An adaptive image authentication scheme for vector quantization compressed image

Jun-Chou Chuanga, Yu-Chen Hub,*

a Department of Computer Science and Communication Engineering, Providence University, 200 Chung Chi Rd., Taichung 43301, Taiwan, ROC
b Department of Computer Science and Information Management, Providence University, 200 Chung Chi Rd., Taichung 43301, Taiwan, ROC

ARTICLE INFO

Article history:
Received 24 November 2010
Accepted 30 March 2011
Available online 2 April 2011

Keywords:
Tamper detection
Image authentication
Watermark
Image compression
Vector quantization
Index table
Random values
Random seed

ABSTRACT

This paper proposes an image authentication scheme which detects illegal modifications for image vector quantization (VQ). In the proposed scheme, the index table is divided into non-overlapping index blocks. The authentication data is generated by using the pseudo random sequence. Our scheme can adaptively determine both the size of the authentication data and the number of the indices in each index block. Then, the selected indices are used to embed the secret data to generate the embedded image.

To authenticate the given VQ compressed image, two sets of the authentication data are needed to perform the tamper detection process. One set is generated by using the pseudo random number sequence. The other set is extracted from the compressed image. The experimental results demonstrate that the proposed scheme achieves acceptable image quality of the embedded image while keeping good detecting accuracy.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Due to the coming of the digital era, it is very convenient to get information from the Internet. Among different media, the digital images are widely used on the Internet. The digital images are generally stored in the compressed formats. From the literature, the image compressed methods can be classified into two main categories: lossless compression and lossy compression. Special-purposed images such as the military images and the medical images are generally compressed by using the lossless image compression techniques. The lossy compression techniques such as block truncation coding, vector quantization (VQ) [1–6], subband coding, fractal coding, JPEG, and JPEG2000 are often used to compress the general-purposed images.

In order to protect the image integrity, the concept of image authentication [8] had been proposed.

The image authentication schemes can be generally classified into two categories: signature-based schemes [9–12] and fragile watermark-based schemes. In a signature-based scheme, the host image is first processed by using the hash function and the hashed result is then encoded via the public key cryptosystem to generate the signature of the image. Finally, the signature of the image is stored. When the image is to be authenticated, the stored signature in a different place is required and it is then compared to the other signature that is generated from the given image to detect the tamper areas.

In a fragile watermark-based scheme [13–23], the watermark bits are embedded into the host image for verifying the authenticity of the image. In other words, there is no need to store additional data in a different place. When one image is to be authenticated by using the fragile watermark scheme, the watermark is extracted from the given image to verify the image integrity.


The proposed tamper detection scheme works on the VQ encoded images. The compressed image of VQ consists of the indices for the image blocks. Those VQ indices are collected together to form the index table. In the proposed scheme, the index table is divided into non-overlapping index blocks. For each index block, we embed the authentication codes that are generated by the pseudo random number sequence into some selected VQ indices. To authenticate the given image, two sets of the authentication data are needed to perform the tamper detection process. One set is generated by using the pseudo random number sequence. The other set is extracted from the compressed image. By comparing these two sets, the tampered areas can be detected.

The rest of this paper is organized as follows. We will review the vector quantization scheme in Section 2. Section 3 will present the proposed scheme including the secret code generation procedure, the secret embedding procedure, and the image detecting procedure. The experimental results and analysis will be discussed in Section 4. Finally, Section 5 will address the conclusions and discussions.

2. The vector quantization scheme

The vector quantization scheme is a lossy image compression for grayscale images. It consists of three main procedures: codebook generation, image encoding, and image decoding. The goal of the codebook generation procedure is to design a set of representative codewords for image encoding/decoding. From the literature, the LBG algorithm [1] that was proposed by Linde et al. is the most commonly used algorithm for codebook generation. Before the LBG algorithm is executed, we need to determine the codebook size \( N \), the size of image block \( n \times n \), and the training set of images.

To generate a codebook of \( N \) codewords, multiple training images are selected. Each training image is divided into non-overlapping image blocks of \( n \times n \) pixels. Then, \( N \) image blocks from the training image blocks are chosen as the initial codewords. The set of initial codewords is also called the initial codebook. After the initial codebook is generated, an iterative process for several rounds is executed to generate the codebook in the LBG algorithm. In each round, the data clustering procedure and the centroid updating procedure are executed.

