МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В. Н. КАРАЗІНА
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
КОМП'ЮТЕРНОЇ ФІЗИКИ ТА ЕНЕРГЕТИКИ

Кафедра інформаційних технологій в фізико-енергетичних системах

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи
за освітньо-кваліфікаційним рівнем бакалавр

«Генерація істинно випадкових чисел на основі джерела природного шуму»

Виконала: студентка 4 курсу, групи НФ-43
Маматова Д.В.
спеціальність:
105 Прикладна фізика та наноматеріали
освітня програма: Прикладна фізика
енергетичних систем
Керівник: к.т.н. Лісін Д.О.

Рецензент:

Харків – 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н.Каразіна

Навчально-науковий інститут комп'ютерної фізики та енергетики
Кафедра інформаційних технологій у фізико-енергетичних системах
Рівень вищої освіти (освітньо-кваліфікаційний рівень) бакалавр
Спеціальність 105 Прикладна фізика та наноматеріали

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Руслан Сухов

підпис

"____"_____2020 року

З А В Д А Н Н Я

НА ДИПЛОМНУ РОБОТУ

Маматова Дар'я Вікторівна

1.Тема роботи «Генерація істинно випадкових чисел на основі джерела природного шуму» керівник роботи: Лісін Денис Олександрович, кандидат технічних наук,

затверджені наказом по університету від "05" грудня 2019 року №0210-05/2249

2. Строк подання студентом роботи 06.06.2020

3. Перелік питань, які потрібно розробити:

1) Розглянути існуючі генератори істинно випадкових чисел;
2) Обрати оптимальне джерело інформаційного шуму;
3) Розробити алгоритм для отримання інформаційного шуму;
4) Розробити програмне забезпечення;
5) Провести тестування програмного забезпечення.

4. План роботи

№ з/п	Назви етапів роботи
1	Вибір теми, узгодження, а також затвердження її на кафедрі
2	Затвердження теми на кафедрі
3	Робота над літературою та іншими джерелами
4	Написання вступу до роботи та першого розділу
5	Написання другого розділу
6	Написання третього розділу
7	Завершення написання дипломної роботи(висновки та література)
8	Подача науковому керівнику
9	Відгук, рецензія, підготовк до захисту
10	Попередній захист
11	Захист

5.Дата видачі завдання

Студент _____ Дар'я МАМАТОВА

підпис

Керівник роботи _____ Денис ЛІСІН

підпис

РЕФЕРАТ

Маматова Д.В. Генерація істинно випадкових чисел на основі джерела природного шуму.

Дипломна робота представлена на 45 сторінках; складається з вступу, 3 розділів та висновків: містить 20 рисунків, 2 таблиці, список використаної літератури з 35 найменувань.

Мета даної роботі – розробка програмного забезпечення для генерації істино випадкових послідовностей на основі шуму отриманого з кольорового зображення.

Об’єкт дослідження – випадкові числа.

Предмет дослідження – генерація істинно випадкових чисел.

Розроблене програмне забезпечення яке генерує послідовність істино випадкових чисел на основі шуму отриманого з кольорового зображення.

Зображення отримано з камери закріпленої на кронштейні з двома сервопріводами, які забезпечують рух камери по двох осях, управління відбувається за допомогою налагоджувальної плати STM32F407.

Алгоритм для програмного забезпечення генерує задану кількість байт чисел, кожні чотири байти генеруються з окремого зображення. При роботі з зображенням виключаються насичені пікселі для отримання неупереджених байтов.

Результати згенеровані у програмному забезпеченні були успішно протестовані на статичних тестах Національного інституту стандартів і технологій.

КЛЮЧОВІ СЛОВА: ГЕНЕРАТОР ІСТИННО ВИПАДКОВИХ ЧІСЕЛ, ВИПАДКОВІ ЧИСЛА, ІНФОРМАЦФНИЙ ШУМ, ЦИФРОВИЙ ШУМ.

РЕФЕРАТ

Маматова Д.В. Генератор истинно случайных чисел на основе источника природного шума.

Дипломная работа представлена на 45 страницах; состоит из введения, 3 глав и выводов; содержит 20 рисунков, 2 таблицы, список использованой литературы из 35 наименований.

Цель данной работы – разработка программного обеспечения для генерации истинно случайных чисел на основе шума полученного из цветного изображения.

Объект исследования – случайные числа.

Предмет исследования – генерация случайных чисел.

Разработано программное обеспечение которое генерирует последовател ьность истинно случайных чисел на основе шума полученного из цветного изображения.

Изображение получено с камеры закрепленной на кронштейне с двумя сервоприводами, которые обеспечивают движение камеры по двум осям, управление происходит с помощью отладочной платы STM32F407.

Алгоритм для программного обеспечения генерирует заданное количество байт чисел, каждые четыре байта генерируются из отдельного изображения. При роботе с изображениями исключаются насыщенные пиксели для получения независимых байтов.

Результаты полученные в программном обеспечении были успешно протестированы на статистических тестах Национального института стандартов и технологий.

КЛЮЧЕВЫЕ СЛОВА: ГЕНЕРАТОР ИСТИННО СЛУЧАНЫХ ЧИСЕЛ, ИСТИННО СЛУЧАЙНЫЕ ЧИСЛА, ИНФОРМАЦИОННЫЙ ШУМ, ЦИФРОВОЙ ШУМ.

ABSTRACT

Mamatova D.V. True random numbers generator based on source of natural information noise.

The diploma project is presented on 45 pages; consists of introduction, 3 chapters and conclusions; contains 20 figures, 2 tables, and a list of used literature of 35 items.

The purpose of this work is to develop software for generation of true random numbers which are based on noise from color image.

The object of study is random numbers.

The subject of the study is generation of random numbers.

Develop software generates sequence of true random numbers based on noise from color image.

Images was taken from camera that is fixed on the bracket. The bracket contains two servos that provide movement along two axes. All parts are controlled by microcontroller STM32F407.

The software algorithm generates sequence with specified numbers of bits, every four bytes generated from new image, the saturated pixels are excluded to receive unbiased bits.

All results successfully passed statistical tests of National Institute of Standard and Technology.

KEYWORDS: TRUE RANDOM NUMBERS GENERATOR, TRUE RANDOM NUMBERS, INFORMATIONAL NOISE, IMAGE NOISE.

ЗМІСТ

ВСТУП.....8
1. ІСНУЮЧИ ГЕНЕРАТОРИ ВИПАДКОВИХ ЧИСЕЛ.....11
1.1 Генератори псевдовипадкових чисел (ГПВЧ).....11
1.2 Використання псевдовипадкових чисел.....15
1.3 Генератори істинно випадкових чисел (ГІВЧ).....17
1.4 Генератори істинно випадкових від Intel.....17
1.5 Використання генераторів істинно випадкових від чисел.....20
2. АЛГОРИТМ ГЕНЕРАЦІЇ ВИПАДКОВИХ ЧИСЕЛ З ШУМУ ЗОБРАЖЕННЯ.....22
2.1 Джерела шуму у сенсорах камери.....22
2.2 Модель шуму.....23
2.3 Наближення для ефективного вибору неупорядженних пікселі.....29
2.4 Алгоритм у псевлокоді.....34
3. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ТА ТЕСТУВАННЯ РЕЗУЛЬТАТІВ.....36
3.1 Програмна реалізвція.....36
3.2 Тестування отриманих результатів.....38
ВИСНОВКИ.....42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....43

ВСТУП

Комп’ютерні та телекомунікаційні системи грають важливу роль у світі сучасних технологій. Комунікації та передача даних через комп’ютер торкаються майже кожного аспекту нашого життя, наприклад, відстеження персональних даних, торгування через інтернет, онлайн-банкінг та надсилання імейлів. Так як через дріт або бездротово передається цінна інформація, потреба в захисту цієї інформації від хакерів росте. Стурбованість безпекою підкреслює важливість у розвитку технологій та методів для трансформації даних з метою приховування її інформаційного змісту щоб запобігти модифікації та несанкціоноване використання.

Генерація випадкових чисел це фундаментальний процес для захисту конфіденційності електронних комунікацій. Це ключовий компонент шифрування який захищає інформацію від нападників роблячи її нечитабельною без належного процесу дешифрування. Оскільки сила шифрування безпосередньо пов’язана з випадковістю бінарних чисел які використовуються в ньому, тому існувала величезна потреба спроектувати та розробити ефективний генератор випадкових чисел, що може виробляти істинно випадкові числа щоб впровадити безпечну криптографічну систему.

Крім кібербезпеки випадкові числа ще використовуються в таких галузях як:

– Соціологічні та наукові дослідження. Підготовка випадкових вибірок при зборі даних, опитуванні думок чи дослідженні фізичних явищ з випадковим вибором результатів експериментів.

– Моделюванні. У комп’ютерному моделюванні фізичних явищ. Крім того, математичне моделювання використовує випадкові числа як один із інструментів численного аналізу.

– Прийняття рішень в автоматизованих експертних системах. Використання випадкових чисел є частиною стратегій прийняття рішень.

Наприклад, для неупередженості вибору квитка студентом на іспиті. Випадковість також використовується в теорії матричних ігор.

– Оптимізація функціональних залежностей. Деякі математичні методи оптимізації використовують стохастичні методи для пошуку екстремумів функцій.

– Розваги та ігри. В комп’ютерних або настільних іграх випадковість дозволяє урізноманітнити ігровий процес.

Генератори випадкових чисел бувають фізичними, або вони можуть бути обчислювальні. Більшість доступних генераторів випадкових чисел програмні. Так як алгоритми використані у програмних ГВЧ фіксовані, числа згенеровані ними здаються випадковими у будь-який час, але насправді дотримуються заданої послідовності. Тому, ці числа не є насправді випадковими та можуть бути легко передбачені при наявності інформації про алгоритм який генерує числа. Випадкові числа згенеровані цими алгоритмами можуть лише імітувати істинну непередбаченість генератора істинно випадкових чисел. Тому ці програмні ГВП прийнято називати “генераторами псевдовипадкових чисел ”.

Також існують фізичний ГВЧ які являються псевдовипадковими.

З іншої сторони існують фізичні ГВЧ які можуть генерувати істинно випадкові там непередбачені бінарні послідовності чисел. Числа згенеровані такими ГВЧ не мають визначеної структури чи порядку, наступне число у послідовності завжди несподіваність, що не може бути передбачено людиною або будь-яким обчислювальним пристроєм знаючи внутрішній алгоритм чи структуру схеми. Такі генератори використовують якийсь тип фізичного шуму як джерело початкових значень для подальшої генерації істинно випадкових чисел. Деякі із поширених фізичних шумів, які використовуються у ГВЧ включають в себе: тепловий шум, джітер годинника та ядерний розпад. Так як шум який породжують ці джерела випадковий, то і числа згенеровані цими джерелами будуть випадковими.

Зростає інтерес в областях цифрового дизайну у розробці простому апаратно основаному генераторі істинно випадкових чисел. Компанії з

виробництва чипів виготовлюють генератори випадкових чисел на основі кремнію , що можуть бути інтегровані у чип процесора для провадження захисту інформації високого рівню у цифрових системах.

Метою даної роботи є розробка швидкого та надійного генератора істино випадкових чисел на основу шуму кольорового зображення, використання всіх типів шумів які існують у системах цифрових камер.

Об’єкт роботи – випадкові числа.

Предмет роботи – генерація випадкових чисел.

Актуальність: З розвитком цифрових екосистем мобільні додатки потребують швидкі та надійні алгоритми для генерації випадкових чисел. В той самий час резолюція і швидкість камер досягла досить високого рівня. Утилізація камер мобільних телефонів чи портативних пристроїв с камерою для шифрування та захисту даних є зручним, швидким там бюджетним методом генерації істинно випадкових чисел.

1. ІСНУЮЧИ ГЕНЕРАТОРИ ВИПАДКОВИХ ЧИСЕЛ

1.1 Генератори псевдовипадкових чисел (ГПВЧ)

Апаратний генератор псевдовипадкових чисел це пристрій здатний згенерувати послідовність бінарних чисел які імітують властивості випадкових чисел. Вхідні початкові значення які подаються на ГПВЧ називається насіння. Вихідна послідовність з'являється випадковою але істинно є непередбачуваною.

