

Вишневий С.В.Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**Журба А.В.**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

СИНТЕЗ ВІДЕОПОСЛІДОВНОСТЕЙ, ЩО МІСТЯТЬ ОДНОРІДНУ ТЕКСТУРУ ІЗ СЕПАРАБЕЛЬНОЮ АВТОКОРЕЛЯЦІЙНОЮ ФУНКЦІЄЮ

В статті розглянуто задачу синтезу відеопослідовностей, в яких всі кадри містять однорідну текстуру із однаковими статистичними характеристиками. Виконано стислий огляд сфер практичного застосування методів генерування текстурних зображень в контексті обробки відеоданих. Проведено дослідження алгоритму синтезу відеопослідовностей, який побудований на основі застосування авторегресійної моделі гаусівського випадкового поля із сепарабельною автокореляційною функцією, що враховує просторово-часову кореляцію відліків. В статті наведено розрахункові вирази, а також блок-схеми алгоритмів, які можуть бути використані для реалізації розглянутого підходу формування відеопослідовностей за допомогою відповідної мови програмування. Використання представленого алгоритму синтезу послідовності текстурних зображень потребує виконання попереднього визначення параметрів авторегресійної моделі шляхом аналізу навчальних даних, які містять однорідну текстуру, в ході якого мають бути оцінені коефіцієнти кореляції вздовж кожної із координат, а також виконано обрахунок середнього значення яскравості та середнього квадратичного відхилення значень пікселів текстури. Моделювання алгоритму реалізовано в програмному середовищі MATLAB. В статті представлено зразки синтезованих текстур, надано показники часових витрат створення відеофайлів для різних форматів кадру, приведено графіки автокореляційних функцій та гістограми отриманих зображень. З'ясовано, що кадри відеопослідовностей характеризуються анізотропною автокореляційною функцією, що в деяких випадках може бути розглянуто як недолік, для подолання якого необхідно ускладнення процедури синтезу текстури шляхом застосування моделі більш високого порядку. Результати роботи можуть бути використані в наукових дослідженнях та практичних задачах, в яких необхідно виконувати синтез відеопослідовностей, що містять текстуру.

Ключові слова: відеопослідовність, текстура, авторегресійна модель, гаусівське випадкове поле, алгоритм, цифрове зображення.

Постановка проблеми. Стрімкий розвиток електронної компонентної бази та значне зростання обчислювальної потужності мікропроцесорних пристроїв сприяло розвитку та широкому впровадженню апаратно-програмних комплексів та систем різного практично-прикладного призначення, що передбачають реєстрацію, передачу, обробку та збереження даних, які представлені у вигляді послідовності цифрових зображень, що застосовуються для вирішення цілого переліку різноманітних задач аналізу даних [1, 2]. Важливою відмінністю відеопослідовності від статичних одиночних зображень є можливість реєструвати дина-

міку зміни об'єкту інтересу. Відеопослідовності загалом характеризуються суттєвою надмірністю [3]. З метою зниження розміру відеоданих застосовуються алгоритми компресії [4–6]. Наприклад, в цифровому телебаченні широке поширення набув стандарт сімейства MPEG, використання якого дає можливість значно зменшити об'єм даних за допомогою використання алгоритму стиснення зображення із втратами [7, 8]. В деяких практичних задачах може виникати необхідність не використовувати стиснення даних задля забезпечення високої якості кадрів, однак при цьому відеодані характеризуються значно більшим розміром.

Серед властивостей об'єктів, які можуть бути представлені на кадрах відеопослідовності, текстура є однією із визначальних особливостей [9, 10]. Текстура може бути розглянута як псевдоперіодичний візерунок, який поширюється в межах деякої області на зображенні. Наявністю текстури можуть характеризуватися послідовності зображень, що отримуються в спеціалізованих медичних діагностичних комплексах, в системах моніторингу та спостереження за ділянками земної поверхні, текстуру також можуть мати об'єкти що містяться на кадрах телевізійних зображень тощо.

З-поміж широкого кола завдань, для вирішення яких використовуються методи цифрового оброблення зображень, своє застосування знаходить синтез текстур. До практичних задач, в яких виникає необхідність застосування алгоритмів синтезу текстур, можна віднести відновлення окремих пошкоджених ділянок в кадрі або видалення небажаних об'єктів [11–13]. Вирішення цього завдання може передбачати аналіз властивостей неушкодженої текстурної області, синтез штучної текстури, і заміщення пошкоджених або небажаних фрагментів в кадрах відеопослідовності згенерованими текстурними зразками. Алгоритми синтезу текстур можуть застосовуватися в задачах стиснення та передачі даних, зокрема, в каналах зв'язку з низькою пропускнуою здатністю замість передачі інформації про значення пікселів деякої текстурної ділянки кадру, можуть бути визначені параметри та характеристики текстури, які передаються на приймальну сторону по каналу зв'язку, а на приймальній стороні виконується синтез відповідних текстурних ділянок кадру [14–16]. Таким чином, відбувається фактичне заміщення реальної текстури синтезованим аналогом. Синтез текстурних відеопослідовностей може застосовуватися в задачах комп'ютерної графіки, в індустрії комп'ютерних ігор, в мультимедійних системах доповненої реальності [17]. Крім того, процедура синтезу текстур може бути задіяна в задачах дослідження алгоритмів та методів обробки цифрових зображень та відеопослідовностей. Застосування алгоритмів синтезу текстур дозволяє виконати генерування даних, які по своїм статистичним властивостям відповідають реальним даним. Отримані текстурні зображення можуть додатково спотворюватися відповідним видом завади із заданими параметрами, які узгоджуються із завадою, вплив якої передбачається в умовах функціонування деякої реальної технічної системи. Синтезовані «ідеальні» дані, а також дані, що спотворені завадою, можуть бути під-

дані обробці відповідним алгоритмом (фільтрації, сегментації тощо) з метою оцінки кількісних показників його роботи [18, 19]. Зміна параметрів моделі текстури дозволяє створити цілий набір різноманітних даних, які по своїм характеристикам співпадають із реальними даними. Це дає змогу проведення детального аналізу відповідного алгоритму обробки відеопослідовностей без необхідності використання великої колекції реальних даних, доступ до яких може вимагати додаткових витрат. Замість цього використовуються синтезовані дані. Таким чином, важливою складовою є вибір алгоритму синтезу текстур; визначення особливостей його практичної реалізації; оцінка часових витрат процесу синтезу текстурних відеопослідовностей; аналіз розміру отриманих відеофайлів тощо.