In the data clustering procedure, the closest codeword of each training image block in the codebook is to be determined. To find out the closest codeword of each training image block in the codebook, a total number of \( N \) squared Euclidean distances are calculated. Then, the closest codeword is determined by the minimal squared Euclidean distance from itself to the training image block. After the data clustering procedure is performed, all these training image blocks are classified into \( N \) groups. In the centroid updating procedure, the mean vector of the training image blocks in each group is computed. These \( N \) calculated image vectors in turn form the new codebook of this round. By repeatedly performing the same process in each round, a representative set of codewords can be generated.

In the image encoding procedure, a given grayscale image is divided into non-overlapping image blocks of \( n \times n \) pixels. Each \( n \times n \) image block can be treated as a \( k \)-dimensional image vector where \( k = n \times n \). To encode each image block, we search the codebook of \( N \) codewords to find the closest codeword for the image block based on the minimal squared Euclidean distance criterion. The codeword in the codebook that has the minimal squared Euclidean distance to the image block is selected. Then, the index of the closest codeword in the codebook is recorded. After each image block is sequentially compressed, the set of the indices, also called the index table, is obtained.

To rebuild the compressed image, the same codebook of \( N \) codewords is stored and used. To recover each image block, the index of \( \log_2 N \) bits of each compressed block is sequentially extracted from the index table. The codeword of each index in the codebook is used to rebuild the compressed image block. In other words, the decoder only needs to perform simple table lookup operations to reconstruct the encoded image.

3. The proposed scheme

The goal of the proposed scheme is to protect the integrity of the grayscale images that are compressed by VQ. To protect the integrity of the VQ compressed images, the index table is divided into non-overlapping blocks. Then, the secret data is embedded into some indices of each index block. The number of indices in each block that will be used to embed the secret data can be adaptively selected. In addition, the number of the secret bits to be embedded into the index can be adaptively determined.

3.1. The secret data generation procedure

Suppose the grayscale image of \( W \times H \) pixels is divided into \( n \times n \) non-overlapping image blocks and it is compressed by VQ using the codebook of \( N \) codewords. The index table of \( w \times h \) indices has already been generated where \( w = W / n \) and \( h = H / n \). Each index is stored in \( \log_2 N \) bits. The index table of \( w \times h \) indices is divided into non-overlapping blocks of \( m \times m \) indices. The total number of index blocks (\( t_{\text{nb}} \)) can be calculated as follows:

\[
t_{\text{nb}} = \frac{w \times h}{m \times m}.
\]  

Let \( \text{ino} \) denote the number of the indices in each block that will be used to embed the secret data where \( 1 \leq \text{ino} \leq m \times m \). Let \( \text{eb} \) denote the embedded bits of each selected index. A total number of \( t_{\text{nb}} \times \text{ino} \) secret data of \( \text{eb} \) bits is to be generated. To generate the secret data, the pseudo random number generator (PRNG) with a predefined seed is used to generate \( \text{tnib} \times \text{ino} \) random values. Each random value \( \text{rv} \) is then converted to a secret code \( q \) of \( \text{eb} \) bits by using the following equation:

\[
q = r \mod 2^\text{eb}.
\]  
The secret code \( q \) will be embedded into the selected index in the data embedding procedure.

3.2. The data embedding procedure

In the data embedding procedure, \( \text{ino} \) secret codes of \( \text{eb} \) bits are to be embedded into each block of \( m \times m \) indices. Note that the indices in the index table are stored in \( \log_2 N \) bits. To select \( \text{ino} \) indices in each index block, the displacement (\( \text{disp} \)) value to select the index in the block is computed as follows:

\[
\text{disp} = \frac{m \times m}{\text{ino}}.
\]  

In the beginning, the first index in the block is selected. By regularly selecting the index with \( \text{disp} \) away from the first index, a total number of \( \text{ino} \) indices in the \( m \times m \) index blocks can be selected. In other words, the indices with number 1, \((1 + \text{disp}), (1 + 2 \times \text{disp}), \ldots \), are sequentially collected until the required number of indices is searched.
Fig. 1 lists an example of index selection for a 2 × 2 block. In Fig. 1(a) disp equals 4 when we want to select one index in the 2 × 2 index block. First, the first index in the block is selected. Then, no more indices are to be selected because only one index is needed to be selected. In Fig. 1(b) the value of disp equals 2. Similarly, the first index is selected. Next, the index to be selected is (1 + 2). In Fig. 1(d) all the four indices in the 2 × 2 index block are selected because disp equals 1. The selected indices in Fig. 1 with different ino values are shaded.

