Motion Estimation with Variable Search Range
Dong-Keun Lim and Yo-Sung Ho
Kwangju Institute of Science and Technology
572 Sangam-Dong Kwangsan-gu, Kwangju, 506-712, Korea

Abstract
In this paper, we examine a new motion compensation method based on the affine transformation and derive a fast algorithm for affine transformation using vector relationship. We also develop a more effective motion estimation method than the conventional image warping method in terms of computational complexity, reconstructed image quality, and the number of coding bits. The performance of the proposed motion compensation method, which combines the affine transformation with the proposed adaptive variable search range, is evaluated experimentally. We simulate our proposed motion compensation method in a DCT-based coder by encoding CIF (Common Intermediate Format) images at bitrates below 64 kbps. The proposed method can reduce the computational complexity below about 50% of the hexagonal matching method, while maintaining the image quality comparable to the hexagonal method.

1. Introduction
One of the important developments for image coding is a mathematical model describing motion of objects [1,2]. Motion estimation and motion compensation have become more and more important, especially for low bit rate video coding. Because of the requirements for simplicity and real-time processing, translational motion models have mainly been investigated for image coding [1]. A translational movement generates a frame-to-frame displacement of the moving object. Several algorithms have been proposed to estimate the amount of displacement between two successive frames. Based on the displacement vector, motion-compensated predictive coding, motion-compensated transform coding, and motion adaptive frame interpolation can be realized.

In low bit-rate video coding, insufficient bit allocation for motion information and other image contents results in severe image degradation such as blocking and mosquito effects. Such degradation is due to inadequate motion models that cannot represent the real-world motions properly with a small number of motion parameters. To develop more effective motion compensation methods for low bit-rate video coding, we address issues of adopting more sophisticated spatial transformations than the simple translation, and of developing a new motion estimation algorithm that is more suitable for these spatial transformations.
The spatial transformations discussed in this paper include the affine transformations [2]. In the spatial transformation, actual motion is predicted with a geometric transformation of the previous image. These transformations can produce a predicted image with less blocking artifacts, and can compensate for different types of motion, including scaling and rotation that the conventional integer matching algorithm cannot cover properly. They also enable significant subjective improvement in motion prediction and consistent reduction of prediction errors. The hexagonal matching method is one of the refinement methods in image warping. It brings better image quality, but it requires a large amount of computations. In this paper, we examine new motion compensation methods based on the affine transformation. We derive fast algorithms for affine transformations using vector relationship. The reconstructed image is formed by a geometric transformation of the previous image based on the mapping of a selected set of image points between the previous and the current images. We also develop a more motion estimation method than the conventional image warping method in terms of computational complexity, comparable image quality, and the number of coding bits. By simulating a video coder which combines the proposed motion compensation algorithm with DCT-based block coding, the proposed scheme demonstrates improved performance at very low bitrates, owing to the fact that the affine transformations maintain an acceptable level of prediction errors even when the number of grid points is reduced to about half of the number of blocks used in the conventional image warping methods.

2. Fast Affine Transformation
In motion compensated video coding, a prediction image is formed from the previous image using a spatial transformation, known as image warping [2-5]. The transformation specifies the spatial relationship between each point in the previous and current images. For each pixel in the prediction image, the corresponding spatial position in the previous image is estimated, and the pixel value at that position, which may not lie on the integer pixel grid, is used as a prediction value for that pixel. Parameters for the transformation should be transmitted to the decoder as an overhead information. The whole prediction process can be divided into three steps: selection of grid points, pixel matching, and block transformation.

