REAL-TIME COMPUTATION OF MOMENT INVARIANTS COMBINED WITH CONTRAST STRETCHING

A. L. C. Barczak, N.H. Reyes, T. Susnjak and M. J. Johnson

Institute of Information and Mathematical Sciences, Massey University
Private Bag 102904, North Shore City 0745, Auckland, New Zealand
phone: + (64) 9 4140800, fax: + (64) 9 441 8181, email: a.l.barczak@massey.ac.nz

ABSTRACT

There is a large body of evidence that demonstrates the practicable use of moment invariants in real-time computer vision applications. Object detection, recognition and tracking are some of them, to name a few. However, the efficacy of moment invariants is highly susceptible to varying illumination conditions, which is inherent in real-world applications. Contrast stretching alleviates the problem, but performing contrast stretching prior to the calculation of moment invariants is computationally intensive and not suitable for real-time use. We address this problem by proposing an efficient real-time method that integrates the calculation of moment invariants up to the 4th order with a contrast stretching operation (all in one go), by utilising Summed-Area Tables (SATs). The method is limited to general contrast stretching, not necessarily covering very distinct illumination sources from different directions; that is, illumination conditions that create extra strong edges. We test the proposed method with a popular benchmarking image database (Amsterdam Library of Object Images) that is publicly available. Such images were acquired in a controlled environment, demonstrating varying lighting conditions. We show empirically that the method works in real-time and accurate enough for practical object detection applications.

1. INTRODUCTION

Moment invariants were observed to be highly sensitive to illumination intensity changes [5]. Although one can use normalisation to overcome simple contrast problems, this prevents the simultaneous realisation of contrast invariance and scaling invariance [5]. Images with poor contrast, exhibiting either too dark or too bright areas, may benefit from a simple contrast stretching operation, as long as the contrast invariance can be achieved simultaneously with rotation invariance, scaling invariance and translation invariance.

Also, in an object detection problem it is often convenient to process sub-windows separately. Sub-windows processing, however, calls for a quick and efficient algorithm that can run in real-time. To this end, SAT is applied as it allows for the calculation of geometric moments in thousands of sub-windows (within the same frame or image) in real-time.

Contrast stretching operations work better when applied locally (to a specific sub-window) rather than globally (to the entire image). Unfortunately, if we apply contrast stretching operations to a specific sub-windows and modify the image, the use of SATs are rendered too slow, as the SATs would have to be recomputed for every sub-window. Ideally, we should be able to compute the SATs only once for the entire image, and still be able to extract the moment invariants from a given sub-window with contrast stretching. In this paper we propose a method that can achieve that. Effectively, the method allows for the computation of moment invariants from any sub-window in real-time, and the moments are invariant to rotation, translation, scaling and contrast.

In order to compare the proposed method with the traditional method, we have used three approaches. Firstly, we extract moments from images without any consideration to contrast. Images with the same objects have different illumination conditions, and one can measure the variance of these extracted moment invariants.

Secondly, we apply the contrast stretching operation to the sub-window where the object is contained, and extract the moments. In most cases, this helps to bring the variance down. This second approach serves as the "control" method for comparison purposes only. In practise, this method is too slow, as already explained above.

Lastly, we extract moments using the novel approximation technique that combines contrast stretching with geometric moments using SATs. This method can achieve quasi-invariant moments for grey-scale images, and works in real-time. Although only an approximation, empirical results show that in most cases this is better than trying to compute moments directly from the *raw* images with no contrast stretching.

2. RELATED WORK

Crow [3] used a simple structure called Summed-area Table (SAT) to speed up the computation of sum of pixels over arbitrary sized sub-windows. Viola and Jones [11] used the idea of SAT's to compute Haar-like features in real-time for face detection applications. In same direction, other researchers realised that they could use SATs to compute moment invariants in real-time. For example, [1] implemented algorithms to compute moment invariants over arbitrary sub-windows, including approximation of circles. They used eleven independent moments up to the 4^{th} order, which differs from the original seven moments by Hu ([7]). The eleven moment invariants were proposed by [4].