To embed each eb-bit secret code q into the log2 N-bits index x, the eb-bit reminder of x can be computed as follows:

\[ r = x \mod 2^{eb} \]  

(4)

If r is equal to q, no change is to be made on the index x. Otherwise, we need to find out the most similar index (si) of x that satisfies \( si \mod 2^{eb} = q \). The diagram of data embedding is listed in Fig. 2.

3.3. The tamper detecting procedure

The goal of the tamper detecting procedure is to detect whether a VQ encoded image is modified or not. To detect the tamper areas of the index table of \( w \times h \) indices, we need to know the random number seed, the size of index block (\( m \times m \)), the number of selected index (ino) for each block, and the number of embedded bits (eb) for each index. In addition, two sets of authentication codes are to be generated. One set is generated by using the random values induces with the random number seed. The other set is extracted from the received index table.

To generate the first set of the authenticated codes, the total number of the index block (tnib) is calculated by Eq. (1). A total number of tnib × ino random values are generated by using the random number seed. Each random value \( rv \) is converted into the authentication code \( q \) of eb bits by Eq. (2). The set of the authentication code is the same as that generated by using the secret data generated procedure when the same random number seed is used here.

The second set of the authentication codes will be extracted from the index table of \( w \times h \) indices. The index table of \( w \times h \) indices is partitioned into non-overlapping blocks of \( m \times m \) indices. In each \( m \times m \) block, ino indices are selected to extract the authentication codes. To select the indices in each index block, the displacement value to select the indices in each block can be computed by Eq. (3). Then, these ino indices in each block can be determined. The authentication code \( ac \) for each selected index \( x \) can be computed as follows:

\[ ac = x \mod 2^{eb} \]  

(5)

When two sets of the authentication codes are available, we can determine whether each selected index \( x \) is tampered or not. Recall that \( ac \) denotes the eb-bit authentication code that is generated from \( x \) and the corresponding authentication code \( q \) is generated from the random value. If \( ac \) is equal to \( q \), we assume that the index \( x \) is not tampered. Otherwise, the index \( x \) is treated as a modified index. The diagram of index tamper detection is shown in Fig. 3.

When these ino selected indices in each \( m \times m \) block are checked, we can determine whether each block is tampered or not. If the number of the tampered indices is greater than or equal to one, the \( m \times m \) block is classified as a tempered block. Otherwise, it is classified as a clear block. The corresponding white pixel and black pixel are used to represent the tempered block and the clear block, respectively. The detected result for the index table can be then generated by collecting the resultant pixels. The size of the detected result is of size \( (w/m) \times (h/m) \). For example, the detected result equals 64 × 64 when the index table and the block size of 128 × 128 with 2 × 2 indices are used, respectively.

The detected result needs to be refined because it is possible that some modified indices may have the same eb-bit remainders as the original indices. The possibility of this situation decreases...
as the \( eb \) value increases. To improve the detecting efficiency, an iterative mechanism is designed to refine the roughly detected result of size \( \left( \frac{w}{m} \right) \times \left( \frac{h}{m} \right) \).

In each round, we need to check whether each white pixel will be changed to a black pixel or not. Four conditions that are listed in Fig. 4 are checked for each while pixel in DR. In Fig. 4 \( p \) denotes the selected white pixel to be processed. In the first condition as shown in Fig. 4(a) if the adjacent up and down pixels of \( p \) are black, \( p \) will be modified to the black pixel. In the second condition as shown in Fig. 4(b) if the adjacent left and right pixels of \( p \) are black, \( p \) will be modified to the black pixel. Similarly, two additional conditions for the 45\(^\circ\) and 135\(^\circ\) splay black pixels of \( p \) are listed in Fig. 4(c) and (d), respectively.

