

УДК 004.932.2: 519.68

## ОДНОБАЙТОВИЙ ПАЛІТРОВИЙ ФОРМАТ ВІДЕОФАЙЛУ

*Кумыш В.Ю.*

*Одеська національна академія зв'язку ім. О.С. Попова,  
65029, Україна, м. Одеса, вул. Ковальська, 1.  
kumish@mail.ru*

## ОДНОБАЙТОВЫЙ ПАЛИТРОВЫЙ ФОРМАТ ВИДЕОФАЙЛУ

*Кумыш В.Ю.*

*Одесская национальная академия связи им. А.С. Попова,  
65029, Украина, г. Одесса, ул. Кузнечная, 1.  
kumish@mail.ru*

## PALETTED VIDEO FILE FORMAT WITH SINGLE-BYTE CODE LENGTH

*Kumysh V.U.*

*O.S. Popov Odessa national academy of telecommunications,  
1 Kovalska St., Odessa 65029, Ukraine.  
kumish@mail.ru*

**Анотація.** В статті удосконалено медіаконтейнер AVI 1.0, в результаті чого в даному форматі забезпечується прийнятна якість візуального сприйняття зображення з  $PSNR_{\text{palette}} = 41,27$  дБ, оскільки для зберігання кожного відеокадра використовується своя адаптивна палітра кольорів. Також запропонованим форматом підтримується повна сумісність зі стандартними декодерами AVI 1.0.

**Ключові слова:** медіаконтейнер, палітровий формат, палітрове кодування, AVI, FOURCC код

**Аннотация.** В статье усовершенствован медиаконтейнер AVI 1.0, в результате чего в данном формате обеспечивается приемлемое качество визуального восприятия изображения с  $PSNR_{\text{palette}} = 41,27$  дБ, поскольку для хранения каждого видеокadra используется своя адаптивная палитра цветов. Также предложенным форматом поддерживается полная совместимость со стандартными декодерами AVI 1.0.

**Ключевые слова:** медиаконтейнер, палитровый формат, палитровое кодирование, AVI, FOURCC код

**Abstract.** Media container AVI 1.0 has been improved in this paper. The obtained format provides an acceptable perceptual quality with  $PSNR_{\text{palette}} = 41,27$  dB since it uses for each video frame its own adaptive color palette. Also, the proposed format supports full compatibility with standard AVI 1.0 decoders.

**Key words:** media container, paletted format, paletted encoding, AVI, FOURCC

### Вступ

Палітрові методи кодування відеопослідовностей, в яких відеокадри кодуються окремо й незалежно один від одного та послідовно записуються у відеопотік або відеофайл, забезпечують низьку обчислювальну складність та високу швидкість процесів кодування/декодування, а також простоту реалізації. Такі методи кодування можуть ефективно застосовуватись в прикладних телевізійних системах (системах відеоспостереження, системах автоматизації медичних досліджень, системах неруйнівного контролю якості та ін.), де змістовний аналіз відеокадрів повинен виконуватись в режимі реального часу.

Палітрові методи кодування використовуються з різними форматами відеофайлів. На-

приклад, Microsoft Run Length Encoding (MRLE) [1] та Microsoft Video 1 (MV1) [2] найчастіше використовуються разом з форматом Audio Video Interleave (AVI), а методи: CDXL [4], FLIC [5] та ін. мають власні формати відеофайлів. Основним недоліком існуючих палітрових методів кодування й форматів відеофайлів є істотні спотворення кадрів з причин невідповідності кольорів зображення кольорам палітр, що використовуються у цих форматах. Формальна оцінка якості візуального сприйняття зображень у форматі AVI, при кодуванні відеокадрів методом MRLE, показала, що пікове співвідношення сигнал/шум, в середньому, становить 26 дБ, що є неприйнятним. Тому проблема удосконалення палітрових форматів для збереження цифрових відеопослідовностей, що забезпечували б прийнятну якість візуального сприйняття, є актуальною.