Аналіз останніх досліджень і публікацій.

Враховуючи велике різноманіття текстур, які можуть зустрічатися на цифрових зображеннях та відеопослідовностях, на поточний момент не існує єдиного підходу, який би задовольняв всі вимоги, що можуть висуватися до методів та алгоритмів синтезу статичних або динамічних текстур. Своє застосування на практиці знаходять алгоритми, які передбачають використання текстурного зразка, що представляє собою деякий блок, який відтворюється в межах області синтезу [20, 21]. Такий підхід може бути використаний для текстур, які містять псевдоперіодичний характер паттерну, однак в цьому випадку треба враховувати можливу необхідність застосування процедур усунення артефактів, які проявляються у вигляді швів між окремими блоками. Широке поширення набули моделі, які ґрунтуються на використанні марковських випадкових полів [22, 23]. В якості недоліків таких моделей можна віднести обчислювальну складність їхньої практичної реалізації. Останнім часом все більшу популярність здобувають алгоритми синтезу текстур, що ґрунтуються на використанні технологій машинного навчання та нейронних мереж, які знаходять своє застосування в різних сферах від комп'ютерної графіки до стиснення та відновлення зображень у відеопослідовностях та статичних цифрових знімках [24–27]. Дані алгоритми дозволяють отримувати текстури, які мають високий рівень реалістичності, однак при практичному використанні таких алгоритмів виникає необхідність проведення попереднього навчання нейронних мереж, що потребує значних обсягів навчальних даних, а також такі алгоритми можуть характеризуватися високими обчислювальними витратами. Для

деяких задач може виникати необхідність використання моделей синтезу текстурних зображень та відеопослідовностей, які характеризуються помірним рівнем складності своєї реалізації та не вимагають значних обчислювальних потужностей при їх використанні, а також не потребують проведення тривалої процедури навчання моделей. На основі використання авторегресійної моделі можуть будуватися алгоритми синтезу текстур для генерування двовимірних зображень [28, 29]. Авторегресійна модель може застосовуватися і для реалізації генерування відеопослідовностей. Параметри моделі можуть визначатися на основі попередньо оціненої автокореляційної функції, дисперсії та середнього значення яскравості початкової вибірки.

Постановка завдання. Метою статті є проведення аналізу алгоритму синтезу відеопослідовностей, що характеризуються наявністю однорідної текстури в межах сформованої множини кадрів, використовуючи для реалізації алгоритму авторегресійну модель генерування гаусівського випадкового поля, що характеризується сепарабельною автокореляційною функцією, та виконання оцінки результатів синтезу відеоданих.

Виклад основного матеріалу. В загальному вигляді відеопослідовність представимо у формі множини кадрів [7, 30]. Окремий кадр в свою чергу може бути розглянутий як двовимірний масив, де значення кожного елементу масиву буде відповідати рівню яскравості пікселя. Такий підхід буде адекватним опису послідовності, яка складається із зображень в градаціях сірого. При переході до кольорового зображення, використовуючи модель RGB, в загальному випадку кожен піксель характеризується значенням інтенсивності в червоному, зеленому та синьому каналах, таким чином це несуттєво ускладнює структуру представлення цифрового зображення. Будемо покладати, що окремий кадр є однорідним текстурним зображенням, причому статистичні характеристики та параметри текстури в межах відеопослідовності є незмінними.

Позначимо піксель відеопослідовності таким чином: $v(x,y,k)$, де v – піксель, значення якого відповідає рівню яскравості елементу зображення; x – координата, яка відповідає за номер стовпця, в якому розміщується піксель в кадрі ($x=1, \dots, N$); y – координата, яка відповідає за номер рядка, в якому розташований піксель в кадрі ($y=1, \dots, M$); k – координата, яка відповідає за номер кадру ($k=1, \dots, K$). З урахуванням прийнятих позначень відеопослідовність схематично представлено на рис. 1.

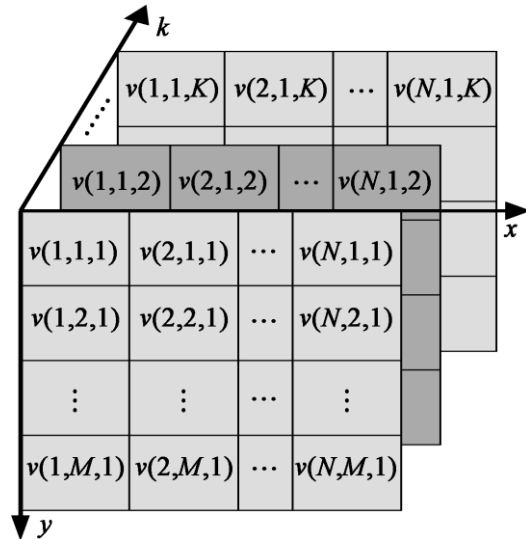


Рис. 1. Схематичне представлення відеопослідовності

В загальному випадку двовимірна авторегресійна модель передбачає проведення обчислень коефіцієнтів моделі та дисперсії шуму збудження, використовуючи для цього систему лінійних рівнянь розміри якої залежать від обраного порядку моделі (a,b) [31]. В роботі [32] приведено загальний вид авторегресійної моделі, яка враховує просторово-часову кореляцію відліків.

