ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2012. No. 4(120)
ELEKTRONIKA IR ELEKTROTECHNIKA

T 120 SYSTEM ENGINEERING, COMPUTER TECHNOLOGY
SISTEMŲ INŽINERIJA, KOMPIUTERINĖS TECHNOLOGIJOS

Energy Consumption and Quality of Approximate Image Transformation

R. Damasevicius, G. Ziberkas
Software Engineering Department, Kaunas University of Technology,
Studentų str. 50, LT-51368, Kaunas, Lithuania, phone: +37037300399,
e-mails: damarobe@soften.ktu.lt, ziber@soften.ktu.lt

crossref http://dx.doi.org/10.5755/j01.eee.120.4.1459

Introduction

When developing software for mobile devices (such as laptops, tablets, PDAs, smart phones, etc.) it is very important to take the amount of actually available hardware resources into account. The resources (CPU speed, RAM size, battery capacity) are usually very constrained. Therefore, it is important to look for the methods allowing to save on the consumption of resources required for the provision of a service at the expense of other characteristics such as service quality [1, 2].

One of the aspects of services provided by a mobile device is the quality of images, videos or graphical user interfaces (GUIs) rendered on the screen of a mobile device [3]. Due to the smaller screen size, the user is often satisfied with poorer image quality. Therefore, when developing GUIs and rendering images (e.g., in mobile photo editing applications), software solutions are required for the implementation of resource-aware algorithms that produce the required result at acceptable quality loss (we do not analyze the hardware-based solutions to the problem here).

This paper analyses the implementation of image transformation algorithm in a mobile environment using the approximate computation methods. Energy consumption vs. image quality trade-offs are analyzed using image quality metrics. Finally, recommendations for mobile application developers are formulated.

Data specialization and approximate computation

We consider two methods for improving efficiency of resource-greedy calculations: 1) data specialization (using caching in look-up tables); and 2) approximate computation (using function approximation).

Data specialization [4] involves separation of early and late computations, where early computations are performed in advance and the results are stored in look-up tables. Such look-up tables can be generated on demand by meta-programs using the meta-programming methods [5].

Another method is to use function approximations such as the Taylor series (see Eq. 1), Padé approximants (see Eq. 2) or Chebyshev polynomials (see Eq. 3) [6]:

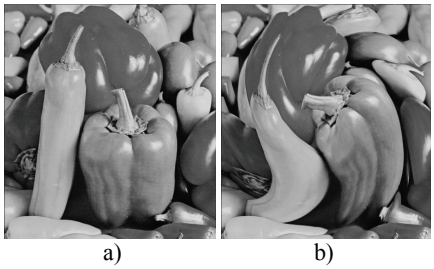
$$f(x)=\sum_{k=0}^{n} a_k x^k \,, \tag{1}$$

$$f(x)=\frac{\sum_{k=0}^{n} a_k x^k}{\sum_{k=0}^{m} b_k x^k}, \tag{2}$$

$$f(x)=\sum_{k=0}^{n} a_k T_k \,. \tag{3}$$

Image transformation using approximately calculated trigonometric functions

To demonstrate image transformation using the approximate computation methods, we have selected the Twirl effect (see Fig. 1). The Twirl effect distorts an image by rotating a layer around its centre. This effect is often provided by image editing software.



a) b)

Fig. 1. Original *Peppers* image (a) and image processed with the *Twirl* effect (b)

The algorithm for implementing the Twirl effect is given in Fig. 2. Using data specialization, we calculate the sine and cosine values, and the difference of adjacent values at a meta-program (i.e., program generator) level and store values in a data array. Also the function for angle reduction and linear interpolation between function argument values is generated (Fig. 3).

```
// image – image under transformation
// (cX, cY) – coordinates of the image center
// width, height -  width and height of the image
// (x, y) – original coordinates of an image point
// (xn, yn) – new coordinates of an image point

foreach point (x, y) in image do
   dx = x - cX
   dy = y - cY
   icX = width * cX
   icY = height * cY
   radius = min(cX, cY)
   radius2 = radius * radius
   distance = dx * dx + dy * dy
   if (distance > radius2) then begin
      xn = x
      yn = y
   end
   else begin
      distance = sqrt(distance)
      a = atan2(dy, dx) +
          angle * (radius - distance) / radius
      xn = cX + distance * cos(a)
      yn = cY + distance * sin(a)
   end if
end foreach
```

