Grafika

Szare jest piękne

Bogusław Jackowski

Wprowadzenie

Przypuszczalnie większość użytkowników komputerów osobistych zdążyła się już zetknąć z problematyką modelowania odcieni szarości. Praktycznie każdy program graficzny, taki jak *PC Paintbrush*, *Aldus PhotoStyler*, *Corel Photo-Paint*, *Micrografx Picture Publisher*, itp., posiada — wśród wielu innych — opcję zwaną *halftoning*. Cóż to oznacza?

Otóż załóżmy, że dysponujemy danymi (pochodzącymi np. ze skanera lub stanowiącymi wynik obliczeń) zawierającymi zapis obrazu w postaci pikseli o różnych poziomach szarości, powiedzmy od 0 do pewnej liczby n. Nie nastręcza kłopotu wizualizacja tych danych na urządzeniu mającym możliwość odtwarzania pikseli z zachowaniem informacji o odcieniach szarości, np. na ekranie obsługiwanym przez kartę graficzną *VGA*. Niestety, są też urządzenia, które mają jedynie piksele dwustanowe — włączony/wyłączony (karta graficzna *Hercules*, typowe drukarki, fotonaświetlarki). Jeśli więc obraz z odcieniami szarości ma zostać przesłany na takie urządzenie, zachodzi konieczność przekodowania „szarych” danych na dane „czarno--białe”. Zabieg ten to właśnie *halftoning*, czyli po polsku *cieniowanie*.

Idea jest prosta: jeden lub więcej szarych pikseli należy zastąpić pewną liczbą pikseli czarnych i białych ułożonych w stosowny wzorek. Okazuje się, że ostateczny efekt w sposób niezwykle istotny zależy od użytego wzorku. Jeśli redukowana jest przy tym ilość informacji, np. jeśli na wejściu mamy piksele o 65-ciu poziomach szarości, a do modelowania używamy kwadratów o rozmiarach 4×4 piksele (17 odcieni), to sposób redukcji też oczywiście ma wpływ.

Niniejszy tekst jest przedrukiem — z niewielkimi zmianami — artykułu opublikowanego w materiałach III Górskiej Międzynarodowej Szkoły PTI (Szczyrk, czerwiec 1991).

Moim celem jest poglądowe omówienie podstawowych problemów, na jakie można się natknąć przy cieniowaniu oraz praktycznych sposobów rozwiązywania tych problemów.

W omawianych przykładach będę korzystał z danych o 65-ciu odcieniach szarości reprezentowanych przy pomocy znaków ASCII w następujący mnemotechniczny sposób:

znak ‘`.`’ oznacza kolor biały (poziom 0),
znak ‘`:`’ — pierwszy poziom szarości,
znak ‘`|`’ — drugi poziom,
cyfry ‘0’...‘9’ — poziomy szarości 3...12,
litery ‘a’...‘z’ — poziomy szarości 13...38,
litery ‘A’...‘Z’ — poziomy szarości 39...64.

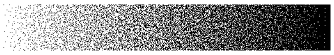
Niniejszy tekst wraz z ilustracjami został wydrukowany na drukarce laserowej typu *Hewlett-Packard* o rozdzielczości 600×600 pikseli na cal, przy czym ilustracje zostały przygotowane z rozdzielczością 300×300 pikseli na cal.

Podejście naiwne

Najprostsze możliwe dane to pasek o narastającym równomiernie zaczernieniu:

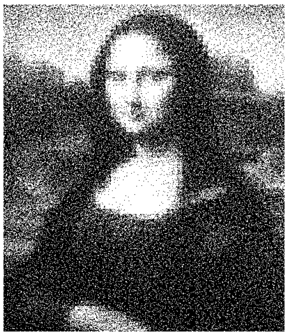
```
.:|0123456789abc   ···   KLMNOPQRSTUVWXYZ
                    ⋮
.:|0123456789abc   ···   KLMNOPQRSTUVWXYZ
```

zaś najprostszy możliwy wzorek odpowiadający poziomowi szarości k to kwadrat 8×8 pikseli zawierający k czarnych pikseli i $64 - k$ białych pikseli rozmieszczonych losowo. Wynik jest — można rzec — ciekawy:



Ilustracja 1

Niestety, w przypadkach bliższych życia ten naiwny sposób zawodzi:



Ilustracja 2

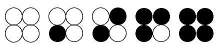
Ilustracja 2 przedstawia rezultat takiego postępowania z danymi zaczerpniętymi z pracy Knutha (D. E. Knuth, *Fonts for Digital Halftones*, TUGboat, Vol 8, 1987, No. 2, str. 135–160; p. lewy łam). Tak przetworzony obraz, o fakturze przypominającej akwatintę, może się przydać w zasadzie do celów specjalnych.

Rozmycie uporządkowane

Stanęliśmy zatem przed problemem w jaki sposób „rozmywać" szare piksele (w literaturze anglojęzycznej używa się określenia *dither*, co oznacza *drżenie*, *dygotanie*). Nietrudno wyobrazić sobie jak można by reprezentować odcienie szarości o poziomach 0–4. W sposób dość naturalny nasuwa się zapełnianie kwadratu 2×2 w kolejności:

2	1
0	3

Innymi słowy odcieniom szarości 0–4 odpowiadałyby piksele poczwórne:



Mniej oczywiste jest rozmywanie większej liczby poziomów szarości. Sposób zaproponowany przez Bayera rekursywnie powiela ten schemat (B. E. Bayer, *An Optimum Method for Two-Level Rendition of Continuous-Tone Pictures*, *Int. Conf. Commun.*, 1973, str. (26-11)–(26-15)). Podejście to można by określić jako *rozmycie uporządkowane* (ang. *ordered dither*) w odróżnieniu od rozmycia przypadkowego, określonego przed chwilą jako „podejście naiwne". Bayer zaproponował następujące związki rekurencyjne dla wyznaczania macierzy $\mathbf{B}^{(k)}$, $k = 1, 2, 4, 8, \ldots$, określających kolejność zapełniania kwadratu $k \times k$:

$$\mathbf{B}^{(1)} = [0]$$

$$\mathbf{B}^{(k)} = \begin{bmatrix} 4\mathbf{B}^{(k/2)} + 2\mathbf{E}^{(k/2)} & 4\mathbf{B}^{(k/2)} + \mathbf{E}^{(k/2)} \\ 4\mathbf{B}^{(k/2)} & 4\mathbf{B}^{(k/2)} + 3\mathbf{E}^{(k/2)} \end{bmatrix}$$

gdzie $\mathbf{E}^{(k)}$ jest macierzą $k \times k$, taką że $\mathbf{E}^{(k)}_{i,j} = 1$.

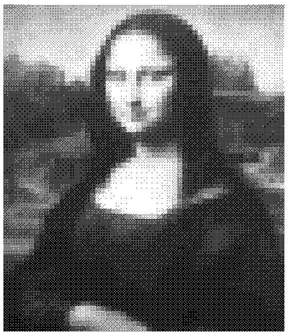
Dla $k = 4$ oraz $k = 8$ otrzymuje się wówczas następującą kolejność zapełniania:

$$\mathbf{B}^{(4)} = \begin{array}{|c|c|c|c|} \hline 10 & 6 & 9 & 5 \\ \hline 2 & 14 & 1 & 13 \\ \hline 8 & 4 & 11 & 7 \\ \hline 0 & 12 & 3 & 15 \\ \hline \end{array}$$

$$\mathbf{B}^{(8)} = \begin{array}{|c|c|c|c|c|c|c|c|}\hline 42&26&38&22&41&25&37&21\\\hline 10&58&6&54&9&57&5&53\\\hline 34&18&46&30&33&17&45&29\\\hline 2&50&14&62&1&49&13&61\\\hline 40&24&36&20&43&27&39&23\\\hline 8&56&4&52&11&59&7&55\\\hline 32&16&44&28&35&19&47&31\\\hline 0&48&12&60&3&51&15&63\\\hline\end{array}$$

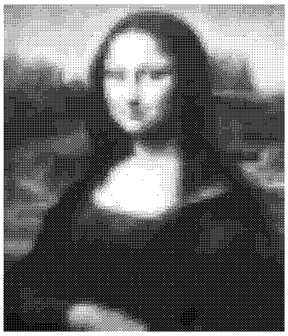
Rozmywanie uporządkowane sprowadza się w istocie do kolejnego zaczerniania możliwie dalekich od siebie pikseli.

