

УДК 681.3.06

**О.Н. Романюк, доцент, канд. техн. наук**

*Вінницький національний технічний університет*

*м. Вінниця, Україна*

*E-mail: ran@svitonline.com*

## **ПІДХІД ДО АПАРАТНОЇ РЕАЛІЗАЦІЇ ЗАФАРБОВУВАННЯ ТРИВИМІРНИХ ГРАФІЧНИХ ОБ'ЄКТІВ**

*Пропонується апаратна реалізація зафарбовування тривимірних об'єктів методом Фонга, який дозволяє суттєво підвищити реалістичність колірною відтворення тривимірних поверхонь, в порівнянні з методом Гуро, апаратно реалізований в сучасних відеокартах.*

При синтезі й рендерингу графічних сцен необхідно вирішувати двоєдину задачу — забезпечення високої реалістичності графічних об'єктів і досягнення прийнятної для конкретної задачі часу формування графічних сцен. Постійне підвищення продуктивності засобів комп'ютерної графіки обумовлює більш жорсткі вимоги до реалістичності синтезованих зображень. На сучасному етапі розвитку комп'ютерної графіки розробка нових високопродуктивних методів і засобів формування реалістичних зображень є пріоритетною задачею, оскільки традиційні підходи не завжди забезпечують необхідну продуктивність і реалізм.

Як показує практика, робота з реалістичними растровими зображеннями тривимірних об'єктів у реальному часі можлива тільки за наявності спеціалізованих апаратних засобів високої продуктивності. Враховуючи, що зафарбовування є одним із найбільш трудомістких етапів формування реалістичних тривимірних зображень, то його апаратна реалізація дозволить суттєво підвищити продуктивність генерації зображень.

Мета статті — розробка підходу до апаратної реалізації зафарбовування за умови використання формули (1).

У більшості сучасних відеокартах реалізовано зафарбовування за методом Гуро [1, 2], згідно з яким розраховуються інтенсивності кольору для вершин трикутників полігональної мережі об'єкта, які потім у процесі rasterization лінійно інтерполюються вздовж ребер і рядків сканування.

До найбільш суттєвих недоліків методу Гуро можна віднести наступні:

1. Метод найбільш придатний тільки для формування матових поверхонь та таких, які погано відбивають світло, тому що кодова інтерполяція інтенсивностей кольору не враховує локальної кривизни поверхні, оскільки інтенсивність кольору має нелінійну залежність від координат пікселів, а в методі Гуро використовується виключно лінійне інтерполювання.

2. Поява артефактів при формуванні динамічних зображень.

При формуванні динамічних зображень об'єкти трансформуються, що обумовлює зміну структури та положення вузлів триангуляційної мережі. Це призводить до різкої зміни інтенсивності кольору зображення від кадру до кадру, оскільки в методі Гуро не враховується локальна кривизна поверхні.

3. Поява ефекту смуг Маха [2], оскільки забезпечується тільки неперервність значень інтенсивностей кольору вздовж границь полігонів.

На даному етапі розвитку комп'ютерної графіки більш реалістичні зображення можна отримати за методом Фонга [1, 2], в якому замість значень інтенсивності кольору інтерполюються вектори нормалей, які потім використовуються у функції тонування [1] для обчислення інтенсивності кольору кожного елемента зображення. Метод характеризується по відношенню до методу Гуро значно більшими обчислювальними витратами, однак при цьому досягається краща локальна апроксимація кривизни поверхні й, як наслідок, отримують більш реалістичні зображення.

Обчислювальна складність методу Фонга обмежує його широке використання в системах комп'ютерної графіки, орієнтованих на формування зображень у реальному масштабі часу. З іншого боку використання методу Гуро обмежене низькою якістю формування зображень.

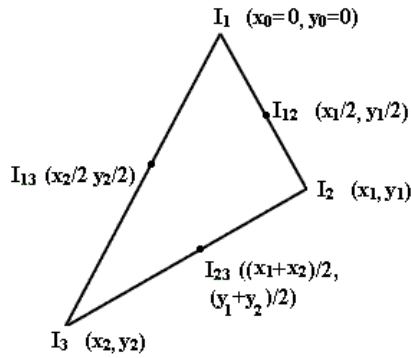
У зв'язку із цим актуальними є питання розробки нових підходів до зафарбовування, які б забезпечили прийнятний компроміс між продуктивністю формування тривимірних зображень і їх реалістичністю. Один із можливих підходів до вирішення проблеми полягає у формуванні поверхні другого порядку [3], точки якої визначають колір піксела в заданому трикутнику.