After checking each white pixel to determine whether it should be modified to a black pixel or not, the total number of modified pixels can be computed. If the number of modified pixels is greater than or equal to 1, the same process for the tamper refinement is executed. Otherwise, the tamper refining process is stopped. The size of the refined result is of \( \left( \frac{w}{m} \right) \times \left( \frac{h}{m} \right) \) pixels. To generate the final detected results for the given index table of \( w \times h \) indices, each pixel in the refined result is expanded into the block of \( m \times m \) pixels. Each pixel in the expanded image block is set to the color of its corresponding pixel in the refined result.
3.4. Discussion

The proposed scheme embeds $eb$-bit secret codes into some selected indices in the index table. The block size $m \times m$ that is used to partition the indices in the index table and the number of embedding bits for each index, $eb$, affect the image quality of the embedded image. If the block size equals $1 \times 1$, each index in the index table is used to embed the secret codes. When a large block size is set and only some indices in each block are selected to embed the secret codes, the better image quality of the embedded image will be obtained. However, the detecting accuracy of the proposed scheme becomes worse when a large block size is set.

The number of embedding bits for each selected index, $eb$, also affects the detecting accuracy. The detecting accuracy increases when a large $eb$ value is set. However, the image quality of the embedded image decreases when a large $eb$ value is used. That is because the searched similar index having the same $eb$-bit remainder as the index may not be really close to the index when a large $eb$ value is set. Therefore, it is a great challenge to balance the image quality and the detecting accuracy in the proposed scheme.

4. Experimental results

Our experiments are performed on Windows XP PC with an Intel Core Duo 2.2 GHz CPU and the 512 MB RAM. The testing programs are implemented by using Bloodshed Dev C++. In our experiments, six grayscale images of 512 x 512 pixels “Airplane”, “Girl”, “Goldhill”, “Lenna”, “Pepper”, and “Toys”, as shown in Fig. 5 are used. Among them, three images “Airplane”, “Goldhill”, and “Toys” are used as the training images to design the VQ codebooks by using the accelerated LBG algorithm [2]. In the simulations, the size of the image block is set to $4 \times 4$. Besides, the termination threshold of the LBG algorithm for codebook generation is set to 0.001. The size of the index table of each VQ-compressed image is of $128 \times 128$ indices.

Fig. 6 lists results of the image qualities of VQ decoded images. It is shown that the image quality of the decoded image increases when a larger codebook is used in VQ. The length of the index equals $\log_2 N$ when the codebook of $N$ codewords is used in VQ. The average image qualities of VQ are greater than 30 dB when the sizes of the codebooks are greater than or equal to 128.

Fig. 8. The embedded images of the proposed scheme when the block size is set to $1 \times 1$.

Fig. 9. Example of the tampered image and the difference image.
4.1. Tamper test for $1 \times 1$ index blocks

In the simulations, two possible block sizes to partition the index table are tested. First, the index table is divided into index blocks of size $1 \times 1$. The average image qualities of the VQ-compressed images and the embedded images of the proposed scheme are listed in Fig. 7 when the index table is divided into $1 \times 1$ index blocks. From the results in Fig. 7 the image qualities of the embedded images decrease as the $eb$ value increases. Compared to the VQ scheme, 2.472 dB, 4.180 dB, and 5.660 dB image quality degradations are found in the proposed scheme when the $eb$ values are 1, 2, and 3, respectively. That is because any modifications on the compressed codes significantly affect the image quality for the VQ compressed images.

The embedded images of the proposed scheme using the codebooks of 256 codewords and 1024 codewords are listed in Fig. 8. The embedded images that were compressed by VQ with the codebook of 1024 codewords provide better visual qualities.
than those with the codebook of 256 codewords. The visual qualities of the embedded images become worse when the \(eb\) value increases with the same codebook.

To generate the tampered image, a rose is added on the hat of each embedded image. In the simulations, the same modification is made on these six embedding images in Fig. 8 to generate the tampered images. The tampered image for the embedded image as shown in Fig. 8(b) is listed in Fig. 9(a). In this test, 8231 pixels are modified when the rose is added. A total number of 579 blocks of \(4 \times 4\) pixels is affected. The difference image between the embedded image and the tampered image is listed in Fig. 9(b).

