Color Constancy

Page 198 - 209
Marc Ebner

Presented by Yuan Xi

School of Electrical Engineering and Computer Science
Kyungpook National Univ.
Evolving color constancy

- Evolution algorithms

Initialization of population

Selection

Reproduction

Variation

Done?

Output Result

Fig. 8.5 Main loop of an evolutionary algorithm. New individuals are created by selecting highly fit individuals and reproducing these individuals. The offspring, however, are usually not exact copies. Some individuals may be altered by genetic operators. In the course of time, the individuals adapt to their environment, i.e. the selection criterion.
Tree-based genetic programming

- Representation of individuals
  - Inner nodes called elementary functions
  - Outer nodes called terminal symbols

Fig. 8.6 In tree-based genetic programming, individuals are represented as trees. A sample individual is shown on the left. This individual represents a symbolic expression that evaluates to $2x - 1$. 
- Set of elementary functions

**Table 1.** Elementary functions used for the experiments.

<table>
<thead>
<tr>
<th>Elementary Function</th>
<th>Arity</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>2</td>
<td>+</td>
</tr>
<tr>
<td>Subtraction</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>Multiplication</td>
<td>2</td>
<td>*</td>
</tr>
<tr>
<td>Protected division</td>
<td>2</td>
<td>/</td>
</tr>
<tr>
<td>Multiply by 2</td>
<td>1</td>
<td>mul2</td>
</tr>
<tr>
<td>Divide by 2</td>
<td>1</td>
<td>div2</td>
</tr>
</tbody>
</table>
- Set of terminal symbols

**Table 1.** Set of terminal symbols.

<table>
<thead>
<tr>
<th>Terminal</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant one</td>
<td>1</td>
</tr>
<tr>
<td>Red input band $c_r(x, y)$</td>
<td>red</td>
</tr>
<tr>
<td>Green input band $c_g(x, y)$</td>
<td>green</td>
</tr>
<tr>
<td>Blue input band $c_b(x, y)$</td>
<td>blue</td>
</tr>
<tr>
<td>Current band $c_i(x, y)$</td>
<td>band</td>
</tr>
<tr>
<td>Estimate from current element</td>
<td>center</td>
</tr>
<tr>
<td>$L_i(x, y)$</td>
<td>left</td>
</tr>
<tr>
<td>Estimate from left element</td>
<td>right</td>
</tr>
<tr>
<td>$L_i(x - 1, y)$</td>
<td>up</td>
</tr>
<tr>
<td>Estimate from right element</td>
<td>down</td>
</tr>
<tr>
<td>$L_i(x + 1, y)$</td>
<td></td>
</tr>
<tr>
<td>Estimate from element above</td>
<td></td>
</tr>
<tr>
<td>$L_i(x, y - 1)$</td>
<td></td>
</tr>
<tr>
<td>Estimate from element below</td>
<td></td>
</tr>
<tr>
<td>$L_i(x, y + 1)$</td>
<td></td>
</tr>
</tbody>
</table>
- Evolving program
  - Form element to population

**Fig. 8.7** A single processing element (a) is connected to four other processing elements. The elements are able to perform simple calculations and can obtain data from the neighboring elements. By combining such processing elements, we construct an $n \times n$ matrix with one processing element per image pixel (b).
- Processing element of constancy

\[
o_i(x, y) = \begin{cases} 
\frac{c_i(x, y)}{L_i(x, y)} & \text{if } L_i(x, y) > 0.001 \\
1 & \text{otherwise}
\end{cases}
\]

(8.9)

where \( i \in \{R, G, B\} \) for 3-channels.