In image warping, we divide an image into local regions (rectangular blocks or triangular patches) and estimate a set of motion parameters for each region. A predicted image of the k-th frame $\hat{l}_k(x,y)$ is formed from the decoded image of the previous frame $\hat{q}_{k-1}(u,v)$. This process can be expressed as

$$\hat{l}_k(x,y) = \hat{q}_{k-1}(u,v) = \hat{q}(f(u,v), g(u,v))$$  (1)
where the geometric relationship between \( x'(x,y) \) and \( x(u,v) \) is defined by the transformation functions \( x = f(u,v) \) and \( y = g(u,v) \).

\[
A'(u_0,v_0) \quad Z'(u_0,v_0) \quad C'(u_2,v_2) \\
Q'(u_0,v_0) \quad P'(u_0,v_0) \quad B'(u_1,v_1) \\
(a) \text{Previous}(k-1) \text{Patch} \\
A(x_0,y_0) \quad Q(x_0,y_0) \quad Z(x_2,y_2) \\
B(x_1,y_1) \quad P(x_2,y_2) \quad C(x_2,y_2) \\
(b) \text{Current}(k) \text{Patch}
\]

Figure 1: Deformation of Patches

In image warping, the affine transformation [3-5] using the matrix operation requires intensive computation. Sometimes, since the determinant of the matrix can be zero, there is no inverse matrix. In this section, we derive a fast affine transformation using vector relationship. At first, we can show that the scale factors, \( p \) and \( q \), are preserved through the deformation process. The affine motion model is denoted as follows:

\[
\begin{align*}
x &= a_1 u + a_2 v + a_3 \\
y &= a_4 u + a_5 v + a_6
\end{align*}
\]

or

\[
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix} =
\begin{bmatrix}
a_1 & a_2 & 0 \\
a_4 & a_5 & 0 \\
a_3 & a_6 & 1
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
1
\end{bmatrix}
\]

From the previous patch and the current patch, as shown in Figure 1, we can get

\[
a_{k-1} = \frac{Q'B'}{A'B'} = \frac{(u_2 - u_0)^2 + (v_2 - v_1)^2}{(u_0 - u_1)^2 + (v_0 - v_1)^2}
\]

or

\[
a_{k-1} = \frac{Q'B'}{A'B'} = \frac{(x_2 - x_1)^2 + (y_2 - y_1)^2}{(x_0 - x_1)^2 + (y_0 - y_1)^2}
\]

For each control point, we obtain its affine transformation from Eq. (2) and Eq. (3).

\[
\begin{align*}
x_0 &= a_1 u_0 + a_2 v_0 + a_3 \\
y_0 &= a_4 u_0 + a_5 v_0 + a_6
\end{align*}
\]

\[
\begin{align*}
x_1 &= a_1 u_1 + a_2 v_1 + a_3 \\
y_1 &= a_4 u_1 + a_5 v_1 + a_6
\end{align*}
\]

\[
\begin{align*}
x_2 &= a_1 u_2 + a_2 v_2 + a_3 \\
y_2 &= a_4 u_2 + a_5 v_2 + a_6
\end{align*}
\]

The affine transformation includes translation, rotation, scaling operation. Generally speaking, since the matrix operation is not commutative, the order of concatenation is very important. If we denote \( S \) as a scaling matrix, \( R \) as a rotation matrix, and \( D \) as a displacement matrix, then we get a composition of three matrices as follows:

\[
\begin{bmatrix}
S & 0 & 0 \\
0 & S & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\cos \theta & \sin \theta & 0 \\
-\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
M_{comp} = S R D
\]

We get six different matrices through Eq. (9), 2x2 submatrix values are not different. It consists of two different cases. From this characteristic, we get

\[
\begin{align*}
a_2^2 &= \frac{(S_2^2 \cos \theta + \sin \theta)^2}{(S_2^2 \cos \theta - \sin \theta)^2} \cdot \frac{(S_2^2 \sin \theta + \cos \theta)^2}{(S_2^2 \sin \theta - \cos \theta)^2} \cdot \frac{(S_2 \cos \theta + \sin \theta)^2}{(S_2 \cos \theta - \sin \theta)^2} \cdot \frac{(S_2 \sin \theta + \cos \theta)^2}{(S_2 \sin \theta - \cos \theta)^2} \\
&= \frac{(u_2 - u_0)^2 + (v_2 - v_1)^2}{(u_0 - u_1)^2 + (v_0 - v_1)^2}
\end{align*}
\]

If we set \( S_u = S_r \) or the same scaling for each axis, we obtain the following relationship using Eq. (10) and Eq. (11).

\[
D_x = D_{k-1} \quad \Rightarrow \quad a_x = a_{k-1}
\]

For the scaling factor \( p \), the same derivation is applied. Therefore, the scale factors \( p \) and \( q \) are preserved after deformation.

