ABSTRACT
A particle filtering based block-wise estimation of true motion in a video sequence is proposed. Parameters of motion of a block in a sequence are defined as a state, and evolution of the state with respect to the block index is tracked with particle filtering. The state is assumed to be dependent on the states of neighboring blocks. Estimated motion fields are consistent and suitable for applications that require true motion such as intermediate frame generation. With the particle filtering, motion can be searched only where they are probable. Hence, the proposed method can estimate motion with a fraction of search points necessary for conventional estimation methods.

1. INTRODUCTION
Estimation of motion in a video sequence is widely used in many video processing applications such as compression, format conversion, and picture quality improvement [1, 2, 3, 4]. In most applications, there are no chances to correct errors after the estimated motion is compensated as in the case of compression. Inaccurately estimated motion introduces spurious errors in final productions of video sequences. Estimation methods that claim to find true motion are used to reduce the errors by making estimated motion vector field consistent [1, 5].

Consistent motion fields can be obtained by regularized estimation methods that put a constraint on the roughness of the estimated motion and disparity fields [5, 6]. A cost function in the estimation procedure consists of a deviation penalty that penalizes differences between two blocks in current and reference images and a roughness penalty function that penalizes the differences between motion and disparity between neighboring blocks.

The regularized estimation methods still search for every point in the search space, which is the same as the full search method. They use different cost functions to find more meaningful estimate. As the number of parameters being estimated becomes large, the search space grows very fast. For example, for estimation of affine motion, there are five parameters for zoom, rotation, and translation. For joint estimation of motion and disparity, there are four parameters for disparity and motion in left and right views. The total number of points to be exhaustively searched by either the full search or the regularized search becomes impractically large.

3D recursive search is another method that provides true motion [1]. The 3D recursive search provides a consistent motion vector field by limiting the search only to the vicinities where the motion vectors of neighboring blocks point. The 3D recursive search has a distinct advantage that it provides a fast search method for the estimation of true motion. It has been widely used in video format conversion methods that require fast estimation of true motion [7, 2]. However, it is not clear how to extend the 3D recursive search to include more parameters or to improve performance.

This paper proposed a new search method for the estimation of true motion based on particle filtering [8]. Parameters of motion of a block in a video sequence are defined as a state. Evolution of the state with respect to the block index is tracked with particle filtering by estimating the posterior probability density of the state. Motion then can be found by taking the expectation of the state with respect to the estimated posterior probability. The state is assumed to be dependent on the states of neighboring blocks, which makes the estimated motion and disparity fields consistent and suitable for applications that require true motion. With the particle filtering, motion can be searched only where they are probable. The number of the search points can be limited to a fraction of those necessary for conventional estimation methods.

Experiments are performed for the estimation of block-wise affine motion in a video sequence. The proposed method can estimate true motion, with which intermediate frames of good subjective image quality can be reconstructed. The number of search points can be chosen to be a fraction of those of conventional methods for similar performance.

Section 2 presents the proposed estimation method. Section 3 provides experimental results with test patterns and real video sequences with discussion. Section 4 concludes the paper.
2. PARTICLE FILTER BASED MOTION ESTIMATION

A video sequence is denoted by \( f(i, j, t) \), where \( i \) is the vertical index, \( j \) is the horizontal index, and \( t \) is the temporal index. A frame of the video sequence, at time \( t = t_0 \) for instance, is an \([I \times J]\) size matrix and denoted by \( f(i, j, t_0) \). We assume that a block of image is available one at a time. A temporal index \( k \) is introduced to account for the availability of the new block. For the block of the size \([b_i \times b_j]\), there are \([M \times N]\) blocks in a frame, where \( M = I/b_i \) and \( N = J/b_j \). The index \( k \) is related to the \((m, n)\)th block in the \( t \)th frame by

\[
k = (t - 1) \cdot MN + (m - 1) \cdot N + n. \tag{1}
\]

To formulate the problem of motion estimation, consider the evolution of the state sequence \( \{x_k, k \in \mathbb{N}\} \), which is associated with the \( k \)th block:

\[
x_k = G(x_{1:k-1}, v_k) \tag{2}
\]

where \( x_{1:k-1} \) is the states up to \((k-1)\)th block, \( G \) is a possibly nonlinear function that describes the dynamics of the states, and \( \{v_k, k \in \mathbb{N}\} \) is an i.i.d. process noise. The goal of the motion estimation is to estimate the state \( x_k \) that contains information about motion in the scene from measurements

\[
z_k = H(x_k, n_k) \tag{3}
\]

where \( H \) is a possibly nonlinear function and \( \{n_k, k \in \mathbb{N}\} \) is an i.i.d. measurement noise.

The state is defined by

\[
x_k = (v_i, v_j, r_i, r_j, \theta)^T \tag{4}
\]

to describe translational, scaling, and rotational motion, where \((v_i, v_j)\) are the translational motion, \((r_i, r_j)\) are the scaling ratios, and \( \theta \) is the angle of rotation. The subscript \( i \) and \( j \) indicate vertical and horizontal directions, respectively, and the superscript \( T \) denotes the transpose operation.

