Image Warping using Adaptive Partial Matching

Dong-Keun Lim and Yo-Sung Ho

Kwangju Institute of Science and Technology
572 Ssangam-dong Kwangsan-gu, Kwangju, 506-712, Korea

Abstract

In this paper, we propose an adaptive partial matching method for motion estimation to reduce the computational complexity, while maintaining the image quality comparable to the hexagonal matching method. The proposed motion compensation method combines the fast affine transformation using vector relationship. We simulate our proposed motion estimation method in DCT-based coder by encoding CIF(common intermediate format) images at the bitrate of below 64kb/s. The quality of reconstructed image with our method is substantially improved compared with the block matching algorithms(BMA), and is comparable to the hexagonal matching method. Computational complexity and coding bits are also reduced significantly relative to the BMA and the conventional image warping methods.

1 Introduction

In low bitrate video coding, insufficient bit allocation for motion information and other image contents results in severe image degradations such as blocking and mosquito effects. Such degradations is due to inadequate motion models that cannot represent real-world motions properly with a small number of motion parameters. To develop more effective motion compensation methods for low bitrate video coding, we address issues of adopting more suitable spatial transformations[1] than the simple translational motion in the block matching algorithm(BMA)[2]. As a more general spatial transformation, we consider the affine transformation, where actual motion is predicted with a geometric transformation of the previous image. This scheme can compensate for different types of motion, including scaling and rotation that the block matching algorithm cannot cover properly. Therefore, the proposed scheme can produce a predicted image with less blocking artifacts. The new scheme enables a significant subjective improvement in the motion prediction and a consistent reduction of prediction errors. The hexagonal matching method is one of the refinement methods in image warping[1,3-5]. It brings better image quality, but it requires a large amount of computations. In this paper, we propose a new motion estimation algorithm that employs an adaptive partial matching with a variable search range. Instead of fixing the search range for coarse motion estimation, we adaptively adjust the search range based on the peak signal to noise ratio(PSNR) of the frame difference(FD).

2 Fast Affine Transformation using Vector Relationship

In image warping, the affine transformation 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 of the affine transformation.

\[
\begin{align*}
A'(x_0, y_0) &\rightarrow Z'(x_1, y_1) \\
Q'(x_2, y_2) &\rightarrow C'(x_2, y_2) \\
A(x_0, y_0) &\rightarrow Z(x_1, y_1) \\
B'(x_1, y_1) &\rightarrow P'(x_1, y_1) \\
Q(x_2, y_2) &\rightarrow Z(x_2, y_2) \\
B(x_2, y_2) &\rightarrow P(x_2, y_2)
\end{align*}
\]

(a) previous(k-1) patch
(b) current(k) patch

Figure 1: Deformation of Patches

The affine motion model is defined as follows:

\[
x = a_{11}u + a_{21}v + a_{31}
y = a_{12}u + a_{22}v + a_{32}
\]  

(1)
\[
\begin{bmatrix}
  x & y & 1
\end{bmatrix} = \begin{bmatrix}
  u & v & 1
\end{bmatrix} \begin{bmatrix}
  a_{11} & a_{12} & 0 \\
  a_{21} & a_{22} & 0 \\
  a_{31} & a_{32} & 1
\end{bmatrix}
\] (2)

From the relationship of the previous and current patches, as shown in Figure 1, we can get
\[
q_{k-1} = \frac{Q'B'}{|A'B'|} = \frac{\sqrt{(u_q - u_l)^2 + (v_q - v_l)^2}}{\sqrt{(u_0 - u_0)^2 + (v_0 - v_l)^2}}
\] (3)
\[
q_k = \frac{QB}{|AB|} = \frac{\sqrt{(x_q - x_l)^2 + (y_q - y_l)^2}}{\sqrt{(x_0 - x_0)^2 + (y_0 - y_1)^2}}
\] (4)

For each control point, we obtain its affine transformation by Eq. (1). When we put Eq. (5) and Eq. (6) into Eq. (4), we can obtain Eq. (7).
\[
x_q = a_{11}u_q + a_{21}v_q + a_{31}
\]
\[
x_0 = a_{11}u_0 + a_{21}v_0 + a_{31}
\]
\[
x_1 = a_{11}u_1 + a_{21}v_1 + a_{31}
\]
\[
y_q = a_{12}u_q + a_{22}v_q + a_{32}
\]
\[
y_0 = a_{12}u_0 + a_{22}v_0 + a_{32}
\]
\[
y_1 = a_{12}u_1 + a_{22}v_1 + a_{32}
\] (5) (6)