В даній роботі для реалізації алгоритму синтезу відеопослідовностей застосовується авторегресійна модель, яка конкретизована для випадку при якому для синтезу першого кадру ($k=1$) використовується модель порядку $(1,1)$, параметри якої мають зрозумілий фізичний смисл, а саме представляють собою коефіцієнти кореляції значень пікселів вздовж координати x та y . Такий підхід відповідає моделі із сепарабельною автокореляційною функцією виду $R(\Delta x, \Delta y) = \sigma_v^2 r_p^{|\Delta x|} r_c^{|\Delta y|}$, де r_p , r_c – коефіцієнти кореляції пікселів вздовж рядка і стовпця [33]. А при синтезі наступних кадрів ($k=2, \dots, K$) враховується також кореляція значень відповідних пікселів вздовж координати k . Отримані дані характеризуються сепарабельною автокореляційною функцією виду $R(\Delta x, \Delta y, \Delta k) = \sigma_v^2 r_p^{|\Delta x|} r_c^{|\Delta y|} r_k^{|\Delta k|}$, де r_k – коефіцієнт кореляції вздовж координати, що позначає номер кадру [34]. Розглянутий в роботі алгоритм дозволяє синтезувати кадри відеопослідовності, які характеризуються однаковістю своїх статистичних характеристик. В разі необхідності виконати симуляцію зміни властивостей сцени відеоряду (наприклад, моделюючи зміну сцени реєстрації, що отримується в деякій телевізійній системі), необхідно зада-

тися новими параметрами моделі і почати процес синтезу кадрів заново.

Загалом, процедура генерування відеопослідовності, передбачає виконання таких етапів: 1) задати коефіцієнти кореляції r_p , r_c , r_k . Визначити середнє квадратичне відхилення значень пікселів та середнє значення яскравості. Зазначені параметри можуть бути розраховані на основі аналізу деякої навчальної відеопослідовності; 2) задати розміри кадру, формат збереження зображення, кількість кадрів у відеопослідовності, та виконати синтез послідовності текстурних зображень; 3) на основі множини зображень сформулювати відеопослідовність та отримати відеофайл.

Для подальших викладок з метою компактного представлення записів, координату k , яка позначає номер кадру, представимо у вигляді нижнього індексу: $v_k(x,y)=v(x,y,k)$, $x=1,\dots,N$; $y=1,\dots,M$; $k=1,\dots,K$. Покладається що кадри відеопослідовності не є незалежними між собою. Перший кадр буде формуватися у вигляді одиночного зображення, при синтезі якого враховується кореляція вздовж координат x та y . При формуванні другого та наступних кадрів береться до уваги також коефіцієнт кореляції r_k .

Алгоритм № 1, який дозволяє синтезувати значення відліків першого кадру ($k=1$), передбачає виконання наступних кроків:

Крок 1: синтез першого пікселя в першому рядку зображення:

$$v_1(1,1) = \sigma_v \omega(1,1), \quad (1)$$

де σ_v – середнє квадратичне відхилення значень пікселів зображення; $\omega(1,1)$ – гаусівська некорельована випадкова величина із нульовим математичним сподіванням та дисперсією, що дорівнює 1.

Крок 2: синтез пікселів з 2-го по N -й, що розміщуються в першому рядку зображення ($x=2,\dots,N$):

$$v_1(x,1) = r_p v_1(x-1,1) + \sigma_v \sqrt{1-r_p^2} \omega(x,1). \quad (2)$$

Крок 3: синтез першого пікселя наступного рядка зображення ($y=2,\dots,M$):

$$v_1(1,y) = r_c v_1(1,y-1) + \sigma_v \sqrt{1-r_c^2} \omega(1,y). \quad (3)$$

Крок 4: синтез всіх наступних пікселів поточного рядка зображення ($x=2,\dots,N$):

$$v_1(x,y) = r_p v_1(x-1,y) + r_c v_1(x,y-1) - r_p r_c v_1(x-1,y-1) + \sigma_v \sqrt{1-r_p^2} \sqrt{1-r_c^2} \omega(x,y). \quad (4)$$

Блок-схема алгоритму приведена на рис. 2, де *randn* представляє собою генератор незалежних гаусовських випадкових величин із нульовим

математичним сподіванням та одиничною дисперсією (використовується в блок-схемі замість параметра $\omega(x,y)$ у виразах (1)–(4)).

Інший підхід реалізується за допомогою *Алгоритму № 2*, блок-схема якого приведена на рис. 3. Синтез значень пікселів зображення передбачає