Mimo tego dość wyrafinowanego sposobu rozmywania pikseli „Knuthowa" *Mona Lisa* nadal nie prezentuje się najpiękniej (ilustracja 3). Wyraźnie widoczne są elementy mozaiki wielkości 8×8 pikseli. Przy zastosowanej rozdzielczości (300 dpi) nieuniknione wydaje się modelowanie szarości za pomocą pikseli 4×4, co oznacza redukcję liczby odcieni z 65 do 17.



Ilustracja 3

Żeby zapobiec zmniejszaniu się obrazu wraz ze zmniejszeniem pikseli, można wstawić między każde dwa wiersze i każde dwie kolumny dodatkowy wiersz i dodatkową kolumnę metodą interpolacji liniowej, a następnie każdy piksel szarości k zamienić na piksel o szarości $k \,\mathbf{mod}\, 4$.



Ilustracja 4

Rezultat (ilustracja 4) jest niewiele lepszy: wprawdzie kontury postaci nie są już takie kanciaste, wyraźnie jednak zauważa się zmniejszenie liczby odcieni szarości. Jest to niewątpliwy defekt.

Algorytm Floyda-Steinberga

Defektu tego można uniknąć przez zastosowanie algorytmu zaproponowanego przez Floyda i Steinberga (R. Floyd, L. Steinberg, *An Adaptive Algorithm for Spatial Gray Scale*, *SID 1975 Digest*, *Int. Symp. Dig. Tech.*, 1975, str. 36–37). Ich pomysł polega na rozłożeniu błędu, jaki się popełnia przy redukcji liczby odcieni szarości, na piksele sąsiednie.

Załóżmy, że na wejściu mamy dane w postaci tablicy $[P_{i,j}]$, $0 \leqslant P_{i,j} \leqslant p$, wielkości $m \times n$. Innymi słowy $[P_{i,j}]$ przedstawia obraz o $p+1$ poziomach szarości. Należy przekształcić tablicę $[P_{i,j}]$ na tablicę $[R_{i,j}]$ takich samych rozmiarów, przy czym $0 \leqslant R_{i,j} \leqslant r < p$, czyli że wynikowa liczba poziomów szarości ma być mniejsza.

W omawianym algorytmie tabela $[P_{i,j}]$ jest przeglądana w porządku leksykograficznym i liczba poziomów szarości jest redukowana w oczywisty sposób:

$$R_{i,j} \leftarrow \mathit{round}\left(\tfrac{r}{p} P_{i,j}\right)$$

Popełnia się przy tym błąd

$$\varepsilon_{i,j} = \tfrac{r}{p} P_{i,j} - \mathit{round}\left(\tfrac{r}{p} P_{i,j}\right)$$

Kluczowym punktem algorytmu jest rozłożenie tego błędu na sąsiednie piksele w kierunkach wschód, południowy wschód i południe z wagami odpowiednio 3/8, 1/4, 3/8, i uwzględnienie tego błędu w dalszej obróbce.