Інтенсивність кольору на поверхні другого порядку можна задати таким поліномом

$$I(x, y) = Ax^2 + By^2 + Cxy + Dx + Ey + F, \quad (1)$$

де  $x, y$  — точки поверхні,  $I(x, y)$  — інтенсивність кольору.

З рівняння видно, що необхідно визначити шість невідомих  $A, B, C, D, E, F$ . Це передбачає формування й розв'язок систем із шести рівнянь. Оскільки для трикутника задаються інтенсивності кольору тільки в трьох його вершинах, то найпростіше до визначити інтенсивності кольору в середніх точках на ребрах трикутника (зображено на малюнку 1), що дасть можливість скласти систему із шести рівнянь.



Малюнок 1 — Вихідний трикутник

Такий підхід до визначення невідомих коефіцієнтів є досить громіздким й вимагає великих обчислювальних витрат. Суттєво спростити задачу можливо за рахунок визначення поверхні інтенсивностей кольору для ортонормованого трикутника [3] і трансформації отриманих результатів на вихідний трикутник. У [3] розглянуто питання обчислення коефіцієнтів рівняння (1) з використанням трьох додаткових точок на ребрах трикутника, у яких інтенсивності кольору приймають екстремальні значення. Такий підхід серед перелічених забезпечує максимальну точність визначення інтенсивностей кольору.

Обчислення за формулою (1) достатньо трудомісткі, оскільки передбачають виконання операцій множення. Розглянемо реалізацію формули (1) з використанням методу кінцевих різниць з метою подальшої апаратної реалізації зафарбовування.

Знайдемо

$$\begin{aligned} I(x+1, y) &= A(x+1)^2 + By^2 + C(x+1)y + D(x+1) + Ey + F = \\ &= (Ax^2 + By^2 + Cxy) + Dx + Ey + G + F + 2Ax + A + Cy + D = \\ &= I(x, y) + 2Ax + A + C + D \end{aligned}$$

Запишемо отримане рівняння у вигляді

$$I(x+1, y) = I(x, y) + P_x,$$

де  $P_x = -2Ax + A + Cy + D$ .

Знайдемо значення наступного приросту інтенсивності кольору

$$P_{x+1} = -2A(x+1) + A + Cy + D = P_x + 2A$$

Аналогічно можна показати, що

$$I(x, y+1) = I(x, y) + 2By + B + Cx + E = I(x, y) + P_y$$

$$I(x, y-1) = I(x, y) - 2Ax + F + Cy - D = I(x, y) + P_{-z}$$

де  $P_{-x} = -2Ax + A - Cy - D$ ,  $P_y = 2By + B + Cx + E$ .

Легко показати, що

$$P_{-x+1} = P_y + 2B$$

$$P_{-x-1} = P_{-x} + 2A$$

При виконанні діагонального крокового переміщення типу  $(x+1, y+1)$

$$\begin{aligned} I(x+1, y+1) &= I(x, y) + (C+2B)y + B + (-2A+C)x + A + D + B + C + E = \\ &= I(x, y) + P_x + P_y - C \end{aligned}$$

У випадку діагонального крокового переміщення типу  $(x-1, y+1)$  значення інтенсивності кольору в потоковій точці буде дорівнювати

$$\begin{aligned} I(x-1, y+1) &= I(x, y) + (-C+2B)y + B + (-2A+C)x + A - D + B - C + E = \\ &= I(x, y) + P_{-x} + P_y - C \end{aligned}$$

Очевидно, що в точці  $(x=0, y=0)$

$$P_x = A + D, P_y = B + E, P_{-x} = A - D.$$

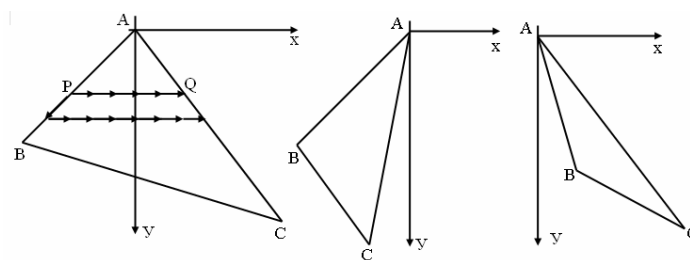
З наведених формул видно, що  $P_x + P_{-x} = 2A$ . Звідси  $P_x = 2A - P_{-x}$ .

