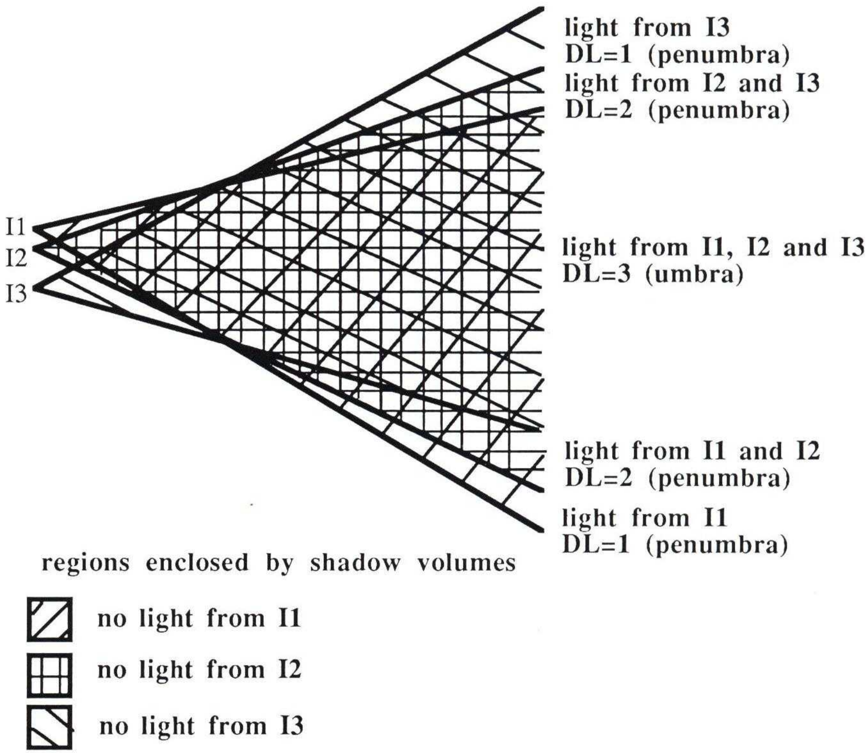

Fig.9.9. An example of distributed light with superposition of shadow volume

9.6.3 Umbrae and penumbrae for linear sources

The shadow caused by a linear source, as described in Sect. 7.1.4, also consists of umbrae and penumbrae. Nishita et al. (1985) propose a method to determine the shadow volumes, which is easy to implement because they require objects to be sets of convex polyhedra.

Consider a linear light source with two end points $\mathbf{Q_1}$ and $\mathbf{Q_2}$, a convex polyhedron V, and a face F farther from the light source than V. We define C_1 and C_2 as the two silhouette contours of V when viewed from $\mathbf{Q_1}$ and $\mathbf{Q_2}$. Let C_1' be the projection of C_1 onto F and C_2' be the projection of C_2 onto F. The umbra area is the intersection of C_1' and C_2'. The penumbra area is the minimum convex polygon surrounding C_1' and C_2'. Similarly, if we consider two shadow volumes U_1 and U_2 corresponding to $\mathbf{Q_1}$ and $\mathbf{Q_2}$, the umbra volume is defined as the intersection of U_1 and U_2 and the penumbra volume is the minimum convex volume surrounding U_1 and U_2.

Shadow detection is performed by using the relationships between the shadow volumes and each of the faces. Faces that are totally invisible from $\mathbf{Q_1}$ and $\mathbf{Q_2}$ are ignored. For faces that are visible from either $\mathbf{Q_1}$ or $\mathbf{Q_2}$, the algorithm to obtain the penumbra and umbra boundaries (stored as penumbra loop and umbra loop) works as follows:

if the penumbra volume of the polyhedron V and the face F do not intersect
then
  {there is no shadow cast on F}
else
  Project the contour lines C_1 and C_2 onto F giving C_1' and C_2'
  if one volume U_k, k=1,2 encloses the other one
  then
    C_k' is the penumbra loop and the other projected contour the umbra loop
  else {volumes U_1 and U_2 intersect each other}
    Determine two points $\mathbf{P_L}$ and $\mathbf{P_R}$, common to C_1' and C_2' {see Nishita et al. 1985}
    Divide C_1' into two separated strings S_{1A} and S_{1B} using $\mathbf{P_L}$ and $\mathbf{P_R}$
    Divide C_2' into two separated strings S_{2A} and S_{2B} using $\mathbf{P_L}$ and $\mathbf{P_R}$
    Penumbra and umbra loops are obtained by connecting two strings S_{jX}, j=1,2 X=A,B

9.6.4 Nishita-Nakamae method for interreflection and shadows

Nishita and Nakamae (1985) introduce a method for indirect lighting with interreflection and shadows results. In order to save time, the penumbrae and umbrae are predetected before scanning for hidden surface removal. As in their original method, first shadow volumes for penumbrae and umbrae formed by a convex polyhedron and a light source are obtained. Then, the penumbrae (or umbrae) on each face are obtained as the intersection areas of penumbra (or umbra) volumes and the face. The procedure works as follows:

1. Input of 3D objects
2. Subdivision of faces into subfaces
3. Classification of faces of each polyhedron for shading into three classes:
   - Faces receiving light from the whole region of the source
   - Faces receiving light from a part of the source
   - Faces receiving no light from the source
4. Obtaining penumbra and umbra volumes
5. Calculation of penumbrae and umbrae on each face
6. Calculation of interreflection of light
7. Determining priority of visibility for a given viewpoint
8. Hidden-surface removal and calculation of direct illuminance at each pixel

The most interesting step is the calculation of interreflection taking into account shadows. As faces in a room are subdivided into subfaces, a test to determine whether or not objects exist between every pair of subsurfaces is required. To reduce the complexity of the test, it is only executed for the four corners of each subsurface. This means that both shadow calculations and illuminance calculation are done at the corner points of subsurfaces. The shadow function v_{ij} for adding shadow influence between corner points $\mathbf{P_i}$ and $\mathbf{P_j}$ is given by:

v_{ij} = 1 if there are no objects between $\mathbf{P_i}$ and $\mathbf{P_j}$
0 if blocked

The weighting coefficients w_j for the illuminance calculation at $\mathbf{P_j}$ is given by:

w_j =
1/4 if $\mathbf{P_j}$ is a corner point of polygon
1/2 if $\mathbf{P_j}$ is a point on the edge of polygon
1 otherwise

Assume a point $\mathbf{P_i}$ on the face F_A, a point $\mathbf{P_j}$ on the face F_B, and the subsurface A_j surrounding $\mathbf{P_j}$. As the subsurfaces are very small, then the form factor F_{ij} (see Eq. 7.33) may be approximated as follows:

$$F_{ij} = \frac{d_{Aj}\ d_{Bi}}{\pi r_{ij}^{4}} A_j \tag{9.1}$$

where d_{Aj} is the perpendicular distance between the face F_A and the point $\mathbf{P_j}$ and d_{Bi} is the perpendicular distance between F_B and $\mathbf{P_i}$.

The procedure for calculating the shadow functions v_{ij} is complex. Assume a line segment $s = \mathbf{P_m P_n}$ consisting of $\mathbf{P_j}$ (j=m to n) on a face of a polyhedron F_B as a linear light source. The shadow function v_{ij} for a point $\mathbf{P_i}$ on a face of another polyhedron F_A is obtained as follows:

for each v_{ij}
  v_{ij}:=1 (j=m to n)
Remove polyhedra behind F_A and F_B
Remove the polyhedra not intersecting the triangle $\mathbf{P_iP_mP_n}$ by using bounding boxes of the triangle and the polyhedra
Obtain the polyhedra intersecting the plane of the triangle $\mathbf{P_iP_mP_n}$
Extract silhouette contours C_k of the polyhedra viewed from $\mathbf{P_i}$
Calculate the intersection between C_k and s
Search the invisible parts PT of s enclosed by C_k when viewed from $\mathbf{P_i}$
for each point **P** of PT
  v_{ij} := 0

Figure 9.10 shows how the shadow function v_{ij} is calculated; examples of w_j values are also shown. An example is shown in Fig.9.11.

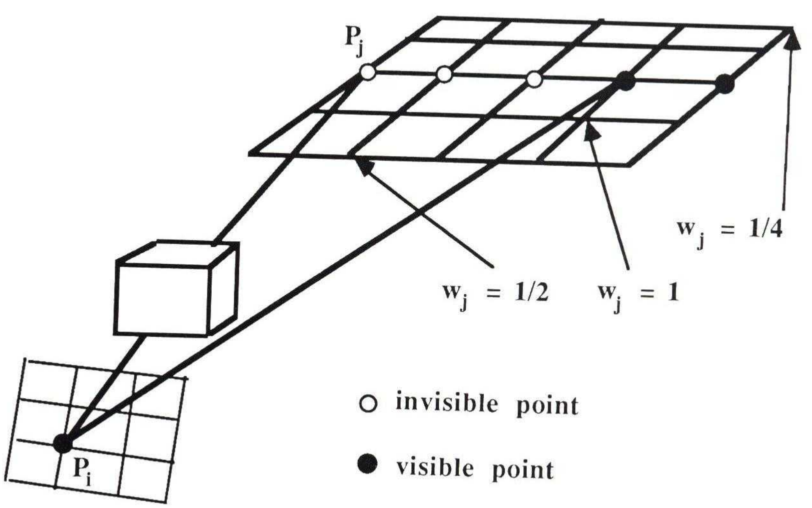

Fig. 9.10. Calculation of the Nishita-Nakamae shadow function

Fig.9.11. Night scenes illuminated by two rectangular sources. Courtesy of T. Nishita, Fukuyama University and E. Nakamae, Hiroshima University