Fig. 2. Algorithm of the *Twirl* effect

```
public double cosSuRedukcija (double x)
{
  double xx = x % 6.283185307179586;
  if (xx < 0)
    xx *= -1;
  byte dalis = (byte) (xx * 0.6366197723675814), indeks;
  xx %= 1.57079632679489 66;
  indeks = (byte) (12.095775674984045 * xx);
  switch (dalis)
  {
    case 0: return kosinus[indeks] + 12.095775674984045 *
      (xx - 0.08267349088394192 * indeks) * skirtu1[indeks];
    case 1: return -sinusai[indeks] - 12.095775674984045 *
      (xx - 0.08267349088394192 * indeks) * skirtum[indeks];
    case 2: return -kosinus[indeks] - 12.095775674984045 *
      (xx - 0.08267349088394192 * indeks) * skirtu1[indeks];
    case 3: return sinusai[indeks] + 12.095775674984045 *
      (xx - 0.08267349088394192 * indeks) * skirtum[indeks];
  }
  return 0;
}
```

Fig. 3. Generated function for cosine calculation using reduction

Evaluation of image quality

Evaluation of image quality is often a subjective process, which requires experience and expert knowledge. However, quantitative image quality metrics [7, 8] also can be used such as:

1) *Normalized Absolute Error* (NAE)

$$NAE = \frac{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|x(i,j)-x'(i,j)\right|}{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}x(i,j)}, \quad (4)$$

here $x(i,j)$ is the color value of the original image at (i,j); $x'(i,j)$ is the color value of the transformed image; and m, n are maximal indices of row and column pixels.

2) *Peak Signal-to-Noise Ratio* (PSNR), measured in decibels (dB)

$$PSNR = 10\lg\frac{65025}{MSE}, \quad (5)$$

here MSE is Mean Square Error and

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[x(i,j)-x'(i,j)\right]^2. \quad (6)$$

If the PSNR value is higher than 30 dB, image quality is excellent; PSNR between 25 dB and 30 dB means good quality; PSNR less than 20 dB means poor quality.

3) *Laplacian Mean Square Error* (LMSE)

$$LMSE = \frac{\sum_{i=1}^{m-2}\sum_{j=1}^{n-2}\left[L(x(i,j))-L(x'(i,j))\right]^2}{\sum_{i=1}^{m-2}\sum_{j=1}^{n-2}L^2(x(i,j))}, \quad (7)$$

here

$$L(x(i,j)) = x(i-1,j)+x(i+1,j)+ \\ +x(i,j-1)+x(i,j+1)-4x(i,j). \quad (8)$$

The larger LMSE value means poorer image quality.

Research method

The research was performed on a laptop computer with Windows XP Professional OS, Intel Core 2 Duo E6420 2.13 GHz CPU, 2 GB RAM, Li-ion battery pack SQU-503 10.8 V, 4800mAh. Windows XP is still installed on a majority of world computers (May, 2011 data). Battery drain was measured using the *Intel Application Energy Command Line Tool* (a part of the *Intel Application Energy Toolkit,* available from http://software.intel.com/en-us/articles/application-energy-toolkit/), which provides total and net energy consumption of an application running on a battery-operated system.

Selected greyscale images (Lena, Mandrill and Peppers) from the USC-SIPI Image Database (available at http://sipi.usc.edu/database/database.php) were transformed by the Twirl effect with different rotation angle values using standard library trigonometric functions and approximate computation functions. Execution time and energy consumption of standard and approximate twirling functions during image generation were measured (image rendering on the screen was not included in the experiment). The obtained images were compared using image quality metrics. Each experiment was repeated 5 times and average values were calculated.

For approximation of sine and cosine functions, meta-programs were developed, which generate Java (a popular language for developing mobile applications) code for a required polynomial order of the Taylor series and the look-up table size.

