

**[0074]** The message  $m$  is provided as input to a translation function  $g$ . The translation function  $g$  translates the message  $m$  into a codeword  $c$  in codeword space **20**. The translation function  $g$  represents a one-to-one mapping of a message  $m$  from message space **10** to a codeword  $c$  in codeword space **20**. Accordingly, for each message  $m$ , there is one corresponding codeword  $c$ . An error-correcting code for use with a binary set of messages  $M$  that are  $k$ -bits in length contains a set of codewords  $C$  including  $2^k$  codewords since there is one codeword  $c$  for each of the  $2^k$  messages. The operation of the translation function  $g$  can be described mathematically as  $g: M \rightarrow C$ . The set of codewords  $C$  in codeword space **20** may be described mathematically as  $C \subseteq \{0, 1\}^n$  where each codeword  $c$  in the set of codewords  $C$  is a binary  $n$ -bit string. Generally, the message  $m$  is different from codeword  $c$  at least because codeword  $c$  contains redundant elements. If a codeword  $c$  contains redundant elements, the length of the codeword  $c$  bit string  $n$  will be greater than the length of the message  $m$  bit string  $k$ .

**[0075]** The codeword  $c$  is transmitted **30** over a communication channel. Noise **35** may be introduced during transmission **30** so that a corrupted codeword  $i$ , which is generally some variation of codeword  $c$ , is received at the receiving end of the communication channel. The corrupted codeword  $i$  is provided as input to a decoding function  $f$ . The decoding function  $f$  reconstructs the codeword  $c$  from the corrupted codeword  $i$ . The redundant elements of the codeword  $c$  allow the decoding function to perform this reconstruction.

**[0076]** The decoding function  $f$  maps a corrupted codeword  $i$  to a codeword  $c$  in the set of codewords  $C$ . A corrupted codeword  $i$  may be an arbitrary  $n$ -bit binary string. When the decoding function  $f$  is successful, it maps a corrupted codeword  $i$  to the nearest codeword  $c$  in the set of codewords  $C$ . In this context, the nearest codeword  $c$  is the codeword  $c$  that is the closest by an appropriate metric from the corrupted codeword.

**[0077]** The task of mapping an arbitrary string to its nearest codeword is known as the maximum likelihood decoding problem. Practical classes of codes with polynomial-time solutions to this broad problem are at present unknown. Conventional decoding functions perform a more limited task in that they successfully decode any word that lies within a certain radius of some codeword. Such decoding functions can be used in embodiments described herein.

**[0078]** Generally, when a decoding function  $f$  fails, it outputs  $\phi$ . (Some error correcting codes may operate somewhat differently. For example, list decoding functions  $f$  yield a set of candidate codewords, rather than a single correct one. The underlying principles remain the same in such settings.) The operation of the decoding function  $f$  can be described mathematically as  $f: \{0, 1\}^n \rightarrow C \cup \{\phi\}$ . The reverse translation function  $g^{-1}$  is used upon receipt of a reconstructed codeword  $c$  to retrieve the original message  $m$ .

**[0079]** The robustness of an error-correcting code depends on the minimum distance of the code. In this description, Hamming distance and Hamming weight will be used as an example of a way to measure the minimum distance of a binary block code. If the Hamming weight of an  $n$ -bit binary string  $u$  is defined to be the number of '1' bits in  $u$  and the Hamming weight of an  $n$ -bit string  $u$  is denoted by  $\|u\|$ , then

the Hamming distance between two binary bit-strings  $u$  and  $v$  is defined to be the number of bits in which the two strings differ. The Hamming distance between two binary bit-strings  $u$  and  $v$  is denoted by  $\|u \oplus v\|$ .

**[0080]** The minimum distance of a convolution code is defined without reference to Hamming distance or Hamming weight. The use of Hamming distance or Hamming weight as an example here does not indicate any intent to limit an embodiment to these metrics as the only appropriate metrics of the minimum distance of an error-correcting code. Another metric for a set of sequences whose elements are nonbinary, for example, would be the  $L^\infty$  norm, a measure of the maximum difference between elements. The  $L^\infty$  difference between the sequence  $u = \{3, 4, 5\}$  and the sequence  $v = \{10, 5, 1\}$  would be 7.

**[0081]** A decoding function  $f$  has a correction threshold of size  $t$  if it can correct any set of up to  $t$  errors. In other words, the decoding function  $f$  can successfully decode any corrupted codeword  $i$  whose errors are less than or equal to the correction threshold  $t$  of the decoding function. The error in a corrupted codeword  $i$  can be described as the offset  $\delta$  from the nearest codeword  $c$ . In a binary block code where the Hamming weight of the corresponding offset  $\delta$  is less than or equal to the bit correction threshold  $t$ , the decoding function  $f$  will successfully decode a corrupted codeword  $i$  to a codeword  $c$  in the set of codewords  $C$ . This concept is expressed mathematically as follows: given  $c \in C$  and  $\delta \in \{0, 1\}^n$  with  $\|\delta\| \leq t$ , then  $f(c + \delta) = c$ .

**[0082]** Generally, the Hamming distance between any two codewords in the set of codewords  $C$  is greater than two times the correction threshold ( $2t$ ). If the Hamming distance between codewords were not greater than  $2t$ , then a corrupted codeword  $i$  would exist that could be decoded into more than one codeword. The neighborhood of a codeword  $c$  comprises the subset of all possible corrupted codewords that the decoding function  $f$  maps to the codeword  $c$ . The decoding function  $f$  is generally such that any corrupted codeword  $i$  in  $f^{-1}(c)$  is closer to the codeword  $c$  than to any other codeword.

**[0083]** For example, given a message  $m$  that is one bit long ( $k=1$ ), a codeword  $c$  that is three bits long ( $n=3$ ), a set of two codewords  $C$  consisting of 000 and 111 ( $C = \{000, 111\}$ ), and a decoding function  $f$  that computes majority, the correction threshold  $t$  for the decoding function  $f$  equals one bit error ( $t=1$ ). The decoding function  $f$  maps a corrupted codeword  $i$  consisting of three binary bits to 000 if at least two bits are 0 and to 111 if at least two bits are 1. The correction threshold  $t$  indicates that the decoding function  $f$  can correct a single bit error because changing a single digit in either 000 or 111 does not change the majority.

**[0084]** The coding efficiency of an error-correcting code is the ratio of the bit length of a message  $m$  to the bit length of a codeword  $c$ . The coding efficiency ( $k/n$ ) measures the degree of redundancy in the error-correcting code. The lower the coding efficiency, the more redundancy in the codewords. The error-correcting code described in this example has a coding efficiency of  $1/3$ . In general, codes that can correct a large number of errors have a low coding efficiency.

**[0085]** Error-correcting codes may be defined for non-binary spaces as well, and it is intended that the principles described here be extended to such spaces.