Вихідна послідовність бінарних чисел ГПВЧ є детермінованою функцією значень насіння, що значить послідовність може бути відтворена пізніше якщо відоме насіння. Термін "псевдовипадкові" посилається на детермінований характер генератора.

ГПВЧ також є періодичними, так як випадковість обмежена генерацією насіння, вихідна послідовність бінарних чисел буде повторюватися через рівні проміжки часу. Період ГПВЧ визначається максимальною довжиною неповторних закономірностей у послідовності. ГПВЧ що містить n бітів внутрішніх ставні не може мати період більше за 2^n, іноді період набагато коротший залежно від даного насіння. Випадковість ГПВЧ може бути збільшена якщо включити у ГПВЧ значно більше бітів чім вимагається споживачем. Проте додаткові біти вимагають додаткове обладнання для ГПВЧ, що є небажаним.

Добрим прикладом апаратного генератора псевдовипадкових чисел є Регістр зсуву з лінійним зв'язком (Linear feedback shift register, LFSR). Регістр зсуву з лінійним зв'язком це набір регістрів зсуву з'єднаних послідовно з виходами інших регістрів зсуву у конфігурації виключного АБО для забезпечення механізму зворотного зв'язку. Коли на вхід регістрів подаються початкові значення та таймер починає роботу, генерується псевдовипадкова закономірність 1 та 0. Рис. 1.1. показує 4-бітний регістр зсуву з лінійним зв'язком який використовує D-тригер та XOR. Такий регістр називається

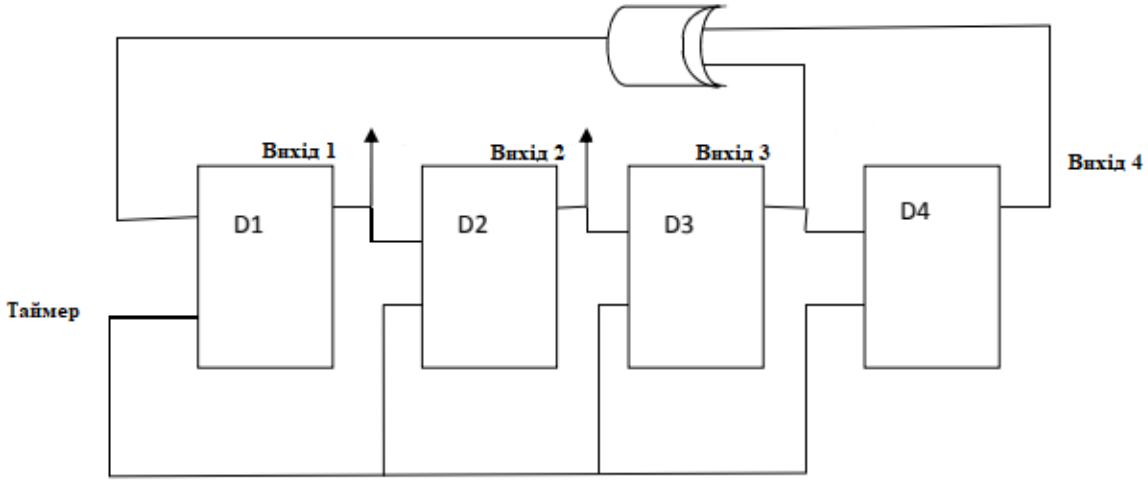


Рис. 1.1. 4-бітний регістр зсуву з лінійним зворотним зв'язком

регістром зсуву з лінійним зв’язком максимальної довжини, якщо він може згенерувати потік випадкових чисел що мають максимальну дожину $2^n - 1$ перш ніж вона починає повторюватися, де n це кількість елементів реєстру в ньому. У максимальній довжині 32-біта регістр зсуву з лінійним зв’язком може згенерувати до 4 мільярдів випадкових чисел перш ніж він почне повторювати послідовність чисел знову.

Дуже ранній та ефективний підхід до генерування випадкових чисел за допомогою ГПВЧ описано у патенті США 5732138 [1], який використовує хаотичну систему як джерело випадковості для отримання випадкових чисел.

У цьому винаході стан хаотичної системи перетворений у послідовність бітів для генерації бінарної послідовності. Цей бінарний рядок потім криптографічно хешируют для отримання другої бінарної послідовності. Другий бінарний рядок використовується як начальний стан насіння псевдовипадкового генератора чисел. Випадкові вихідні дані можуть бути використані для захисту інформації.

Хаотична система – це система, яка з часом випадково змінює стан. Тут хаотична система використана як джерело випадковості для перетворення стану систему у бінарну послідовність. У винаході про який йшла мова раніше, як хаотична система була використані лавові лампи. Лавова лампа це система з

двома різними типами рідини, які мають різні кольори та різні хімічні властивості, ці рідини поєднуються в закритому контейнері. Колірний малюнок рідини змінюється у непередбачуваній манері з часом. На Рис. 1.2. показано як малюнок рідини змінюється за t секунд. У винаході використовують чотири лава лампи, зміну їх кольорового малюнку перетворюють в цифровий вигляд за допомогою цифрової камери. Цифрова камера знаходиться перед лавовими лампами та записує фотографію як цифрове зображення. Записане зображення можна описати як прямокутний масив, де кожне значення відповідає кольоровому малюнку в цій точні. Прямокутний масив складається з пікселів, це бінарні значення відповідного кольору та інтенсивності.

Бінарний рядок, згенерований з зображення перетворюють у другий бінарний рядок за допомогою криптографічної хеш-функції. Тут використовується Secure Hash Standard – криптографічна хеш-функція за двома режимами, розроблена Національним інститутом стандартів і технологій.



Лавова лампа після t секунд

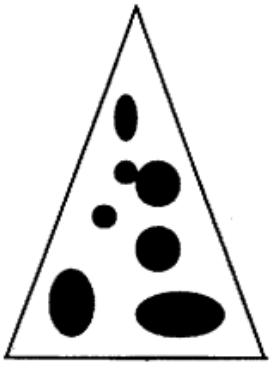


Рис. 1.2. Лавова Лампа

Криптографічна хеш-функція приймає бінарні рядки отриманні з зображення та перетворює їх у інші бінарні рядки. Наприклад, функція бере рядок x=00001101100 та перетворює його у рядок hash(x)=10110010011. Криптографія хеш-функція має наступні властивості :

- x та hash(x) можуть не мати однакову довжину. Користувач буде спостерігати тільки hash(x), тому буде важно визначити стан хаотичної системи x при наявності тільки вихідних даних.
- значення hash(x) та hash(x+1) абсолютно різні. Криптографічна функція розроблена так щоб прогнозування майбутнього результату на основі теперішнього було майже неможливо.
- Для різних значень x та y є можливість що hash(x) = hash(y).

ГПВЧ отримує насіння згенероване шляхом застосування криптографічної хеш-функції на вхідних даних, що були отримання з джерела хаосу. Потім ГПВЧ приймає потрібну кількість байтів за насіння або початковий стан, та застосовує детерміновану трансформацію для отримання нового стану та випадкових чисел. Це шаг повторюється доки вихідні данні не починають повторюватися. Для подальшої генерації потрібно нове насіння для ГПВЧ, яке можна отримати як показано на Рис. 1.3., це блок-схема з шагами для генерації випадкових чисел.

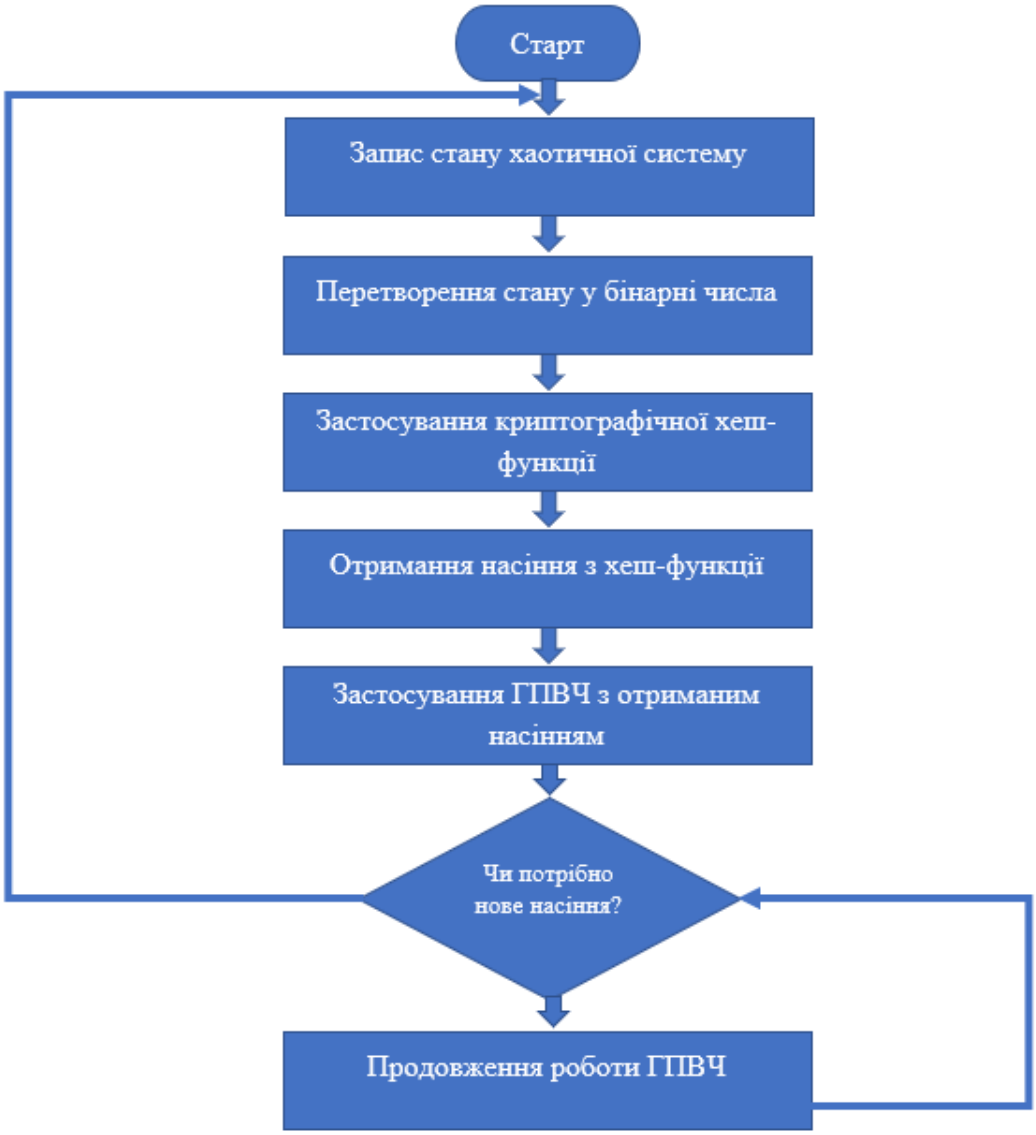


Рис.1.3. Блок-схема

1.2 Використання псевдовипадкових чисел

ГПВЧ не підходить для використання коли непередбачуваність бінарних чисел – сама важлива ознака, наприклад генерація ключів шифрування та дешифрування. Але особливості ГПВЧ роблять їх підходящими для застосування у таких галузях як симуляція, моделювання, fault grading та цифрових комунікацій.

Texas Instruments використовують ГПВЧ у fault grading [2]. ГПВЧ генерує закономірності, які, як було доведено, забезпечують високе покриття

несправностей. Після того, як інженер-конструктор завершує перевірку дизайну та функціональності для ASIC, LFSR розроблено, та його вихідні данні пов'язані з вхідними даними ASIC. Як показано на Рис. 1.4. [2] виходи LFSR мультиплексовані з входами ASIC таким чином, щоб логіка ASIC могла бути імітована як виходами LFSR так і нормальними вхідними даними.