Now we can derive a vector transformation by exploiting the above geometric conservation property. In Figure 3, since line BC and line QZ have the same gradient and line BA and line FZ also have the same gradient, we can express the following relationship:

\[
\begin{align*}
\begin{bmatrix}
X_Y - Z_x \\
X_Z - X_x \\
X_W - X_z
\end{bmatrix} &= \begin{bmatrix}
Z_x - Y_y \\
Z_y - X_z \\
Z_z - X_w
\end{bmatrix} = b_{BA} \\
&= a_{BC} \\
&= a_{AB} \\
\end{align*}
\]

Using the above results, we can get

\[
\begin{align*}
x_z &= p(x_z - x_x) + q(y_z - x_z) + x_x \\
y_z &= p(y_z - x_z) + q(y_z - x_z) + y_x
\end{align*}
\]

which implies a vector transformation.
\[ Z = p(C - B) + q(A - B) + \beta = (1 - d - q)\beta + zC + qA \quad (16) \]

In Figure 2, the vector \( Z \) is calculated as follows:
\[ Z = \beta + p\bar{B}C + q\bar{A} = \beta + p(C - \bar{B}) + q(A - \bar{B}) \quad (17) \]

Note that Eq. (16) and Eq. (17) are equivalent. It does not require any matrix inversion or multiplication operations. Therefore, using the vector transformation, we can get a faster transformation than the matrix operations.

3. Motion Estimation with Variable Search Range

The PSNR (peak signal-to-noise ratio) value of the BMA (block matching algorithm) can be improved by increasing the SR (search range); however, it is saturated at some point. Unlike BMA, the image warping method does not estimate the motion vector using the actual prediction error. Figure 3 demonstrates that PSNR of image warping is not always increased by increasing the search range for several BS (block size), which is different from the BMA case. If PSNR of FD (frame difference) is smaller, it requires larger SR.

Figure 4 shows the search range distribution for three different image sequences. To find an optimal search range, we employ a second-order polynomial curve fitting. For refinement of the motion vector of each control point, we can apply the hexagonal matching method.

If the samples are spread to pursue an efficient local representation without any built-in topological constraints, the resulting samples can have a nearly chaotic structure. Starting from a regular patch, each control point is moved toward the direction of minimizing the interpolation error while maintaining the topology of the original mesh. The goal of using the spatial transformation in the mesh-based procedure is to minimize the interpolation errors from node values. However, the minimization of this error alone may cause certain nodes to get close to each other. In the extreme case, certain elements can overlap or destroy the desired topological relation among the nodes.

As shown in Figure 5, most of the block mean square errors (BMSE) are distributed at the lower part of the mean BMSE. The upper values of the mean BMSE represent large errors between consecutive images during motion estimation and compensation. On the other hand, the lower values of the mean BMSE represent small errors, human eyes cannot distinguish their differences. For this reasoning, we only compensate for blocks of large errors.

The hexagonal matching method [3] produces better image quality than the BMA or the conventional image warping.
method. However, it requires a large amount of computational complexity. Although it improves image quality, it is not adequate for the real-time encoding process. There are some simpler methods than the hexagonal matching method with slight sacrifice in image quality. We cannot improve the image quality only by increasing the search range beyond necessity. At this point, the optimal search range is determined.