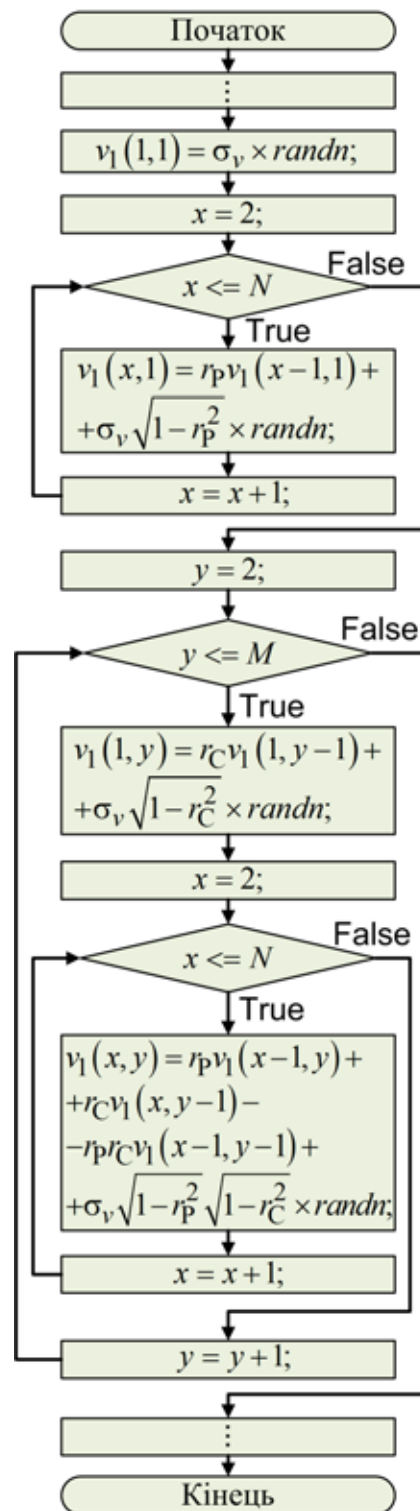


Рис. 2. Алгоритм №1

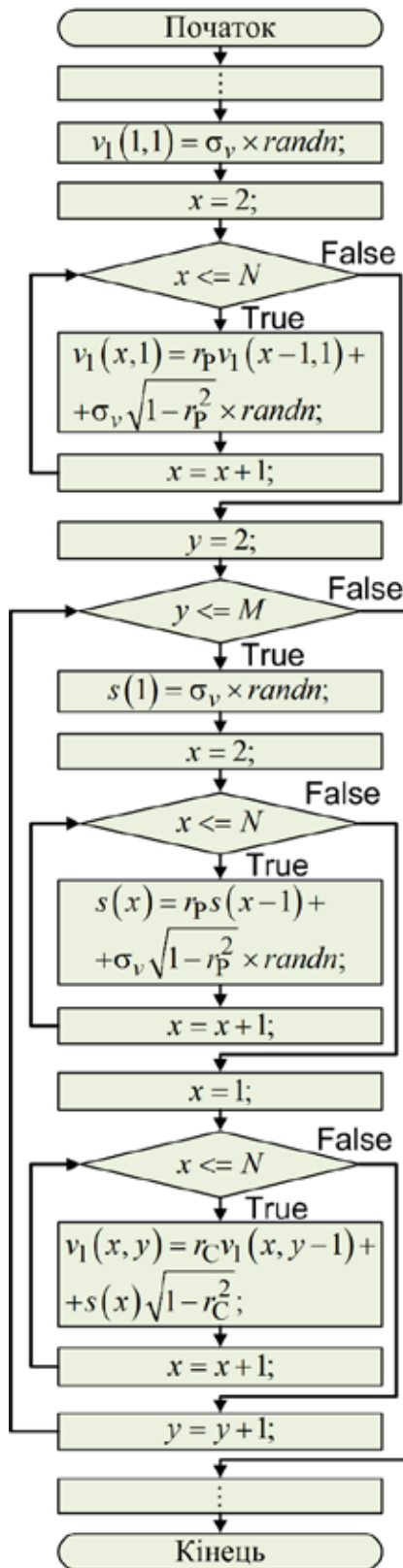


Рис. 3. Алгоритм № 2

залучення додаткового одновимірного масиву, який позначений як $s(x)$, $x = 1, \dots, N$. Одновимірний масив $s(x)$ виконує роль допоміжного рядка

зображення, значення елементів якого використовуються для синтезу пікселів, що розміщуються в рядках з 2-го по M -й в першому кадрі відеопослідовності. Значення елементів масиву $s(x)$ синтезуються кожного разу заново при генеруванні значень пікселів чергового рядка першого кадру, таким чином, елементи масиву $s(x)$ постійно оновлюються, причому процедура формування елементів масиву $s(x)$ відповідає синтезу пікселів першого рядка в першому кадрі відеопослідовності. Значення статистичних параметрів (коефіцієнт кореляції та середнє квадратичне відхилення), які використовуються у виразі обрахунку значень елементів масиву $s(x)$, повинні обиратися такими як і для формування значень пікселів першого рядка першого кадру відеопослідовності. Загалом, результати генерування значень пікселів зображення, які формуються за допомогою Алгоритму № 1 або Алгоритму № 2, при умові обрання однакових вхідних параметрів для кожного із алгоритмів, повинні володіти однаковими статистичними характеристиками. Нижче в статті проводиться дослідження швидкості роботи обох алгоритмів при їх застосуванні для синтезу множини однорідних текстурних зображень, на основі яких формується відеопослідовність.

Після синтезу першого зображення, що виступає першим кадром ($k=1$), процедура формування наступного кадру передбачає використання додаткового двовимірного масиву для зберігання додаткового зображення, яке синтезується на кожному кроці генерування кадрів з 2-го по K -й, застосовуючи для цього обраний раніше алгоритм – Алгоритм № 1 або Алгоритм № 2, використовуючи ті самі значення параметрів, які поклалися при генеруванні зображення першого кадру. Якщо позначити додаткове зображення як $q(x, y)$, $x=1, \dots, N$, $y=1, \dots, M$, тоді пікселі кадру $v_k(x, y)$, $k=2, \dots, K$ будуть формуватися таким чином:

$$v_k(x, y) = r_k v_{k-1}(x, y) + q(x, y) \sqrt{1 - r_k^2}. \quad (5)$$

Сама процедура синтезу кадрів відеопослідовності може бути представлена за допомогою блок-схеми алгоритму, яка приведена на рис. 4.

Дослідження алгоритмів проводилося в MATLAB. В ході синтезу множини зображень з метою їх використання для формування відеопослідовності не обов'язково застосовувати трьохвимірний масив $v_k(x, y)$, де відповідні виміри позначають номери стовпця та рядку зображення та номер кадру згідно із прийнятими позначеннями на рис. 1. Натомість при моделюванні було використано 4 двовимірних масиви: перший