### Аналіз існуючих форматів відеофайлів

Відеодані зберігаються в спеціалізованих файлах – медіаконтейнерах, що можуть одночасно містити різні потоки (відео, аудіо та текстові). Специфікації медіаконтейнерів визначають спосіб збереження даних в межах файлу й відповідний формат файлу, універсальні для різних методів кодування. В залежності від формату файлу може забезпечуватися підтримка: множинних потоків, прискореного переходу між кадрами та розширеного контенту (користувальницької інформації та мета даних). До спеціальних можливостей медіаконтейнерів можна віднести підтримку: кадрів з двонаправленим передбаченням, змінної швидкості передавання аудіопотоку й змінної частоти кадрів.

Одним з найефективніших мережних медіаконтейнерів є Ogg Media (OGM) [6]. Даний формат за забезпечує корекцію помилок, надійне транспортування й контроль цілісності файлу. До недоліків цього формату можна віднести складність структури інформаційної частини файлу, надмірність службових даних при формуванні заголовків елементів інформаційної частини та додаткові зворотні перетворення елементів інформаційної частини при синтезі відеокадрів.

Об'єм службових даних й кількість зміщень по файлу при одночасному відтворенні декількох потоків мінімізовані у медіаконтейнері Audio Video Interleave (AVI) [3]. Медіаконтейнер AVI спирається на спеціалізовану структуру даних chunk (ск), визначену стандартом EA IFF 85 (Interchange File Format, IFF). Для ідентифікації chunk'ів використовуються сигнатури Four Character Code (FOURCC): наприклад, формат AVI 1.0 передбачає два складених різновиди chunk'ів RIFF й LIST.

Загалом медіаконтейнер AVI – це складений chunk RIFF, що має ідентифікатор 'AVI' і містить в собі: заголовок – chunk LIST з ідентифікатором 'hdrl' та інформаційну частину – chunk LIST з ідентифікатором 'movi'. Заголовок 'hdrl' включає головний заголовок файлу 'avih' з глобальною інформацією про файл. Також структура 'hdrl' передбачає для кожного потоку даних окремий список заголовків з кодом 'strl', що включає заголовок потоку 'strh' та дані про формат потоку 'strf'. Структура 'strf' для відеопотоку без використання додаткових алгоритмів стиснення, тобто послідовності відеокадрів у растровому апаратно незалежному форматі BMP, повторює програмний заголовок BMP та додатково містить системну палітру.

Інформаційна частина файлу може містити потоки даних різних типів: растрові дані зображення, звук й текст. Блоки цих даних, що відтворюються одночасно зберігаються в спільний chunk LIST з FOURCC кодом 'rec', що забезпечує ефективну роботу з пристроями вводу/виводу, оскільки передбачає такий режим зчитування файлу в буфер, коли за кожну операцію повністю зчитується структура 'rec'. Це дозволяє без затримки відтворювати всі інформаційні блоки, що відносяться до одного кадра.

Для кодування відео у палітровому режимі медіаконтейнер AVI 1.0 використовує систе-

мну палітру кольорів й алгоритми кодування MRLE та MV1. При такому підході неможливо забезпечити прийнятну якість візуального сприйняття зображення, оскільки представлення кадрів в кольорах системної палітри неможливе без втрат та спотворень. Це зумовлено невідповідністю кольорів зображення кольорам системної палітри.

В [7] був запропонований адаптивний палітровий метод кодування цифрових відеопослідовностей з постійною довжиною коду, який для зберігання кожного відеокadra використовує свою палітру кольорів, отриману за допомогою алгоритму адаптивного квантування кольорів. Даний метод дозволяє скоротити об'єм файлу в 2,98 рази і зберегти прийнятну якість візуального сприйняття зображення. Виходячи зі сказаного вище, метою роботи є удосконалення медіа-контейнера AVI 1.0 на підґрунті адаптивного палітрового методу кодування [7] для забезпечення прийнятної якості візуального сприйняття.

