Chapter 4

Optimization through Clustering

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, presented in this chapter. Clusters can be used to hierarchically organize mixels [Cohen, 1993, p. 294] and to optimize the use of resources, by grouping multiple sources together into a single representative source [Herder and Cohen, 1997]. 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.

The basic idea of clustering is illustrated in Figure 4.1. 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, as described in Section 4.3. The sources in the lower part of the Figure 4.1 cannot be clustered because of different motion

direction (i.e., different Doppler shift).

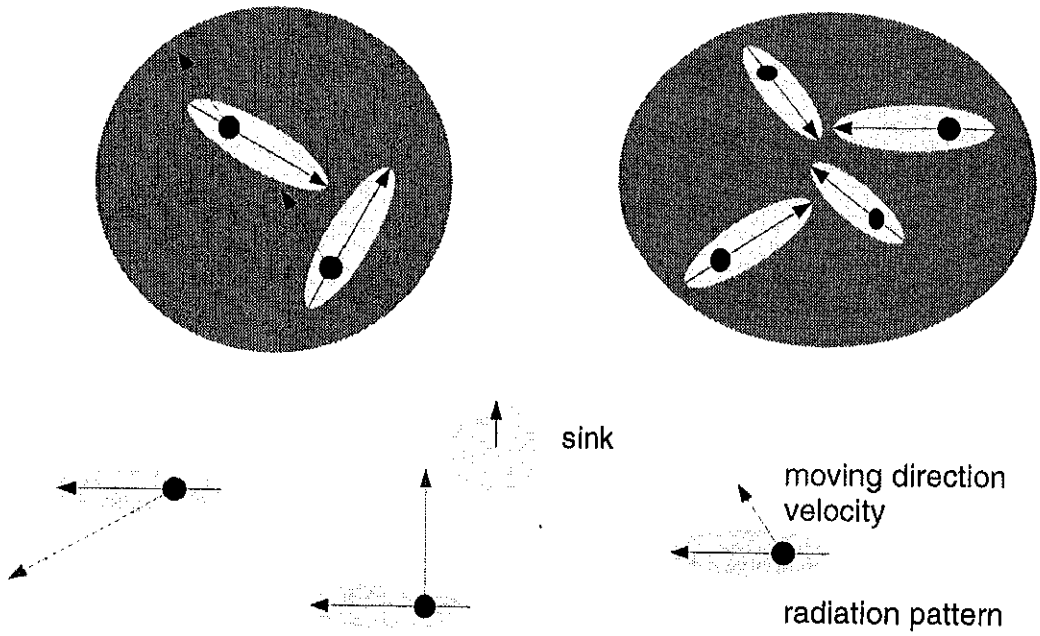

Figure 4.1: Clustering of sources in resolution cone with similar moving direction and speed: the cluster in the left upper corner shows two cars chasing each other in the distance in direction away from the sink; the cluster in the right upper corner represents a stationary group of people talking; the other sound sources cannot be clustered because of different motion direction, or because they do not fit into one resolution cone

4.1 Clustering algorithm

The sound resource allocation algorithm described in Chapter 2 can be extended and improved by introducing sound source clustering. Figure 4.2 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 4 is presented in pseudo-code. 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 [Makous