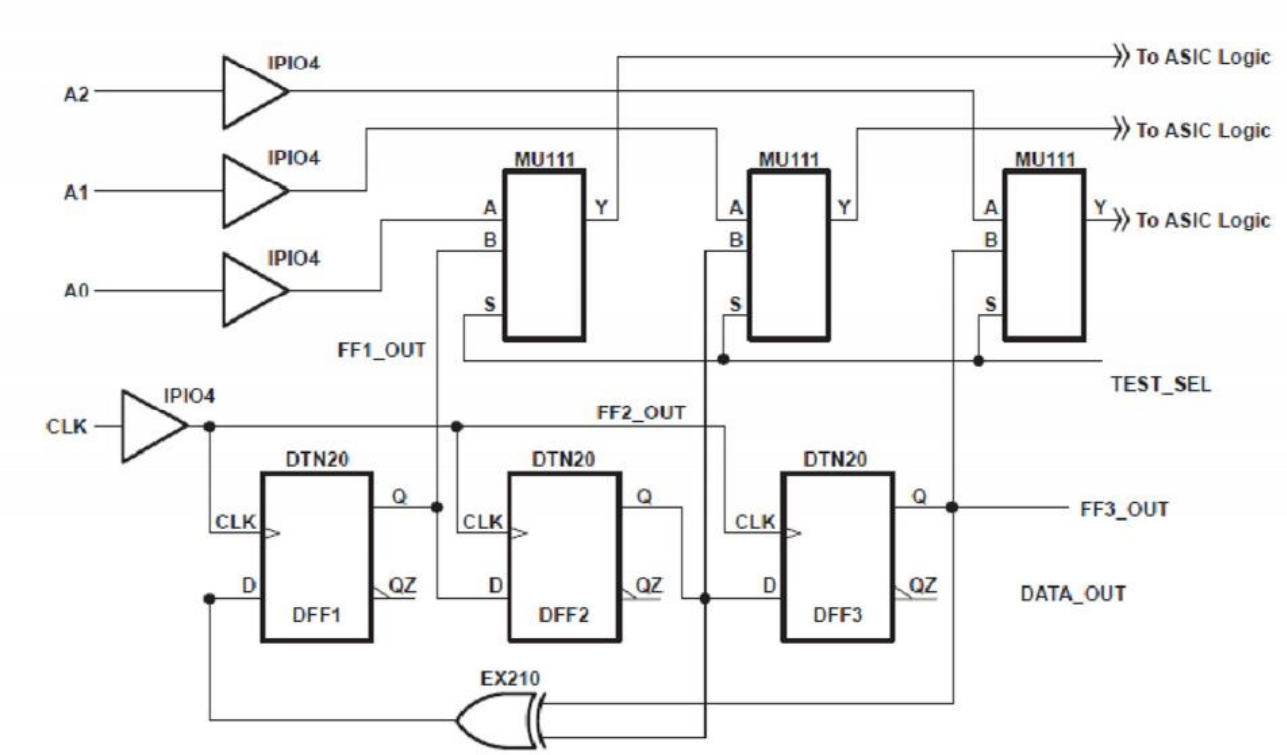


Рис. 1.4. Виходи LFSR мультіплексовані з входами ASIC

Спочатку конструктор бере підсхему LFSR та симулюїть її на самій собі для спостерігання вихідих закономірностей порождених LFSR. Можна легко спостерігати ефект разних насінь на згенеровані закономірности. Важливо знати, чи усі $2^n - 1$ послідовності, які може згенрувати LFSR використовуються у вихідій послідовності [4]. Конструктор буде брати вихідні послідовності згенеровані на LFSR та використовувати іх для підтвредження, що LFSR виробляє коректний сигнал, коли симулюється повна схема. Далі, пристрій встановлюється у тестовий режим та LFSR генерує вихідні дані у ASIC. Вихідні данні повної схеми відібраються кожен тактовий цикл для

генерації очікуваних результатів від ASIC з початковими даними забезпеченими LFSR [6].

Після того, як перевіренно що LFSR виробляє коректний вхідні дані для схеми, результуючий симуляційний вектор може бути використований для перевірки конструкції на дефекти.

1.3 Генератори істинно випадкових чисел (ГІВЧ)

Апаратні генератори істинно випадкових чисел це електронні прибори, що можуть генерувати істинно випадкові та непередбачувані бінарні числа [7]. Вихідні послідовності, отриманні з ГІВЧ, мають довільний та недетермінований характер, це означає, що вихідну бінарну послідовність неможливо відтворити навіть якщо відомі внутрішня конструкція та насіння генератора. Оскільки повна непередбачуваність є ключовим аспектом генератора випадкових чисел, то насіння надане генератору, повинно також бути випадковим. На щастя, зібрати справжню випадковість не так важко, якщо використовувати природній хаос. Деякі приклади фізичних джерел випадковості це тепловий шум, дробовий шум [8], атмосферний шум, цифровий шум та джиттер часів [9]. Для генерації випадкових чисел у ГІВЧ може бути подано насіння з любого випадково фізичного процесу.

Інша ознака ГІВЧ це їх неперіодичний характер (на відміну від ГПВЧ), при поданні одного й того ж насіння у генератор ніколи не будуть згенеровані однакові послідовності.

1.4 Генератори істинно випадкових від Intel

Компанії з виробництва чіпів, такі як Intel ,включають у свої продукти генератори випадкових чисел. У [10] показаний апаратний генератор випадкових чисел Intel для використання у криптографічних додатках. На Рис. 1.5. показана структурна схема генератора випадкових чисел від Intel [10]. Апаратні генератори випадкових чисел основані на непередбачуваних властивостях аналогових пристроїв, таких як тепловий шум або джиттер часів.

Генератор Intel збирає тепловий шум резисторів шляхом підсилення на них напруги. Одна з суттєвих проблем пов’язаних з методом підсилення шуму це те , що випадкові компоненти шуму пов’язані з локальними
псевдовипадковими джерелами шуму, такими як температура та флуктуації у джерелах живлення. Вплив цих джерел можливо мінімізувати шляхом віднімання сигналів відібраних з двох сусідніх резисторів.

Генератор випадкових чисел Intel використовує два кільцевих осцилятора, один швидкий, а інший набагато повільніший. Джерело теплового

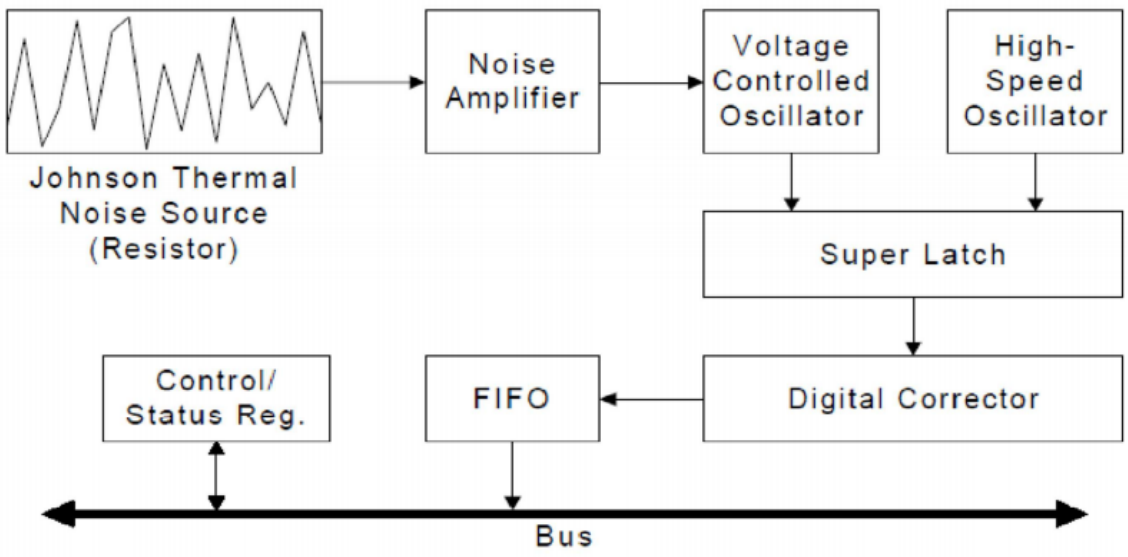


Рис. 1.5. Генератор випадкових чисел Intel

шуму використовується для модуляції частоти більш повільного годинника. Змінний, шумо-модульованний повільний годинник використовується для запуску швидкого годинника. Дрейф між повільним та швидким годинником забезпечує джерело випадкових бінарних чисел. Сигнали обох осциляторів фіксуються і потом подаються на цифровий коректор. Коректор фон Неймана конвертує пари бітів у вихідні біті перетворюючи пару (0, 1) у вихідний біт 1, (1, 0) у вихідний біт 0, та не генеруючи вихідні біті для пар (0, 0) або (1, 1).

Цей підхід включає в себе аналогову конструкцію генератора випадкових чисел на основі кільцевого генератора. Аналогова схема споживає багато енергії і площі. Щороку виробники чіпів придумають процеси для вироблення у менших вагах, а також упаковуючи більшу кількість транзисторів на невеликій площі. Intel вигадали генератор випадкових чисел який базується тільки на цифровому обладнанні.

Цифровий генератор випадкових чисел включає у себе пару інверторів, де вихід кожного інвертора підключено до входу кожного інвертора. Як що на виході першого інвертора 0, тоді вхід другого інвертора також 0 , а тоді його вихід 0. Як що вихід першого інвертора 1, то вихід другого буде 0.

Але ці інвертори встановленні з резисторами у таким чином, що переключення між транзисторами призводять вході і виходи обох інверторів у логічний стан 1. Рис. 1.6. із [11] показує, що два транзистора підключенні до годинника, який регулярно вмикає або вимикає інвертори[11].

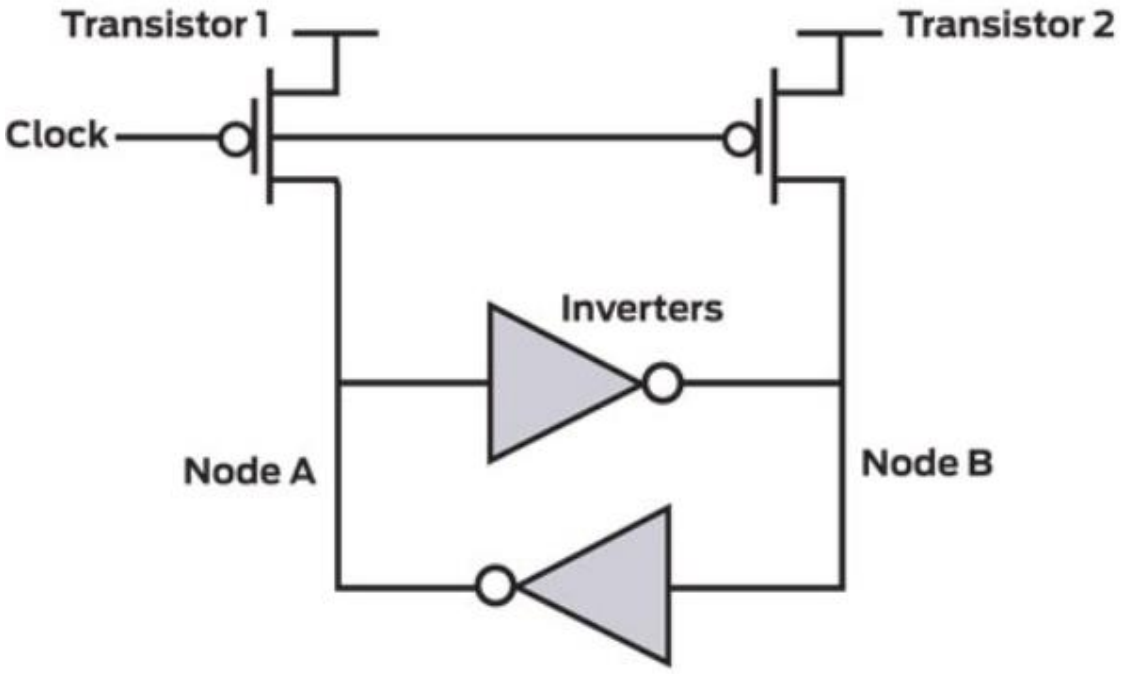


Рис. 1.6. Intel ГВЧ

Два інвертора в схемі нестабільні, коли усі входи та виходи знаходяться у однаковому стані. Але вони стабільні, коли у них протилежні стани. Коли транзистори виключені, обидна інвертора, що примушені знаходитися у стані з однаковими входами та виходами, прагнуть прийняті стабільний стан. Незважаючи на те що інвертори однакової конструкції, завжди є тонка різниця в швидкості їх реакції або характеристиках міцності. Тепловий шум, що присутній в ланцюзі інверторів визначає який стан буде на інверторах. Один біт випадкової інформації генерується за кожен тактовий цикл.