Algorytm Floyda-Steinberga można zapisać następująco:

dane: *tablica liczb całkowitych* $P_{m\times n}$
liczby całkowite $p > 0$ *oraz* $r > 0$
$0 \leqslant P_{i,j} \leqslant p,\ r < p$

inicjalizacja:
  for $i := 1$ **to** m **do**
    for $j := 1$ **to** n **do** $\varepsilon_{i,j} := 0$;

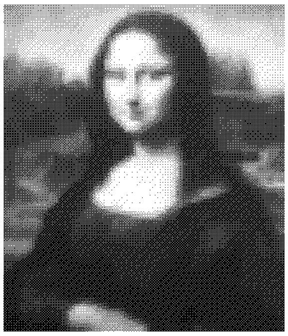
główny algorytm:
  for $i := 1$ **to** m **do**
    for $j := 1$ **to** n **do**
    begin
      $R_{i,j} := \mathit{round}\left(\tfrac{r}{p} P_{i,j} + \varepsilon_{i,j}\right)$;
      if $R_{i,j} < 0$ **then** $R_{i,j} := 0$;

$$
\begin{aligned}
&\textbf{if } R_{i,j} > r \textbf{ then } R_{i,j} := r;\\
&\varepsilon_{i,j} := \frac{r}{p} P_{i,j} - R_{i,j};\\
&\varepsilon_{i,j+1} := \varepsilon_{i,j+1} + \frac{3}{8}\,\varepsilon_{i,j};\\
&\varepsilon_{i+1,j+1} := \varepsilon_{i+1,j+1} + \frac{1}{4}\,\varepsilon_{i,j};\\
&\varepsilon_{i+1,j} := \varepsilon_{i+1,j} + \frac{3}{8}\,\varepsilon_{i,j};
\end{aligned}
$$

end;

wynik: *tablica* $R_{m\times n}$, $0 \leqslant R_{i,j} \leqslant r$;

Tym razem poprawa jakości obrazu (ilustracja 5) jest istotna. Dzięki sprytnemu algorytmowi Floyda-Steinberga bogactwo odcieni szarości zostało zachowane. Natomiast strukturę rastra można wciąż dostrzec i jest ona dość dokuczliwa ze względu na własności ludzkiego wzroku, wyczulonego na linie poziome i pionowe. Z tego właśnie powodu prawie wszystkie zdjęcia w gazetach mają raster układający się wzdłuż przekątnych.

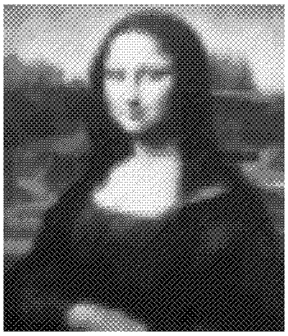


Ilustracja 5

Zbliżony efekt można uzyskać umieszczając jak na szachownicy lewo- i prawoskośne piksele 4×4, to znaczy korzystając na przemian z tablic $\mathbf{B}_r^{(4)}$ oraz $\mathbf{B}_l^{(4)}$ zdefiniowanych następująco:

$$
\mathbf{B}_r^{(4)} = \begin{array}{|c|c|c|c|}\hline 10 & 6 & 9 & 5\\\hline 2 & 14 & 1 & 13\\\hline 8 & 4 & 11 & 7\\\hline 0 & 12 & 3 & 15\\\hline\end{array}
\qquad
\mathbf{B}_l^{(4)} = \begin{array}{|c|c|c|c|}\hline 5 & 9 & 6 & 10\\\hline 13 & 1 & 14 & 2\\\hline 7 & 11 & 4 & 8\\\hline 15 & 3 & 12 & 0\\\hline\end{array}
$$

Ponadto kłuje w oczy zbyt jasna plama na czole i dekolcie *Mony Lisy*. Tę usterkę daje się stosunkowo łatwo usunąć, wystarczy *par force* zmniejszyć zakres szarości z $0\ldots16$ na, powiedzmy, $1\ldots16$. Jest regułą, że kolor biały źle się prezentuje na tego typu obrazach, tym gorzej im mniejsza jest rozdzielczość urządzenia, i należy go w zasadzie unikać (o ile nie jest to efekt zamierzony). Ilustracja 6 przedstawia skutek uwzględnienia tych zmian.