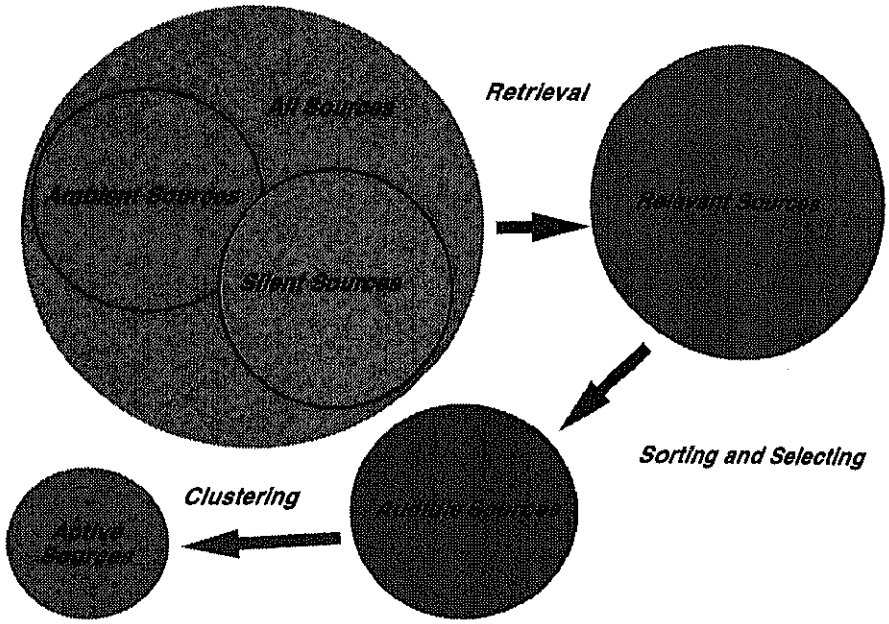

Figure 4.2: Clustering reduces the number of required spatialization channels

and Middlebrooks, 1990]). Error can be calculated for direction, distance, and Doppler shift. A cluster is valid for only one sink. Figure 4.3 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.

Clustering Algorithm 4 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 `representative` 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, or calculated as suggested in the next section.