For refinement we apply the hexagonal matching method on the entire image. We calculate the PSNR of FD and block MSE. The order of each block is sorted by for block MSE. Blocks that have larger block MSE have higher priority. We determine an adaptive search range using the optimal second-order polynomial curve and the PSNR of FD.

4. Simulation Results

Computer simulations have been performed on the CCITT monochrome test sequences, "MISS AMERICA", "CLAIRED", and "SALESMAN" for 88 frames each, which consist of 352x288 luminance pixels (CIF format). The frame rate, which is originally 30Hz, was set at 10Hz (the 1st, 4th, 7th, .... 88th frames were used) in the simulation. In the simulation of motion compensation, the original image of the previous frame was used as a reference frame to synthesize the predicted current frame. The image quality was evaluated by the peak signal-to-noise ratio (PSNR) defined as follows:

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I(m,n) - \tilde{I}(m,n))^2$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad [dB]$$  \hspace{1cm} (18)

where M=352 and N=288, BS(Block Size)=16 for CIF.

Table 1. Average Performance of Affine Motion Compensation

<table>
<thead>
<tr>
<th></th>
<th>BMA</th>
<th>Hexa7</th>
<th>AFMC 75%</th>
<th>AFMC 100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (dB)</td>
<td>37.80</td>
<td>38.38</td>
<td>38.30</td>
<td>38.51</td>
</tr>
<tr>
<td>DCT/Q (bytes)</td>
<td>1618</td>
<td>1491</td>
<td>1498</td>
<td>1492</td>
</tr>
<tr>
<td>refine_CP</td>
<td>396</td>
<td>396</td>
<td>357</td>
<td>357</td>
</tr>
<tr>
<td>trans_CP</td>
<td>396</td>
<td>396</td>
<td>297</td>
<td>357</td>
</tr>
<tr>
<td>Overhead (bits)</td>
<td>0</td>
<td>0</td>
<td>400</td>
<td>400</td>
</tr>
<tr>
<td>Opt_SR</td>
<td>7</td>
<td>7</td>
<td>4.31</td>
<td>4.31</td>
</tr>
<tr>
<td>Coding bits</td>
<td>16112</td>
<td>15096</td>
<td>14760</td>
<td>15192</td>
</tr>
<tr>
<td>complexity (frame)</td>
<td>22809600</td>
<td>6847355</td>
<td>6847358</td>
<td></td>
</tr>
<tr>
<td>complexity (frame)</td>
<td>22809600</td>
<td>7165125</td>
<td>26824089</td>
<td>32243097</td>
</tr>
</tbody>
</table>

Note: AFMC × % represents the affine motion compensation using × % transmitted motion vector for control point.

As shown in Table 1, the proposed method provides better performances in terms of reconstructed image quality, number of coding bits, and computational complexity compared with block matching algorithm (BMA) and the hexagonal matching method (Hexa7). In the table, the reconstructed image quality is denoted as PSNR and DCT/QU represents the amount of bytes to express the DCT and quantized residual image. The term of refine_CP is the number of CP that requires for refinement, and trans_CP means the number of CP that requires for decoding. Overhead bits are used to represent the rate_MSE_map. Computational complexity can be classified into the coarse motion estimation and the refinement. The affine motion estimation employs a fast affine transformation.

5. Conclusion

In this paper, we have derived a fast spatial transformation using vector relationship. There are some differences between vector relationship and spatial transformation. We also have developed a more effective motion estimation method than the conventional image warping method in terms of computational complexity, image quality, and the amount of coding bits. Computer simulations show that our proposed scheme is suitable for very low bit-rate video coding. We have obtained good results due to the fact that the affine transformation maintain the acceptable level of prediction error even when the number of the control points is reduced to about half of the number of blocks used in the other methods. Furthermore, our scheme enables a layer coding and a motion tracking by selecting transmission rate.

Acknowledgments

This work was supported in part by the Korea Science and Engineering Foundation(KOSEF) through the Ultra-Fast Fiber-Optics Networks Research Center at Kwangju Institute of Science and Technology.

References