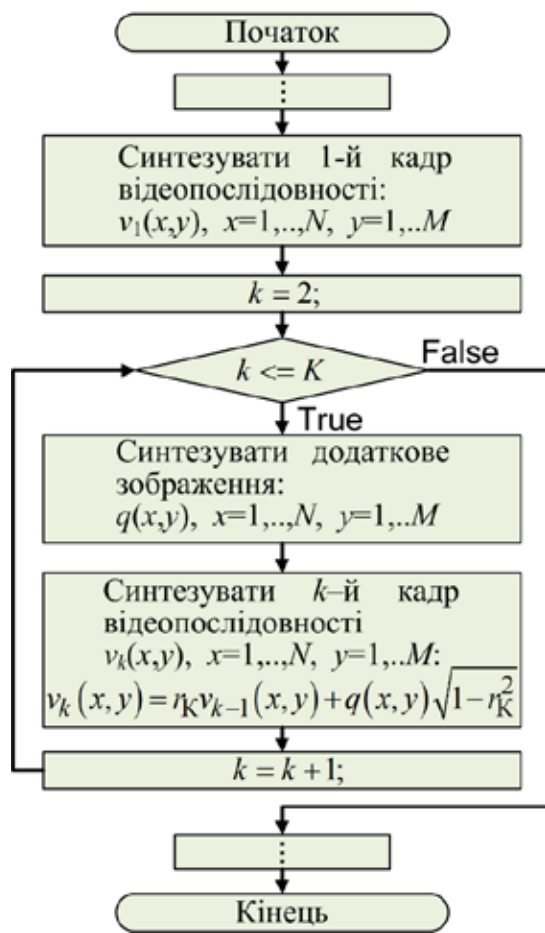


Рис. 4. Алгоритм синтезу кадрів відеопослідовності

масив використовувався для зберігання пікселів зображення, що покладалося поточним кадром (масив A); другий масив використовувався для зберігання зображення, що покладалося попереднім кадром послідовності (масив B); третій масив застосовувався для зберігання додаткового зображення (масив C), яке використовувалося для формування кадрів з 2-го по K -й. При синтезі значень елементів масивів A , B , C було обрано тип даних з плаваючою комою, а сформовані зображення A , B , C загалом представляли собою центровані гаусівські випадкові поля із заданим середньоквадратичним відхиленням σ_v . З метою формування зображень, що характеризуються відмінним від 0 середнім значенням яскравості пікселів, використовувалася константа $mean$, яка відігравала роль середньої яскравості зображення. Значення параметрів σ_v та $mean$ обирались таким чином, щоб після додавання $mean$ до значень елементів масиву A отримані значення потрапляли в діапазон $[0;1]$. Після цього за допомогою функції $im2uint8()$ значення пікселів перетворювалися до цілих значень із діапазону $[0;255]$ і зберігалися в двовимірний

масив D , який по суті представляв собою 8-бітне зображення, що записувалося на жорсткий диск ПК. При цьому значення елементів масиву A не змінювалися, а описана процедура реалізовувалася в MATLAB таким чином:

$$D = im2uint8(A + mean); \quad (6)$$

Процес моделювання в MATLAB відбувався так:

1) для першого кадру ($k=1$) за допомогою обраного алгоритму (Алгоритм № 1 або Алгоритм № 2) синтезувалося перше зображення і зберігалось в масив A . За допомогою виразу (6) формувався масив D , який представляв собою 8-бітне зображення, що за допомогою функції $imwrite()$ зберігалось на жорсткий диск;

2) для кадрів $k=2, \dots, K$ синтез виконувався в циклі згідно з алгоритмом, який показано на рис. 4. На початку кожної ітерації за допомогою раніше обраного алгоритму (Алгоритм № 1 або Алгоритм № 2) виконувався синтез додаткового зображення C . Крім того на початку кожної ітерації значення елементів масиву A присвоювалися відповідним елементам масиву B (оновлюючи тим самим вміст масиву B , який призначений для зберігання попереднього кадру послідовності). За допомогою виразу аналогічного (5) виконувався розрахунок пікселів зображення поточного кадру (де $v_{k-1}(x,y)$ – це піксель попереднього кадру, тобто відповідний елемент масиву B ; $q(x,y)$ – це піксель додаткового зображення, тобто відповідний елемент масиву C ; $v_k(x,y)$ – піксель поточного кадру, тобто відповідний елемент масиву A). За допомогою виразу (6) формувався масив D , і отримане зображення записувалося на диск. Після цього відбувався перехід до наступної ітерації.

На рис. 5 показано результати моделювання для різних параметрів моделі, розміри текстурних зображень 100×140 пікселів. На рис. 5.а приведені (зліва направо) автокореляційна функція текстурного зображення, 1-й кадр, 25-й кадр та 100-й кадр відеопослідовності, а також гістограма 1-го кадру; при синтезі послідовності зображень покладалася такі значення параметрів моделі: $\sigma_v = 0,15$, $r_p = 0,98$, $r_c = 0,98$, $r_k = 0,99$. На рис. 5.б показані аналогічні результати для таких параметрів моделі: $\sigma_v = 0,12$, $r_p = 0,9$, $r_c = 0,9$, $r_k = 0,95$. На рис. 5.в приведені результати для випадку: $\sigma_v = 0,1$, $r_p = 0,7$, $r_c = 0,7$, $r_k = 0,9$. Середнє значення пікселів для всіх приведених варіантів покладалося рівним 0,5, що при переведенні значень пікселів за допомогою функції $im2uint8()$ до цілого типу даних, відповідає середньому значенню яскравості 128.