These eleven moment invariants can be computed directly from the 2-D geometric moments:

$$m_{pq} = \sum_x \sum_y x^p y^q i(x,y) \tag{1}$$

Where m_{pq} is the geometric moment of order pq, and $i(x,y)$ is a digital image. Using Summed-area Tables (SAT) one can speed-up the computation of moments over arbitrary sub-windows. In order to create these SATs, a simple recursive method can be used for each order pq:

$$M_{pq}(x,y) = M_{pq}(x-1,y) + M_{pq}(x,y-1) - M_{pq}(x-1,y-1) + i(x,y)x^p y^q \quad (2)$$

Where $M_{pq}(x,y)$ is a point of the SAT that contains the sum of all points between the origin $(0,0)$ and (x,y).

For a certain sub-window, four look-up points suffice to find the sum of the pixels that belong to it. For the moments up to the 4^{th} order one needs 15 SATs [1].

2.1 Linear Transformations

There are cases where transformations have to be applied separately to each sub-window. There are two ways to compute the moments with the transformed image.

The first is to compute the transform for a particular sub-window, recompute the SATs, and then compute the moments. In this case there is no advantage in using SATs, and if the number of sub-windows is large enough, it significantly slows the application.

The second way is to combine the transformation with the moments computation, in such a way that with the support of the SATs we can get an equivalent result.

This is possible if either the parameters of the linear transformation are known in advance, or if they depend on local information that the SATs can compute. For example, one can use $M_{0,0}(x,y)$ and compute the mean of the pixels.

Let $\bar{i}(x,y)$ be the image after a transformation:

$$\bar{i}(x,y) = ai(x,y) + b \quad (3)$$

2-D moments can be rewritten splitting the position components:

$$m_{pq} = \sum_n i(x,y)C_{pq} \quad (4)$$

Where $C_{pq} = x^p y^q$. The moments after the transformation are:

$$\bar{m}_{pq} = \sum_n \bar{i}(x,y)C_{pq} = \sum_n (ai(x,y)+b)C_{pq} \quad (5)$$

$$\bar{m}_{pq} = \sum_n (ai(x,y)C_{pq} + bC_{pq}) \quad (6)$$

$$\bar{m}_{pq} = a\sum_n (i(x,y)C_{pq}) + \sum_n (bC_{pq}) \quad (7)$$

but $\sum_n (i(x,y)C_{pq} = m_{pq}$, and therefore:

$$\bar{m}_{pq} = am_{pq} + b\sum_n C_{pq} \quad (8)$$

m_{pq} are the geometric moments of the image before the transformation, computed directly from the *raw* image. The expression $\sum_n C_{pq}$ can be pre-computed at the initialisation stage using SATs with “unit” images (all pixel values set to 1) and do not need to be repeated for the duration of the sequence of images. To find the value for each order, it suffices to know the 4 points that define the sub-window. For moments up to the 4_{th} order, we need 15 SATs for m_{pq} and 15 SATs for $\sum_n C_{pq}$. Using rectangular sub-windows, it requires 120 look-ups. In addition, the use of SATs allows the computation over arbitrary sub-window shapes (e.g., [1] used approximations to circular sub-windows).

Digital pixel values may overflow after a linear transformation of the form presented above, and therefore the equation 8 may become invalid [5]. In practise, however, only a few points suffer from overflow, and the equation may be used as a good approximation. This will be demonstrated with empirical experiments where the linear transformation is a function of the mean (μ) and the variance (σ) of the sub-window.

2.2 Approximating a Linear Transformation for Lighting Contrast Stretching

In object detection applications, it is more effective to apply contrast stretching operations to individual subwindows, instead of to the entire image. Lienhart et. al. [9] implemented a simple contrast stretching for Haar-like features, creating an extra SAT for the sum of the squares of the pixel values, so the variance can be easily obtained per sub-window.

A simple well-known contrast stretching uses maximum and minimum values of the image (or sub-window):

$$\bar{i}(x,y) = k\left(\frac{i(x,y) - min}{max - min}\right) \quad (9)$$

Where k is the maximum value represented in the digital pixel (e.g., 255 for greyscale with 8 bits), *min* and *max* are the minimum and maximum pixel values for the sub-window.