Аналіз отриманих формул показує, що при зафарбовуванні базовою мікрооперацією буде нагромаджувальне додавання за умови, що невідомі коефіцієнти у формулі (1) розраховані на верхньому (програму) рівні.

Растрування внутрішньої ділянки поверхні, обмеженої трикутником, будемо виконувати з верхньої вершини, зліва направо (зображено на малюнку 2). При такому підході вздовж горизонтальних рядків

растеризації будуть виконуватися виключно крокові прирости типу  $+1x$ , а крокові прирости типу  $-1y$  будуть взагалі відсутні. Це суттєво спрощує рендеринг ділянки, обмеженої трикутником.

Якщо зафарбовування здійснюється від лівого ребра трикутника, то кількість різних варіантів розміщення трикутника відносної локальної системи координат є достатньо обмеженим (зображено на малюнку 1).



Малюнок 2 — Типи трикутників

Розглянемо питання апаратної реалізації зафарбовування з використанням отриманих виразів.

Пристрій для зафарбовування включає адресний блок та блок визначення інтенсивності кольору. Адресний блок виконує растеризацію ділянки, обмеженої трикутником, включаючи його ребра. Він формує адреси пікселів в екранній системі координат. На виході блоку визначення інтенсивностей кольору розраховуються значення інтенсивностей кольору.

Розглянемо питання апаратної реалізації адресного блоку.

Структурну схему адресного блоку наведено на мал. 3, яка включає регістр RG1 приросту  $\Delta x$ , регістр RG2 приросту  $\Delta y$ , регістр RG3 ознак вектора, лінійні інтерполятори ЛІ1, ЛІ2 [16], блок координатних приростів БКП, адресні регістри RG4, RG5, регістр RG<sub>p</sub>, координатні лічильники СТ<sub>x</sub>, СТ<sub>y</sub>. Рядковий інтерполятор включає рядковий лічильник СТ<sub>p</sub> та блок координатних приростів БКП.

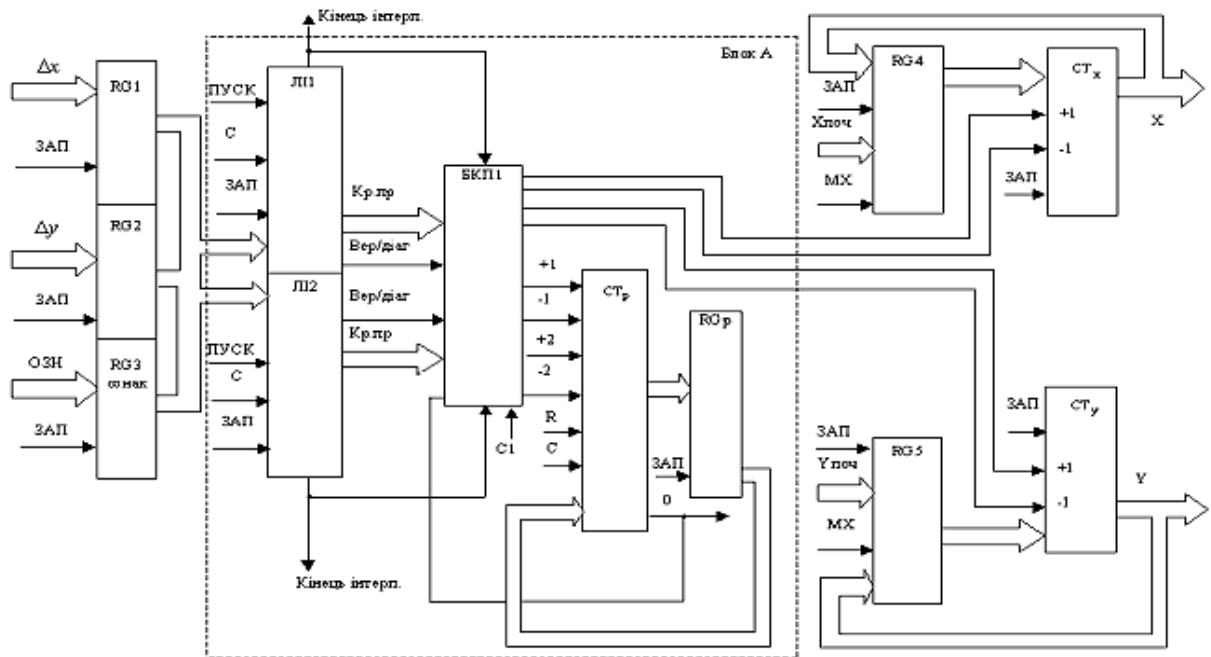
У циклі підготування в лінійний інтерполятор ЛІ1 через регістри RG1, RG2, RG3 завантажуються прирости  $\Delta x$ ,  $\Delta y$  ведучого ребра (ребра, від якого здійснюється растеризація) та його ознаки (знак  $\Delta x$ , знак  $\Delta y$ , ознака мажоритарності приростів  $\Delta x$ ,  $\Delta y$ ), а в лінійний інтерполятор ЛІ2 — прирости  $\Delta x$ ,  $\Delta y$  та ознаки ребра, яке має з ведучим ребром спільну верхню точку.