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:
\[
S = \begin{bmatrix}
  S_u & 0 & 0 \\
  0 & S_v & 0 \\
  0 & 0 & 1
\end{bmatrix}
\] (8)
\[
R = \begin{bmatrix}
  \cos \theta & \sin \theta & 0 \\
  -\sin \theta & \cos \theta & 0 \\
  0 & 0 & 1
\end{bmatrix}
\] (9)
\[
D = \begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
 -T_u & -T_v & 1
\end{bmatrix}
\] (10)

\[
M_{\text{comp}} = 3P_3(D, R, S) = \begin{bmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{21} & a_{22} & a_{23} \\
  a_{31} & a_{32} & a_{33}
\end{bmatrix}
\] (11)

We get six different matrices in Eq. (11). The 2x2 submatrix can have two different cases:
\[
\begin{bmatrix}
  a_{11} & a_{12} \\
  a_{21} & a_{22}
\end{bmatrix} = \begin{bmatrix}
  S_u \cos \theta & S_v \sin \theta \\
 -S_v \sin \theta & S_u \cos \theta
\end{bmatrix}
\] (12)

We put Eq. (12) or Eq. (13) into Eq. (7). If we set \(S_u = S_v\) or the same scaling for each axis, we obtain Eq. (14). With this condition, the affine transformation is a special case of bilinear transformation. This implies that a certain spatial transformation may preserve the scale factors between previous patch and current patch through deformation.
\[
p_k = p_{k-1}, \quad q_k = q_{k-1}
\] (14)

Therefore, the scale factors \(p\) and \(q\) are preserved through deformation.

**Figure 2: Vector Diagram**

Now we can derive a vector transformation by exploiting the above geometric property. In Figure 2, since line BC and line QZ have the same gradient and line BA and line PZ also have the same gradient, we can express the following relationship:
\[
a_{p_z} = \frac{y_b - y_a}{x_b - x_a} = \frac{y_b - y_a}{x_b - x_a} = a_{b a}
\]
\[
a_{q_z} = \frac{y_b - y_a}{x_b - x_c} = \frac{y_b - y_c}{x_b - x_c} = a_{b c}
\] (15)
\[
(y_b - y_a)x_z - (x_b - x_a)y_z = (y_b - y_a)x_p - (x_b - x_a)y_p
\]
\[
(y_b - y_c)x_z - (x_b - x_c)y_z = (y_b - y_c)x_p - (x_b - x_c)y_p
\] (16)

Using the above results, we can get
\[
x_z = p(x_a - x_b) + q(x_a - x_b) + x_b
\]
\[
y_z = p(y_c - y_b) + q(y_a - y_b) + y_b
\] (17)

which implies a vector transformation:
\[
Z = p(C - B) + q(A - B) + B = (1 - p - q)B + pC + qA
\] (18)
\[ q_k^2 = \frac{(a_{11}^2 + a_{21}^2)(u_2 - u_1)^2 + 2(a_{11}a_{21} + a_{12}a_{22})(u_2 - u_1)(v_2 - v_1) + (a_{11}^2 + a_{21}^2)(v_2 - v_1)^2}{(a_{11}^2 + a_{12}^2)(u_0 - u_1)^2 + 2(a_{11}a_{21} + a_{12}a_{22})(u_0 - u_1)(v_0 - v_1) + (a_{21}^2 + a_{22}^2)(v_0 - v_1)^2} \]  

(7)

Figure 3: PSNR vs. Search Range

In Figure 2, the vector \( \vec{Z} \) is calculated as follows:

\[ \vec{Z} = \vec{B} + p\vec{B}\vec{C} + q\vec{B}\vec{A} = \vec{B} + p(\vec{C} - \vec{B}) + q(\vec{A} - \vec{B}) \]

\[ = (1-p-q)\vec{B} + p\vec{C} + q\vec{A} \]  

(19)

Note that Eq. (18) and Eq. (19) are the same. It does not require any matrix inversion or multiplication operations. Therefore, using the vector transformation, we can get a faster transformation than the matrix operation.

3 Adaptive Partial Matching Method

The performance of the block matching algorithm for motion estimation is improved by increasing the motion search range(SR); however, it can be 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 different block sizes(BS), which is quite different from BMA.

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

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 prediction errors between consecutive images during the motion estimation and compensation operations. On the other hand, the lower values of the mean BMSE represent small errors, which cannot be noticed by human eyes. For this reasoning, we only compensate for blocks of large errors.

The hexagonal matching method[3] brings better image quality than the BMA or the conventional image warping method. However, it requires a large amount of computational complexity. Therefore, it is
not transmitted, but used for refinement.