The problem is to extract the maximum and minimum points from an SATs. Obviously, this cannot be achieved by examining SATs that only contain sum of areas, and no detailed information about the image. Fortunately, one can approximate the minimum value to $\mu - c\sigma$ and the maximum value to $\mu + c\sigma$, where c is a tunable constant. Both μ and σ can be easily computed for any arbitrary sub-window using SATs [9]. Rewriting equation 9:

$$\bar{i}(x,y) = \frac{k(i(x,y) - (\mu - c\sigma))}{\mu + c\sigma - (\mu - c\sigma)} = \frac{k(i(x,y) - \mu + c\sigma)}{2c\sigma} \quad (10)$$

For a gray-scale image, from equation 10, with the restrictions $0 \leq \bar{i}(x,y) \leq 255$ for a greyscale 8 bits image, we have:

$$\bar{i}(x,y) = \frac{255(i(x,y) - \mu + c\sigma)}{2c\sigma}, c \in \Re^+ \quad (11)$$

In our experiments we tested constant c values between 1.0 to 2.0. Based on equation 5, the moments computed on the original image can be re-written as:

$$\bar{m}_{pq} = \frac{255}{2c\sigma}[m_{pq} + (c\sigma - \mu)\sum_n C_{pq}] \quad (12)$$

Equation 12 requires an extra SAT with the sum of the square of the pixel values, so we can compute σ over a sub-window. Therefore, we need a total of 124 look-up points (4 points per SAT), as we have 15 SATs for m_{pq}, 15 SATs for $\sum_n C_{pq}$ and one SAT for the sum of the square of the pixels.

Which value for the moments should be considered correct? When we carry out the contrast stretching in digital images, we are forced to round and cut-off pixel values to a certain interval, and therefore the values for the combined computation will vary widely when compared to the moments computed directly from the stretched image. Equation 12 does not round nor cut-off pixel values until the end, and therefore would be much closer to the moments computed using pixels represented by floating point values.

What we have to show is that the variance among the moments computed with equation 12 is smaller than those values computed with the *raw* images, making the feature extraction process more reliable under different illumination conditions.

3. EXPERIMENTS AND ANALYSIS

In this section we present results gathered using the Amsterdam Library of Object Images (ALOI) [6], specifically the set aloi_grey_red2_col. This set contains 1000 objects with 12 different illumination conditions.

For these tests, we implemented three algorithms using the approaches described in section 1. For every object, we computed 11 moment invariants (from ψ_1 to ψ_{11}) for the 12 different illumination conditions that the ALOI set contains. We measured the relative variance (σ/μ) for each case, and plotted graphs for all ALOI images (e.g. figure 4).

We computed 11 moments with the help of 31 SATs and equation 12. The first 5 moments (from ψ_1 to ψ_5) correspond to Hu's set ϕ_1 and ϕ_4-ϕ_7 ([7]). The other 6 moment invariants equations (from ψ_6 to ψ_{11}) can be derived from [5], or can be found explicitly in [1].

Through the naked eye, only subtle differences can be seen between figures 1, 2 and 3. Nevertheless, as a result of the application of the proposed contrast stretching, as depicted in figure 2, a more pronounced object shape was produced as compared to figure 3. More importantly, as can be observed in the calculation of moment invariants (figure 4), the proposed algorithm produced more stable moments corresponding to the target object. This is deemed vital for object detection tasks.

In figure 4 one can see three curves corresponding to the three methods used to compute the 11 moment invariants, namely "*raw*" (no contrast stretching), "*stretched*" (apply contrast stretching first, then compute the moments) and "*combined*" (the proposed method). The curve *raw* shows that the illumination conditions causes some variability for the computed moments of the same object within the same sub-windows for all the 12 illumination conditions. The curve *stretched* shows that the variance has been minimised, even though the computation could not be done in real-time if we were scanning several thousand sub-windows. The curve *combined* shows that the values are very close to the *stretched* line, the moments being quasi-invariant to illumination intensity.

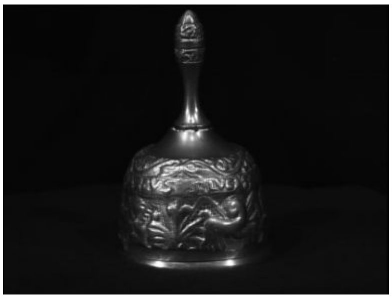

Figure 1: Original image (ALOI image 20) without contrast stretching.

