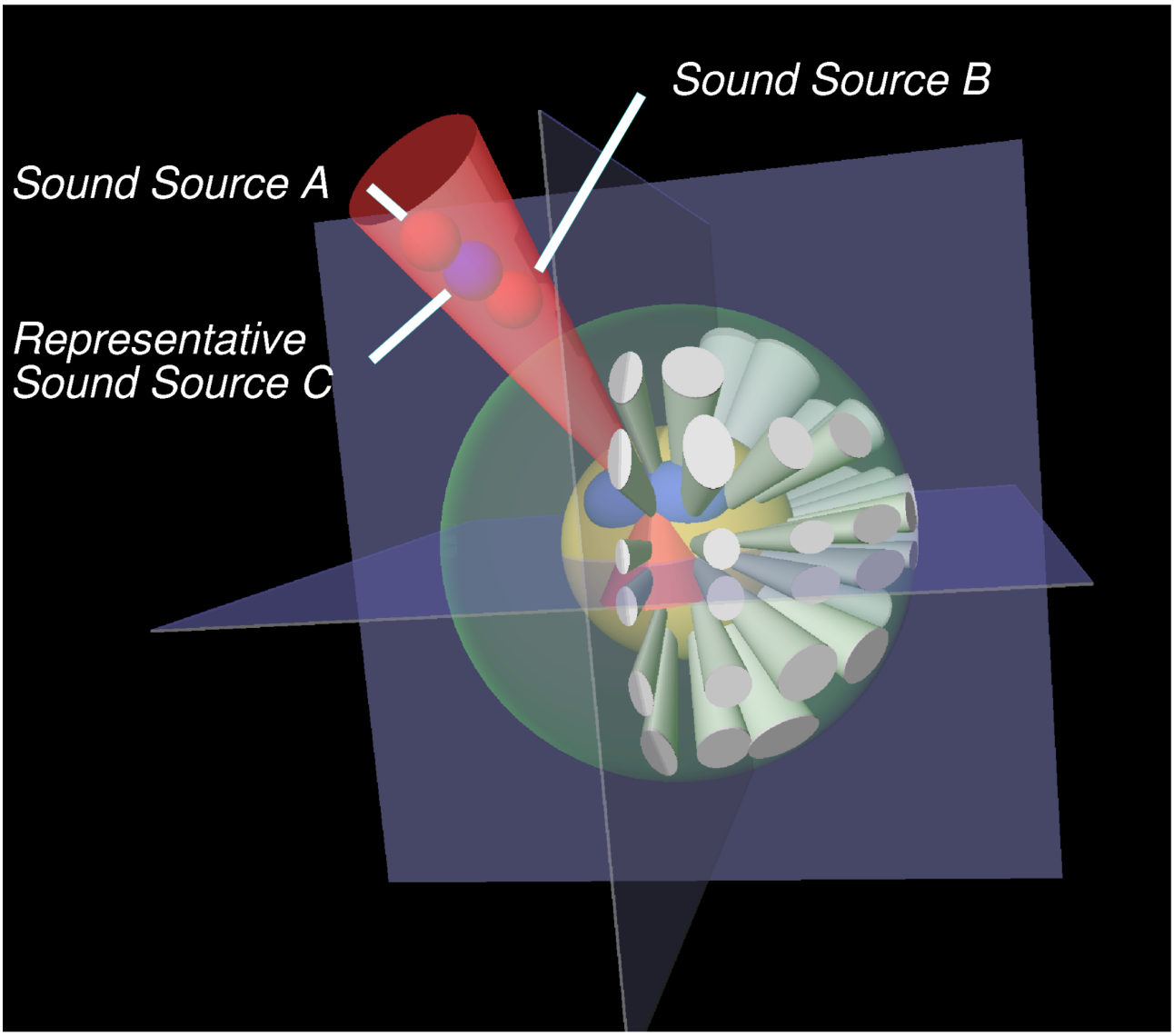

Figure 3: Two sound sources A and B are clustered together within the resolution cone of the representative virtual sound source C; Resolution cones denotes horizontal and vertical unsigned localization errors for a broadband signal; ellipses axes denote error in azimuth and elevation

$$\texttt{inDirection}(sink, rep, sources)\ \&$$
$$\texttt{inDistance}(sink, rep, sources)\ \&$$
$$\texttt{inDoppler}(sink, rep, sources) \quad (2)$$

The boolean `inDirection` function returns `true` if the sources are in the resolution cone of the representative for a given sink. The azimuth and elevation limit values for the specific direction of the representative are calculated by interpolation of experimentally determined limit values.