Щоб зробити вихідні послідовності ще більш випадковими Intel розробив трьох каскадну схему, що включає у себе генератор випадкових чисел, нормалізатор (conditioner) та генератор псевдовипадкових чисел. На Рис. 1.7. структурна схема для трьох етапів процесу [11].



Рис. 1.7. Структурна схема для трьох етапів процесу [11]

Генератор випадкових чисел від Intel може генерувати випадкові числа зі швидкістю близько трьох гігабіт в секунду.

1.5 Використання генераторів істинно випадкових від чисел

Генератори істинно випадкових чисел підходять для багатьох застосувань, таких як фінансовані владою лотереї, азартні ігри, симуляції та криптографічний захист систем безпеки. Генератори істинно випадкових чисел це фундаментальний ключ до усіх вимог безпеки щодо захисту конференційної інформації у комп’ютері та у світі телекомунікацій. Тому важливо, щоб

апаратні ГІВЧ для генерації бінарних чисел не мали ризик невдачі, якщо відомий алгоритм або насіння.

У підсумку розглянемо характеристики генераторів істинно випадкових чисел та їх відмінності від генераторів псевдовипадкових чисел.

Таблиця 1

Відмінна характеристика ГІВЧ від ГПВЧ

Характеристики	**Генератори псевдовипадкових чисел**	**Генератори істинно випадкових чисел**
Недетерміновані	Ні	Так
Непередбачувані	Ні	Так
Можна відтворити	Так	Ні
Періодичні	Так	Ні

2. АЛГОРИТМ ГЕНЕРАЦІЇ ВИПАДКОВИХ ЧИСЕЛ З ШУМУ ЗОБРАЖЕННЯ

2.1 Джерела шуму у сенсорах камери

Просте використання даних камери – перетворити параметри яскравості у кожній точці (значенню пікселя) цифрового зображення у послідовність. Якщо камера вказує на випадкове місце, то послідовність буде ймовірно випадковою [12]. Однак для веб-камери і особливо для камери мобільного телефону сценарії можуть сильно відрізнятися. Вони можуть бути направлені на користувача ноутбука, нерухому підлогу чи кишеню. Перед тим, як використати дані камери, треба зрозуміти, що ми насправді отримуємо з датчика зображення. Ми використовуємо наведений нижче поглиблений тест, щоб показати, що прямий потік бітів з камери не є випадковим [13].

Справжня яскравість отримана з сенсорів пікселя [14] :

$$b = b_{NF} + noise \qquad (2.1)$$

де b це кінцева яскравість, b_{NF} це яскравість у якій немає шуму, $noise$ це всі адитивні шуми.

Дробовий шум походить з квантової природи світла [15] та може бути описаний розподілом Пуассона. Якщо світло, що падає на датчик, камери має потік фотонів I, повна кількість фотонів отримана за час t буде It, сигнал шуму матиме значення $\sqrt{It}$ [16].

Темновий струм походить від термоелектронної емісії всередині кремнієвого шару сенсора. Він також підчиняється статистиці Пуассона. Повний шум за час t буде $\sqrt{N_d t}$. [17].

Шум читання N_r походить від процесу читання у мікросхемі. Це одноразовий шум, який попадається лише під час читання. [18]

Повний шум може бути представлений $\sqrt{It + N_d t + N_r^2}$, де сигнал It. Тоді співвідношення сигнал/шум(ССШ) дорівнюватиме $\frac{It}{\sqrt{It+N_d t+N_r^2}}$. [19].

Виробники камер та фотографи прагнуть збільшити співвідношення сигнал/шум, але шуми завжди існують у певній мірі [20]. У добрих світових кондиціях шум менш помітний, але навіть прикриваючи об’єктив пальцями, можна показати, що значення пікселів яскравості коливаються [21]. Ці шуми хоча не є оптимальними для високоякісних зображень , насправді вітаються в нашому алгоритмі оскільки вони виступають як пул фізичної ентропії.

2.2 Модель шуму

Наступні зображення і тести засновані на інтегрованій камері VivoBook X540SA, а також на задній камері Honor 10 lite. Ми очікуємо подібних результатів з будь-якими сучасними цифровими камерами.

На Рис. 2.1 показано гістограми розподілу яскравості в одному знімку. У простих інтегрованих камерах фіксується діафрагма та частота кадрів. Для отримання чіткого зображення є вбудований механізм підвищення чутливості (ISO) при темних кондиціях та зниження чутливості при яскравому світлі. Як

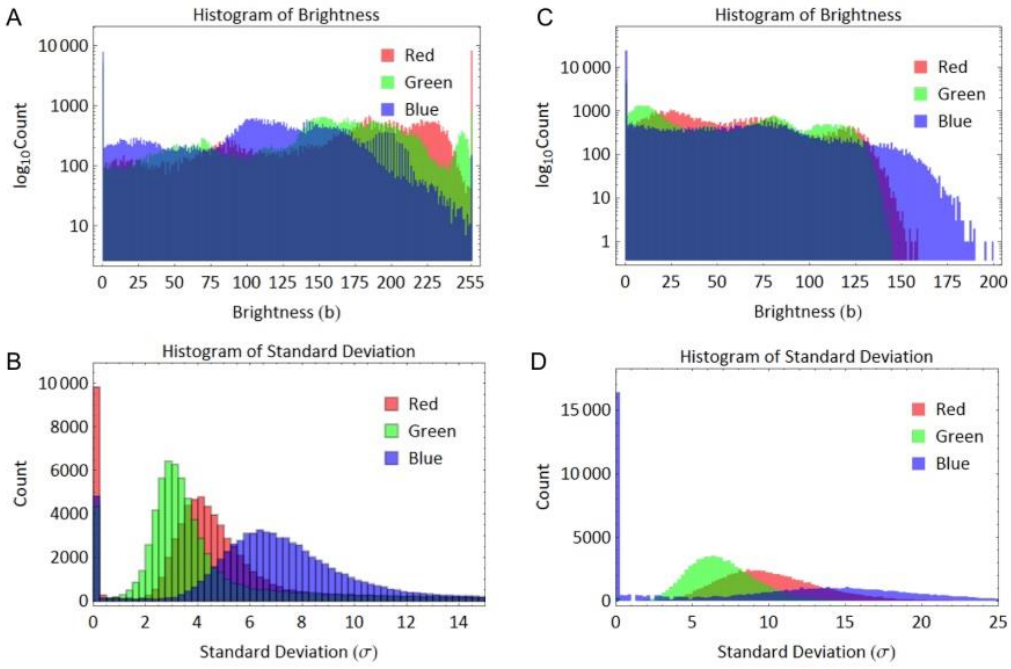


Рис.2.1. Гістограми яскравості та стандартного відхилення у цифрових зображеннях. Статистика у лівій колонці – з зображень зроблених при яскравих кондиціях, правій – темних

результат, типова гістограма зображення повинна розподілятися на тінь, середину та ключові регіони. Однак ми все ж можемо отримати зони надмірної експозиції або недоексопзиції при зйомці зображення коли присутні яскраві та темні компоненти. У цьому випадку гістограма яскравості буде мати піки у максимальних значеннях, та відповідні пікселі називаються «насиченими». На гістограма Рис.2.1., вона побудована по $\log_{10} Count$, насичені пікселі мають дуже високу чисельність на таких зображення. У порівнянні з яскравістю, стандартне відхилення кожного пікселя має менший діапазон. У темних кондиціях сенсор буде підвищувати чутливість [22] це призведе до більшого відхилення, що створює більше невизначеності на кожному пікселі. Більшість стандартних відхилень коливається у діапазоні від 2 до 25. Декілька великих стандартних відхилень може походити від рухів предметів перед камерою. Значна кількість стандартних відхилень рівних 0, може бути результатом мертвих або насичених пікселів. Якщо ми підрахуємо кількість яскравості у діапазоні між 0 і 255, вона буде приблизно дорівнювати кількості стандартних відхилень зі значенням 0.

Розподілення ймовірності яскравості представлених пікселів у 600 кадрах показано на Рис.2.2. для ярких або темних кондицій. Ці зразки пікселів обираються випадковим чином в простими числами. Вони можуть бути наближені їх відповідною функцією густини ймовірності (PDF) нормального розподілення з параметрами (μ, σ). Однак приближення не можливе в двох крайних випадках, коли їх рахунок буде накопичуватися на граничних значеннях.

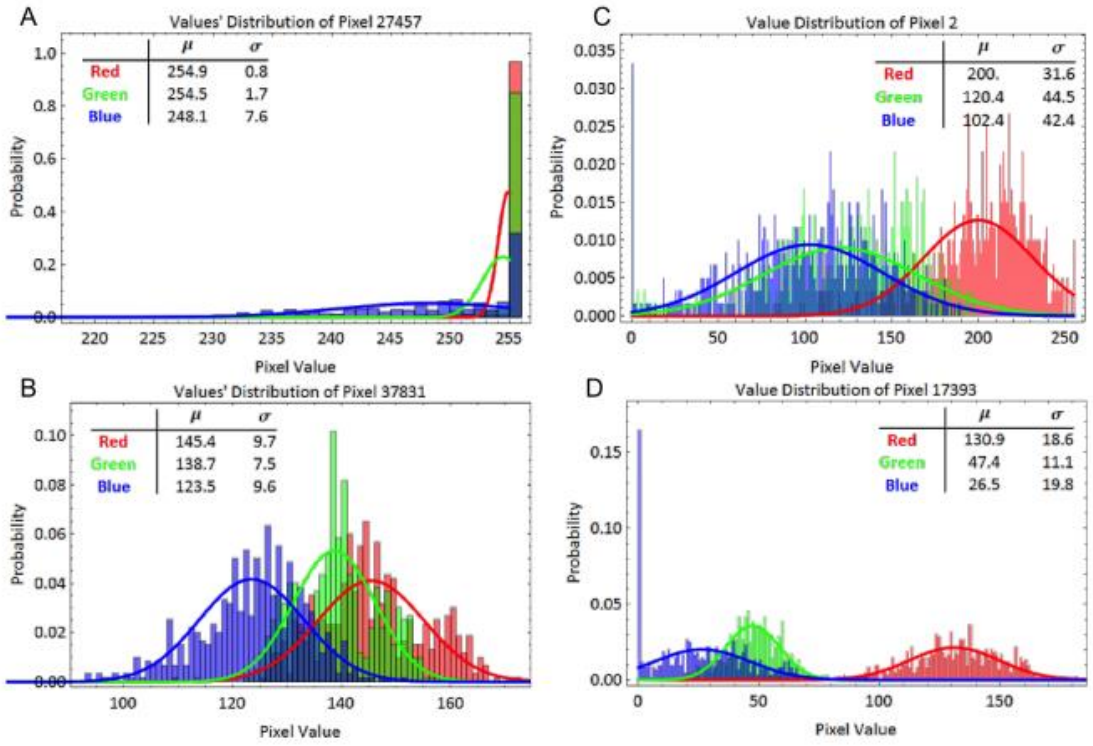


Рис. 2.2. Розподілення ймовірності яскравості представлених пікселів у 600 кадрах. Зображення використані для генерації розподілу ймовірностей у лівому стовпчику були зняти при яскравому освітленні, та у правому стовпчику ті які були зняти у темних кондиція

Загальна поведінка шумів зазвичай моделюється як Additive Zero Mean Gaussian Noise (AWGN) [23] :

$$PDF(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}}\, exp\,(\frac{-(x-\mu)^2}{2\sigma^2}) \qquad (2.2)$$

де x спостерігається, μ це середнє значення та σ це стандартне відхилення розподілу.