На інформаційні входи  $x_{\text{поч}}$ ,  $y_{\text{поч}}$  регістрів RG4, RG5 подаються відповідно адреси  $x$ ,  $y$  верхньої вершини трикутника, які заносяться в лічильники СТ<sub>x</sub>, СТ<sub>y</sub>. Лічильник СТ<sub>p</sub> онулюється.

Після цього в регістри RG1, RG2, RG3 заносяться відповідно прирости координат  $x$ ,  $y$  та ознаки ребра, яке має з ведучим ребром спільну нижню точку.

Вказані вище дії відносяться до циклу підготування.

Растеризація ділянки, обмеженої трикутником, починається з верхньої вершини трикутника. За сигналом “Пуск” починають роботу лінійні інтерполятори ЛІ1 та ЛІ2. Вказані блоки працюють незалежно один від одного й формують крокову траєкторію провідного та відповідного йому непровідного ребра (потокowego ребра, розміщеного в даний час зліва від провідного ребра). При формуванні блоками ЛІ1 та ЛІ2 діагонального (вертикального) крокового приросту робота відповідного інтерполятора блокується. До зупинки обох інтерполяторів у рядковому лічильнику формується операнд, який визначає відстань між точками провідного й непровідного ребра в рядку растеризації. Для цього блок крокових приростів рядкового інтерполятора за значеннями крокових приростів, які формуються на виходах лінійних інтерполяторів ЛІ1 та ЛІ2, формує сигнали  $+1$ ,  $-1$ ,  $+2$ ,  $-2$ . Після блокування інтерполяторів ЛІ1 та ЛІ2 рядковий інтерполятор формує імпульсну послідовність крокових приростів рядка растеризації. З кожним тактом вміст рядкового лічильника зменшується на одиницю. При досягненні лічильником нульового стану формування горизонтального рядка в площині трикутника завершується і передача крокових приростів із рядкового інтерполятора в блок крокових приростів блокується. Після того, коли лічильник СТ<sub>p</sub> досягне нульового стану, в нього заноситься вміст регістра RG<sub>p</sub>.



Малюнок 3 — Структурна схема адресного блоку

Для переходу до наступного рядка rasterизації в координатні лічильники  $СТ_x$ ,  $СТ_y$  заносять координати  $x$ ,  $y$  точки провідного ребра, з якої здійснювалось зафарбовування рядка rasterизації.

Указані дії виконуються до моменту завершення лінійним інтерполятором ЛП1 формування провідного ребра або ЛП2 — непровідного ребра. Вказане визначається за сигналами “Кінець інтерполяції”, які формуються на виходах відповідних інтерполяторів. Після цього з регістрів  $RG_1$ ,  $RG_2$ ,  $RG_3$  в інтерполятор, який завершив роботу, заносять прирости та ознаки ребра, яке не використовувалося при rasterизації. Після завершення формування вказаного ребра процес визначення адрес точок трикутника завершено.