If the block size is smaller than objects in motion, one can assume that the motion of the current block is similar to those of neighboring blocks. Hence, the states are modeled such that they depend on neighbors. The probabilistic model that describes the state evolution is chosen such that the current state depends on the states pointed by the set \( C(k) \), or

\[
p(x_k|x_{1:k-1}) = p(x_k|x_c, c \in C(k)), \tag{5}
\]

where \( C(k) \) is the set of indices defined by

\[
C(k) = \{k - N - 1, k - N, k - N + 1, k - 1, k - MN, k - MN + 1, k - MN + N - 1, k - MN + N, k - MN + N + 1\}. \tag{6}
\]

Note that upper and left blocks pointed by the set are in the current frame whereas collocated, right, and lower blocks are in the previous frame. The causality remains valid while accessing neighbors.

Without knowing which neighboring states the current state is more closely related, the prior probability in (5) is modeled by a Gaussian mixture

\[
p(x_k|x_c, c \in C(k)) = \sum_{c \in C(k)} \pi_c \mathcal{N}(x_k; x_c, \sigma^2_c), \tag{7}
\]

where \( \pi_c \) is the mixing probabilities, and \( \mathcal{N}(x_k; x_c, \sigma^2_c) \) is the Gaussian distribution with mean \( x_c \) and variance \( \sigma^2_c \). The mixing probabilities, \( \pi_c \), and the variance, \( \sigma^2_c \) are set to constants for all mixing densities.

With all the mixing densities being equally likely, the state model in (2) can be rewritten as

\[
x_k = g(x_{1:k-1}; \gamma) + v_k, \tag{8}
\]

where \( \gamma \) is the result of a \( C \)-faced fair coin toss and \( C \) is the cardinality of the set \( C \). The nonlinear function \( f \) selects one of the neighboring states based on the result of the coin toss \( \gamma \) such that

\[
g(x_{1:k-1}) = x_c, \text{ for } \gamma = c. \tag{9}
\]

The measurements \( z_k \) one can have is the distance between the current block and the block in the reference frame accessed by the state. In this work, the sum of absolute difference (SAD) is used as a measure for comparisons to existing works.

The distance between the \((m, n)\)th block in the current frame, \( f(i, j, t) \), and the block in the reference frame, \( f(i, j, t - 1) \), accessed by the state is obtained by

\[
h(x_k) = \sum_{(i, j) \in B} |f(i' + i, j' + j, t - 1) - f(i + i, j + j, t)| \tag{10}
\]

where \( i_k = (m-1)b_i \) and \( j_k = (n-1)b_j \) are starting indices of the current block, the indices \((i', j')\) and \((i, j)\) are related by the motion model

\[
\begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} r_i & 0 \\ 0 & r_j \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} v_i \\ v_j \end{bmatrix} \tag{11}
\]

and \( B \) is the set of indices in a block given by

\[
B = \{(i, j) | i \in [0, b_i), j \in [0, b_j)\}. \tag{12}
\]

When \((i', j')\) are not integers, the bilinear interpolation is used to find pixel values.

The observation model in (3) can be rewritten by

\[
z_k = h(x_k) + n_k. \tag{13}
\]

The probabilistic model that describes the measurement is the the distance measure in (10) follows the Gaussian distribution. The posterior probability is modeled by

\[
p(z_k|x_k) = \mathcal{N}(z; 0, \sigma^2_n). \tag{14}
\]
The Bayesian approach of motion estimation is to estimate what is the probability of state $x_k$ having specific values given the history of measurements up to $k$, $z_{1:k}$ [8]. In other words, the goal of the Bayesian motion estimation is to find the posterior probability $p(x_k | z_{1:k})$.

It is assumed that the prior probability $p(x_0 | z_0) \equiv p(x_0)$ is known, where $z_0$ means no measurement. Then the posterior can be estimated in a recursive manner through the prediction stage

$$p(x_k | z_{1:k-1}) = \int p(x_k | x_c, c \in C(k))p(x_c, c \in C(k) | z_{1:k-1})dx_c,$$

where the integral is $C$ dimensional integral over the states in $C(k)$, and the update stage

$$p(x_k | z_{1:k}) \propto p(z_k | x_k)p(x_k | z_{1:k-1}).$$

(16)

All the information on motion can be obtained by taking expectation of the state with respect to the posterior probability by

$$\hat{x}_k = E\{x_k\}. \quad \text{(17)}$$

3. EXPERIMENTS AND DISCUSSIONS

The performance of the particle search is evaluated by reconstructing frames of a video sequence using the estimated motion. Results are compared to those of the full search, two-stage search of local affine motion [9] and the 3D recursive search. For a given video sequence, motion parameters are estimated between $f(i, j, t)$ and $f(i, j, t-1)$, which are used to reconstruct the $t$th frame, $\hat{f}(i, j, t)$, and the $(t-0.5)$th frame, $\hat{f}(i, j, t-0.5)$.