Ilustracja 6

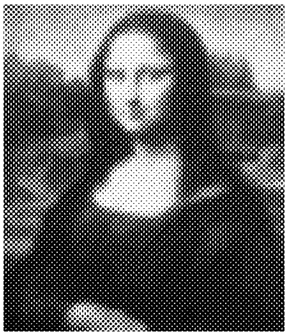
Rzeczą gustu jest ocena, która ilustracja jest ładniejsza — 5 czy 6. Raster wciąż się rzuca w oczy, obraz jest w obu przypadkach wyraźnie za ciemny. Niestety, przy tak małej rozdzielczości (na 1 cal przypada 75 pikseli wielkości 4×4) trudno się spodziewać czegoś istotnie lepszego.

Rozmycie przekątniowe

Okazuje się jednak, że widoczność rastra nie musi oznaczać złej jakości obrazu. Tak jest w przypadku metody zaproponowanej przez Garda (R. L. Gard, *Digital Picture Processing Techniques for the Publishing Industry*, *Computer Graphics and Image Processing*, **5**, (1976), str. 151–171). Jego podejście oparte jest o inną koncepcję rozmywania — zamiast zaczerniać jak najdalsze od siebie piksele, Gard najpierw zaczernia przekątne:

$$
\mathbf{G}_r^{(4)} = \begin{array}{|c|c|c|c|}\hline 14 & 10 & 5 & 1\\\hline 12 & 8 & 7 & 3\\\hline 2 & 6 & 9 & 13\\\hline 0 & 4 & 11 & 15\\\hline\end{array}
\qquad
\mathbf{G}_l^{(4)} = \begin{array}{|c|c|c|c|}\hline 1 & 5 & 10 & 14\\\hline 3 & 7 & 8 & 12\\\hline 13 & 9 & 6 & 2\\\hline 15 & 11 & 4 & 0\\\hline\end{array}
$$

Piksele lewo- i prawoskośne powinny być układane w szachownicę, tak jak to uczyniliśmy dopiero co z pikselami $\mathbf{B}_r^{(4)}$ oraz $\mathbf{B}_l^{(4)}$.



Ilustracja 7

Rezultat (ilustracja 7) jest zdecydowanie najlepszy z dotychczasowych. Raster — dzięki ułożeniu wzdłuż przekątnych — nie drażni wzroku, a przy większych obrazach lub większej rozdzielczości przestaje być zauważalny. Poprawiła się też skala odcieni szarości, obraz jest nieco jaśniejszy, dzięki czemu widoczne są szczegóły w jego dolnej części.

Krótki wypad w stronę generowania obrazów

Zgromadzone doświadczenia zachęcają do próby wykorzystania ich przy cieniowaniu uzyskanego drogą obliczeń obrazu obiektu trójwymymiarowego, np. kuli. Jasność $I(x,y,z)$ powierzchni kuli w punkcie (x,y,z) można z niezłym przybliżeniem określić na podstawie półempirycznego wzoru (T. Pavlidis, *Grafika i przetwarzanie obrazów*, *WNT*, 1987, str. 284 – 286):

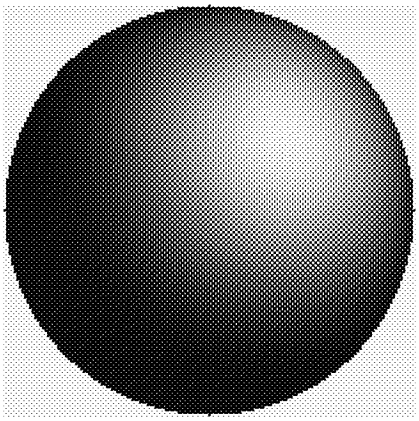
$$I(x,y,z) = c_d\,\mathbf{s}\diamond\mathbf{n} + c_r\left(\frac{\mathbf{s}+\mathbf{d}}{2}\diamond\mathbf{n}\right)^{\beta}$$

gdzie c_d, c_r — współczynniki odpowiednio rozproszenia i odbicia charakteryzujące daną powierzchnię (do obliczeń przyjęte zostały wartości $c_d = 1$, $c_r = 20$), $\mathbf{n}$ — wektor normalny do powierzchni w punkcie (x,y,z), $\mathbf{s}$ — wektor wyznaczający kierunek źródła światła, $\mathbf{d}$ — wektor wyznaczający kierunek obserwatora, β — współczynnik empiryczny (do obliczeń przyjęta została wartość $\beta = 4$), zaś operacja $\mathbf{u}\diamond\mathbf{v}$ jest zdefiniowana jako

$$\mathbf{u}\diamond\mathbf{v} = max\left(0, \frac{\mathbf{u}\cdot\mathbf{v}}{\mathbf{u}\,\mathbf{v}}\right)$$