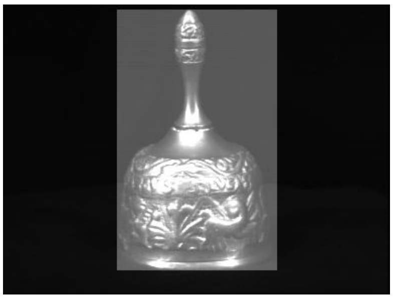

Figure 2: ALOI image 20: Contrast stretching applying equation 11 to the subwindow that encompass the object.

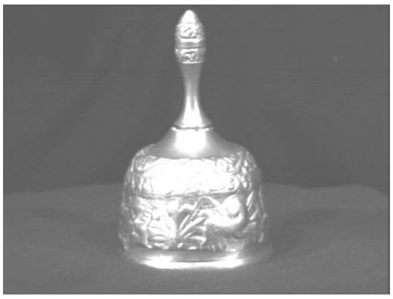

Figure 3: ALOI image 20: Contrast stretching applying equation 11 over the entire image. Note that the details of the object are not as clear as the image in figure 2. Also, the contrast stretching is influenced by any extreme pixels in the background even if located further from the object.

The only tunable parameter for the contrast stretching is the constant C (see equation 12). In figure 5, several values of C (1.0, 1.5, 1.75 and 2.0) for ALOI image 20 are shown, in order to compare the results. In this particular instance of ALOI image 20 (figure 4), the lowest C value worked best. For other ALOI images, other values of C achieved the best results.

For the purpose of comparison, we also plotted figure 6 to analyse the variance for all the figures of ALOI. The variance improves with the use of the *combined* method for most cases, and the most influenced moment is ψ_1. This is no surprise, given that this is the lowest order moment of the set, deemed to be more robust to noise.

Figure 7 shows the mean of the relative variance for the 11 moments using the proposed method (*combined*), for various values of C ($C = 1.0, 1.5, 1.75 and 2.0$). The figure shows that the best C for the mean over the ALOI set was $C = 2.0$, even though there is very little variation between $C = 2.0$, $C = 1.5$ $C = 1.75$. For $C = 1.0$, the relative variance is slightly higher. The parameter C between the values shown here does not influence the invariance in a clear way.

For most images contained in ALOI we observed that the 11 moments relative variance was just above 1% on average. When applying the proposed method the relative variance was reduced to a range between 0.1% to 0.8%. In rare cases (about 10% of the moments), we observed that the variance actually was slightly above the variance of the raw image. This is explained in section 3.1.

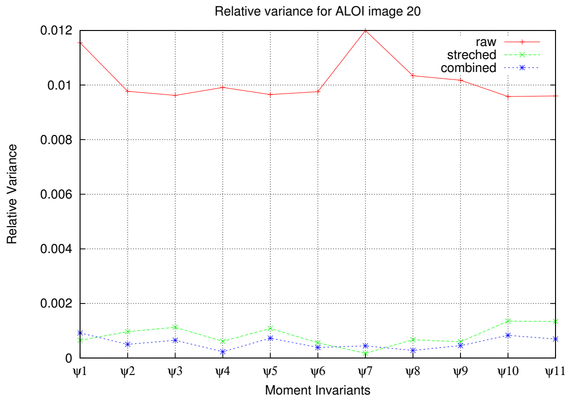

Figure 4: Relative variance (σ/μ) for 11 moments in ALOI image 20 with $C = 1.0$.

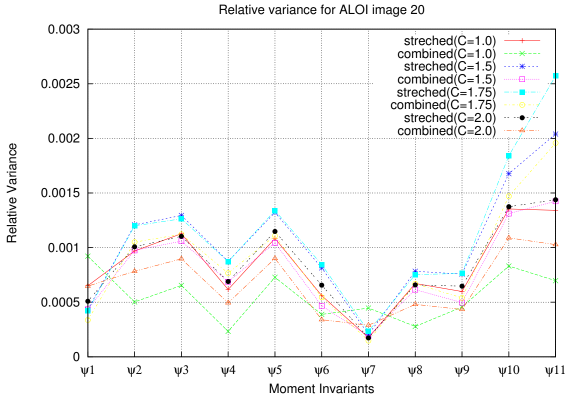

Figure 5: Comparison between the methods *stretched* and *combined*: relative variance (σ/μ) for 11 moments in ALOI image 20, with various values of C.