### Однобайтовий палітровий формат відеофайлу

Палітровий метод кодування цифрових відеопослідовностей з постійною довжиною коду [7] полягає в наступному:

1. зменшення колірної надлишковості кадрів відеопослідовності шляхом адаптивного квантування зображень [8];
2. представлення кожного окремого кадра разом зі своєю незалежною адаптивною палітрою у однобайтовому палітровому графічному форматі з постійною довжиною коду [9]:
  - a. кодування трьох- або чотирьохбайтових значень кольору елементів квантованого зображення однобайтовими індексами адаптивної палітри;
  - b. запис адаптивної палітри у заголовок файлу;
  - c. запис закодованого зображення однобайтовими індексами адаптивної палітри в основну частину файлу;
3. зберігання кадрів відеопослідовності, представлених у однобайтовому палітровому форматі, у мультимедійному контейнері.

Схема кодера для даного методу кодування наведена на рис. 1.

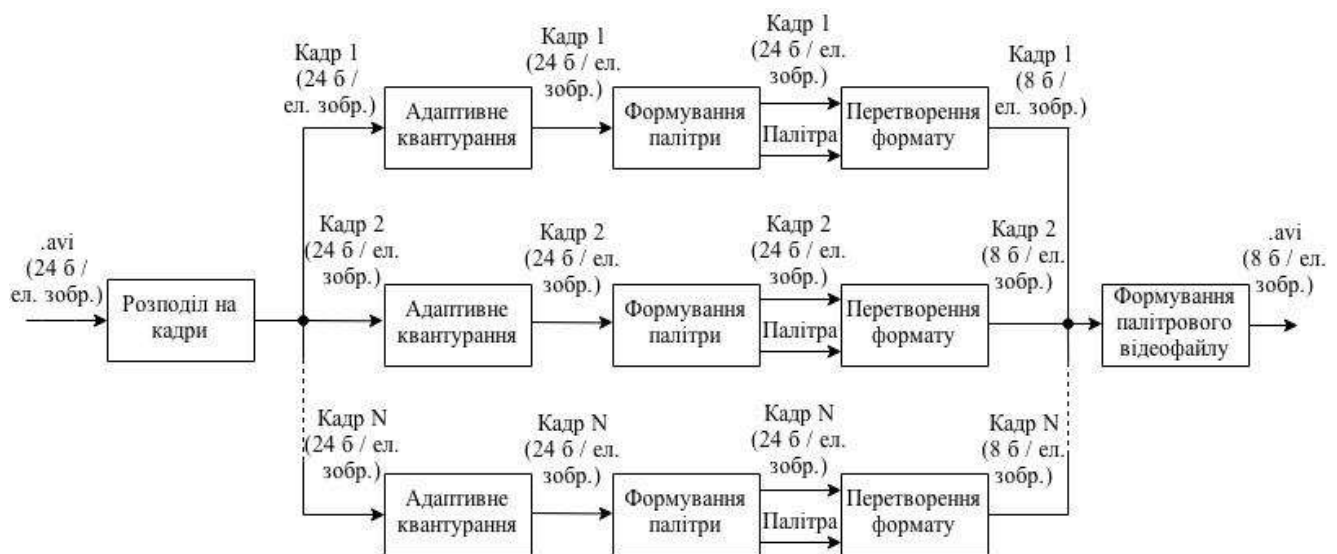


Рисунок 1 – Схема адаптивного палітрового кодера відеопослідовностей

Формування палітрового відеофайлу в удосконаленому форматі виконується наступним чином. Закодовані відеодані зберігаються в інформаційній частині файлу в окремому потоці. Для даного потоку у заголовку відеофайлу заповнюється список заголовків 'strl', який містить заголовок потоку 'strh' й структуру формату відеопотоку 'strf'. У структуру 'strf' заносяться програмний заголовок й адаптивна палітра першого кадра. Отже у структурі 'strf' міститься інформація про ширину й висоту кадра, глибину кольору, розподільну здатність та кількість кольорів у палітрі.

