Abstract

A common procedure in digital postproduction is rotoscoping, the segmentation of independently moving foreground elements from background in a sequence of images. Still often carried out manually, rotoscoping is time-consuming and requires great skill in determining the boundary between foreground and background. Errors lead to a bubbling artefact in the final composited sequence. The industry is interested in automated rotoscoping. Any automatic segmentation method must correctly locate the boundary and be robust given rapid motion and non-static backgrounds. A cellular neural network for segmentation is presented that labels pixels by colour, estimated motion and neighbouring labels. The method is accurate, labour-saving and many times faster than manual rotoscoping.

Keywords: segmentation, cellular neural networks, motion, colour

1. Introduction

Segmentation of foreground and background is a common procedure in film post-production, permitting compositing of foreground elements against new backgrounds when conventional blue-screen or chromakey techniques are not possible. Rotoscoping, the identification of the boundary between foreground and background, is still often performed manually, requiring a skilled and patient operator. Processing times are very long and inaccuracies in boundary identification can result in the unwelcome bubbling artefact, where the boundary vibrates rapidly over the sequence.

An automatic segmentation algorithm is hence highly desirable. The requirements of the industry are strict: the boundary be accurately located; segmentation must be accurate in the presence of the rapid motion that is common in film sequences; bubbling must be eliminated; the algorithm must be fast; and operator intervention must be minimal.

Many characteristics of image data have been employed in segmenting images into background and foreground elements. Of these, motion has particular appeal [1], [6], [10]. Despite being a strong cue for distinguishing foreground from background, motion is insufficient constraint when applied at single pixels, resulting in large numbers of misclassified pixels [1]. The colour distributions of background and foreground, where these have little overlap, provide an alternative approach [9], [11]. More robust results are obtained by combining motion and colour [2], [3].

2. Segmentation from motion and colour

Assuming the motion of each pixel from image $I_{t-1}$ to $I_t$ is identical to its motion from $I_t$ to $I_{t+1}$, the motion of the largest independently moving region (global motion) of $I_t$ may be robustly recovered. This region is most often the background, which, when assumed distant and rigid, has visual motion modelable to high accuracy. Global-motion compensation aligns background pixels of $I_{t-1}$ and $I_{t+1}$ with those in $I_t$ to high accuracy, while motion-compensated foreground pixels tend to exhibit high error. The images representing the magnitude of errors (residuals) in aligning pixels of $I_{t-1}$ and $I_{t+1}$ with those in $I_t$ using the computed global motion are termed the backward and forward displaced frame differences (DFDs) respectively [7].

Since global-motion estimation models background motion accurately, pixels with low error in both DFDs tend to belong to the background. Pixels may therefore be hypothesised to belong to one of four classes based on the magnitude of error in the two DFDs: background pixels (label $B$) have low residuals in both DFDs; foreground pixels (label $F$) have high residuals in both DFDs; covered pixels (label $C$) are foreground in $I_{t-1}$ and background in $I_{t+1}$ and have high backward DFD residuals and low forward DFD residuals; uncovered (label $U$) pixels change from background to foreground and have low backward DFD residuals and high forward DFD residuals. Such pixels belong to the background; use of the covered and uncovered classes prevents
their misclassification, due to high residuals, as foreground.

A crude manual segmentation of $I_0$ provides a colour mask (figure 1) from which pdfs of the chromaticity components $I$ and $Q$ for foreground and background are obtained (figure 2). Colour information and constraints on the labels of neighbouring pixels augment the evidence for a pixel’s label. Labelling is performed by a maximum a posteriori relaxation scheme based on this evidence.

3. Segmentation by neural networks

Combining colour information with motion improves the accuracy of the boundary and reduces misclassifications. Segmentation masks still however contain large numbers of mislabelled pixels, while speckle obscures the foreground-background boundary (see figure 3). The algorithm is also slow. An alternative is put forward here using a cellular neural network (CNN) designed for segmentation based on motion and colour evidence. The VLSI implementation of CNNs is highly parallelisable and potentially much faster.