Pierwszy składnik formuły opisuje rozproszenie światła. W przypadku idealnym światło rozproszone rozchodzi się we wszystkich kierunkach z takim samym natężeniem, zależnym jedynie od kąta padania światła na powierzchnię. Drugi człon pozwala na uwzględnienie połyskliwości powierzchni i stanowi półempiryczne uogólnienie prawa stwierdzającego, że kąt padania musi być równy kątowi odbicia (im większa wartość współczynnika β, tym lepiej człon ten opisuje klasyczne prawo odbicia).

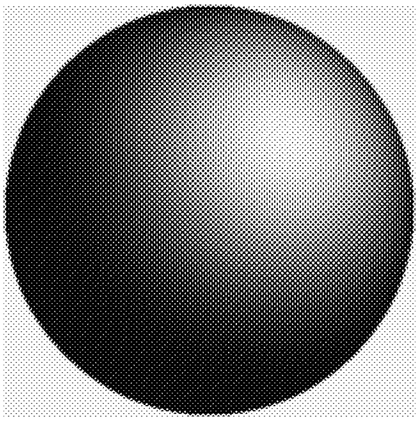
Za pomocą powyższej zależności zostało wyznaczone oświetlenie kuli o poziomach szarości 0 . . . 255, następnie liczba poziomów została zredukowana do 17 metodą Floyda-Steinberga, wreszcie została użyta metoda Garda — rezultat tego postępowania przedstawia ilustracja 8. Wynik jest trochę nie taki, jak by się chciało: kula wyszła nieco kanciasta.



Ilustracja 8

Efektu tego można się pozbyć przez staranniejsze określanie szarości pikseli granicznych. Na przykład wystarczy dla takich pikseli wyznaczyć stosunek pola tej części piksela, która znajduje się wewnątrz obszaru (tu: koła) do pola części znajdującej się poza obszarem i wyznaczyć efektywne oświetlenie piksela granicznego jako odpowiednią średnią ważoną.

Ilustracja 9 przedstawia obraz tej samej kuli z brzegiem wygładzonym taką właśnie metodą (w literaturze anglojęzycznej zabieg ten znany jest jako *anti-aliasing*).



Ilustracja 9

Uwagi końcowe

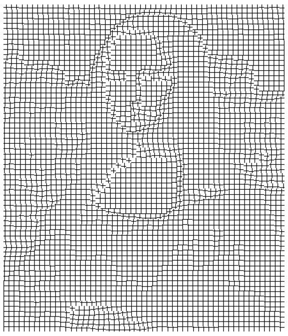
Omówione techniki odtwarzania obrazów z odcieniami szarości na urządzeniach czarno-białych w zasadzie wystarczają w większości praktycznych zastosowań. Czasem jednak pewne modyfikacje są niezbędne. Na przykład Knuth w cytowanej pracy stosuje algorytm Floyda-Steinberga z innymi wagami, a błąd przenosi także w kierunku południowozachodnim. Ponadto uwzględnia osobliwości drukarek laserowych, polegające na tym, że tak piksele Bayera, jak i piksele Garda zaczernione w połowie dają optycznie poziom czerni większy niż 50%. Uwzględnienie tego efektu wymaga wyznaczania wartości $R_{i,j}$ nie za pomocą funkcji *round*, jak w opisanej tu wersji algorytmu, ale za pomocą nieco bardziej zawiłej procedury. Tym, którzy chcieliby poznać zagadnienie bardziej szczegółowo, oprócz literatury cytowanej w tekście można polecić jeszcze dwie przeglądowe pozycje literaturowe:

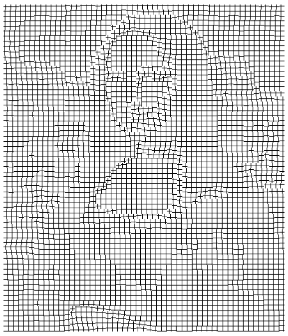
J. F. Jarvis, C. N. Judice, W. N. Ninke, *A Survey of Techniques for the Display of Continuous Tone Pictures on Bilevel Displays*, *Computer Graphics and Image Processing*, **5**, 1976, str. 13–40.

D. F. Rogers, *Procedural Elements for Computer Graphics*, *McGraw-Hill*, 1985.

Na zakończenie nie całkiem poważne spostrzeżenie. Ilustracja 2, jak już wspomniałem, może być uznana za obraz przetworzony specjalną techniką do specjalnych celów. Tropem tym można pójść dalej i — zamiast skupiać się na jak najdoskonalszym cieniowaniu — użyć zupełnie abstrakcyjnych schematów rozmywania. Oto przykład, co można uzyskać w ten sposób:

Ilustracja 10





i ostatnia.

Nietrudno zgadnąć, jaki schemat „rozmywania” został tu zastosowany. Za podsunięcie innych ciekawych efektów, dających się uzyskać tą metodą, będę niezmiernie wdzięczny.

Zarząd GUST-u

W wyniku wyborów przeprowadzonych na walnym zebraniu GUST-u (2 maja Bachotek) w skład zarządu GUST-u na drugą kadencję weszli:

1. Tomasz Przechlewski (Gdańsk) – prezes,
2. Jerzy Ludwichowski (Toruń) – wiceprezes,
3. Marek Ryćko (Warszawa) – wiceprezes,
4. Bogusław Lichoński (Gdańsk) – skarbnik,
5. Jolanta Szelatyńska (Toruń) – sekretarz,
6. Maria Budzik (Poznań),
7. Włodzimierz Bzyl (Gdańsk),
8. Bogusław Jackowski (Gdańsk),
9. Ewa Koisar (Warszawa),
10. Mariusz Olko (Warszawa),
11. Stanisław Wawrykiewicz (Gdańsk),

W 1995 r. odbyły się dwa spotkania zarządu w nowym składzie (2 maja w Bachotku oraz 16 września w Toruniu). Podjęto na nich następujące ważniejsze decyzje:

1. Zmieniono siedzibę GUST-u. Nowy adres grupy ustalono na: Centrum Fizyki Teoretycznej PAN, Al. Lotników 32/46, 00-668 Warszawa.
2. Postanowiono dofinansować ze środków GUST (1800 zł) prace Bogusław Jackowskiego dt. udostępnienia na zasadach *public domain* fontów PostScript-owych.
3. Staszek Wawrykiewicz został koordynatorem grupy roboczej dt. projektu dystrybucji TEX-a.

GUST ma swoją domenę!

Połączone wysiłki Krzysia Leszczyńskiego i Jurka Ludwichowskiego zaowocowały utworzeniem domeny GUST-u. „Na dziś” możemy korzystać z dwóch adresów. `ftp.gust.org.pl` to serwis `ftp` łączący z komputerem Politechniki Gdańskiej, na którym znajduje się archiwum TEX-owe utrzymywane przez Staszka Wawrykiewicza (tzw. ŚTAN), przy współpracy Włodzimierza J. Martina. Jest to najbardziej kompletne archiwum dt. oprogramowania TEX-owego do składania po polsku. Natomiast `www.gust.org.pl` to adres strony domowej GUST-u na komputerze Uniwersytetu Mikołaja Kopernika w Toruniu. Strona ta zawiera aktualne informacje o działalności grupy, konferencjach itp. W przygotowaniu uruchomienie innych adresów, jak np. `biuletyn@gust.org.pl` czy `sekretariat@gust.org.pl`. (TP)