As a comparison, a *chebyfit* function from the *Python* library *mpmath* is used to generate the near-optimal Chebyshev polynomials in the defined interval, and the *pade* function is used to calculate the Padé rational approxi-

mants of a required order of polynomials in the numerator and denominator of a rational fraction.

Research results and evaluation

The research results are presented in Fig. 4–Fig. 7. The markings used in the charts are explained in Table 1.

Table 1. Summary of approximate computation functions and their marking in figures

No.	Marking	Exaplanation
1	◇	2-term *Taylor* polynomial at $x = 0$
2	□	4-term *Taylor* polynomial at $x = 0$
3	△	6-term *Taylor* polynomial at $x = 0$
4	×	2-term *Chebyshev* polynomial in [0, π / 2]
5	✳	4-term *Chebyshev* polynomial in [0, π / 2]
6	●	6-term *Chebyshev* polynomial in [0, π / 2]
7	+	*Padé* approximant in [0, π / 2], 2nd order polynomial in numerator and 2nd order polynomial in denominator
8	-	*Padé* approximant in [0, π / 2], 4nd order polynomial in numerator and 2nd order polynomial in denominator
9	—	*Padé* approximant at $x = \pi / 4$, 2nd order polynomial in numerator and 2nd order polynomial in denominator
10	◆	Look-up table for 4 values in [0, π / 2]
11	■	Look-up table for 10 values in [0, π / 2]
12	▲	Look-up table for 20 values in [0, π / 2]

Energy savings using the approximately computed trigonometric functions as compared to the standard library-based implementations are summarized in Fig. 4.

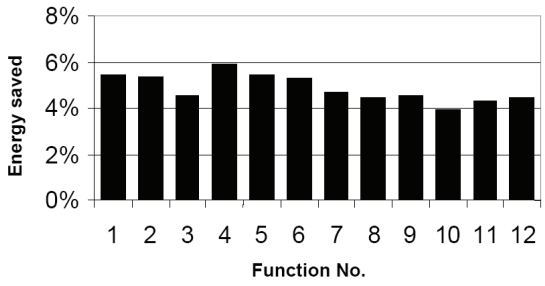


Fig. 4. Energy savings using approximate functions (function no. on x-axis, see in Table 1) as compared to standard Java *Math* library functions

Evaluation of image quality loss calculated using the NAE (Eq. 4), PSNR (Eq. 5) and LMSE (Eq. 7) metrics is given in Figs. 5-7.

After evaluating the energy consumption measurements and the results of the image quality metrics, we claim that the best ratio of image quality vs. energy consumption can be achieved using the 4-term and 6-term Chebyshev polynomials. Other approximate computation methods also can be used, however using less accurate methods may lead to various image rendering artefacts, which cannot be attributed to the image transformation algorithm itself. Moreover it should be noted that quality loss when rendering images on small-sized screens of mobile devices in most cases is almost unnoticeable by the human eye (see Fig. 8).

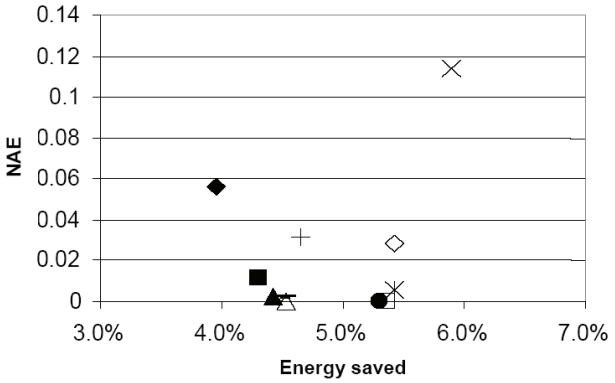


Fig. 5. Image quality according to the NAE metric vs. energy savings for 12 approximate computation functions from Table 1