4 Simulation Results

The simulation was performed on the CCITT monochrome test sequence "MISS AMERICA", "CLAIRE", and "SALESMAN" for 88 frames each, each of which consists of 352x288 luminance pixels. 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 the reference frame to synthesize the predicted image. The image quality was evaluated by the peak signal-to-noise ratio (PSNR) defined as follows:

\[
MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I_k(m,n) - \hat{I}_k(m,n))^2
\]

\[
PSNR = 10 \log_{10} \frac{255^2}{MSE} [dB]
\]

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

As shown in Table 1 and Table 2, the proposed method provides better performances in terms of reconstructed image quality, number of coding bits, and computational complexity, compared to BMA and the conventional image warping methods. The measurement of computation complexity is different for each method, since each requires a different operation. Hexa7, Forward, HGI require the computation of MSE for each block; however, the gradient method requires the computation of the matrix inversion and matrix multiplications. We arrange the result by the number of multiplications, since the point of comparison is not exactly equivalent.
Table 1: Average Performance of Adaptive Partial Matching

<table>
<thead>
<tr>
<th></th>
<th>FSM</th>
<th>AFMC 25%</th>
<th>AFMC 50%</th>
<th>AFMC 75%</th>
<th>AFMC 100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (dB)</td>
<td>37.7961</td>
<td>37.0281</td>
<td>37.9972</td>
<td>38.2938</td>
<td>38.5093</td>
</tr>
<tr>
<td>DCT/Q (byte)</td>
<td>1618</td>
<td>1689</td>
<td>1510</td>
<td>1498</td>
<td>1492</td>
</tr>
<tr>
<td>refine_CP</td>
<td>396</td>
<td>155</td>
<td>269</td>
<td>357</td>
<td>357</td>
</tr>
<tr>
<td>trans_CP</td>
<td>396</td>
<td>99</td>
<td>198</td>
<td>297</td>
<td>357</td>
</tr>
<tr>
<td>Overhead (bits)</td>
<td>0</td>
<td>400</td>
<td>400</td>
<td>400</td>
<td>400</td>
</tr>
<tr>
<td>Optimal_SR</td>
<td>7 (Fixed)</td>
<td>4.31</td>
<td>4.31</td>
<td>4.31</td>
<td>4.31</td>
</tr>
<tr>
<td>Transmitted bits</td>
<td>16112</td>
<td>14704</td>
<td>14064</td>
<td>14760</td>
<td>15192</td>
</tr>
<tr>
<td>Complexity (coarse)</td>
<td>301830</td>
<td>5179483</td>
<td>6847355</td>
<td>6847358</td>
<td></td>
</tr>
<tr>
<td>Complexity (refine)</td>
<td>22809600</td>
<td>8941363</td>
<td>17882726</td>
<td>26824089</td>
<td>32243097</td>
</tr>
</tbody>
</table>

AFMC x%: Affine motion compensation using x% transmitted motion vector for control points

Table 2: Average Performance of Image Warping Methods

<table>
<thead>
<tr>
<th></th>
<th>Hexa7</th>
<th>Gradient</th>
<th>Forward</th>
<th>HGI</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (dB)</td>
<td>38.3894</td>
<td>37.0660</td>
<td>38.3729</td>
<td>37.8021</td>
<td>38.5093</td>
</tr>
<tr>
<td>DCT/Q (byte)</td>
<td>1491</td>
<td>1676</td>
<td>1505</td>
<td>1723</td>
<td>1492</td>
</tr>
<tr>
<td>refine_CP</td>
<td>396</td>
<td>396</td>
<td>396</td>
<td>515</td>
<td>357</td>
</tr>
<tr>
<td>trans_CP</td>
<td>396</td>
<td>396</td>
<td>396</td>
<td>515</td>
<td>357</td>
</tr>
<tr>
<td>Overhead (bits)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>396</td>
<td>400</td>
</tr>
<tr>
<td>Optimal_SR</td>
<td>7 (Fixed)</td>
<td>.</td>
<td>7</td>
<td>7</td>
<td>4.31</td>
</tr>
<tr>
<td>Transmitted bits</td>
<td>15096</td>
<td>16576</td>
<td>15208</td>
<td>17904</td>
<td>15192</td>
</tr>
<tr>
<td>Complexity (coarse)</td>
<td>22809600</td>
<td>792x792</td>
<td>Inverse Matrix</td>
<td>22995879</td>
<td>6847358</td>
</tr>
<tr>
<td>Complexity (refine)</td>
<td>71651328</td>
<td>71651328</td>
<td>71752704</td>
<td>97491968</td>
<td>32243097</td>
</tr>
</tbody>
</table>

Hexa7: Hexagonal Matching Method (SR = ±7)[3], Gradient: Gradient Constraint Method [3]

5 Conclusion

In this paper, we have developed a more effective motion estimation method than the conventional image warping methods in terms of reconstructed image quality, computational complexity, and the number of coding bits. Computer simulations demonstrate that our proposed scheme is suitable for very low bit-rate video coding. We also have derived a fast algorithm for the affine transformation using the vector relationship. We have obtained good results due to the fact that the affine transformation maintains 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 layered coding and motion tracking by selecting different transmission rates.

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