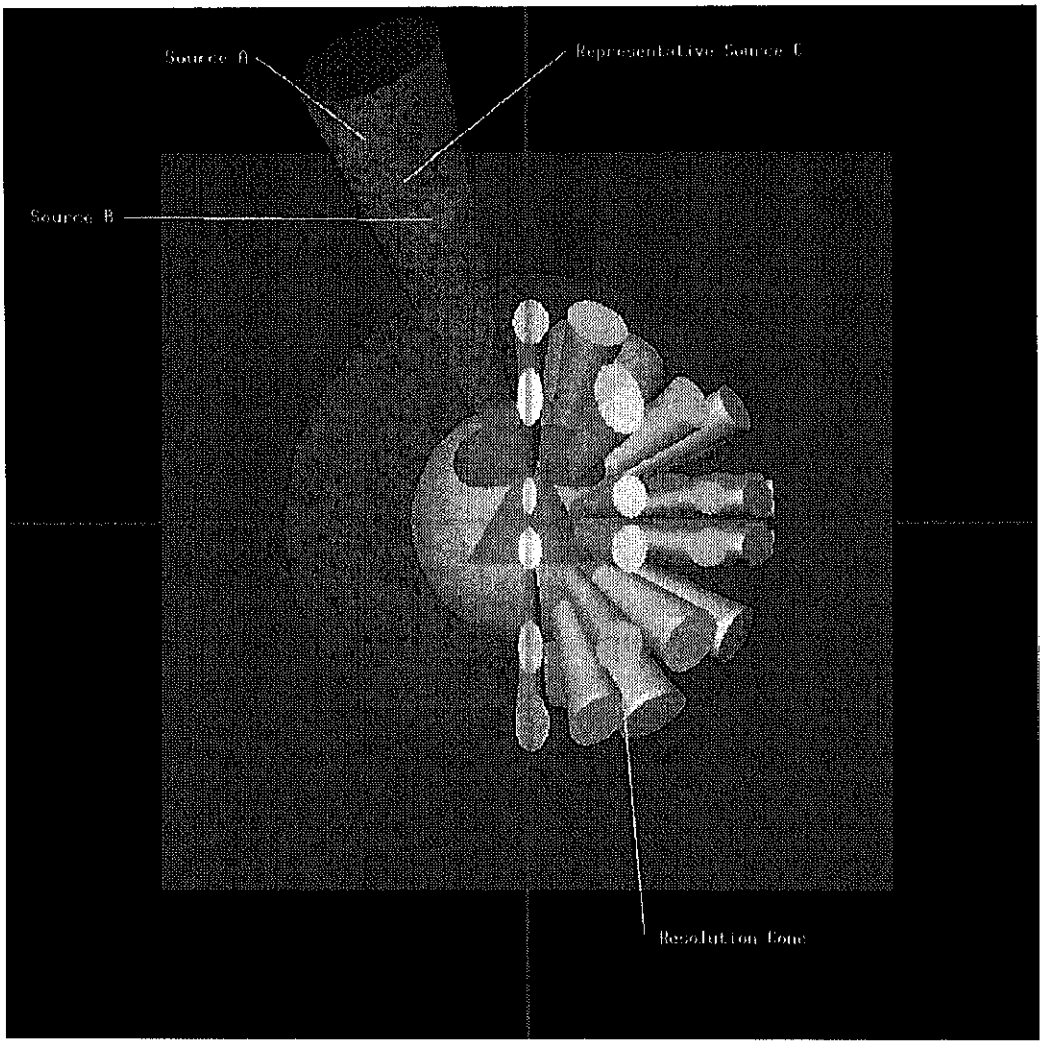

Figure 4.3: Two sound sources A and B are clustered together within the resolution cone of the representative virtual sound source C