A set of test video sequences are chosen to include test patterns generated with known parameters and actual test video sequences. For the particle search, the search ranges are set to $[-16, 16], [0.9, 1.1], [-2\pi, 2\pi]$ for translational, scaling, and rotational motion, respectively. The variances of the process noise and observation noise are determined by a pilot test with test pattern sequences with known motion. Once the variances are set, the same values are used throughout the experiments. The search ranges of other search methods are set to $[-16, 16]$ for both vertical and horizontal directions. The second stage of the two stage search uses the sets $\{-0.5, 0.0, 0.5\}$, $\{0.9, 1.0, 1.1\}$, and $\{-0.2\pi, 0.0, 0.2\pi\}$ for the search of translational, scaling, and rotational motion, respectively.

The performance of the particle search is evaluated with test video sequences. A set of test video sequences are chosen to include CIF format, standard definition (SD), and high definition (HD) format sequences. Table 1 shows the average PSNR values for the selected video sequences.

Fig. 1 shows reconstructed frames of rotation sequence using motion vectors estimated by various search methods. First, the frame $\hat{f}(i, j, t)$ is reconstructed using the estimated motion estimated with only 32 particles can reconstruct the intermediate frame with superior subjective image quality.

Fig. 1. Frames of rotation sequence (rotation of lionhead image by 1.0 degree per frame) reconstructed using the estimated motion. [top: the reconstructed 29th frame (estimated motion vector field is overlayed), bottom: the reconstructed intermediate frame between the 28th and 29th frames]. (a) two stage search, (b) particle search with 32 particles. True motion estimated with only 32 particles can reconstruct the intermediate frame with superior subjective image quality.
Table 1. Average PSNR between the Original Frames and the Frames Reconstructed Using the Estimated Motion in dB.

<table>
<thead>
<tr>
<th>method</th>
<th>particles (search points)</th>
<th>rotation SD</th>
<th>scaling SD</th>
<th>mobile CIF</th>
<th>flower CIF</th>
<th>bus CIF</th>
<th>tennis CIF</th>
<th>football CIF</th>
<th>shields HD</th>
<th>stokholm HD</th>
</tr>
</thead>
<tbody>
<tr>
<td>full search</td>
<td>1089</td>
<td>29.31</td>
<td>32.02</td>
<td>24.76</td>
<td>28.60</td>
<td>26.60</td>
<td>30.54</td>
<td>26.62</td>
<td>31.23</td>
<td>32.06</td>
</tr>
<tr>
<td>two stage search</td>
<td>1170</td>
<td>31.59</td>
<td>34.25</td>
<td>27.77</td>
<td>30.74</td>
<td>28.76</td>
<td>31.66</td>
<td>28.07</td>
<td>33.70</td>
<td>33.83</td>
</tr>
<tr>
<td>3D recursive search</td>
<td>81</td>
<td>27.99</td>
<td>29.07</td>
<td>24.24</td>
<td>28.54</td>
<td>25.50</td>
<td>29.68</td>
<td>23.90</td>
<td>31.05</td>
<td>31.85</td>
</tr>
<tr>
<td>particle search</td>
<td>128</td>
<td>28.49</td>
<td>34.55</td>
<td>28.08</td>
<td>31.23</td>
<td>27.62</td>
<td>29.22</td>
<td>24.22</td>
<td>33.34</td>
<td>33.35</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>29.17</td>
<td>34.36</td>
<td>27.95</td>
<td>30.90</td>
<td>27.41</td>
<td>28.91</td>
<td>23.96</td>
<td>33.16</td>
<td>33.20</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>27.42</td>
<td>33.72</td>
<td>27.62</td>
<td>30.35</td>
<td>27.18</td>
<td>28.44</td>
<td>23.58</td>
<td>32.82</td>
<td>32.90</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>26.53</td>
<td>31.98</td>
<td>26.95</td>
<td>29.35</td>
<td>26.78</td>
<td>27.28</td>
<td>22.79</td>
<td>32.16</td>
<td>32.30</td>
</tr>
</tbody>
</table>

motion. In this case, the full search and the two stage search provide higher PSNR values than the particle search. The 3D recursive search reports the lowest PSNR values. No obvious error can be seen in all the pictures. Second, the intermediate frame $\hat{f}(i, j, t - 0.5)$ is reconstructed using the half of the estimated motion. This is a typical set up used for the frame rate conversion applications, where the frame rate of an input video sequence is doubled by inserting a motion compensated frame in between two input frames [7, 10]. The intermediate frames reconstructed by the two stage search show spurious errors. It can be easily seen the structures in the picture are broken. In contrast, the intermediate frame reconstructed by the particle search shows good subjective image quality without any broken structures.

The particle search uses the information about the motion of neighboring blocks in defining the prior. With a correct modeling of the prior, the particles will sample the posterior probability of the state only where they are more probable. For example, only 32 particles, or equivalently 32 search points, are need for the reconstruction of the intermediate frame in Fig. 1.

4. CONCLUSION

This paper presents a novel estimation method for block-wise estimation of true motion. The basic idea is to assume that a block of an image is being available one at a time, and to track the motion of the blocks with the particle filtering. The proposed method has two advantages over conventional estimation methods. The estimated motion are consistent and closer to true motion and disparity. The number of search points can be limited to small numbers even when the number of parameters being estimated is large.

5. REFERENCES