Figure 1. Hand-segmented Dolores image

Figure 2. Background (top) and foreground (bottom) chromaticity pdfs for the (colour) image in figure 1. Horizontal axes range over all values of chromaticity components $I$ and $Q$.

Figure 3. Motion-based segmentation mask

The CNN, developed by Chua and Yang [5], is a collection of regularly spaced cells that interact with their nearest neighbours. Each cell has an input from its neighbours that may alter its state and alter its output. A CNN models a large-scale non-linear analogue circuit for real-time signal processing. Outputs are binarised; it can be shown that the states converge over time; and cells may be four-connected like the pixels of an image. CNNs are therefore suited to image-processing tasks such as feature extraction [4], [8].

3.1. Cost function

The dynamics of a CNN are described by an ordinary differential equation, which may be approximated by a difference equation with a small time step. This dynamic equation is essentially a convolution. The optimisation procedure that determines the set of states in the limit is equivalent to finding the weights of a convolution that performs the required image-processing process. The filter is determined by minimising a suitable cost function.

For the purposes of segmentation, the cost function must be designed so that, on minimisation, the values of its constituent variables provide the labelling of pixels that best segments foreground from background. In order for minimisation to be possible, this function must be quadratic and have output in the interval $[0, 1]$.

Four neurons are assigned to each pixel of the original image, one for each label in the set $\Lambda = \{B, F, C, U\}$. Each neuron in a pixel competes with the others for the allocation of its label to the pixel and is subject to several constraints that control the labelling of adjacent pixels. The following
The four terms of the cost function constrain the probability $p_{ik}$ that $k$ is the most suitable label for pixel $i$. The coefficient $\pi_{ik} \in [0, 1]$ is the selection pressure, an input bias on the neurons that determines whether probabilities are to be selected ($\pi_{ik} \geq 0.5$) or suppressed ($\pi_{ik} < 0.5$). The selection pressure is determined from the evidence that a pixel belongs to a given class (its motion residuals and colour); eg, for the class $C$, $\pi_{ik}$ is defined as:

$$\pi_{ik} = \frac{p(k = C|a_i)}{p(e_i^1|\lambda_{UN})p(e_i^2|\lambda_{CH})p(c_i|\lambda_{GR})}$$

(2)

where $a_i = (e_i^1, e_i^2, c_i)$ is the evidence vector for the pixel. Components $e_i^1$ and $e_i^2$ are the motion residuals at the pixel in the backward and forward DFDs respectively and $c_i$ is the pixel’s colour. The labels $\lambda_{UN}$ and $\lambda_{CH}$ are those assigned to unchanged pixels (those having low residual in a DFD) and changed pixels (those with high residual) respectively. The label $\lambda_{GR}$ is that assigned to pixels in the background of the colour mask.

The spatial compatibility term favours compatible labels of neighbouring pixels over those that are incompatible. Eg, neighbouring foreground pixels are highly compatible, while adjacent foreground and background pixels are incompatible. The term $C_{ij}^k$ is the compatibility between pixels $i$ and $j$ with respective labels $k$ and $l$. This term helps ensure compactness of image elements, smoothness of the boundary and elimination of speckle.

The temporal compatibility term favours compatibility between the label of a pixel in the current and previous images. Compatibility is measured by $T_{kl}$ where $k$ is a potential label in the current image and $l$ is the pixel’s label in the previous. This term reduces the bubbling artefact.

The error term constrains the sum of a pixel’s label probabilities to be as near as possible to 1. Squaring the deviation from 1 ensures that this term is positive and also makes the resulting cost function quadratic, as required.

The relative effects of the terms may be adjusted by varying the weights $\alpha$, $\beta$ and $\gamma$.