Кольори адаптивної палітри записуються у вигляді чотирибайтових значень: трьох байт RGB компонент та четвертого байту, що може використовуватися як альфа-канал. Далі заповнюється інформаційна частина відеофайлу. Відеопотік містить для кожного кадра його адаптивну палітру та растрові дані, закодовані відповідними номерами кольорів адаптивної палітри. Для відокремлення палітри від растрових даних відеопотік формується з chunk'ів двох різних типів. Вони позначаються FOURCC-кодами, де перші 2 байти позначають номер потоку у файлі, а інші 2 байти – тип даних: db (Device-independent bitmap, DIB) – апаратно незалежний формат зображення відеокадра, pc (Palette change) – палітра.

Під час декодування відеопослідовності однобайтові коди растрових даних кадра замінюються на відповідні значення кольору в адаптивній палітрі.

Даний спосіб формування відеопослідовності передбачає наступний порядок заповнення заголовків відеофайлу. На першому етапі заповнюється головний заголовок файлу. Для цього визначаються ширина  $w$  і висота  $h$  кадра та кількість кадрів в потоці  $N$ . Далі записується максимальна швидкість потоку, що для частоти 25 кадрів/с дорівнює  $v_{\max} = 25 \cdot V_{\text{buf}}$  (Б/с), де  $V_{\text{buf}}$  – об'єм буферу. В поле *Об'єм буферу*  $V_{\text{buf}}$  записується загальний об'єм даних палітри максимальної довжини  $V_{\text{pc}_{\max}}$  та растрових даних одного кадра  $V_{\text{db}}$ :  $V_{\text{buf}} = V_{\text{pc}_{\max}} + V_{\text{db}}$  (Б). Даного об'єму буферу достатньо, щоб програмному забезпеченню декодера не потрібно було додатково виділяти місце в оперативній пам'яті, оскільки це зменшить його продуктивність.

За головним заголовком файлу слідує заголовок потоку. У поле *Тип потоку* вноситься FOURCC код 'vids', що визначає тип даних потоку, як відеодані. У поле *Код декодера* записується FOURCC код апаратно незалежного формату зображення відеокадра без кодування 'DIB' (0x20424944). У поле *Прапори потоку* вноситься код 0x0001000c, який позначає, що кадри потоку зберігаються у палітровому вигляді, і для відтворення кожного кадра використовується його окрема адаптивна палітра кольорів.

На наступному етапі необхідно задати формат потоку. У поле *Довжина коду* записується значення 8 біт. Поле *Тип стиснення* містить FOURCC код, що вказує алгоритм стиснення, що використовується у кодері (якщо відеофайл формується без стиснення дане поле приймає значення 0). Далі необхідно вказати розмір стиснутого зображення в байтах, що при відсутності ентропійного стиснення містить, безпосередньо, об'єм растрових даних  $V_{\text{db}} = w \cdot h$  (Б). У поле *Кількість кольорів у палітрі* записується розмір палітри першого відеокадра  $K_1$ . Далі записується адаптивна палітра першого кадра й завершувач JUNK, необхідний для вирівнювання блоків даних за розміром сектору на жорсткому магнітному диску. Завершувач JUNK складає 512 байт й заповнюється нулями. Таким чином, розмір заголовку становить  $V_h = 4096$  байт.

В кінці записується інформаційна частина файлу. Отриманий формат будемо називати однобайтовий палітровий формат відеофайлу. Основні поля заголовків та елементи запропонованого формату наведені в табл. 1.

Таблиця 1 – Основні поля заголовків та елементи однобайтового палітрового формату відео