3.1 Numerical Instabilities of Moment Invariants

It is a well known problem that moment invariants based on geometric moments are prone to numerical instabilities, sensitivity to noise (see for example [8], and problems such as moments vanishing when the objects in the image are perfectly symmetric. Examples are cited in the literature, and many of these problems are sumarised in [5]. Another example is the comparison between Hu's moment invariants and Fourier descriptors for gesture recognition, which found moment invariants to be less robust to noise [2].

As an example of the problems that can be encountered, we present figures 8,9 and 10, which show ALOI image 1 and the contrast stretching operations.

The results for ALOI image 1 (figure 8) show that for one of the moments the relative variance was slightly above the equivalent one for the raw image (figure 11). This is a problem related to the sensitivity to noise and can also be influenced by the very shape of the object (e.g. symmetric shape), as shown in [5]. Despite this setback, we argue that the proposed method is still useful for practical applications such as object detection, as other moments still present a lower variance after the contrast stretching operation.

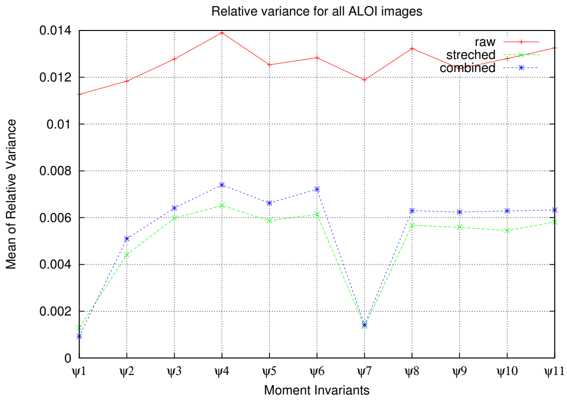

Figure 6: Comparison between the three methods (*raw*, *stretched* and *combined*): mean of the relative variance (σ/μ) for 11 moments in all ALOI images with $C = 1.0$ for *raw*, *stretched* and *combined* methods.

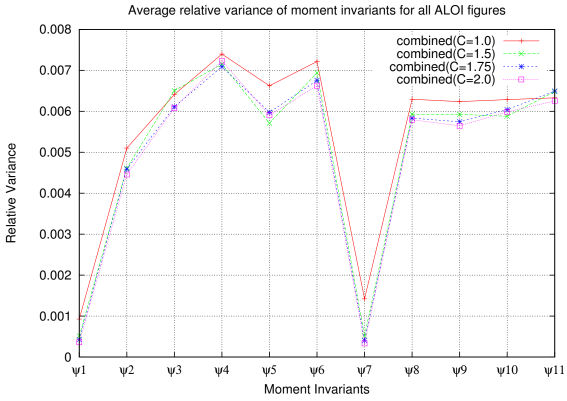

Figure 7: Proposed method("*combined*"): the mean of the relative variance for 11 moments in all ALOI images with various values of C.

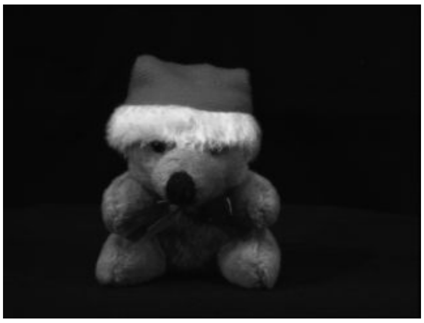

Figure 8: Original image (ALOI image 1) without contrast stretching.

3.2 Real-time Computations

Using the same approach Viola and Jones used for their object detection approach, we counted the number of subwindows and measured the runtime, estimating the frame rates.

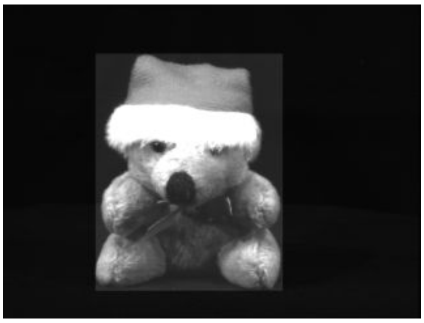

Figure 9: ALOI image 1: Contrast stretching applying equation 11 to the subwindow that encompass the object.