Пристрій для визначення інтенсивностей кольору (зображено на малюнку 4) включає регістри  $RG_1$  —  $RG_7$ , блоки вентилів  $BV_1$  —  $BV_{10}$ , мультиплексори  $MX_1$  —  $MX_3$ , нагромаджувальні суматори  $PSm_1$  —  $PSM_4$ .

Елементи  $RG_1$ ,  $RG_2$ ,  $BV_1$ ,  $BV_2$  і нагромаджувальний суматор  $PSm_1$  уведено для розрахунку приросту інтенсивності  $P_x = -2Ax + A - Cy - D$  за умови, що приріст координат лівого ребра трикутника від’ємний відносно локальної системи координат. У подальшому необхідно враховувати, що потокове значення  $x$  від’ємне.

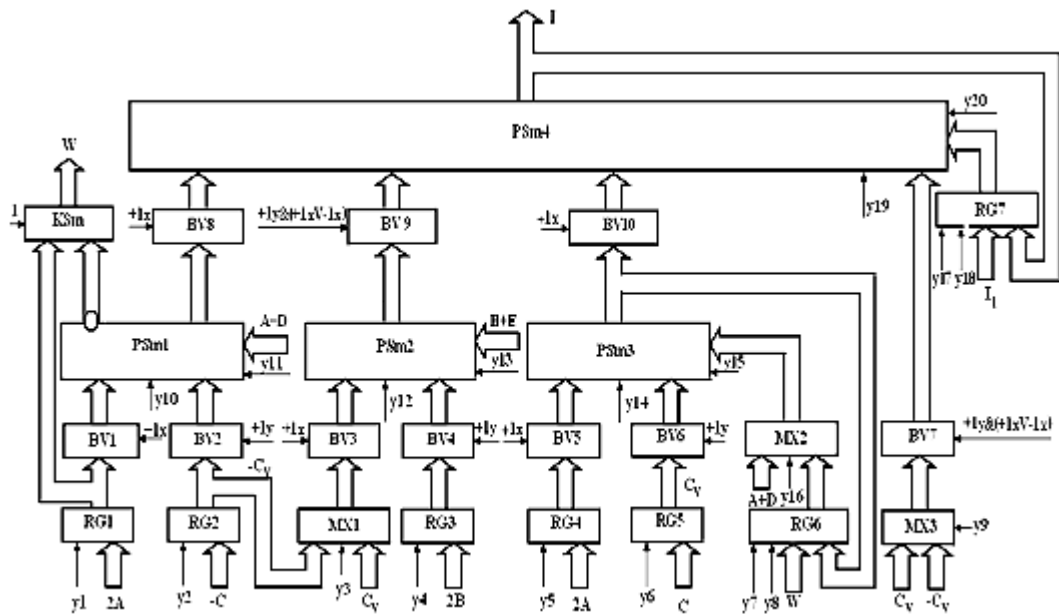
У початковий момент часу в регістр нагромаджувального суматору  $PSm_1$  заноситься початкове значення приросту  $P_x$ , яке дорівнює  $A - D$ , а в регістри  $RG_1$  і  $RG_2$  відповідно значення  $2A_i - C$ . У подальшому до вмісту нагромаджувального суматору додається значення:

- $2A$  при виконанні горизонтального кроку  $-1x$ ;
- $-C$  при виконанні вертикального кроку  $+1y$ ;
- $2A - C$  при виконанні діагонального кроку  $-1x, +1y$ .

У першому випадку активізується блок вентилів  $BV_1$ , у другому —  $BV_1$ , а в третьому —  $BV_1$  і  $BV_2$ .

Елементи  $MX_1$ ,  $RG_3$ ,  $BV_3$ ,  $BV_4$  і нагромаджувальний суматор  $PSm_2$  уведено для розрахунку приросту інтенсивності  $P_x = 2By + B + Cx + E$ . Для цього в регістр нагромаджувального суматору  $PSm_2$  записується значення  $B + E$ , яке приймає  $P_y$  при  $x = 0$  і  $y = 0$ . У подальшому при виконанні вертикального кроку  $+1y$  до вмісту нагромаджувального суматору додається значення  $2B$ , яке зберігається в регістрі  $RG_3$ . При виконанні горизонтальних кроків  $-1x, +1x$  до вмісту нагромаджувального суматору додаються відповідно значення  $-C, C$ , які комутуються на вхід блоку  $BV_3$  комутатором  $MX_1$ .