The boolean `inDistance` function returns `true` if sources are in the range limits of the representative for a given sink. The range limit values for a specific distance are calculated by interpolation of experimentally determined limit values.

The boolean `inDoppler` function compares the Doppler shift of all sources relative to the Doppler shift of the representative. If the difference in Doppler shift is not perceivable, then the function returns `true`. Again the limit values are based on experimentally determined limit values.

3 Psychoacoustic evaluation of the clustering algorithm

User studies can evaluate and validate the performance of a sound spatialization system, including a sound spatialization backend and a procedure for sound spatialization resource management. Since the developed resource management is based on human perception, an evaluation using objective tests would not measure its effectiveness. Testing system performance using subjective tests is not easy, and difficult to reproduce because the test conditions are difficult to control. Auditory experiments and results are comprehensively described in [Bla96]. Subjects' abilities to localize sound vary across subjects, experimental conditions and tasks. Localization ability depends on the stimulus surrounding sounds and room features. Performance tests can only be done for specific tasks of certain applications.

The validity of the presented clustering algorithm can be demonstrated for specific configurations. Here the same spatialization backend was used, but spatialization was done with and without clustering enabled.

3.1 Method

A scene with three sound sources was prepared. A timed script activated the sound sources using MIDI commands. In this evaluation study, the spatialization backend Acoustetron II was used. As sound generator, 4 MIDI synthesizer Roland SoundCanvas SC-55mkIIs were used. The sounds were a bird (instrument 124), a telephone (instrument 125), and a gun shot (instrument 128). The sources were triggered with 500 ms delay in between so that the onsets did not overlap. Two of the MIDI synthesizers produced identical signals for reverberation, which was passed via a mixer to a reverberator Yamaha REV 500 as monaural signal. The reverberant signal mixed with directionalized sound from the spatialization backend. The configuration of the reverberator was setup to simulate a medium sized room. (Parameters were Effect only, 24 ms predelay, 1 s reverb time high-ratio 0.4, and ER level 100.) The reverberator improved externalization [Beg94, p. 97], and was used to produce ambient sound for sound sources which could not be spatialized.

stimuli	spatialization channels	number of clusters	ambient sound sources
no restrictions	3	0	0
clustering	2	1	0
ambient	2	0	1

Table 1: Stimuli use of spatialization resources

stimuli	label	x	y	sound
no restrictions	so. 1	7.87	-7.87	gun shot
	so. 2	7.87	7.87	phone ringing
	so. 3	7.87	11.81	bird call
clustering	so. 1	7.87	-7.87	gun shot
	cl. 1	7.87	9.84	bird call and phone ringing
ambient	so. 1	7.87	-7.87	gun shot
	so. 2	7.87	7.87	phone ringing
	amb.	-	-	bird call

Table 2: Stimuli source description (using the coordinate system of the CRE API)

Listening was done using headphones in an ane-

choic chamber. Five listeners participated. One trial consisted out of a ABA sequence. A stimuli was either three sound source processed with three spatialization channels (N), processed using the developed clustering algorithm (C), or processed using two spatialization channels and one sound source was presented ambient (A). This is summarized in Table 2 (coordinates are represented using the CRE API [CRE94]). The total number of trials was 72; each stimulus combination was presented 8 times. The nine trial combinations are listed in Table 3. The listeners were asked to rate the dissimilarity of the spatial imagery. They marked "1" when the spatial images were judged equal and "5" for the largest difference. The stimuli combinations with itself were included to check if users response is randomly. Also the ratings for a stimuli pair in different order should give similar ratings.

stimulus	abbreviation	first	second
1	NCN	non-restricted	clustered
2	CNC	clustered	non-restricted
3	NAN	non-restricted	ambient
4	ANA	ambient	non-restricted
5	CAC	clustered	ambient
6	ACA	ambient	clustered
7	NNN	non-restricted	non-restricted
8	CCC	clustered	clustered
9	AAA	ambient	ambient

Table 3: Trial combinations

3.2 Results and discussion

The average ratings for all listeners for each stimuli are shown in Table 4. The mean ratings on the diagonal show that not always the listener detected that the same stimulus was presented three times.

Average dissimilarity ratings regardless of order of listening is shown in Figure 4. The average dissimilarity for clustered and ambient processing (CA) was 4.45, for non-restricted to ambient processing (NA) was 4.4375, and for non-restricted to clustered processing was 2.2. The average dissimilarly intercomparison of all three stimuli was 1.1917.

		second		
		N	C	A
first	N	1.075	2.225	4.500
	C	2.175	1.000	4.375
	A	4.375	4.525	1.500

Table 4: Dissimilarity between intervals: non-restricted (N), clustered (C), and ambient (A)

Sound spatialization with no restrictions in the number of spatialization channels or using clustering for (based on the specific configuration) were rated very dissimilar to the processing with one ambient sound source. The dissimilarity judgments between processing with no restrictions and clustering were ranked half compared to the ratings for ambient processing with the others.