Зміст поля	Зсув, Б	Довжина поля, Б
Заголовок RIFF файлу		
Розмір структури RIFF, Б	4	4
Загальний заголовок файлу AVI		
Тривалість кадра, мкс	32	4
Максимальна швидкість потоку, Б/с	36	4
Загальна кількість кадрів	48	4
Об'єм буферу, Б	60	4
Ширина логічного екрану в елементах зображення	64	4
Висота логічного екрану в елементах зображення	68	4
Заголовок потоку		
Тип потоку	108	4
Код деко деру	112	4
Прапори потоку	116	4
Частота кадрів, кадр/с	132	4
Довжина потоку	140	4
Об'єм буферу	144	4
Дані щодо формату потоку		
Ширина зображення в елементах зображення	176	4
Висота зображення в елементах зображення	180	4
Довжина коду, б	186	2
Тип стиснення	188	4
Розмір стиснутого зображення, Б	192	4
Кількість кольорів у палітрі	204	4
Палітра першого кадра	216	1024
Інформаційна частина файлу		
Ідентифікатор LIST	4084	4
розмір структури LIST, Б	4088	4
ідентифікатор 'movi'	4092	4
Ідентифікатор LIST 1-го відеокадра	$V_h = 4096$	4
розмір структури LIST, Б	$V_h + 4$	4
ідентифікатор 'res'	$V_h + 8$	4
ідентифікатор '00pc'	$V_h + 12$	4
розмір структури '00pc', Б	$V_h + 16$	4
палітра	$V_h + 20$	$K_1 \cdot 4$
ідентифікатор '00db'	$V_h + 20 + K_1 \cdot 4$	4
розмір структури '00db', Б	$V_h + 24 + K_1 \cdot 4$	4
растрові дані	$V_h + 28 + K_1 \cdot 4$	$w \cdot h$
Наступні $N - 1$ кадрів, починаючи з ідентифікатору 'res'	$V_h + V_{\text{cinf}_i}$	$V_{\text{cinf}} - V_{\text{cinf}_i}$

Слід зазначити, що в запропонованому форматі забезпечується повна сумісність зі стандартними декодерами формату AVI, тобто при зберіганні відеопотоку в даному форматі з роз-

ширенням файлу .avi, його перегляд може здійснюватися існуючим програмним забезпеченням без змін в декодері.

Для реалізації запропонованого формату було розроблене програмне забезпечення для представлення відеопослідовності у однобайтовому палітровому форматі.

### Дослідження запропонованого палітрового формату відеофайлу

Проаналізуємо розмір відеофайлу та ступень стиснення в однобайтовому палітровому форматі відеофайлу (табл. 2).

Таблиця 2 – Результати зберігання відеопослідовностей в однобайтовому палітровому форматі

Ім'я файлу	Кількість кадрів	Розмір вхідного відеофайлу, Б	Розмір відео файлу в удосконаленому форматі, Б	Коефіцієнт стиснення
testament1	350	435 468 800	145 524 308	2,99
testament2	400	497 677 824	166 312 908	2,99
testament3	350	435 468 800	145 524 308	2,99
testament4	325	404 364 288	135 130 008	2,99
testament5	475	590 991 872	197 495 808	2,99
Girl	295	67 748 864	22 892 596	2,96
Street	314	72 126 976	24 371 784	2,96

За результатами аналізу можна заключити, що коефіцієнт стиснення у запропонованому форматі, в середньому, складає 2,98 рази.

Порівняємо показники якості при збереженні відеопослідовності у оригінальному медіа-контейнері AVI 1.0, що використовує алгоритм кодування MRLE, та в удосконаленому форматі (табл. 3). Для об'єктивної оцінки якості кадрів будемо використовувати середню квадратичну помилку (Mean Squared Error, MSE) і середнє, по компонентам RGB, пікове відношення сигнал/шум (Peak Signal-to-noise Ratio, PSNR):

$$MSE = \frac{1}{3} \sum_p \left[ \frac{1}{MN} \sum_x \sum_y (c_p(x, y) - c'_p(x, y))^2 \right], \quad (5)$$

$$PSNR = 10 \cdot \lg \frac{255^2}{MSE}, \quad (6)$$

де  $c_p(x, y)$ ,  $c'_p(x, y)$  – інтенсивність колірної компоненти елемента зображення з координатами  $(x, y)$  у вхідному та закодованому зображеннях, відповідно,  $p \in \{R, G, B\}$ ,  $c_p \in [0, 255]$ ;  $M, N$  – розміри зображення.