Algorithm 4 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 representativeSourceSet
    remove source from workSet
    for each source in workSet do
      representativeSource ⇐
            representative(representativeSourceSet + source)
      if  withinNotPerceivableLimits(sink,representativeSource,
                                     representativeSourceSet + source)
      then
        add source to representativeSourceSet
        remove source from workSet
      end if
    end for
    add representative(representativeSourceSet) to representativeSources
    including mixing data
  end while
end for
```

$$\texttt{representative}(sources) \;=\; \frac{1}{n}\sum_{i=1}^{n} source_i \tag{4.1}$$

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

$$\begin{aligned}&\texttt{withinNotPerceivableLimits}(sink, representative, sources) = \\ &\quad \texttt{withinNotPerceivableDirection}(sink, representative, sources)\ \& \\ &\quad \texttt{withinNotPerceivableDistance}(sink, representative, sources)\ \& \\ &\quad \texttt{withinNotPerceivableDoppler}(sink, representative, sources)\end{aligned} \tag{4.2}$$

The boolean `withinNotPerceivableDirection` 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 `withinNotPerceivableDistance` 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 `withinNotPerceivableDoppler` 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.

4.2 Determining a representative sound source location for a cluster

A cluster of sound sources can be represented by one (representative) source which is then passed to a spatialization backend. A straightforward approach is to calculate the first moment of all sources in the cluster [Suzuki, 1997]. This does not consider the shape of the perceptual space. Section 3.8 explains the cone of confusion, in which sound sources on the rings centric to the binaural axis cannot be distinguished just by interaural time delay and interaural intensity difference. Usual studies of localization errors use the polar coordinate system using azimuth and elevation. If localization errors are presented using the coordinate system by [Morimoto and Aokata, 1984] based on a lateral and rising angle, then accuracy can be explained by two mutually independent cues. A similar coordinate system is suggested in this section.

A representative virtual source for two sound sources on such a ring should be also on the ring. Taking this requirement into account suggests using a cylindrical coordinate system, as shown on the left of Figure 4.5. A location *loc* (see Figure 4.4) is represented as a triple consisting of the distance y along the interaural axis, the length r of an orthogonal vector to y from the interaural axis to the location of the source, and the angle of this vector relative to the line-of-sight vector.

$$loc = \begin{pmatrix} y \\ r \\ \varphi \end{pmatrix} \tag{4.3}$$

In general, a representative source location is defined as

$$\overline{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{4.4}$$

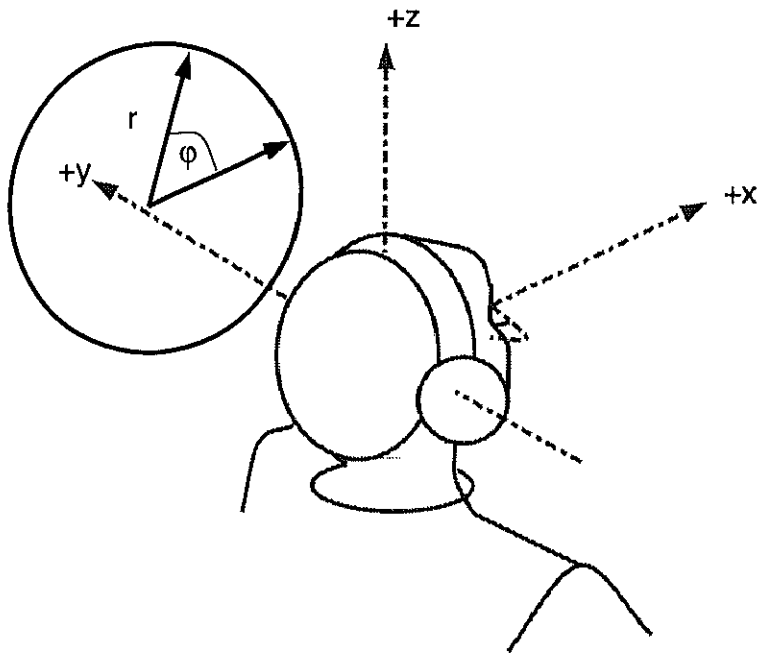

Figure 4.4: Listener inside the cylindrical coordinate system

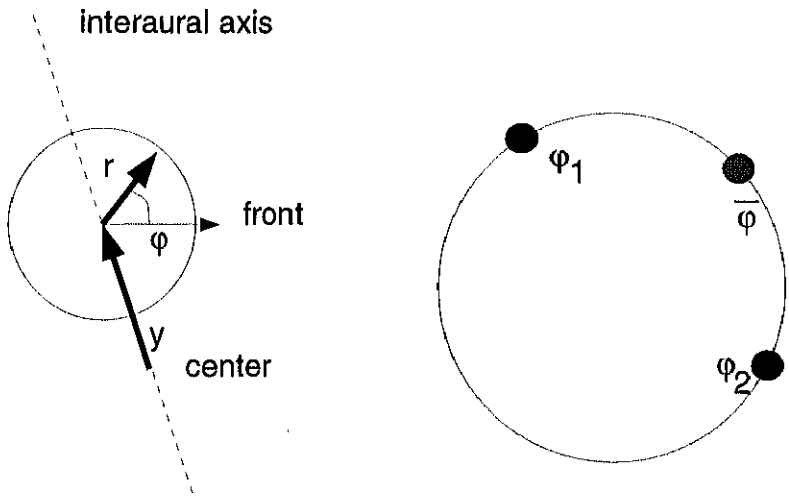

Figure 4.5: left: Cylindrical coordinate system; right: Representative source location

$$\overline{r} = \frac{1}{n}\sum_{i=1}^{n} r_i \tag{4.5}$$

To determine the angle $\overline{\varphi}$, several cases must be distinguished. Before that, some definitions are introduced.

Let

$$x_i = (\cos(\varphi_i), \sin(\varphi_i)) \tag{4.6}$$

The vector x_i for a given angle φ_i is on the unit circle. The first moment of the x_i corresponding to the angle $\overline{\varphi}$ is used to resolve the ambiguity and to find the minimal dividing angle. In the case that two sound sources are on the same ring, then $y_1 = y_2 = \overline{y}$ and $r_1 = r_2 = \overline{r}$ and the representative source is on the same ring.