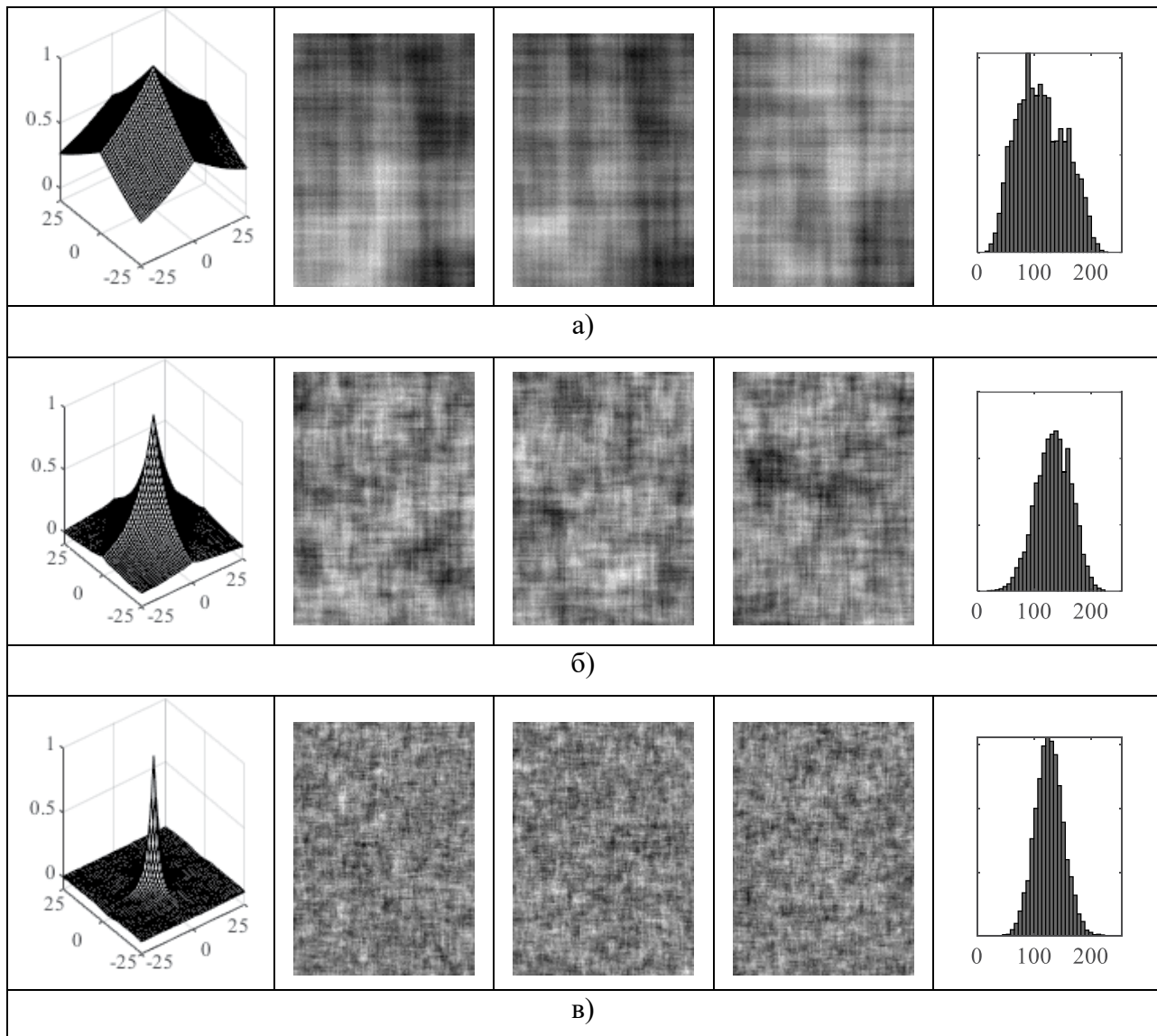


Рис. 5. Результати моделювання

В таблиці 1 приведені часові витрати синтезу тестових відеопослідовностей при використанні *Алгоритму № 1* та *Алгоритму № 2*, які застосовувалися для створення множини однорідних текстурних зображень, що в подальшому використовувалися для формування відеофайлів відповідного формату (розміри відеофайлів приведені в табл. 1). Покладалися такі параметри моделі: $\sigma_v = 0,12$, $mean = 0,5$, $r_p = 0,9$, $r_c = 0,9$, $r_k = 0,95$. Дослідження роботи алгоритмів проводилося в MATLAB на ПК із такими характеристиками: Intel i7-3770K 3.5 ГГц, 32 ГБ RAM, NVIDIA GeForce GTX 1660 Super, Win7. Розміри окремого зображення для кожного окремого експерименту обиралися: 720×576, 1024×768, 1920×1080. Для кожного формату зображення TIFF, JPEG, PNG, BMP формувалося 1500 зображень. Синтезована

послідовність зображень зберігалася на жорсткому диску. За допомогою функції *VideoWriter()* формувався відеофайл тривалістю 1 хвилину, кількість кадрів на секунду 25. При створенні відеофайлів обиралися такі параметри: 1) тип файлу 'MPEG-4', Quality=100, даний формат передбачає використання компресії, що дозволяє суттєво зменшити розмір відеоданих; 2) 'Grayscale AVI' для створення файлу відеопослідовності в градіях сірого, без компресії.

Загалом, результати моделювання демонструють незначні відмінності в швидкодії синтезу послідовності зображень за допомогою *Алгоритму № 1* та *Алгоритму № 2*. Відмінності в часі синтезу множини зображень при виборі різних графічних форматів може бути пояснена необхідністю проведення додаткових процедур комп-

Часові витрати синтезу відеопослідовностей

Розмір кадру: Стовпці × Рядки	Середня тривалість синтезу одного кадру (сек)		Тривалість синтезу всієї послідовності зображень (сек)		Сформований відеофайл. Розмір файлу / тривалість створення файлу (сек)	
	Алг.№ 1	Алг.№ 2	Алг.№ 1	Алг.№ 2	MPEG-4	Grayscale AVI
TIFF						
720×576	0,0351	0,0352	52,6345	52,7615	67,5 МБ	593 МБ
					18,1435	4,5941
1024×768	0,0731	0,0698	109,7112	104,7633	129 МБ	1,09 ГБ
					33,3483	5,6866
1920×1080	0,1838	0,1734	275,6510	260,1296	344 МБ	2,89 ГБ
					87,3245	14,2475
JPEG						
720×576	0,0405	0,0398	60,7552	59,6310	75,8 МБ	593 МБ
					19,7234	8,3854
1024×768	0,0821	0,0781	123,2212	117,1154	145 МБ	1,09 ГБ
					35,6519	14,0599
1920×1080	0,2113	0,2005	316,8998	300,7562	385 МБ	2,89 ГБ
					90,7549	35,4442
PNG						
720×576	0,0557	0,0539	83,5559	80,7756	67,5 МБ	593 МБ
					18,9263	9,6163
1024×768	0,1097	0,1048	164,6081	157,1832	129 МБ	1,09 ГБ
					34,5060	16,0793
1920×1080	0,2823	0,2727	423,5250	409,1239	344 МБ	2,89 ГБ
					86,1868	39,8819
BMP						
720×576	0,0337	0,0331	50,4858	49,6514	67,5 МБ	593 МБ
					19,0803	5,0061
1024×768	0,0704	0,0646	105,5438	96,9165	130 МБ	1,09 ГБ
					33,9088	5,7949
1920×1080	0,1735	0,1672	260,3223	250,7643	344 МБ	2,89 ГБ
					87,0921	13,1189

ресії даних. Формат BMP не передбачає проведення компресії, тому послідовність зображень в даному форматі синтезувалася найшвидше, однак при цьому розмір окремого зображення був найбільшим. Вибір відповідного графічного формату, розміру кадру відеопослідовності, кількості кадрів за секунду може бути продиктовано вимогами до процедури синтезу відеопослідовностей як при виконанні наукових досліджень, наприклад, в ході вивчення методів та алгоритмів обробки відеопослідовностей, так і для реалізації відповідних практичних задач.