Середнє значення μ можна розглядати як яскравість без шуму b_{NF}, при цьому шум додається до базової яскравості b. Тоді $PDF(noise, 0, \sigma)$ стає $PDF(b, b_{NF}, \sigma)$, що переймає усі статистичні особливості шуму [24].

Ми можемо отримати випадкові біти зі значень розподілу, як показано на Рис.2.3. Якщо глибина випадкового біту – 3, то можливо отримати останні три біта від яскравості.

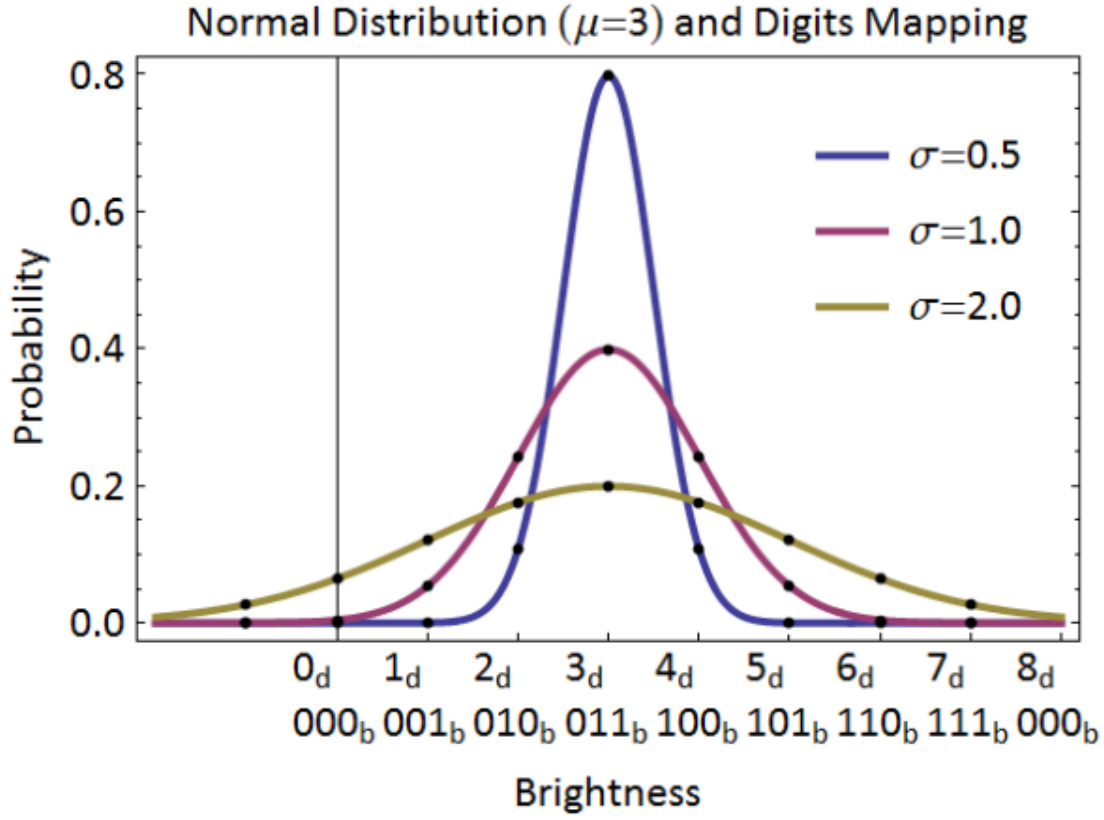


Рис. 2.3. Нормальний розподіл Гаусса відображено у бінарних значеннях. Густина ймовірності Гаусса побудована для $\mu = 3, \sigma = 0.5, 1.0, 2.0$. Останні біти яскравості показані ниже десяткових чисел. Чорні точки представляють кожне стандартне відхилення, σ

Математичне сподівання значення останніх бітів буде дорівнювати :

$$E(bit) = \sum_{b=\mu-4\sigma}^{\mu+4\sigma} PDF(b, \mu, \sigma)(b \bmod 2) \qquad (2.3)$$

Припускаючи густини ймовірності розподілення Гаусса, 99% зразків буде знаходитися у діапазоні $(\mu - 3\sigma, \mu + 3\sigma)$. При підсумовуванні діапазону з $\mu - 4\sigma$ до $\mu + 4\sigma$ необхідно отримати характерну форму «дзвіночка» [24]. Для невеликих σ, сума замінюється на інтеграцію під час числового оцінювання.

В ідеалі, із усіх пікселі повинні генеруватися неупереджені випадкові біти. З ідеального нормального розподілу з симетричним діапазоном очкується значення $\mu = 0.5$. Але це не завжди так у випадку цифрових камер з обмеженим динамічним діапазоном. Значення поза діапазоном відсікаються до граничних

значень. При розрахунку суми, обрізана яскравість $b_{clipped}$ приймає таку форму:

$$b_{clipped}(b) = \begin{cases} b, & 0 \le b \le 255 \\ 0, & b < 0 \\ 255, & b > 255 \end{cases} \quad (2.4)$$

Зміщене очікуване значення тоді стає :

$$E_{shifted}(bit) = \sum_{b=\mu-4\sigma}^{\mu+4\sigma} PDF(b,\mu,\sigma)(b_{clipped}(b) \bmod 2) \quad (2.5)$$

Завдяки вирізанню, уся надмірна ймовірність додається до граничного значення. Справжня середня яскравість та стандартне відхилення зміщені від ідеального гауссового розподілу (μ, σ). Ми можемо розрахувати $(\mu_{shifted}, \sigma_{shifted})$:

$$\mu_{shifted} = \sum_{b=\mu-4\sigma}^{\mu+4\sigma} b_{clipped}(b) PDF(b,\mu,\sigma) \quad (2.6)$$

$$\sigma_{shifted} = \sqrt{\sum_{b=\mu-4\sigma}^{\mu+4\sigma} (b_{clipped}(b) - \mu_{shifted})^{2} PDF(b,\mu,\sigma)} \quad (2.7)$$

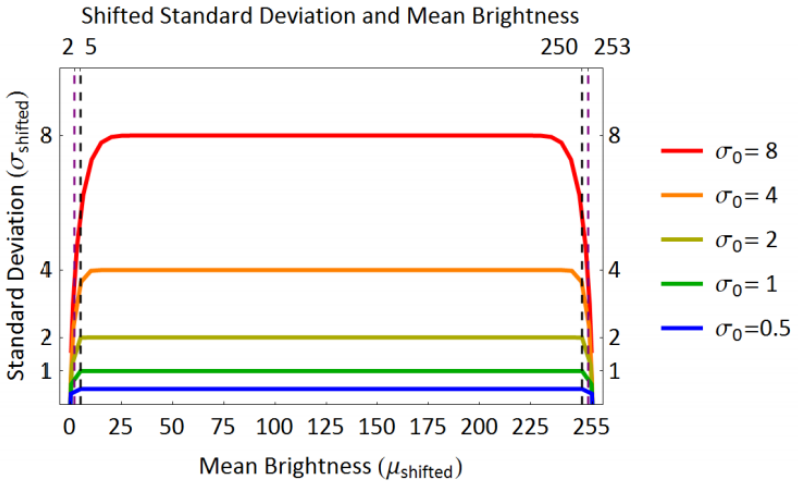


Рис.2.4. Зміщені середня яскравість та стандартне зміщення з обрізаної яскравості, μ у діапазоні з -10 до 265. Вертикальна вісь побудована по $\sqrt{\sigma}$, де $\sigma = 0.5,1,2,3,4,6,8$

Значення $(\mu_{shifted}, \sigma_{shifted})$ ідеального $PDF(b, \mu, \sigma)$ побудовані на Рис 2.4.

Тепер можливо розрахувати $E_{shifted}$ та $(\mu_{shifted}, \sigma_{shifted})$ ідеального розподілу. Їх взаємозв'язок показано на Рис 2.5. Очікування між 0.4995 та 0.5005 помічено сірим, що дозволяє показати упередженість $\frac{0.0005}{0.5} = \frac{1}{1000}$. Взагалі сірі крапки з неупередженої трапецієподібної області(UTR) це функція (μ, σ), в оточенні упередженої нижньої області (BBR) та упередженої бічної області (BSR).

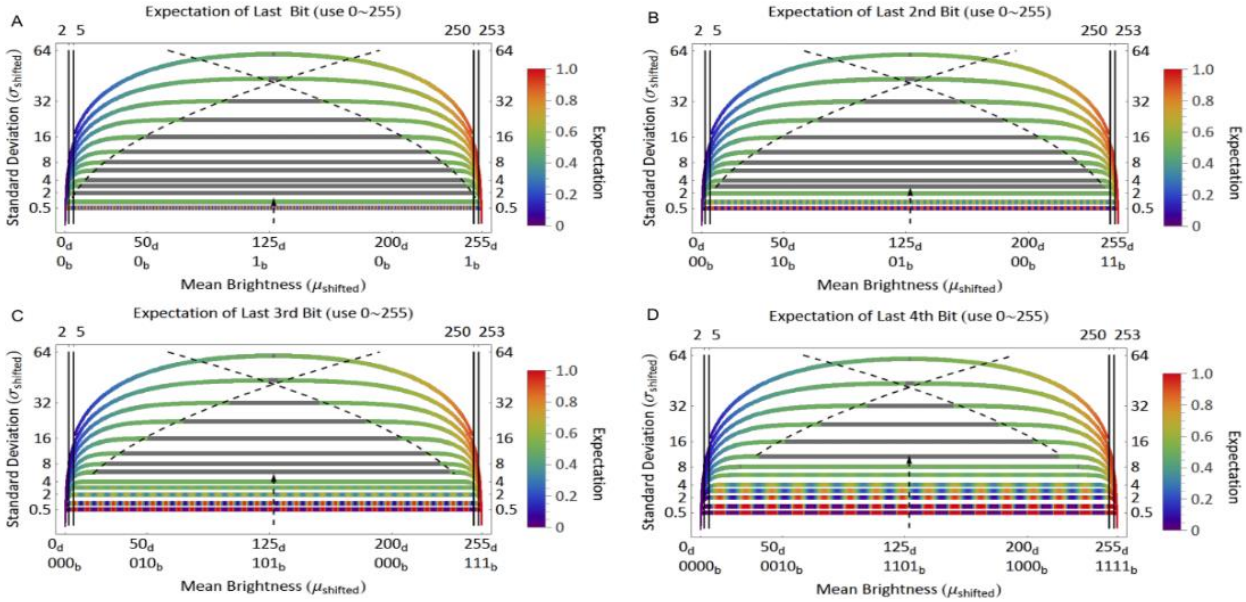


Рис.2.5. Взаємозв'язки між сирим очікуванням, зміщеною середньою яскравістю, та зміщеним стандартним відхиленням. μ у діапазоні з -65 до 320. Вертикальна вісь побудована по $\sqrt{\sigma}$. Межа центральної області повинна бути прямо у лінійній осі. Значення між 0.4995 та 0.5005 помічені сірим. Штрихові лінії показують границі, приближені до нормального σ

BBR походить з надзвичайного маленьких σ (викликаних низьким ISO для яркого освітлення), що не можуть впровадити достатньо випадкової інформації для потрібної глибини біту. З потоковою допустимою похибкою у $\frac{1}{1000}$, нижня границя може бути апроксимована як:

$$\sigma_{min} \geq \frac{2^{depth}}{1.4} \tag{2.8}$$

BSR походить з порушеної симетрії та додаткових ймовірностей на значеннях насичених пікселів. З потоковою допустимою похибкою у $\frac{1}{1000}$, бічна границя може бути апроксимована як:

$$\sigma_{side}(\mu) = \begin{cases} \frac{\mu}{3} & 0 \leq \mu < 127.5 \\ \frac{255-\mu}{3} & 127.5 \leq \mu \leq 255 \end{cases} \tag{2.9}$$

Для отримання неупереджених даних, потрібно вибрати пікселі належні до неупереджених регіонів.