З отриманих результатів випливає, що значення пікового співвідношення сигнал/шум PSNR між закодованими й оригінальними відеопослідовностями для алгоритму MRLE, в середньому, по всім тестовим відео [10], склало  $PSNR_{MRLE} = 25,97$  дБ. Для запропонованого формату максимальне значення склало  $PSNR_{palette\ max} = 46,55$  дБ, а мінімальне  $PSNR_{palette\ min} = 36,91$  дБ. В середньому,  $PSNR_{palette} = 41,27$  дБ, що на 15,3 дБ вище ніж у алгоритму MRLE.

Таблиця 3 – Формальні показники якості відео у медіаконтейнері AVI

Ім'я файлу	MSE		PSNR, дБ	
	MRLE	Адаптивний палітровий метод кодування	MRLE	Адаптивний палітровий метод кодування
testament1	194,97	2,52	25,23	44,27
testament2	172,32	3,59	25,77	42,59
testament3	137,94	7,90	26,75	39,19
testament4	134,71	5,41	26,84	40,81
testament5	174,83	1,46	25,71	46,55
Girl	184,66	9,84	25,62	38,60
Street	170,07	14,13	25,87	36,91

Вхідне та закодовані зображення наведені на рис. 2.



Рисунок 2 – Порівняння якості візуального сприйняття відео у медіаконтейнері AVI:

а – кадр вхідної відеопослідовності; б – кадр, закодований алгоритмом MRLE;  
в – кадр, закодований адаптивним палітровим методом

З рис. 2,б видно, що після кодування MRLE отримуємо спотворене постеризоване зображення. У запропонованому форматі, постеризація помітна тільки на зображенні стакану.

За допомогою емуляторів ЕОМ з різними технічними характеристиками була проведена оцінка часу: кодування, зчитування, декодування та синтезу декодованого зображення. Слід зазначити що, в процесі вимірювання кожна з перелічених операцій запускалась окремо, незалежно одна від одної. Для оцінки продуктивності використовувалась відеопослідовність з розміром кадра  $720 \times 576$  та довжиною палітри  $K = 256$ .

Як видно з табл. 4, сумарний час зчитування закодованого зображення, його декодування і відображення для терміналу з найслабшими технічними характеристиками складає 57 мс. Оскільки при синтезі 25 кадрів на секунду на зчитування, декодування і відображення повинно витрачатись не більше 40 мс, в розробленому застосуванні дані операції програмно об'єднані, що дозволяє їм виконуватись швидше, а сумарний час їх виконання складає 37 мс. Для того, щоб уникнути затримок після синтезу кожного кадра на зчитування та декодування наступного, в розробленому застосуванні передбачено буфер ОЗП. В даному буфері зберігається черга попередньо зчитаних, декодованих та підготовлених для синтезу кадрів на екрані. Розмір буферу розраховується динамічно й залежить від часу зчитування одного кадра відеопослідовності.

Таблиця 4 – Оцінка часових показників запропонованого формату для одноядерних процесорів

Об'єм ОЗП, Мб	512			1024		
	800	1000	1500	800	1000	1500
Частота процесора, МГц	800	1000	1500	800	1000	1500
Час зчитування, мс	6	6	6	7	6	6
Час декодування, мс	30	29	30	30	30	30
Час кодування, мс	34	32	32	31	31	30
Час синтезу, мс	21	21	21	18	18	18

### Висновки

В результаті удосконалення медіаконтейнера AVI 1.0 запропоновано однобайтовий палітровий формат відеофайлу. Даний формат забезпечує прийнятну якість візуального сприйняття зображення з  $PSNR_{palette} = 41,27$  дБ, оскільки використовує адаптивний метод кодування цифрових відеопослідовностей та зберігає кожний окремий відеокадр у кольорах своєї адаптивної палітри.

В запропонованому форматі забезпечується повна сумісність зі стандартними декодерами медіаконтейнера AVI 1.0. Цей формат має невисокі системні вимоги: частота головного процесора 800 МГц, об'єм оперативної пам'яті 512 МБ і може використовуватись у мобільних терміналах зв'язку.