Дещо складнішим є блок для формування приростів  $P_x = 2Ax + A + Cy + D$ , оскільки він найбільш функціонально завантажений, так як формує прирости інтенсивностей уздовж горизонтальних рядків rasterизації і ребер з  $\Delta x \geq 0$ .



Малюнок 4 — Структурна схема блоку визначення інтенсивностей кольору

Формування рядка rasterизації трикутника здійснюється після формування вертикального або діагонального кроку потокового лівого ребра трикутника. Якщо  $\Delta x_{AB} \geq 0$ , то після виконання зазначених кроків на виході нагромаджувального суматора PSm1 формується приріст  $P_{-x}$ . Для рядка rasterизації трикутника використовуються тільки прирости  $P_x$ , тому для знаходження  $P_x = 2A - P_{-x}$  уведено комбінаційний суматор KSm, який працює в режимі віднімання. Отримане на його виході значення W запам'ятовується в регістрі RG6, а потім у нагромаджувальному суматорі PSm3. У подальшому з кожним кроком  $+1x$  уздовж рядка rasterизації до вмісту суматора PSm3 додається значення  $2A$ .

Після відпрацювання рядка rasterизації відновлюють приріст інтенсивності початкової точки потокового рядка rasterизації, для чого вміст регістру RG6 через мультиплексор MX2 заносять у регістр нагромаджувального суматора PSm3.

У випадку коли приріст  $\Delta x_{AB}$  лівого потокового ребра трикутника додатний, то приріст  $P_x$  формується згідно виразу  $2Ax + A + Cy + D$ . Для цього попередньо нагромаджувальний суматор PSm3 приймає значення  $A+D$ , а в подальшому до його вмісту додають значення  $2A$ ,  $C$ ,  $2A+C$  відповідно при формуванні горизонтального, вертикального або діагонального кроків.

На виході нагромаджувального суматора PSm4 формуються інтенсивності кольору точок ділянки, обмеженої трикутником. У початковий момент часу в RG6 заноситься значення інтенсивності кольору  $I_1$ , яке під дією сигналу  $y_{20}$  записується в нагромаджувальний суматор PSm4. У подальшому до вмісту нагромаджувального суматора PSm4 додається один чи два із приростів  $P_x, P_y, P_z$  залежно від типу координатного приросту. При формуванні діагонального кроку типів  $(+1x, +1y)$ ,  $(+1x, -1y)$  до вмісту нагромаджувального суматора PSm4 додаються відповідно операнди  $C$  і  $-C$ , які передаються на вхід нагромаджувального суматора через мультиплексор MX3 і блок вентилів BV7.

Після формування інтенсивності кольору останньої точки рядка rasterизації трикутника нагромаджувальний суматор PSm4 приймає значення операнду з регістру RG7, у якому зберігають інтенсивність кольору першої точки потокового рядка rasterизації.

При переході від потокового лівого ребра трикутника до наступного випадок, коли приріст координат ребра трикутника вздовж осі абсцис — від'ємний, виключається. Можливий тільки випадок, коли приріст координат уздовж осі абсцис наступного лівого ребра трикутника додатний. Це виключає необхідність перезавантаження регістрів RG1 — RG5.

Застосування методу кінцевих різниць дозволило виключити з обчислювального процесу довготривалі операції та звести обчислення тільки до використання мікрооперацій нагромаджувального додавання. Запропоновано структури адресного блоку та блоку для розрахунку інтенсивностей кольору.

Отримані результати можуть бути використані для акселерації зафарбовування в сучасних відеокартах, що складатиме перспективу подальших досліджень у цьому напрямку.

#### Бібліографічний список

1. Херн Д. Компьютерная графика и стандарт OpenGL / Д. Херн, М. Бейкер — М.: Вильямс, 2005. — 1168 с.
2. Цисарж В.В. Математические методы компьютерной графики / В. Цисарж, Р. Марусик — К.: Факт, 2004. — 464 с.
3. Romanyuk O. A Method for Accelerated Computation of Color Intensities for Shading of Three-Dimensional Graphics Objects /Romanyuk O., Hast A. // Współczesne problemy informatyki. Algorytmy i modelowanie. Wydawnictwo Wyższej Szkoły Menedżerskiej. Legnica, 2006 — p. 213 — 226.

*Поступила в редакцию 28.11.2007 г.*