2.3 Наближення для ефективного вибору неупоряджених пікселі

Оскільки неупереджені регіони були теоретично визначенні, можлинво обирати пікселі за їх значеннями (μ, σ). Однак, μ та σ – статистичні властивості, які потребують дані з декількох знімків. Ці значення потребують значну кількість часу для їх збору, що може перешкоджати роботі, а також камера може рухатися, що буде призводити до зміни значень (μ, σ) у пікселів. Також розрахунок належності точки до многокутника (Point-In-Polygon) буде сповільнювати процес. [25]

Деякі емпіричні наближення можуть бути зроблені шляхом спостереження точкової діаграми (μ, σ) за 20 кадрів з цифрової камери Рис.2.6. Усі пікселі з сенсорів червоного каналу построєні за їх координатами (μ, σ). Відповідний графік щільності показує відносну ймовірність кожної конформації. Зелений та синій канал показують подібні особливості, які призводять до однакових висновків.

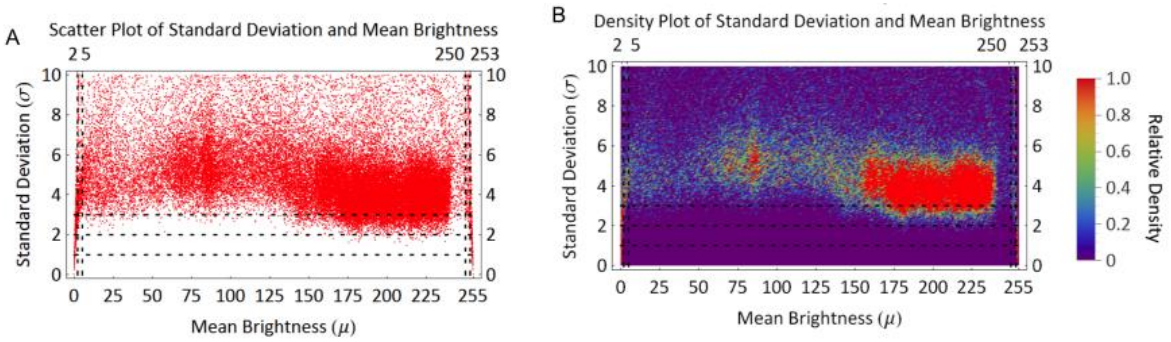


Рисунок 2.6 Точкова діаграма із зміщеним (μ, σ) накладена на рис 2.5. Надзвичайно малі σ з джерела сигналу достатньо рідкісні, багато з них знаходяться біля границь та походять з відсічення

Малі та великі значення σ досить рідкісні. Великі σ можуть виникнути при будь-яких значеннях μ, коли надзвичайно малі σ виникають лише при граничних значення. Це трапляється тому, що будь-які коливання поза діапазону будуть рахуватися як насичені граничні значення. Згадавши взаємозв'язок σ та μ Рис 2.5, можливо об'єднати його з Рис 2.6. Результат, зображений на Рис 2.7. показує, що надзвичайно малі σ можуть бути виключені з діапазону μ, який ви, за рахунок його зменшення.

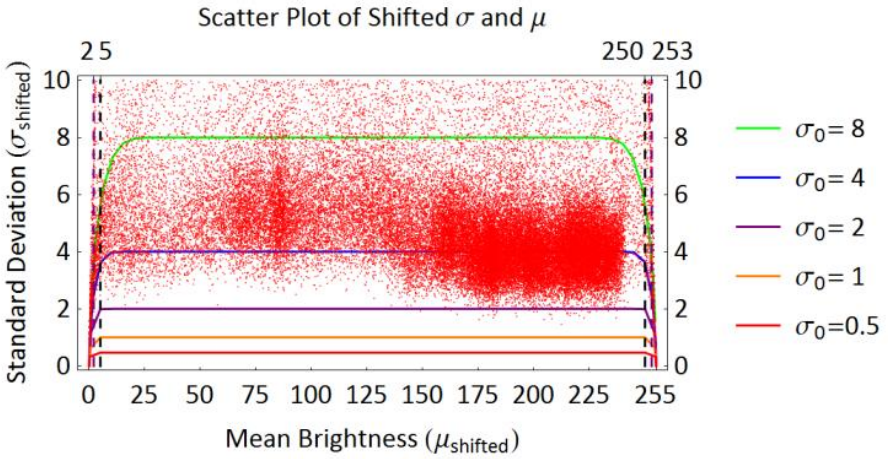


Рис.2.7 Точкова діаграма з відповідним графіком щільності (з права) стандартного відхилення та середнього значенням за 20 кадрів. Пунктирні лінії позначають крайні значення σ та μ

Однак відхилення від граничних значень призведе до ймовірності значення 0, тоді загальна ймовірність більше не буде 1. Тому ми повинні переоцінити очікуване значення, котре стає середнім значення ансамблю, таким що, відрізана функція щільності ймовірності прийме форму (2.10).

$$PDF_{clipped}(b,\mu,\sigma) = \begin{cases} PDF(b,\mu,\sigma) & min \le b \le max \\ 0 & b < min \\ 0 & b > max \end{cases} \quad (2.10)$$

де min може приймати деякі невеликі значення (напр. 1~5) та $max = 255 - min$, для симетрії діапазону.

Останні цифри для середніх значень ансамблю можуть бути розраховано як:

$$< bit > = \frac{\sum_{b=\mu-4\sigma}^{\mu+4\sigma} PDF_{clipped}(b,\mu,\sigma)(b_{clipped}(b)\ mod\ 2)}{\sum_{b=\mu-4\sigma}^{\mu+4\sigma} PDF_{clipped}(b,\mu,\sigma)} \quad (2.11)$$

Перераховане середнє значення ансамблю показано на Рис.2.8. Треба зауважити, що зміщені μ та σ грають роль індексів пікселів, тому вони не перераховуються для нового графіку.

Хоча головна мета останньої операції – уникнути BBR поблизу границь, нові BSR також значно уменшилися порівняно з оригінальними BSR. Найбільші упереджені регіони раніше знаходилися у верхньому лівому/верхньому правому куті (Рис 2.5), тепер змістилися у невеликий нижній кут Рис 2.8. Нижня границя не зачіпається, в той час як нова бічна границя може бути апроксимована як :

$$\sigma_{side}(\mu) = \begin{cases} \frac{\mu - min}{2} & 0 \le \mu < 127.5 \\ \frac{255 - \mu - mim}{2} & 127.5 \le \mu \le 255 \end{cases} \quad (2.12)$$

Подальше зменшення діапазону яскравості суттєво не виключає BSR, та викидає забагато корисних пікселів. Тому обирається діапазон між 2 та 253 у подальшому аналізі у алгоритмі.

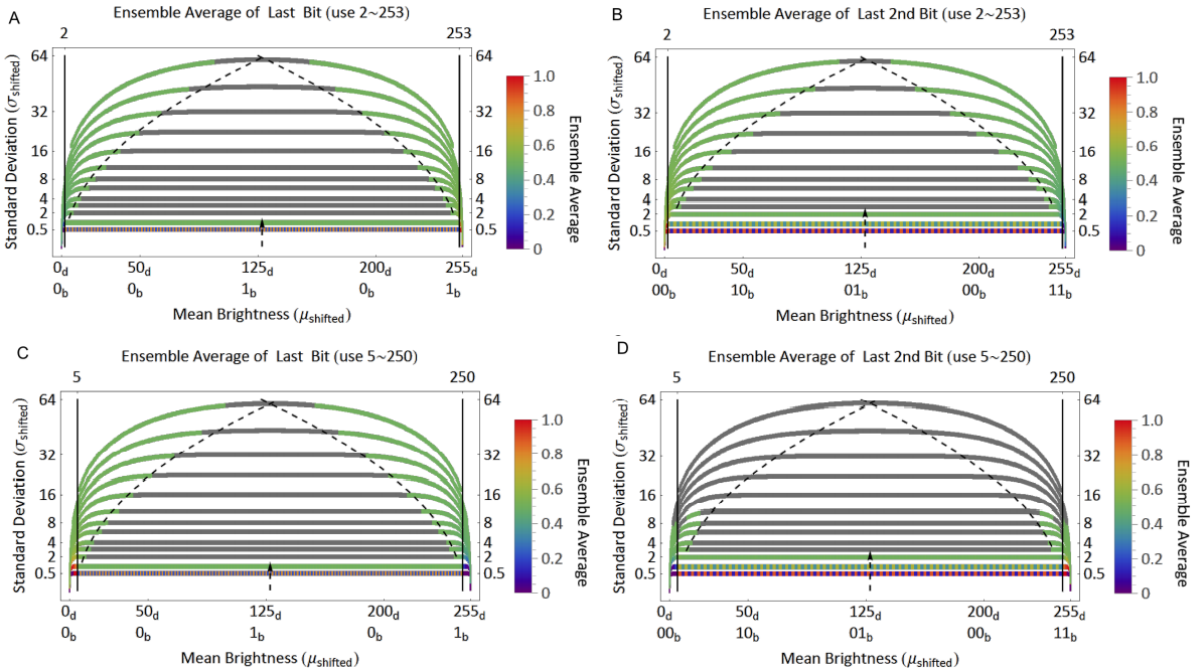


Рисунок 2.8 Взаємозв'язок муж новим очікуванням, зміщеною середньою яскравістю та зміщеним стандартним очікуванням після виключення різних наборів граничних значень. μ у діапазоні з -65 до 320. Вертикальна вісь побудована по $\sqrt{\sigma}$. Значення між 0.4995 та 0.5005 помічені сірим. Штрихові лінії показують границі приближені до нормального σ

На Рис.2.9. показано середнє значення останніх бітів за 20 кадрів, ми можемо спостерігати, що упередження значно зменшилося за рахунок виключення граничних значень. Щоб вирішити, скільки випадкових бітів отримувати з одного пікселя, ми можемо використовувати встановлений зв'язок :

$$depth \leq 1.4\, log_2\, \sigma_{min} \qquad (2.13)$$

у Таблиці 2 показано мінімальне σ, необхідне для певної глибини неупереджених випадкових біт.

Таблиця 2

Таблиця обмежень

$depth$	1	2	3	4	5
σ_{min}	1.43	2.86	5.71	11.43	22.86

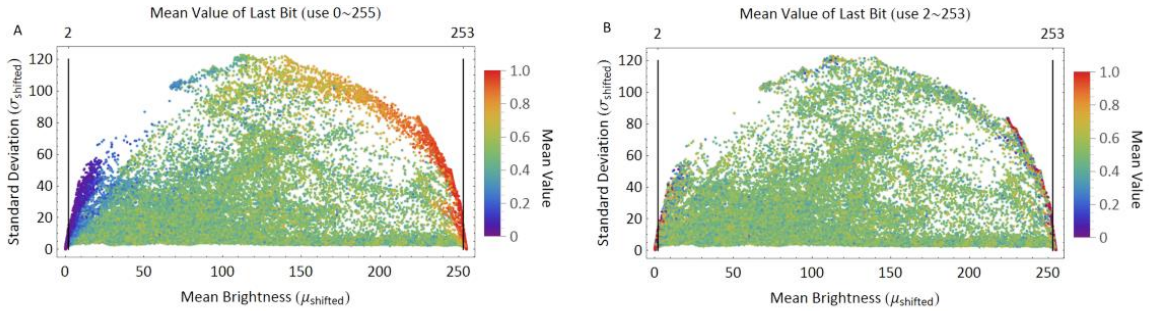


Рисунок 2.9. Середнє значення останніх бітів у 20 квадрах. Після виключення граничної яскравості, упередження значно зменшилося. Зауважимо, що зміщені μ та σ грають роль індексів пікселів, тому вони не перераховуються для ного графіку

Як обговорювалося раніше, стандартне відхилення залежать від чутливості, яка підлаштовується під стан освітлення драйверами камери [27]. Драйвери мають механізм налаштування чутливості до яскравості, тобто зміну σ з середнім μ у усього зображення. Універсальний драйвер повинен адаптуватися до різних умов освітлення шляхом попереднього обчислення середнього значення μ зображення [28].