The detected results of the proposed tamper detection procedure without refinement are listed in Fig. 10. From the results shown in Fig. 10(a)–(c) it is shown that the number of the white spots within the modified area decreases when the \(eb\) value increases. That is because the probability that the original index and the modified index have the same \(eb\)-bit remainders decreases when the \(eb\) value increases. The similar situation can be found from Fig. 10(d)–(f). From the results as shown in Fig. 10 it is clear that the total number of tampered images increases with the increment of the \(eb\) value.

The detected results of the proposed tamper detection procedure with the iterative refinement mechanism are shown in Fig. 11. No white spots are found within the modified area in these six detected images. Compared to the difference image as shown in Fig. 9(b) the tampered area is clearly detected. However, some modified indices in the boundary of the tampered area cannot be detected by using the proposed tamper detection procedure.

<table>
<thead>
<tr>
<th>(eb)</th>
<th>Size</th>
<th>(N = 256)</th>
<th>(N = 1024)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No refinement</td>
<td>With refinement</td>
<td>No refinement</td>
</tr>
<tr>
<td>1</td>
<td>265</td>
<td>500</td>
<td>255</td>
</tr>
<tr>
<td>2</td>
<td>387</td>
<td>515</td>
<td>397</td>
</tr>
<tr>
<td>3</td>
<td>450</td>
<td>524</td>
<td>462</td>
</tr>
</tbody>
</table>

Table 1: Total number of the tampered indices of the proposed scheme for \(1 \times 1\) index block.

![Fig. 12. Average image qualities of the compressed images of VQ and the embedded images of the proposed scheme.](image-url)

![Fig. 13. Results of the tamper detection procedure without refinement when \(ino\) equals 1.](image-url)
The total number of the tampered indices for each tampered image is listed in Table 1. Theoretically, the probability that the original index and the modified index have the same $eb$-bit remainder equals $2^{-eb}$. In other words, the detection accuracy for the proposed tamper detection procedure without refinement equals 50%, 75%, and 87.5% when the $eb$ values are 1, 2, and 3, respectively. The detection accuracy of the proposed scheme without refinement as shown in Table 1 approximately consists with the theoretical analysis.

4.2. Tamper test for $2 \times 2$ index blocks

In the second tamper test, the index table is divided into blocks of $2 \times 2$ indices. The average image qualities of the compressed images of VQ and the embedded images of the proposed scheme are listed in Fig. 12. From the results, the image qualities of the embedded images decrease as $ino$ and $eb$ values increase. Compared to the VQ scheme, 0.810 dB, 1.566 dB, and 2.345 dB image quality degradations are found in the proposed scheme with $ino$.
equal to 1 when the \( eb \) values are 1, 2, and 3, respectively. In addition, 1.436 dB, 2.614 dB, and 3.753 dB image quality losses are found in the proposed scheme with \( ino \) equal to 2 when the \( eb \) values are 1, 2, and 3, respectively.

The same modification as described in Section 4.1 is used here to generate the tampered images. The roughly detected images and the refined detected images of the proposed scheme when one index is selected in each \( 2 \times 2 \) index block are shown in Figs. 13 and 14, respectively. According to the results, the total numbers of detected pixels in the detected results in Fig. 13 are less than those in Fig. 10. That is because only one index is selected to embed the secret code in each index block of size \( 2 \times 2 \) to detect the tampered results in Fig. 13. Besides, the detected results in Fig. 11 are better than those in Fig. 14.

The roughly detected images and the refined detected images of the proposed scheme when two indices are selected in each \( 2 \times 2 \) index block are shown in Figs. 15 and 16, respectively. According to the results, the numbers of tampered indices in the rough images in Fig. 15 are greater than those in Fig. 13. In addition, the detected areas in Fig. 16 are more similar to the tampered area as shown in Fig. 9 than those in Fig. 14.

The total number of the tampered indices for each tampered image in the proposed scheme when the index block size equals \( 2 \times 2 \) are listed in Table 2. The numbers of the tamper indices of the proposed scheme without refinement are less than those in the proposed scheme with tamper refinement. The number of tampered indices increases when the \( eb \) value increases. In addition, the codebook size used in the proposed scheme does not affect the detection accuracy. Compared to the results of the first tamper test as shown in Table 1 and the proposed scheme with \( ino \) equal to 4, we find that the numbers of tampered indices found are approximately the same. That is because all the indices in the index table are used to embed the secret codes.