Expanding the error term and collecting up like terms (interpreting both the first and third terms as input biases) gives the following cost function:

$$\gamma n - \sum_{i \in I} \sum_{k \in \Lambda} \chi_{ik} p_{ik} \sum_{i \in I} \sum_{k \in \Lambda} \sum_{j \in N(i)} \sum_{l \in \Lambda} \Gamma_{ij}^{kl} p_{ik} p_{jl}$$

(3)

where:

$$N(i) = N'(i) \cup \{i\}$$

(4)

$$\chi_{ik} = \pi_{ik} + \beta \sum_{l \in \Lambda} T_{kl} p_{il} + 2\gamma$$

(5)

$$\Gamma_{ij}^{kl} = \begin{cases} -\alpha C_{ij}^{kl} & j \in N'(i) \\ \gamma & j = i, k = l \\ 2\gamma & j = i, k \neq l \end{cases}$$

(6)

The dynamic equation of a CNN with cost function as in equation 3 may be approximated by a difference equation:

$$\frac{du_{ik}}{dt} = \frac{1}{C} \left[ \chi_{ik} - \frac{u_{ik}}{R} + \sum_{i \in I} \sum_{j \in N(i)} \Gamma_{ij}^{kl} g(u_{jl}) \right]$$

(7)

where $C$ and $R$ are respectively the capacitance and resistance of the CNN (when viewed as a circuit), $u_{jl}$ is the state of the neuron promoting label $l$ for pixel $j$ and $g(\cdot)$ is a sigmoid function converting state $u_{jl}$ into probability $p_{jl}$.

The cost function is minimised iteratively by applying the following algorithm for all $i \in I$ and for all $k \in \Lambda$:
1. Initialise each $\pi_{ik}$ as in equation 2;  
2. Initialise each $u_{ik}$ to $g^{-1}(\pi_{ik})$;  
3. Update $u_{ik}$ by applying a Runge-Kutta update procedure to the dynamic equation;  
4. Set $\pi_{ik} = g(u_{ik})$;  
5. Evaluate the cost function $\epsilon_{ik}^t$ for the given probabilities (t is the current iteration);  
6. If $|\epsilon_{ik}^{t+1} - \epsilon_{ik}^t|/|\epsilon_{ik}^{t+1}| > \text{tolerance (eg 1%)},$ go to 3.

4. Results

Figure 4 gives label images for an image of the Dolores. The upper image in figure 4 shows the initial labelling while the lower image shows labels obtained after 100 iterations. Mid-grey pixels are foreground, white pixels are background and others belong to the covered or uncovered classes. The relative motion of foreground and background is approximately diagonal and several pixels per frame in size. Motion later in the sequence is more rapid.

![Graph 4](image)

**Figure 5. Convergence of cost function**

In the limit, the label mask features few misclassified foreground pixels and very few misclassified background pixels. Holes remaining in the foreground are large, so may be rectified manually by the operator and maintained through the sequence by blob tracking.

In the final masks, pixels labelled as uncovered or covered are relabelled as background, the class to which these pixels actually belong.

Graph 5 gives the fraction of the original cost against the iteration number. The rapid minimisation of the cost function is apparent. Convergence of the cost function to a minimum is guaranteed [4].

5. Conclusions

The results indicate that a CNN accurately segments foreground from background. Speckle from misclassified pixels is almost eliminated, clearly revealing the location of the boundary. The temporal compatibility term ensures gradual changes to the boundary over time and is effective in removing the bubbling artefact.

The elimination of speckle ensures that any holes remaining in the foreground are large and hence easily identified by the operator for reclassification as foreground.

The covered and uncovered classes render the algorithm completely robust to relative foreground-background motion of any magnitude and prevent background pixels near the boundary from being misclassified as foreground.

User intervention is reduced from the need to rotoscope every image of the sequence precisely to a rough rotoscoping of the first image only and the identification of large holes in the foreground. Minimal demands on the operator, along with the parallelisability of the CNN algorithm, make this segmentation procedure extremely fast.

References