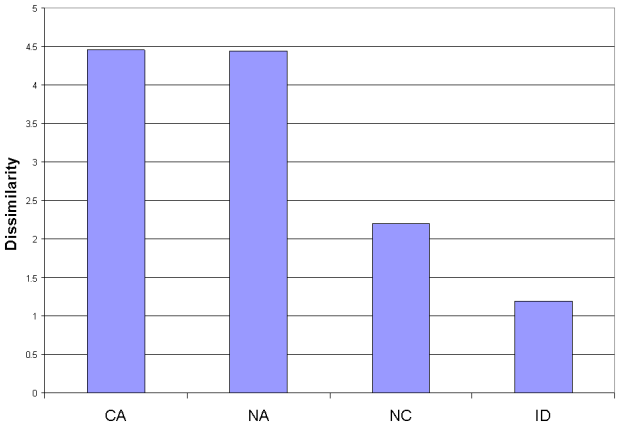

Figure 4: Dissimilarity for non-restricted (N), clustered (C), and ambient (A) processing

Assuming spatialization resource limitations and specific configuration, processing using clustering improves the spatial imagery. Clustering, as implemented, did not give the same spatial image to processing without limitations.

4 Conclusion

The advantages and disadvantages of clustering can be summarized as:

- far better use of spatialization resources,
- freeing resources for other tasks such as visualization,

- improved spatialization fidelity in case of limited resources,
- perceptual artifacts might occur during switching (source assignment to different cluster),
- costs for mixing the audio streams might reduce the gains through clustering, and
- sound spatialization errors might occur through averaging object attributes.

The clustering algorithm applies psychoacoustic data (localization errors dependent on listener orientation) as optimization criteria. The advantages and disadvantages were discussed. The developed sound spatialization resource manager improves spatialization fidelity under runtime constraints. Application programmers and virtual reality scene designers are freed from the burden of assigning and predicting the sound sources.

The algorithm is implemented within the Sound Spatialization Framework [Her98] which is freely available as binary distribution.

5 Future research

The resolution cones used in the experiments were based on localization errors for a broadband signal. In case of voice signals, expected localization errors are higher. Using this information for the clustering algorithm can improve the effectiveness of the resource management process. The effectiveness of the cluster algorithm was only evaluate for one static configuration using a controlled listening experiment. For further evaluation, other configurations, including spatialization backends based on different technology are required.

Acknowledgments

The author thanks Michael Cohen and William L. Martens for fruitful discussions.

References

[Beg94] Durand R. Begault, *3-D Sound for Virtual Reality and Multimedia*, Academic Press, 1994, ISBN 0-12-084735-3.

[Bla96] Jens Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*, revised ed., MIT Press, 1996, ISBN 0-262-02413-6.

[CRE94] Crystal River Engineering, Inc., *Cre_tron library reference manual*, August 1994, Revision B.

[HC97a] Jens Herder and Michael Cohen, *Enhancing perspicuity of objects in virtual reality environments*, CT'97 — Second Int. Cognitive Technology Conf., IEEE, IEEE Press, August 1997, ISBN 0-8186-8084-9, pp. 228–237.

[HC97b] Jens Herder and Michael Cohen, *Sound Spatialization Resource Management in Virtual Reality Environments*, ASVA'97 — Int. Symp. on Simulation, Visualization and Auralization for Acoustic Research and Education (Tokyo, Japan), ASJ, April 1997, pp. 407–414.

[Her98] Jens Herder, *Sound Spatialization Framework*, Web site, University of Aizu, Japan, 1998, `http://www-ci.u-aizu.ac.jp/SF/`.

[Her99a] Jens Herder, *A sound spatialization resource management framework*, Dissertation, University of Tsukuba, July 1999.

[Her99b] Jens Herder, *Visualization of a clustering algorithm of sound sources based on localization errors*, Second Int. Conf. on Human and Computer (Aizu-Wakamatsu, Japan), University of Aizu, September 1999.

[MM90] James C. Makous and John C. Middlebrooks, *Two-dimensional sound localization by human listeners*, JASA **87** (1990), no. 5, 2188–2200.

[OH93] Russell Ovan and William S. Havens, *Intelligent mediation: An architecture for the real-time allocation of interface resources*, Proceedings of the 1993 International Workshop on Intelligent User Interfaces, ACM SIGCHI, ACM Press, 1993, Orlando, Florida, January 4-7, 1993, pp. 55–61.