Висновки. В роботі проведено дослідження алгоритму синтезу відеопослідовностей, який побудований на основі авторегресійної моделі для генерування гаусівського випадкового поля, що характеризується сепарабельною автокореляційною функцією. Приведено результати моделювання роботи алгоритму в програмному серед-

овищі MATLAB. Надано приклади синтезованих текстур, які отримані для відповідних вхідних параметрів моделі.

Враховуючи обраний порядок моделі, отримані зображення характеризуються анізотропною автокореляційною функцією, що проявляється візуально у вигляді певної сітчастої структури, яка стає більш помітною при наближенні значень коефіцієнтів кореляції r_p та r_c до 1. Таким чином, це обмежує можливості розглянутого алгоритму до синтезу переліку текстур, які можуть зустрічатися на реальних відеопослідовностях. Для подолання зазначеного недоліку може бути ускладнена процедура синтезу значень пікселів кадру шляхом переходу до авторегресійної моделі вищих порядків, однак при цьому також ускладнюється методика розрахунку параметрів моделі при аналізі навчальних даних та збільшуються часові витрати на синтез текстурних відеопослідовностей.

Перевагою представленого алгоритму є невисока складність обчислювальних процедур, які необхідно виконувати в ході синтезу зображень, а також можливість швидкого розрахунку (на основі аналізу тестових текстур) необхідних параметрів (середньоквадратичного відхи-

лення і середнього значення яскравості пікселів зображення, та коефіцієнтів кореляції відліків зображення вздовж кожної із координат), які використовуються в ході моделювання кадрів відеопослідовності, що забезпечує простоту практичної реалізації алгоритму.

Список літератури:

1. Szeliski R. Computer Vision: Algorithms and Applications. Springer, 2022. 925 p.
2. Кульчицька О. Ю., Січко Т.В. Цифрова обробка зображень та відео. *Прикладні інформаційні технології* : тези доп. всеукр.наук.-практ. конф. м. Вінниця, 29 квітня 2020 р. Вінниця, 2020. С. 110–111.
3. Tan L, Jiang J. Digital signal processing: Fundamentals and Applications. Academic Press Elsevier, 2013. 876 p.
4. Deshpande R.G., Ragha L.L. Performance Analysis of Various Video Compression Standards. *International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)* : proceedings. Jalgaon, 22–24 Dec. 2016. Jalgaon, 2016. P. 148–151.
5. Devi N., Singh V., T. Vinay. A Study of Different Video Compression Techniques and Latest Standards. *International Journal of Trend in Research and Development*. 2017. Vol. 4 (3). P. 312–313.
6. Pawlowski P., Piniarski, Dabrowski A. Selection and Tests of Lossless and Lossy Video Codecs for Advanced Driver-assistance Systems. *IEEE 2018 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)* : proceedings. Poznan, 19–21 Sept. 2018. Poznan, 2018. P. 344–349.
7. Lundstom L.-I. Understanding Digital Television. An Introduction to DVB Systems with Satellite, Cable, Broadband, and Terrestrial TV Distribution. Focal Press, 2006. 316 p.
8. Benoit H. Digital Television. Satellite, Cable, Terrestrial, IPTV, Mobile TV in the DVB Framework. Focal Press, 2008. 289 p.
9. Nixon M.S., Aguado A.S. Feature Extraction and Image Processing for Computer Vision. Elsevier Academic Press, 2019. 650 p.
10. Solomon C., Breckon T. Fundamentals of Digital Image Processing. A Practical Approach with Examples in MATLAB. Wiley-Blackwell, 2011. 352 p.
11. Ding D., Ram S., Rodriguez J.J. Image Inpainting Using Nonlocal Texture Matching and Nonlinear Filtering. *IEEE Transactions on Image Processing*. 2019. Vol. 28, No. 4. P. 1705–1719.
12. Lu S., Han C. An Improved Image Inpainting Method Based on Patch Matching and Texture Synthesis. *IEEE Chinese Control And Decision Conference (CCDC)* : proceedings. Nanchang, China, 03–05 Jun. 2019. Nanchang, 2019. P. 5933–5938.
13. Zhang L., Chang M. Image Inpainting for Object Removal Based on Adaptive Two-Round Search Strategy. *IEEE Access*. 2020. Vol. 8. P. 94357–94372.
14. Thakur U.S., Ray B. Image coding using parametric texture synthesis. *IEEE 18th International Workshop on Multimedia Signal Processing (MMSP)* : proceedings. Montreal, 21–23 Sept. 2016. Montreal, 2016.
15. Balle J., Stojanovic A., Ohm J.-R. Models for Static and Dynamic Texture Synthesis in Image and Video Compression. *IEEE Journal of Selected Topics in Signal Processing*. 2011. Vol. 5, No. 7. P. 1353–1365.
16. Yang K., Liu D., Chen Z., Wu F., Li W. Spatiotemporal Generative Adversarial Network-Based Dynamic Texture Synthesis for Surveillance Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*. 2021. Vol. 32, No. 1. P. 359–373.
17. Li. Guilin, Chen P., Guo S. Texture Blending for Photorealistic Composition on Mobile AR Platform. *15th International Symposium on Medical Information and Communication Technology (ISMICT)* : proceedings. Xiamen, 14–16 Apr. 2021. Xiamen, 2021. P. 92–97.
18. Vishnevyy S.V., Zhuk S.Ya. Two-Stage mutual causal filtration and segmentation of heterogeneous images. *Radioelectronics and Communications Systems*. 2011. Vol. 54, No. 10. P. 554–565.
19. Gonzalez R.C., Woods R.E. Digital Image Processing. Pearson. 2018. 1022 p.
20. Ramya S., Shibu N.V., Malathi P. A Survey on Texture Synthesis and Its Approaches. *International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)* : proceedings. Kovilpatti, 07–09 Jan. 2016. Kovilpatti, 2016. P. 1–4.
21. Huang Z., Lin X., Chen C. Fast Texture Synthesis for Discrete Example-Based Elements. *IEEE Access*. 2020. Vol. 8, P. 76683–76691.
22. Luo G., Zhu Y., Guo B. Fast MRF-Based Hole Filling for View Synthesis. *IEEE Signal Processing Letters*. 2018. Vol. 25, No. 1. P. 75–79.
23. Haindl M., Havlicek V. BTF Compound Texture Model with Fast Iterative Non-parametric Control Field Synthesis. *14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* : proceedings. Las Palmas de Gran Canaria, 26–29 Nov. 2018. Las Palmas de Gran Canaria, 2018. P. 98–105.