The first momment vector $\overline{x}$ is defined as follows:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.7}$$

Let $\tilde{x}$ be the projection of $\overline{x}$ to the unit circle.

Set by definition:

$$\alpha = \arctan\left(\frac{\sum_{i=1}^{n} \sin\varphi_i}{\sum_{i=1}^{n} \cos\varphi_i}\right) \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \tag{4.8}$$

Case 1 $\overline{x} = 0$:
Set $\overline{\varphi} = 0$, i.e., use (1,0) $= \tilde{x}$

Case 2 $\overline{x} \neq 0$:
$\overline{x} = \frac{1}{n}(\sum_{i=1}^{n} \cos\varphi_i, \sum_{i=1}^{n} \sin\varphi_i)$

Case 2.1 $\sum_{i=1}^{n} \cos\varphi_i = 0$:

Case 2.1.1 $\sum_{i=1}^{n} \sin\varphi_i > 0$:
Set $\overline{\varphi} = \frac{\pi}{2}$, i.e., use $(0,1) = \tilde{x}$

Case 2.1.2 $\sum_{i=1}^{n} \sin\varphi_i < 0$:
Set $\overline{\varphi} = \frac{3\pi}{2}$, i.e., use $(0,-1) = \tilde{x}$

Case 2.2 $\sum_{i=1}^{n} \cos\varphi_i \neq 0$:

Case 2.2.1 $\overline{x}_1 > 0, \overline{x}_2 \geq 0$:
$\overline{\varphi} = \alpha \in [0, \frac{\pi}{2})$

Case 2.2.2 $\overline{x}_1 < 0$:
$\overline{\varphi} = \alpha + \pi \in (\frac{\pi}{2}, \frac{3\pi}{2}), \alpha \in (-\frac{\pi}{2}, \frac{\pi}{2})$

Case 2.2.3 $\overline{x}_1 > 0, \overline{x}_2 < 0$:
$\overline{\varphi} = \alpha + 2\pi \in (-\frac{3\pi}{2}, 2\pi), \alpha \in (-\frac{\pi}{2}, 0)$

In the case the sound sources are in the horizontal plane (i.e., φ for all sources is 0), then the representative source location is the first central moment of the vectors in this plane as a straightforward calculation shows.

4.3 Object monitoring

Object monitoring is necessary for two reasons. For a clustering heuristic, information about speed and motion direction is necessary. If not considered, the Doppler effect would ignored. In other words, objects with different Doppler shifts should not be combined into a cluster. Certain kinds of sound production are atomic, which means they cannot be decomposed without getting a different perception (e.g., attack portion of hitting a string). This situation occurs, e.g., using the MIDI protocol between "`note on`" and "`note off`" commands.

Taking this into account, the number of resource assignment (to cluster, to channel, to ambient, ...) switches should be minimized. Object monitoring allows look-ahead and resource reservation. The following object attributes can be monitored or calculated:

- speed;
- moving direction;
- changes of speed;
- changes of moving direction;
- probability of attribute change;
- maximum, minimum, and average values[1] for direction;
- intensity;
- location; and
- priority.

The probability of an attribute change can be determined by counting attribute changes and dividing through the passed time.

4.4 Discussion

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.

[1] see Application Programmer Interface in Chapter 5

Bibliography

[Cohen, 1993] Michael Cohen. Throwing, pitching, and catching sound: Audio windowing models and modes. *IJMMS: the Journal of Person-Computer Interaction*, 39(2):269–304, August 1993. ISSN 0020-7373.

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

[Makous and Middlebrooks, 1990] James C. Makous and John C. Middlebrooks. Two-dimensional sound localization by human listeners. *JASA*, 87(5):2188–2200, May 1990.

[Morimoto and Aokata, 1984] Masayuki Morimoto and Hitoshi Aokata. Localization cues of sound sources in the upper hemisphere. J. *Acous. Soc. Jap.*, 5(3):165–173, 1984.

[Suzuki, 1997] Taku Suzuki. Spatialization resource management for MIDI mixels. Bachelor thesis, University of Aizu, 1997.