Two totally identical documents would generate a square image with a single dark band along the prime upper-left, lower-right diagonal.

Figure 9-7 indicates that the start of both documents is similar; following that, there is some original material in the lower document. The two areas mirrored across the diagonal suggest some simple rearrangement of the Web upper document in the lower document, following which there is a long similar section. The document ends with some original material in the lower document before another small similar section.

Projecting the dark areas onto a horizontal line indicates how much of the upper document appears in the lower one. Likewise, a vertical projection indicates how much of the lower document appears in the upper.

**Tool: SIM** SIM, or Software Similarity Tester, is used to detect the similarity between two computer programs. It examines the correctness, style, and uniqueness of the program.

Each input C source file is passed to the lexical analyzer to generate a compact structure in terms of streams of integers known as tokens. Each token symbolizes either arithmetical or logical operations like a punctuation symbol, a C macro, a keyword, a numeric or string constant, a comment, or an identifier.

After the two source files get tokenized, the token stream of the second program is divided into sections. Each section represents a module of the original program, and each module gets compared with the token stream of the first program separately. This technique enables SIM to detect similarities even if the positions in the module are changed.

When the tokens are compared, their alignment is scored as follows:

- A match involving two identifier tokens scores 2; other matches score 1.
- A gap scores –2.
- A mismatch involving two identifiers scores 0; other mismatches score –2.

The total alignment score is calculated from the individual score for each block, and then it is normalized. This process is shown in Figure 9-8.

**Tool: PLAGUE** PLAGUE, or Plagiarism in University Environments, is an open repository of resources assisting students and academics in detecting plagiarism in software code and protecting themselves against it.