



Fig. 3. The neighborhood and corresponding difference code data for vertex 314. The first entry, 6, is the degree of the vertex. Other entries are the offsets of the neighbors.

vertices in a preprocessing phase which we will discuss later.

Once the vertices are relabeled, the link of a vertex can be represented by concatenating the code for its degree to the codes for the differences of its neighbors. (See Figure 3 for an example.) If a vertex has a link consisting of multiple cycles or paths (as can occur in a pseudomanifold), this link set can be represented by putting the cycles/paths one after the other with a count before each. If data is associated with some of the simplices, this can be interleaved with the codes for the neighbors. The resulting vertex encodings are stored in fixed-length blocks; if an encoding is larger than will fit in one block, multiple blocks may be formed into a linked list to hold the encoding. Our data structure makes use of a hashing technique to minimize the size of the pointers used in these linked lists.

When the data structure is queried, the code for the corresponding vertex is decompressed. When an update is made, the code for the corresponding vertices is decompressed, modified, and then compressed again.

Compression of the 3D data structure is similar except that the data structure stores the link around representative edges rather than around vertices. For each vertex the data structure stores a list of that vertex's representative out-edges, with pointers to the links of those out-edges. These pointers are compressed using the same hashing technique as above.

5.2. *Generating Labels*

If all the vertices are known before the algorithm begins, our algorithm can relabel them using a technique based on x - y cuts. Given a set of points, the technique first finds which of the x and y axes has the greatest diameter. It finds the approximate median in that coordinate and partitions the points on either side of that median. The points on one side are labeled first, then the points on the other side. This is