### ЛІТЕРАТУРА:

1. RGB Pixel Formats [Електронний ресурс].– Режим доступу: <http://www.fourcc.org/rgb.php>.–Дата доступу: 14.06.2013.
2. Description of the Microsoft Video-1 Decoding Algorithm [Електронний ресурс].– Режим доступу: <http://multimedia.cx/video1.txt>.–Дата доступу: 14.06.2013.
3. AVI RIFF File Reference, [Електронний ресурс] – <http://msdn.microsoft.com/en-us/library/aa451196.aspx> – Дата доступу 18.12.2014.
4. Патент на винахід “Apparatus and method for transferring interleaved data objects in mass storage devices into separate destinations in memory” № 5293606 від 08.03.1994 автора С. Е. Sassenrath.
5. Kent J. The FLIC File format / J. Kent // Dr. Dobb's Journal, 1992.– v. 18.– № 3.
6. The Ogg encapsulation format version 0: IETF RFC 3533 / S. Pfeiffer.– Вид. офіц. – увед. 2003-05-01.– RFC Editor, 2003.– 10 с.
7. Кумиш, В. Ю. Палітровий метод ефективного кодування цифрових відеозображень [Текст] / В. Ю. Кумиш // Восточно-европейский журнал передовых технологий. – 2013. – №4/ 2 (64). – С. 19- 22.
8. Загребнюк В. И. Адаптивное цветное квантование изображений / В. И. Загребнюк, А. М. Яворский // К: Электроника и связь, 2008.– № 5. – С. 30.
9. Загребнюк, В. І. Однобайтові палітрові графічні формати для зберігання цифрових кольорових зображень [Текст] / В. І. Загребнюк, В. Ю. Кумиш // Восточно-европейский журнал передовых технологий. – 2009. – № 4/2. – С. 63–66.
10. Technische Universität München, Institute für Data Processing. TUM Multi Format Test Set [Електронний ресурс].– Режим доступу: <http://www.ldv.ei.tum.de/videolab>.–Дата доступу: 14.06.2013.

### REFERENCES

1. “RGB Pixel Formats.” RGB Pixel Formats. Web. 11 Feb. 2015. <<http://www.fourcc.org/rgb.php>>.



2. "Description of the Microsoft Video-1 Decoding Algorithm." Web. 1 Nov. 2014. <<http://multimedia.cx/video1.txt>>.
3. "AVI RIFF File Reference (Windows CE 5.0)." AVI RIFF File Reference (Windows CE 5.0). Web. 1 Nov. 2014. <<http://msdn.microsoft.com/en-us/library/aa451196.aspx>>.
4. Carl E., Sassenrath. "Apparatus and Method for Transferring Interleaved Data Objects in Mass Storage Devices into Separate Destinations in Memory US005293606A." United States Patent, 8 Mar. 1994. Web. 1 Nov. 2014. <<http://patentimages.storage.googleapis.com/pdfs/US5293606.pdf>>.
5. Kent, J. "The FLIC File Format." Dr. Dobbs's Journal 18.3 (1992). Web.
6. Pfeiffer, S. "The Ogg Encapsulation Format Version 0. RFC 3533." The Internet Engineering Task Force (IETF). IETF, 1 May 2003. Web. 1 Nov. 2014. <<https://www.ietf.org/rfc/rfc3533.txt>>.
7. Kumysh, V.U. "Palette Method of Efficient Digital Video Coding." Eastern-European Journal of Enterprise Technologies EEJET 4.2(64) (2013): 19-22. Print.
8. Zagrebnyuk, V.I., and A.M. Javorsky. "Adaptive Color Image Quantization." Electronics and Communications 5(46) (2008): 30-34. Print.
9. Zagrebnyuk, V.I., and V.U. Kumysh "Single-byte palette image file formats to store digital color images." Eastern-European Journal of Enterprise Technologies EEJET 4(2) (2009): 63-66. Print.
10. Diepold, Klaus, and Christian Keimel. "Lehrstuhl Für Datenverarbeitung: TUM Multi Format Test Set." Technische Universität München, Institute für Data Processing. 1 Jan. 2011. Web. 12 Nov. 2014. <<http://www.ldv.ei.tum.de/videolab>>.