Оскільки камера завжди може рухатися, ми маємо періодично усереднювати яскравість зображення. Стандартний підхід – використовувати лише останній біт, щоб отримати значення з більшості пікселів. Більш гнучкий підхід – використовувати останні три або чотири біта, але стандартне відхилення має бути великим, що є практичним тільки для темних умов. Враховуючи, що середа, у якій знаходиться камера, може бути непередбачуваною і суттєво змінюватися, ми використовуємо лише останній біт у нашому алгоритмі, щоб забезпечити найвищу якість випадковості значень

при будь-яких умов освітлення, навіть якщо ми навмисно обмежуємо кількість отриманих бітів.

2.4 Алгоритм у псевлокоді

Тепер ми представляємо загальний алгоритм отримання випадкових чисел із зображень, що були зроблени цифровою камерою зі трьома каналами кольору. Було використано цифрову камеру для отримання зображення. Для кожного зображення обрані пікселі з яскравістю у діапазоні з 2 до 253 (виключаючи границі). Цикл для кожного зображення продовжується доки не отримується бажане число випадкових бітів. Усі біти з одного кадру стають підсписком загальної послідовності.

Однак сира послідовність не є ідеально випадковою. BSR існує для кожного кадра. Якщо зображення не має достатню експозицію, упередження може бути великим у кожному кадрі. Оскільки алгоритм повинен постійно фотографувати протягом коротких проміжків часу, ми можемо перегортати біти кожні два кадри. Щоб рівномірно розподілити виправлення по всій послідовності, послідовність перетасовується записуючи її у порядку raw-major, потім читаючи у порядку column-major як квадратну матрицю. Розмірність матриці може бути визначене як $\sqrt{\text{Потрібна довжина}}$. Додаткові біти з кожного кадру додаються до хвоста послідовності, щоб приховати інформацію про розмірність. Нижче описаний алгоритм у псевдокоді :

Ініціалізація камери та змінних

Виділенні квадратного масиву

Генерація ;

while (ОтриманніЧисла< ПотрібніЧисла**) do**

Зробити знімок;

Вибрати яскравість ∈ $[2, 253]$ **;**

Взяти останні біти як Підсписок;

If **(Кадр парний) перегорнути біту у Підсписку;**

Додати Підсписок до Фінального списку у порядку raw-major

ОтриманніЧисла = ОтриманніЧисла + Підсписок.Довижина

end while

Вивод вихідних данних

Загальна складність алгоритму$O(N)$. Зауважимо, що оскільки операції у кожному кадру можуть бути виконані паралельно [29], то при правильно імплементовано на паралельній архітектурі [30] (наприклад CPU), складність може буди зменшена до $O(\sqrt{N})$.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ТА ТЕСТУВАННЯ РЕЗУЛЬТАТІВ

3.1 Програмна реалізвція

Була розроблено програмне забезпечення на основі алгоритму, що був описаний у другому розділі. Для розробки програмної реалізації алгоритму було використана середа розробки MATLAB R2018b. У інтерфейсі програми

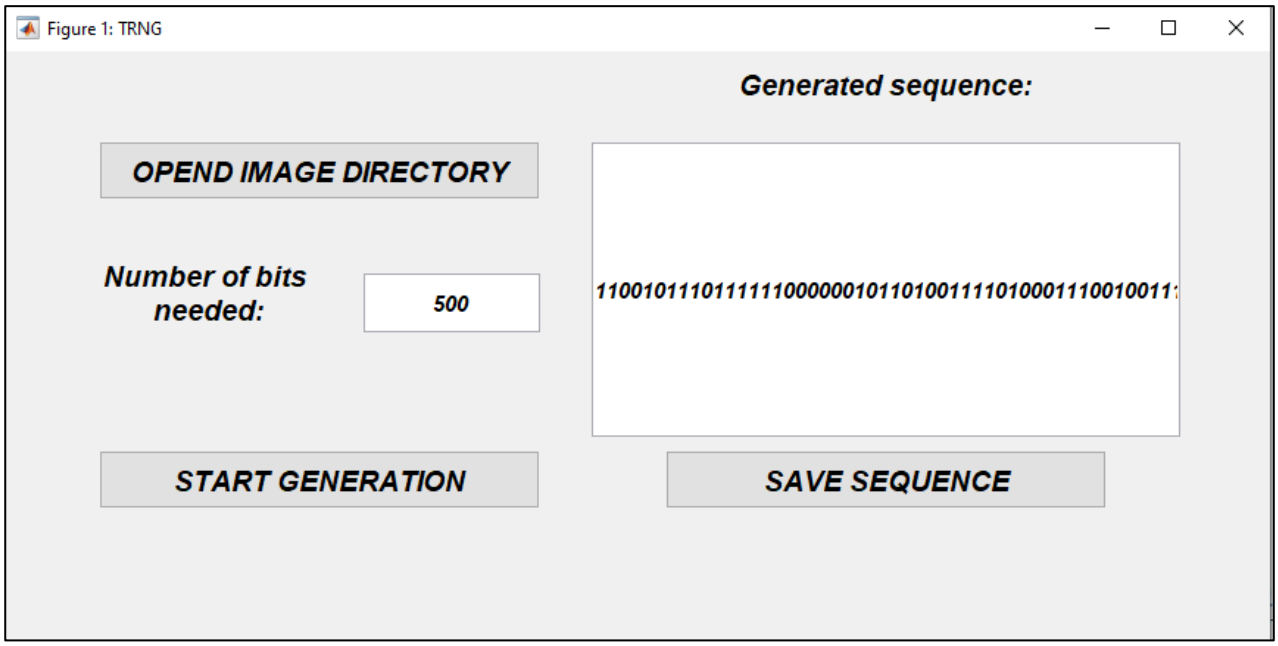


Рис. 3.1. Інтерфейс розробленої програми для генерації випадкових чисел

Рис.3.1. є можливо задати бажану кількість випадкових чисел, та директорію з зображеннями для генерації. Важливо зауважити, що вибір пікселів на зображенні отримується за допомогою вбудованого у MATLAB генератора рівномірно розподілених випадкових чисел.

Для коректної роботи алгоритму потрібна камера, що буде робити знімки за короткий проміжок часу, тому було вирішено використовувати модуль камери, який буде керуватися відладочною платою.

Модуль камери OV2640 закріплено на кронштейні, який рухається завдяки пріклиплених до нього двух сервоприводів SG90, уся конструкція керується за допомогою відладочної плати STM32F407.

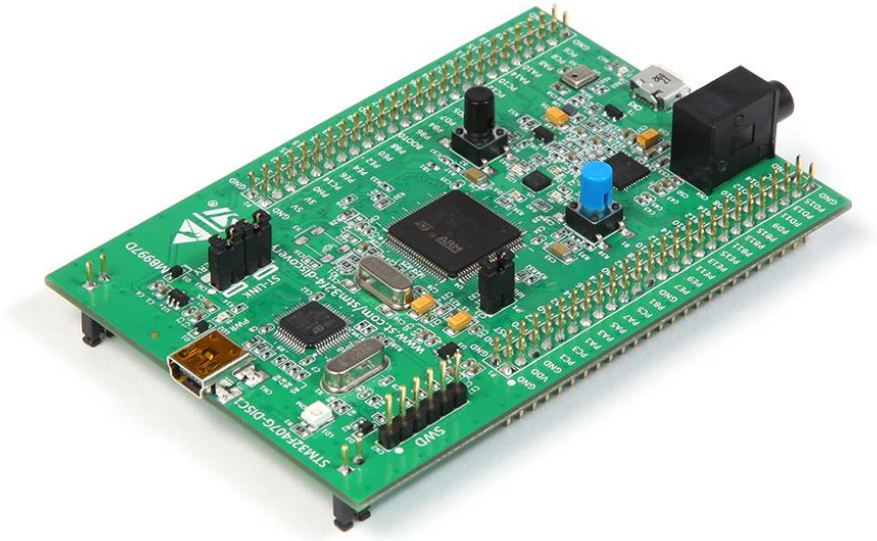


Рис. 3.2. Відладочна плати STM32F407

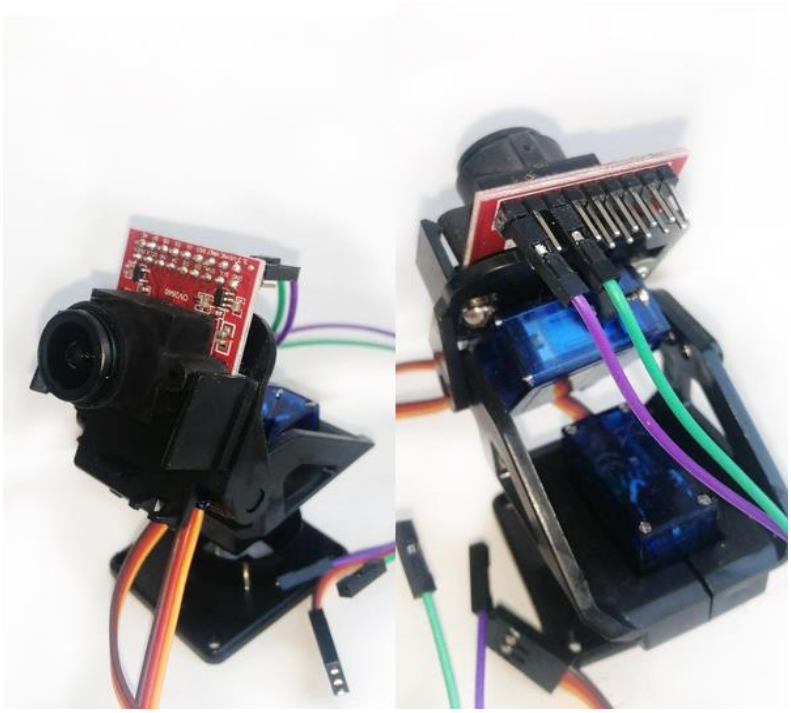


Рис. 3.3 Модуль камери OV2640 на кронштейні з двома сервоприводами SG90

3.2 Тестування отриманих результатів

Існують різні статистичні тести, що створені для перевірки ступеню випадковості послідовностей згенерованих ГІВЧ або ГПВЧ. Ці тести допомагають виявити наявність вразливостей які може мати генератор випадкових чисел. Наприклад, один з базових тестів для випадкових чисел – це перевірка ймовірностей появи 1 чи 0 у послідовності; їх ймовірність повинна бути майже однакова.

Для тестування отриманих результатів був обраний набор тестів NIST 800-22 [31]. Цей набор має п’ятнадцять статистичних тестів для тестування довгої послідовності бінарних чисел генерованих будь-яким апаратним чи програмним забезпеченням. Якщо послідовність не проходить всі статистичні тести вона вважається псевдовипадковою, але якщо послідовність не проходить до 10% статистичних тестів вона є істинно випадковою. Але, ці тести запроваджують лише ймовірнісний підхід до випадкових чисел та не є математичними доведенням того, що бінарні послідовності є істинно випадковими.

Статистичні тести NIST 800-22 та їх опис :

1. Frequency (Monobit) Test: Цей тест дає фундаментальний доказ відсутності випадковості у бінарній послідовності. При непроходженні є велика ймовірність не проходження інших тесті. У тесті підраховується кількість 0 та 1 у всій довгій послідовності. Якщо кількість 0 та 1 не є приблизно однаковою, то тест не буде пройденим.

2. Frequency Test within a Block: Цей тест підраховує кількість одиниць у блоку. Мета цього тесту – з’ясувати, чи кількість 1 у блоку M-бітів приблизно дорівнює половині довжини блоку.

3. Runs Test: Цей тест вимірює загальну кількість runs у послідовності, де run визначається як безперервний потік однакових бітів (0 або 1) наявний у послідовності. Run довжиною n має n однакових бітів.

4. Test for the Longest Run of Ones in a Block: Цей тест підраховує найдовший run із 1 у блоці заданого розміру. Тест не буде пройдено, якщо

найдовший run із 1 у блоці не буде відповідати очікуваному значенні у випадковій послідовності.

5. Binary Matrix Rank Test: Бінарна послідовність розташована у формі рядків і строк матриці. Потім обчислюється ранг матриці для перевірки залежності серед бітових рядків фіксованої довжини вихідної послідовності. Тест не буде пройдено, якщо буде зафіксовано відхиленні у розподіленні рангів, що не відповідає випадковій послідовності.