- Illuminant estimation

\[
L_i(x, y) = \text{program}(L_i(x, y), L_i(x-1, y), L_i(x+1, y), L_i(x, y-1), L_i(x-1, y+1), c)
\]

(8.10)

where program is defined by tree-structure of individual,
\( c \) is original pixel,
and, initially, estimate \( L_i(x, y) \) is set to the color of input image.
- Three fitness cases in genetics

Fig. 8.8 Genetic operators used in the genetic programming experiments.
• Program code for best individual selection

$$(\text{div2} (+ (+ (\text{div2} (+ (\text{div2} \text{ down}) (\text{div2} (+ (\text{div2} (+ \text{band} (\text{div2} \text{ down})))) (\text{div2} (+ \text{band} (\text{div2} \text{ center}))))) (\text{div2} \text{ down})))))) (* (\text{div2} \text{ down}) (/ \text{right} (\text{div2} (+ \text{down} (+ (\text{div2} \text{ down}) (\text{div2} (+ (\text{div2} (+ \text{band} (\text{div2} \text{ down})))) (\text{div2} (+ \text{band} (\text{div2} (+ \text{band} (\text{div2} \text{ center}))))) (\text{div2} (+ (\text{div2} (+ \text{div2} \text{ right}) (\text{div2} (+ \text{band} (\text{div2} \text{ down}))))) (* (\text{div2} \text{ down}) (/ \text{right} (\text{div2} (+ \text{down} (+ (\text{div2} \text{ down}) (\text{div2} (+ (\text{div2} (+ \text{band} (\text{div2} \text{ center})))) (\text{div2} (+ \text{band} \text{band})))))))) \text{down})))$$
• Fitness statistics for ten runs with different seeds for number of generator
- Resulting of evolving color constancy

<table>
<thead>
<tr>
<th>Reflectance Images:</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>Actual Illuminant:</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
</tbody>
</table>

**Fig. 8.9** Performance of best evolved individual on three different fitness cases. Images in first row show reflectance images. Images in second row show virtual illuminant. Images in third row show input images presented to individual. Images in fourth row show illuminant that was estimated by the evolved individual. In last row, estimated reflectance is shown.
Fig. 8.9 Performance of best evolved individual on three different fitness cases. Images in first row show reflectance images. Images in second row show virtual illuminant. Images in third row show input images presented to individual. Images in fourth row show illuminant that was estimated by the evolved individual. In last row, estimated reflectance is shown.
Analysis of chromatic signals

- Most effective choice of chromatic contrast
  - Passing through white point
  - Spanned by red-green and blue-yellow

Fig. 8.10 Three-dimensional chromaticity space of a color cell.
– Getting stable hue descriptor

Fig. 8.11  Linear combination of the output of several neurons in order to arrive at receptors that respond to any desired hue.
Neural architecture based on double opponent cells

- Communication between neurons

\[ n_i = \sum_j w_{ij} f_j \]  

(8.11)

where \( f_j \) is the firing rate of neuron \( j \) and \( w_{ij} \) is weight connecting neuron \( i \) with \( j \).

- Internal cell potential \( E_i \)

\[ \frac{dE_i}{dt} = n_i - s(E_i - E_r) \]  

(8.12)

where \( E_r \), resulting potential, and, \( s \) is a scaling factor that can set to 1.
• Behavior of cell simulated using Euler’s method

\[ \Delta E_i = (n_i - E_i - E_r) \Delta t \]  \hspace{1cm} (8.13)

- Modeling of color opponent and double opponent cells

**Fig. 8.12** Modeling of color opponent and double opponent cells.
- Representation of color using 2-D coordinate system
  - Output of color opponent cells
    
    \[ z = w_1x - w_2 |y| \]  
    \( (8.14) \)
    
    where \( x \) is output of R-G channel, \( y \) output of B-Y channel.

  - 2-D coordinate
    
    \[
    \begin{pmatrix}
    x' \\
    y'
    \end{pmatrix}
    =
    \begin{pmatrix}
    \cos \theta & \sin \theta \\
    -\sin \theta & \cos \theta
    \end{pmatrix}
    \begin{pmatrix}
    x \\
    y
    \end{pmatrix}
    \]  
    \( (8.15) \)

  - Output of cell computing as:
    
    \[ z = w_1(x \cos \theta + y \sin \theta) - w_2 |y \cos \theta - x \sin \theta| \]  
    \( (8.16) \)
- Neural network for color classification and color constancy.