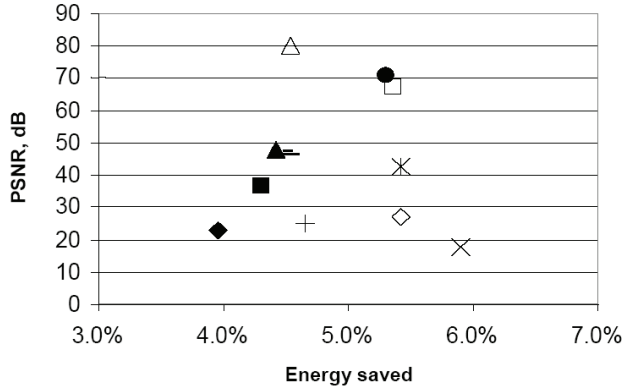


Fig. 6. Image quality according to the PSNR metric vs. energy saving for 12 approximate computation functions from Table 1

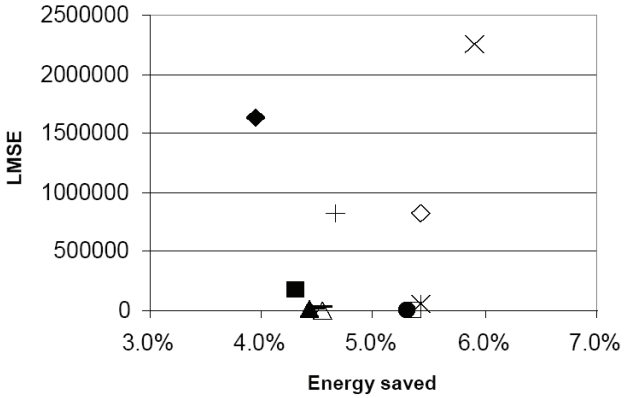
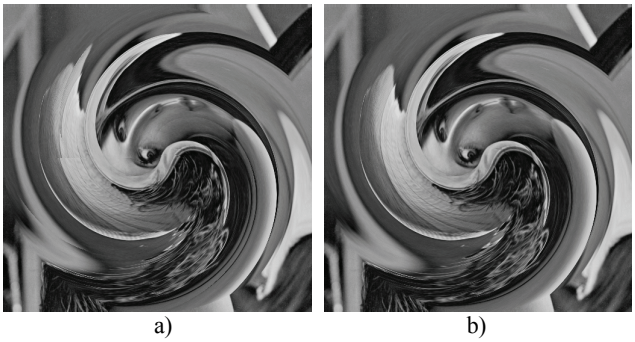


Fig. 7. Image quality according to the LMSE metric vs. energy saving for 12 approximate computation functions from Table 1



a) b)

Fig. 8. Lena images after applying the approximate Twirl effect using 2-term (a) and 6-term (b) Chebyshev polynomial

Related work

Related research is performed on the hardware level, e.g. using domain-specific accelerators of energy-aware graphics operations [9], and on the software level using the energy-optimal combination of parameter values of image compression and resize operations [10]. Data specialization methods are also applied for improving characteristics of various Java programs, including in image processing algorithms [11]. The energy efficiency of the application-specific graphical operations performed on a mobile device is addressed in [12].

Conclusions and future work

The approximate computation methods can be applied in the image processing algorithms with user-acceptable loss of image quality. This is especially relevant in mobile devices, where system factors (e.g., smaller screen size) and external factors (e.g., high sunlight) prevent a user from enjoying images at a highest quality. The designer of mobile applications can evaluate the quality of pictures and interfaces rendered on the screen using image quality metrics (such as NAE, PSNR and LMSE) and select approximate calculation methods that produce the required result without noticeable loss of quality while at the same time allow saving energy consumed by a mobile device.

Using the approximate calculation methods allow to save up to 6% of energy in a mobile environment when transforming the greyscale images using the *Twirl* effect. The best results can be achieved using 4-term and 6-term Chebyshev approximations of the trigonometric functions used in image transformation algorithms instead of standard library-based implementations.

Future work will involve experimental research of wide range of image processing operations.

Acknowledgements

The authors wish to acknowledge the significant contribution of Mindaugas Petkevičius in the collection of the experimental data.