6. Discrete Fourier Transform (Spectral) Test: Цей тест спостерігає висоту піків у дискретному перетворенні Фур'є. Мета цього тесту – виявити періодичні закономірності, що схожі, у послідовності. Тест не буде пройдено, якщо кількість піків, що перевищують порогове значення, дуже високе.

7. Non-overlapping Template Matching Test: Основна мета цього тесту – виявлення кількості зустрічей заздалегідь заданих неперіодичних шаблонів у блоку бітів визначеного розміру. Якщо шаблон не будо знайдено, то пошук починається з наступної позиції біту у блоці. Якщо шаблон було знайдено, то пошук починається заново у новому блоці, який знаходиться на біт після попереднього. Тест не буде пройдено, якщо було знайдено забагато шаблонів у послідовності.

8. Overlapping Template Matching Test: Цей тест дуже схожий на Non-overlapping Template Matching Test. Відмінність від тест 7 полягає у тому, якщо було знайдено шаблон, новий блок зсовується на 1 біт та пошук починається знову. Тест не буде пройдено, якщо було знайдено забагато шаблонів у послідовності.

9. Maurer's "Universal Statistical" Test: Цей тест фіксує кількість бітів між схожими закономірностями у бінарній послідовності. Мета цього тесту знайти послідовність, яка може бути стиснута без втрати інформації. Тест не буде пройдено, якщо послідовність може бути стиснута [33].

10. Linear Complexity Test: Цей тест обчислює довжину регістру зсуву з лінійним зв'язком (LFSR), який був би потрібен для створення даної послідовності бітів. Випадкова послідовність повинна бути досить складною,

щоб характеризуватися довгим LFSR[31]. Тест не буде пройдено, якщо розмір потрібного LFSR буде дуже малим.

11. Serial Test: Цей тест шукає наявність усіх можливих перекриваючих одна одну закономірностей бітів у бінарній послідовності. Число зустрічей кожної перекриваючої закономірностей повинно бути майже однакове. Тест не буде пройдено, якщо частота перекриваючих закономірностей не є рівномірною.

12. Approximate Entropy Test: Цей тест порівнює усі випадки перекриття шаблонів m-біт шаблонами (m+1)-біт у всій бінарній послідовності. Тест не буде пройдено, якщо частота перекриваючих блоків двох послідовних довжин (m та m+1) не є характеристичною для випадкової послідовності.

13. Cumulative Sums Test: Тест кумулятивних сум розраховується шляхом передачі потоку [0,1] у відповідну [-1,+1] послідовність використовуючи $x_i = 2a_i - 1$; де a_i це оригінальний бітовий шаблон. Кумулятивна сума випадкової ходьби обчислюється з часткових сум в новій послідовності. Тест не буде пройдено, якщо екскурсії по випадковій ходьбі занадто великі або занадто малі щодо очікуваної поведінки випадкових послідовностей.

14. Random Excursions Test: У цьому кумулятивна сума, розраховується шляхом приймання випадкової ходьби, яка починається в якійсь точці, яка вважається місцем походження, і повертається до цієї точці. Кумулятивна сума знаходиться аналогічним чином до тесту 13, сума виводиться після того, як послідовність [0,1] передається у відповідну послідовність [-1,+1]. Цей тест фактично складається з серії восьми тестів, це відображає кількість переходів кожного із станів: -4, -3, -2, -1 та +1, +2, +3, +4 у випадкову ходьбу. Тест не буде пройдено, якщо присутні відхилення у очікуваної кількості відвідувань різних станів випадкову ходьбу.

15. Random Excursions Variant Test: Цей тест визначає загальну кількість відвідувань [34] для конкретних станів у кумулятивну суму випадкової ходьби. У тесті вимірюється відхилення для вісімнадцяти станів [-9, -8, …, -1, +1, +2,

…, +9]. Тест не буде пройдено, якщо тест виявить відхилення від очікуваного значення кількості відвідувань різних станів у випадковій ходьбі.

Статистичні тести NIST потребують велику довжину послідовності, тому була генерована послідовність розміру у порідку 10^5. Результати тестів показано на рис.3.4, ГІВЧ зроблений у даній роботі успішно пройшов усі 15 тестів.

STATISTICAL TEST	P-VALUE (min)	PROPORTION	RESULT
Frequency	0.375313	0.986	Passed
Block frequency	0.674543	0.990	Passed
Runs	0.773405	0.988	Passed
Longest Run	0.087162	0.988	Passed
Rank	0.291091	0.991	Passed
Discrete Fourier transform	0.134355	0.984	Passed
Non-overlapping templates	0.002186	0.986	Passed
Overlapping templates	0.989425	0.990	Passed
Universal	0.705466	0.991	Passed
Linear complexity	0.883171	0.989	Passed
Serial	0.123038	0.995	Passed
Approximate entropy	0.607993	0.993	Passed
Cumulative sums	0.514124	0.991	Passed
Random excursions	0.023140	0.9886	Passed
Random excursions variant	0.015993	0.9869	Passed

Рис. 3.4 Пройдені статистичні тести NIST

ВИСНОВКИ

Мета та задачі, поставлені в цій роботі, були виконані. А саме:

- Були розглянути існуючі генератори істинно випадкових чисел;
- Було обране оптимальне джерело інформаційного шуму;
- Розроблено алгоритм для отримання інформаційного шуму;
- Розроблено програмне забезпечення;
- Проведено тестування програмного забезпечення.

Недоліки використання цифрової камери як джерела випадкових чисел – упереджені закономірності в зображеннях та заблокованих блоків сенсорів. Сира послідовність без подальшої обробки може мати фіксовані шаблони шуму. У даній роботі, представлений алгоритм, який використовує пікселі з всіх трьох кольорових каналів. Алгоритм отримує останні кулька бітів, на які впливають шуми і вони є вільними від упереджень в пейзажі. Заблоковані функції сенсора фіксуються зад допомогою перевертання бітів та перестановки даних у матрицю. Корисна довжина пікселя завжди змінюється для кожного кадру, тому у даний алгоритм забезпечує ефект перетасовки, що призводить до кращої якості випадковості в послідовності [35].

Є кілька переваг цього алгоритму. По перше, алгоритм спирається на Гаусові властивості шумових сигналів, а не модель камери. Він може працювати на більшості існуючих пристрів з цифровою камерою. По-друге, немає ніяких особливих вимог до аксесуарів або умов роботи. По-третє алгоритм безпосередньо генерує істинно випадкові числа, зберігаючи обчислення акуратними та простими у реалізації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Landon Curt Noll, Robert G. Mende, Sanjeev Sisodiya, “Method for Seeding a Pseudo-Random Number Generator with a Cryptographic Hash of a Digitization of a Chaotic System”, US patent 5,732,138, January 29, 1996
2. John Koeter, “What’s an LFSR?”, Texas Instrument, December 1996
3. Benjamin Jun and Paul Kocher, “The Intel Random Number Generator”, Cryptography Research, Inc. White Paper Prepared for Intel Corporation, April 22, 1999
4. Greg Taylor, George Cox “Behind Intel's New Random-Number Generator” IEEE Spectrum, September 2011
5. Greg Taylor, George Cox,“Intel Makes a Digital Coin Tosser for Future Processors”, IEEE Spectrum, September 2011
6. Mitchum, S and R. H. Klenke, “FPGA Based Digital True Random Number Generator”, 2009
7. Mitchum, S.T.; Klenke, R.H., “Divergent path random number generators”, Southeastcon, 2009. SOUTHEASTCON '09. IEEE, March 2009, p.109 – 114.
8. Pierre-Yvan Liardet, “Random Number Generating Circuit and Process” US Patent 6581078, January 4, 2000
9. Katsunori Ikake, “Random Number Generator”, US Patent 7124157, June 7, 2002
10. Kohlbrenner, Paul and Alfke, Peter, “Efficient Shift Registers, LFSR Counters and Long Pseudo-Random Sequence Generators”, Xilinx Publication XAPP052, July 7, 1996
11. Mitchum, S.T.; Klenke, R.H., “Design and Fabrication of a Digitally Synthesized, Digitally Controlled Ring Oscillator”, Proceedings of the Third IASTED International Conference on Circuits, Signals and Systems, October 24-26, 2005.
12. Hector Zenil. Randomness Through Computation: Some Answers, More Questions World Scientific, 2011.
13. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. Journal of Computer and System Sciences, 28(2):270–299, 1984.

14. David B Stephenson, Valentina Pavan, and Roxana Bojariu. Is the north atlantic oscillation a random walk? International Journal of Climatology, 20(1):1–18, 2000.
15. Reuven Y Rubinstein and Dirk P Kroese. Simulation and the Monte Carlo method, volume 707. John Wiley & Sons, 2011.
16. Julia Kempe. Quantum random walks: an introductory overview. Contemporary Physics, 44(4):307–327, 2003.
17. Martin Karplus and J Andrew McCammon. Molecular dynamics simulations of biomolecules. Nature Structural and Molecular Biology, 9(9):646–652, 2002.
18. Jessica D Leuchter, Adam T Green, Julian Gilyard, Cecilia G Rambarat, and Samuel S Cho. Coarse-grained and atomistic md simulations of rna and dna folding. Israel Journal of Chemistry, 2014.
19. Z. Gutterman, B. Pinkas, and T. Reinman. Analysis of the linux random number generator. In Security and Privacy, 2006 IEEE Symposium on, pages 15 pp.–385.
20. George Marsaglia. Random number generators. Journal of Modern Applied Statistical Methods, 2(1), 2003.
21. Anthony J Martino and G Michael Morris. Optical random number generator based on photoevent locations. Applied optics, 30(8):981– 989, 1991.
22. L.C. Noll, R.G. Mende, and S. Sisodiya. Method for seeding a pseudorandom number generator with a cryptographic hash of a digitization of a chaotic sys 1998.
23. Bruno Sanguinetti, Anthony Martin, Hugo Zbinden, and Nicolas Gisin. Quantum random number generation on a mobile phone. arXiv preprint arXiv:1405.0435, 2014.
24. Xuping Zhang, Li Qi, Zhiqiang Tang, and Yixin Zhang. Portable true random number generator for personal encryption application based on smartphone camera. Electronics Letters, 50(24):1841–1843, 2014
25. Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, and Elaine Barker. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, DTIC Document, 2001.

26. Y Reibel, M Jung, M Bouhifd, B Cunin, and C Draman. Ccd or cmos camera noise characterisation. The European Physical Journal Applied Physics, 21(01):75–80, 2003.
27. David Dussault and Paul Hoess. Noise performance comparison of iccd with ccd and emccd cameras. In Optical Science and Technology, the SPIE 49th Annual Meeting, pages 195–204. International Society for Optics and Photonics.
28. Angelo Bosco, Arcangelo Bruna, Giuseppe Messina, and Giuseppe Spampinato. Fast method for noise level estimation and integrated noise reduction. Consumer Electronics, IEEE Transactions on, 51(3):1028– 1033, 2005.
29. Kai Hormann and Alexander Agathos. The point in polygon problem for arbitrary polygons. Computational Geometry, 20(3):131–144, 2001.
30. Gonzalez R. C., & Woods R. E. (2002) "Digital Image Processing," second ed., Prentice Hall, Englewood, Cliffs, NJ.
31. Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, San Vo, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications" National Institute of Standards and Technology publication 800-22, revised: April 2010
32. Boyat, A. and Joshi, B. K. (2013) "Image Denoising using Wavelet Transform and Median Filtering', IEEE Nirma University International Conference on Engineering," Ahemdabad.
33. Koli M. & Balaji S., (2013) "Literature survey on impulse noise reduction," Signal & Image Processing : An International Journal (SIPIJ) Vol.4, No.5.
34. Zhang L., Dong W., Zhang D. & Shi G. (2010) "Two stage denoising by principal component analysis with local pixel grouping," Elsevier Pattern Recognition, Vol. 43, Issue 4, pp. 1531-1549.
35. Schowengerdt R. A. (1983) "Techniques for Image Processing and classifications in Remote Sensing," First Edition Academic Press.