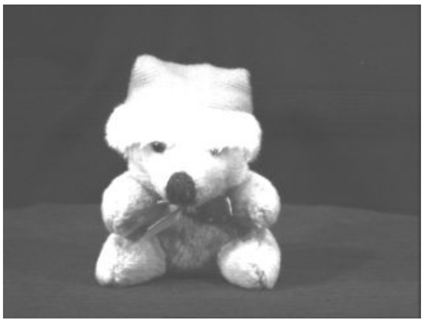

Figure 10: ALOI image 1: Contrast stretching applying equation 11 over the entire image.

Table 1 shows the number of sub-windows computed for a frame 384x288 (same as used in Viola-Jones [11]) using a 2GHz Intel, ordered by the number of sub-windows scanned. Inevitably, the performance of this approach will depend on frame size, scaling and translation factors (which define the number of sub-windows), and the processor speed. The moment/SAT code (without the contrast stretching) was implemented in an earlier work using a GPU [10]. This shows the potential of the combined approach for real-time applications

4. CONCLUSIONS AND FUTURE WORK

We presented and tested a real-time method for the combined computation of moment invariants and contrast stretching. The proposed method is simultaneously invariant to rotation, scaling, translation, and contrast. We have used the ALOI image database to benchmark the method and found that over 90% of the moments that have undergone the contrast stretching operation using the proposed method have improved their invariance to illumination. Our future goal is to apply this method in dynamic environments, using video cameras for object detection and recognition tasks. We also plan to investigate the conditions in which the moment invariants become less robust to noise and symmetry.

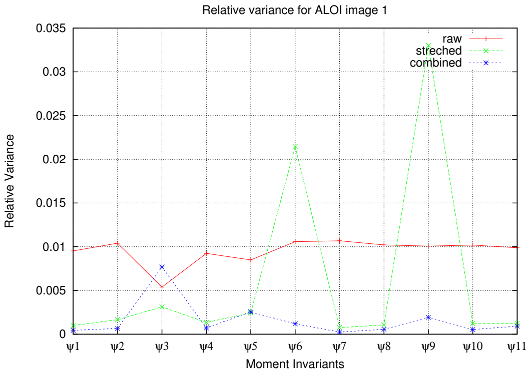

Figure 11: Relative variance (σ/μ) for 11 moments in ALOI image 1 with $C = 1.0$.

Table 1: Frame rates for the combined approach

scaling factor	translation factor	sub-windows	frames/sec
1.5	4	17017	17.49
1.4	4	20063	15.62
1.2	4	34281	11.12
1.5	2	67686	6.61
1.4	2	79460	5.87
1.2	2	135647	3.70

REFERENCES

[1] A. L. C. Barczak and M. J. Johnson. A new rapid feature extraction method for computer vision based on moments. In *IVCNZ 2006*, pages 395–400, Auckland, NZ, November 2006.

[2] S. Conseil, S. Bourennane, and L. Martin. Comparison of fourier descriptors and hu moments for hand posture recognition. In *15th EUSIPCO*, pages 1960–1964, Poznan, Poland, 2007.

[3] F. C. Crow. Summed-area tables for texture mapping. *Computer Graphics*, 18 (3):207–212, July 1984.

[4] J. Flusser. On the independence of rotation moment invariants. *Pattern Recognition*, 33:1405–1410, 2000.

[5] J. Flusser, B. Zitova, and T. Suk. *Moments and Moment Invariants in Pattern Recognition*. Wiley Publishing, 2009.

[6] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The Amsterdam library of object images. *Int. J. Comput. Vis.*, 61(1):103–112, 2005.

[7] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8:179–187, 1962.

[8] K. Kotoulas and I. Andreadis. Image analysis using moment. In *5th Int. Conf. on Technology and Automation*, pages 360–364, Thessaloniki, Greece, 2005.

[9] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *ICIP02*, pages I: 900–903, Rochester, NY, September 2002.

[10] C. Messom and A. Barczak. Stream processing of geometric and central moments using high precision summed area tables. In *ICONIP08*, pages 1095–1102, Auckland, New Zealand, 2008.

[11] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.