References

1. **Mayo R. N., Ranganathan P.** Energy Consumption in Mobile Devices: Why Future Systems Need Requirements–Aware Energy Scale–Down // Proc. of 3rd Int. Workshop on Power–Aware Computer Systems (PACS'2003). – San Diego, CA, USA, Springer, 2004. – Vol. 3164. – P. 26–40.
2. **Toldinas E., Stuikys V., Damasevicius R., Ziberkas G.** Application–level energy consumption in communication models for handhelds // Electronics and Electrical Engineering – Kaunas: Technologija, 2009. – No. 6(94). – P. 73–76.
3. **Lin C.–H., Liu J.–C., Liao C.–W.** Energy analysis of multimedia video decoding on mobile handheld devices // Comput. Stand. Interfaces, 2010. – No. 1–2(32). – P. 10–17.
4. **Knoblock T. B., Ruf E.** Data Specialization // SIGPLAN Notices, 1996. – No. 5(31). – P. 215–225.
5. **Damasevicius R., Stuikys V., Toldinas E.** Embedded program specialization for multiple criteria trade–offs // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 8(88). – P. 9–14.
6. **Green R.** Faster Math Functions // Proc. of Game Developers Conference. – San Francisco, CA, USA, 2003.
7. **Mrak M., Grgic S., Grgic M.** Picture quality measures in image compression systems // Proc. of IEEE EUROCON'2003. – Ljubljana, Slovenia, 2003. – Vol. 1. – P. 233–236.
8. **Vysniauskas V.** Triangle based Image Magnification // Electronics and Electrical Engineering. – Kaunas: Technologija, 2006. – No. 6(70). – P. 45–48.
9. **Dasika G. S., Fan K., Mahlke S. A.** Power–efficient medical image processing using PUMA // Proc. of the IEEE 7th Symposium on Application Specific Processors. – San Francisco, CA, USA, 2009. – P. 29–34.
10. **Hussain S. A., Razzak M. I., Minhas A. A., Sher M., Tahir G. R.** Energy Efficient Image Compression in Wireless Sensor Networks // International Journal of Recent Trends in Engineering (IJRTE), 2009. – No. 1(2). – P. 117–120.
11. **Schultz U. P., Lawall J. L., Consel C.** Automatic program specialization for Java // ACM Transactions on Programming Languages and Systems, 2003. – No. 4(25). – P. 452–499.
12. **López M. B., Nykänen H., Hannuksela J., Silvén O., Vehviläinen M.** Accelerating image recognition on mobile devices using GPGPU // Parallel Processing for Imaging Applications (SPIE Proceedings), 2011. – Vol. 7872.

Received 2011 07 05
Accepted after revision 2011 10 03

R. Damasevicius, G. Ziberkas. Energy Consumption and Quality of Approximate Image Transformation // Electronics and Electrical Engineering. – Kaunas: Technologija, 2012. – No. 4(120). – P. 79–82.

The paper analyses a problem of effective image processing in mobile devices with limited resources. We analyse various approximate calculation methods for improving energy characteristics of components of image processing algorithms that are especially energy and CPU resource greedy. We research an image transformation applying the Twirl effect using data specialization (caching in look-up tables) and approximate computation of the trigonometric functions using Taylor series, Padé rational fractions and Chebyshev polynomials. Quality of transformed images is evaluated using standard image quality metrics. Based on the experimental results, the recommendations for mobile application developers are formulated. Ill. 8, bibl. 12, tabl. 1 (in English; abstracts in English and Lithuanian).

R. Damaševičius, G. Ziberkas. Apytikrei vaizdo transformacijai naudojama energija ir kokybė // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2012. – Nr. 4(120). – P. 79–82.

Analizuojama efektyvaus vaizdų apdorojimo mobiliuosiuose įrenginiuose su ribotais ištekliais problema. Nagrinėjami apytikslio skaičiavimo metodai, taikomi skaičiavimams imlių vaizdų apdorojimo algoritmų komponentų energijos naudojimo charakteristikoms pagerinti. Tiriamas vaizdų transformavimas taikant susukimo efektą duomenims specializuoti (žinomoms reikšmėms saugoti duomenų lentelėse) ir aproksimuoti Teiloro, Čebyšovo ir Padė polinomais. Gautų vaizdų kokybė įvertinama naudojant standartines vaizdų kokybės metrikas. Pateikiamos rekomendacijos mobilios aplinkos taikomųjų vaizdų apdorojimo programų programuotojams. Il. 8, bibl. 12, lent. 1 (anglų kalba; santraukos anglų ir lietuvių k.).