24. Wang L., Chen W., Yang W., Bi F., Yu F.R. A State-of-the-Art Review on Image Synthesis With Generative Adversarial Networks. *IEEE Access*. 2020. Vol. 8. P. 63514–63537.
25. Fadaeddini A., Majidi B., Eshghi M. A Case Study of Generative Adversarial Networks for Procedural Synthesis of Original Textures in Video Games. *IEEE 2nd National and 1st International Digital Games Research Conference: Trends, Technologies, and Applications (DGRC)* : proceedings. Tehran, 29–30 Nov. 2018. Tehran, 2018. P. 118–122.
26. Wang Z., Li M., Xia G. Conditional Generative ConvNets for Exemplar-Based Texture Synthesis. *IEEE Transactions on Image Processing*. 2021. Vol. 30. P. 2461–2475.
27. Zhang L., Liu Y. Image Generation Based on Texture Guided VAE-AGAN for Regions of Interest Detection in Remote Sensing Images. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* : proceedings. Toronto, 06–11 Jun. 2021. Toronto, 2021. P. 2310–2314.
28. Maarouf A., Hachouf F., Kharfouchi S. Exemplar-based Texture Synthesis Using Two Random Coefficients Autoregressive Models. *Image Analysis and Stereology*. 2023. Vol. 42., No. 1. P. 37–49.
29. Vaishali D., Ramesh R., Anita Christaline J. 2 D autoregressive model for texture analysis and synthesis. *International Conference on Communication and Signal Processing* : proceedings. Melmaruvathur, 03–05 Apr. 2014. Melmaruvathur, 2014. P. 1135–1139.
30. Zaporozhets D.A., Vyshnevyi S.V. Filtration of image sequences distorted by noise using two-stage processing technique. *Radioengineering Problems, Signals, Devices and Systems* : proceedings of the XII-th international scientific and technical conference, Kyiv, 13–15 Dec. 2023. Kyiv, 2023. P. 8–10.
31. Khandelia A., Gorecha S., Lall, B., Chaudhury S., Manthur M. Parametric Video Compression Scheme using AR based Texture Synthesis. *Sixth Indian Conference on Computer Vision, Graphics and Image Processing* : proceedings. Bhubaneswar, 16–19 Dec. 2008. Bhubaneswar, 2008. P. 219–225.
32. Racape F., Doshkov D., Koppel M., Ndjiki-Nya P. 2D+T Autoregressive Framework for Video Texture Completion. *IEEE International Conference on Image Processing (ICIP)* : proceedings. Paris, 27–30 Oct. 2014. Paris, 2014. P. 4657–4661.
33. Dolloff J., Doucette P. The Sequential Generation of Gaussian Random Fields for Applications in the Geospatial Sciences. *ISPRS International Journal of Geo-Information*. 2014. Vol. 3(2). P. 817–852.
34. Xiao Te, Zhang L., Li D., Cao Z. Efficient 3-D Random Field Simulation with Separable Correlation Functions. *7th International Symposium on Geotechnical Safety and Risk (ISGSR)* : proceedings. Taipei, 11–13 Dec. 2019. Taipei, 2019. P. 358–363.

Vyshnevyi S.V., Zhurba A.V. SYNTHESIS OF VIDEO SEQUENCES CONTAINING A HOMOGENEOUS TEXTURE WITH A SEPARABLE AUTOCORRELATION FUNCTION

The article addresses the problem of synthesizing video sequences in which all frames contain a homogeneous texture with identical statistical characteristics. A brief review of practical applications of texture image generation methods in the context of video data processing is presented. The study focuses on a video sequence synthesis algorithm based on the use of an autoregressive model of a Gaussian random field with a separable autocorrelation function that accounts for the spatial-temporal correlation of samples. The article provides analytical expressions and block diagrams of algorithms that can be used to implement the proposed video sequence generation approach using an appropriate programming language. The application of the presented texture sequence synthesis algorithm requires prior estimation of the autoregressive model parameters through analysis of training data containing homogeneous texture. This involves estimating the correlation coefficients along each coordinate, as well as calculating the mean brightness and the standard deviation of the pixel values of the texture. The algorithm was modeled in the MATLAB software environment. The article presents samples of synthesized textures, time performance indicators for video file generation across different frame formats, autocorrelation function plots, and histograms of the resulting images. It was found that the frames of the synthesized video sequences are characterized by an anisotropic autocorrelation function, which in some cases may be considered a drawback. To address this issue, it is necessary to increase the complexity of the texture synthesis procedure by applying a higher-order model. The results of this study can be used in scientific research and practical applications where the synthesis of video sequences containing texture is required.

Key words: video sequence, texture, autoregressive model, Gaussian random field, algorithm, digital image.

Дата надходження статті: 29.07.2025

Дата прийняття статті: 28.08.2025

Опубліковано: 27.10.2025