next up previous
Next: Feature Based Image **Up:** No Title **Previous:** Introduction

Intensity Based Image Mosaicing

We know that two images in a panoramic image sequence, or two images belonging to a planar scene, are related by an affine transormation. In particular, the image coordinates of any point in the scene, in two different images are related by an equation of the form:

$$\begin{aligned} x_i^{'} &= \frac{m_0 x_i + m_1 y_i + m_2}{m_6 x_i + m_7 y_i + 1} \\ y_i^{'} &= \frac{m_3 x_i + m_4 y_i + m_5}{m_6 x_i + m_7 y_i + 1} \end{aligned} \qquad (1)$$

The problem now becomes that of determining the transformation parameters m_i between every two adjacent images, in order to merge the set of images into a single complete image.

The idea would then be to choose the parameters m_i such that the sum of squared difference between all pixels between the two images is minimized. That is :

$$E = \sum_i \left[I^{'}\left(x_i^{'}, y_i^{'}\right) - I\left(x_i, y_i\right) \right]^2 = \sum_i e_i^2 \qquad (2)$$

The problem is that of non-linear minimization which can be solved by an iterative algorithm. The algorithm that was used in the present work, namely, the Levenberg Marquardt algorithm, involved computation of the approximate Hessian matrix at each stage of the iteration. This in turn, involved the computation of partial derivatives of the error with respect to the parameters being solved for. The following shows the derivation:

$$e_i = I^{'}\left(x_i^{'}, y_i^{'}\right) - I\left(x_i, y_i\right) \qquad (3)$$

As $I\left(x_i, y_i\right)$ is a constant,

$$\frac{\partial e_i}{\partial m_k} = \frac{\partial I^{'}}{\partial x^{'}} \frac{\partial x^{'}}{\partial m_k} + \frac{\partial I^{'}}{\partial y^{'}} \frac{\partial y^{'}}{\partial m_k} \qquad (4)$$

Using the above formula, the partial derivatives were calculated. They are:

$$\begin{aligned} \frac{\partial e_i}{\partial m_0} &= \frac{x_i}{D_i}\frac{\partial I^{'}}{\partial x^{'}}; \frac{\partial e_i}{\partial m_1} = \frac{y_i}{D_i}\frac{\partial I^{'}}{\partial x^{'}}; \frac{\partial e_i}{\partial m_2} = \frac{1}{D_i}\frac{\partial I^{'}}{\partial x^{'}} \\ \frac{\partial e_i}{\partial m_3} &= \frac{x_i}{D_i}\frac{\partial I^{'}}{\partial y^{'}}; \frac{\partial e_i}{\partial m_4} = \frac{y_i}{D_i}\frac{\partial I^{'}}{\partial y^{'}}; \frac{\partial e_i}{\partial m_5} = \frac{1}{D_i}\frac{\partial I^{'}}{\partial y^{'}} \end{aligned}$$

$$\frac{\partial e_i}{\partial m_6} = -\frac{x_i}{D_i}\left(x_i'\frac{\partial I'}{\partial x'} + y_i'\frac{\partial I'}{\partial y'}\right)$$
$$\frac{\partial e_i}{\partial m_7} = -\frac{y_i}{D_i}\left(x_i'\frac{\partial I'}{\partial x'} + y_i'\frac{\partial I'}{\partial y'}\right) \qquad (5)$$

The Hessian matrix $A = [a_{kl}]$ (which is symmetric), and the gradient $b = \{b_k\}$ at each stage has entries

$$a_{kl} = \sum_i \frac{\partial e_i}{\partial m_k}\frac{\partial e_i}{\partial m_l}$$
$$b_k = -2\sum_i e_i \frac{\partial e_i}{\partial m_k} \qquad (6)$$

The equations solved at each stage belong to the linear system:

$$(A + \lambda I)\Delta m = b \qquad (7)$$

where λ is a tuning parameter that is adjusted according to the change in the sum of squared differences at each stage. The following describes the algorithm in more detail:

1. Choose a small start value for λ. (Current work used $\lambda = 0.0001$)
2. For each pixel in the reference image (x_i, y_i), do:
    1. Compute the transformed coordinates (x_i', y_i') using the current estimate of the parameters m_k
    2. Compute the error in intensity for each pixel location
    3. Compute the intensity gradients in the unregistered image, namely, $\partial I'/\partial x'$ and $\partial I'/\partial y'$ at (x_i', y_i')
    4. Compute the partial derivatives $\partial e_i/\partial m_k$
    5. Add the pixel's contribution to the Hessian and gradient matrices **A** and **b**.
3. Solve the system of equations $(A + \lambda I)\Delta m = b$ and obtain the increment:

$$\Delta m \longleftarrow (A + \lambda I)^{-1} b \qquad (8)$$

4. Refine the current guess to obtain the new guess:

$$m \longleftarrow m + \Delta m \qquad (9)$$

5. If the sum of squared differences has decreased, reduce λ by a factor (typically 10). Else, increase λ a factor.
6. If the value of λ has reduced to a small amount (typically 10^{-14}) stop. Else, go to **2**.

The above algorithm was quite robust, with λ falling to the minimum in about 20 iterations.

next up previous

Next: Feature Based Image **Up:** No Title **Previous:** Introduction

---

Generated by latex2html-95.1
Fri Jul 12 16:03:24 EDT 1996