When the size of index block is set to \( 2 \times 2 \) in the proposed scheme, the false detection problem is found. Recall that 8231 pixels are modified in each embedded image when the rose is added.

### Table 2

<table>
<thead>
<tr>
<th>( eb )</th>
<th>( ino = 1 )</th>
<th>( ino = 2 )</th>
<th>( ino = 3 )</th>
<th>( ino = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N = 256 )</td>
<td>No refinement</td>
<td>With refinement</td>
<td>No refinement</td>
<td>With refinement</td>
</tr>
<tr>
<td>1</td>
<td>67</td>
<td>412</td>
<td>59</td>
<td>460</td>
</tr>
<tr>
<td>( ino = 2 )</td>
<td>119</td>
<td>540</td>
<td>128</td>
<td>568</td>
</tr>
<tr>
<td>( ino = 3 )</td>
<td>201</td>
<td>564</td>
<td>207</td>
<td>576</td>
</tr>
<tr>
<td>( ino = 4 )</td>
<td>250</td>
<td>592</td>
<td>263</td>
<td>608</td>
</tr>
<tr>
<td>( N = 1024 )</td>
<td>No refinement</td>
<td>With refinement</td>
<td>No refinement</td>
<td>With refinement</td>
</tr>
<tr>
<td>2</td>
<td>102</td>
<td>464</td>
<td>94</td>
<td>516</td>
</tr>
<tr>
<td>( ino = 2 )</td>
<td>187</td>
<td>568</td>
<td>194</td>
<td>584</td>
</tr>
<tr>
<td>( ino = 3 )</td>
<td>291</td>
<td>572</td>
<td>294</td>
<td>604</td>
</tr>
<tr>
<td>( ino = 4 )</td>
<td>379</td>
<td>600</td>
<td>415</td>
<td>624</td>
</tr>
<tr>
<td>3</td>
<td>115</td>
<td>528</td>
<td>112</td>
<td>556</td>
</tr>
<tr>
<td>( ino = 2 )</td>
<td>215</td>
<td>580</td>
<td>227</td>
<td>576</td>
</tr>
<tr>
<td>( ino = 3 )</td>
<td>350</td>
<td>592</td>
<td>341</td>
<td>596</td>
</tr>
<tr>
<td>( ino = 4 )</td>
<td>438</td>
<td>604</td>
<td>480</td>
<td>616</td>
</tr>
</tbody>
</table>
to generate the tamper image. By doing so, a total number of 579 blocks with $4 \times 4$ pixels is affected. When the number of tampered indices found in the proposed scheme is greater than 579, some unchanged indices are classified intotampered indices. In the proposed tamper detection procedure, each index block of size $2 \times 2$ is classified as a modified block when the number of the tampered indices is greater than or equal to 1. Some indices in the modified index block may not be actually modified. According to the results, the false detection problem becomes worse when the $eb$ value increases.

5. Conclusions and discussion

In this paper, we proposed a novel tamper detection scheme for VQ compressed images. As well-known, the index tables are the VQ compressed results of the grayscale images. In the proposed scheme, the secret codes that were generated by using the random values were embedded into the selected indices in the index table. The number of embedded bits for each index and the number of indices in each index block can be adaptively determined by users to reach a compromise between the embedded image quality and the detection accuracy in the proposed scheme.

From the first tamper test, the image qualities of the embedded images decrease when the number of the embedded bits increases with the size of the index block set to $1 \times 1$. The clear shapes of the tampered roses can be detected for the tampered images when different $eb$ values are used. To provide good image qualities of the embedded images, we suggest that the number of embedded bits for each selected index should be set to 1 when the size of the index block is set to $1 \times 1$ in the proposed scheme.

From the second tamper test, better image qualities of the embedded images achieve when the size of the index block is set to $2 \times 2$ compared to the first tamper test. Similarly, the image qualities of the embedded images decrease when the number of selected indices increases. To compromise between the embedded image quality and the detection accuracy, it is suggested that the number of embedded bits for each selected index should be set to 2 when the size of the index block is set to $2 \times 2$ in the proposed scheme.

Acknowledgments

This research was supported by Providence University, Taichung, Taiwan under Contract the National Science Council, Taipei, ROC under Contract NSC 99-2221-E-126-004-MY2 and NSC 99-2632-E-126-001-MY3.

References