УДК 551.581.1
О.С. Лімонов, *к.т.н.,* **Б.В. Перелигін**, *к.т.н.*, **Т.М. Пустовіт,** *інж.*
Одеський державний екологічний університет

РЕАЛІЗАЦІЯ ФІЛЬТРА КАЛМАНА ДЛЯ ВІДНОВЛЕННЯ ІЗОГІПС ПРИ ПОБУДОВІ КАРТ ЦИФРОВОГО РЕЛЬЄФУ

В статті досліджений варіант реалізації фільтра Калмана для відновлення ліній рівних висот - ізогіпс.
Ключові слова: *фільтр Калмана, ізогіпси, карти цифрового рельєфу, Open CV, алгоритм відновлення.*

Вступ. Використання фільтра Калмана для відновлення ізогіпс в відомій літературі розглянуто недостатньо.

Матеріали і методи дослідження: досліджується реалізація фільтра Калмана з алгоритмом відновлення ізогіпсів карти цифрового рельєфу.

Мета статті: дослідити використання фільтра Калмана для відновлення ізогіпс при побудові карти цифрового рельєфу.

Виклад основного матеріалу. ***Реалізація фільтра Калмана через бібліотеку OpenCV.*** OpenCV (англ. Open Source Computer Vision Library, бібліотека комп'ютерного зору з відкритим початковим кодом) – бібліотека алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Реалізована на C/C++, також розробляється для Python, Ruby, Matlab, Lua і інших мов. Може вільно використовуватися в академічних і комерційних цілях ─ поширюється в умовах ліцензії BSD. Фактично, OpenCV – це набір типів даних, функцій і класів для обробки зображень алгоритмами комп'ютерного зору.

Основні модулі бібліотеки:

1) cxcore – ядро містить базові структури даних і алгоритми:
   - базові операції над багатовимірними числовими масивами;
   - матрична алгебра, математичні функції, генератори випадкових чисел;
   - запис/відновлення структур даних, відновлення/запис XML;
   - базові функції двовимірних (2D) графіків;
2) CV – модуль обробки зображень і комп'ютерного зору:
   - базові операції над зображеннями (фільтрація, геометричні перетворення, перетворення колірних просторів і т. п.);
   - аналіз зображень (вибір відмітних ознак, морфологія, пошук контурів, гістограми);
   - аналіз руху, стеження за об'єктами;
   - виявлення об'єктів;
   - калібрування камер, елементи відновлення просторової структури;
3) HighGUI – модуль для введення/виведення зображень і відео, створення призначеного для користувача інтерфейсу:
   - захоплення відео з камер і з відео файлів, читання/запис статичних зображень;
   - функції для організації простого GUI (усі демо застосування використовують HighGUI);
4) Cvaux – експериментальні і застарілі функції:
   - просторовий зір: стерео калібрування, саме калібрування;
   - пошук стерео-відповідності, кліки в графах;
   - знаходження і опис рис об'єктів;

5) CvCam – захоплення відео дозволяє здійснювати захоплення відео з цифрових відео-камер.

Для реалізації програми було використане середовище розробки Microsoft Visual Studio 2008.

Морфологічні перетворення зображення. Основні морфологічні перетворення вхідного зображення реалізовані з використанням операцій звуження (Erode) та розширення (Dilate) [1,2]. OpenCV реалізує їх двома функціями:

1) cvErode() – розмиває (операція звуження) зображення з використанням фільтра (ядра) один або кілька разів. Зображення формується з локальних мінімумів – тобто будуть збільшуватися темні області);
2) cvDilate() – розтягує (операція розширення) зображення з використанням фільтра (ядра) один або кілька разів. Зображення формується з локальних максимумів – тобто будуть збільшуватися світлі області).

У cvErode () ядро накладається на зображення і на місці якоря (центр ядра) залишається мінімальне значення, що лежить під ядром (у разі cvDilate () – навпаки – максимальне).

Ерозія (розмивання/звуження) зображення використовується для позбавлення від випадкових вкраплень на зображенні. Ідея полягає в тому, що вкраплення при розмиванні усунуться, тоді як великі і відповідно більш візуально-значущі регіони залишаються.

Розтягування (розширення) так само усуває шум і сприяє об'єднанню областей зображення, які були розділені шумом, тінями. Застосування невеликого розтягування має сплавити ці області в одну.

Морфологічні операції застосовуємо над двійковими зображеннями, які виходять після порогового перетворення і є власно зображеннями ізогіпс.

Створення ядра довільної форми здійснюється функцією cvCreateStructuringElementEx ().

Форму ядра визначаємо через values, яка містить маску, що визначає які сусідні пікселі повинні враховуватися. Values – показник на масив, де ненульові елементи визначають значущі пікселі. Якщо $values \neq NULL$ всі елементи вважаються ненульовими.

Вибираючи різну структуру ядра, можна вирішувати різні задачі обробки зображень:

- придушення шумів;
- виділення меж об'єкта;
- виділення скелетону об'єкта.

Знаходження розірваних контурів. Знаходження розірваних контурів потрібно для того, щоб знайти кінцеві точки розірваних ліній. Для цього потрібна функція cvStartFindContours(). За основу функції узят детектор кордонів Кенні.

Кордони (межі) – це такі криві на зображенні, уздовж яких відбувається різка зміна яскравості або інших видів неоднорідностей.

Простіше кажучи, край – це різкий перехід/зміна яскравості.

Причини виникнення кордонів:

- зміна освітленості;
- зміна кольору;
- зміна глибини сцени (орієнтації поверхні).

Виходить, що границі відображають важливі особливості зображення, і тому метою перетворення зображення в набір кривих є:

- виділення істотних характеристик зображення;
- скорочення обсягу інформації для подальшого аналізу.

В роботі використовуємо метод виділення кордонів за допомогою детектора кордонів Кенні.

Кроки детектора:

– забрати шум і зайві деталі з зображення;
– розрахувати градієнт зображення;
– зробити края тонкими (edge thinning);
– зв'язати края в контур (edge linking).

Детектор використовує фільтр на основі першої похідної від гауссіану. Оскільки він сприйнятний до шумів, краще не застосовувати даний метод на необроблених зображеннях.

Кордони на зображенні можуть перебувати в різних напрямках, тому алгоритм Кенні використовує чотири фільтра для виявлення горизонтальних, вертикальних і діагональних кордонів. Скористаємось оператором виявлення кордонів і отримаємо значення для першої похідної в горизонтальному напрямку (Gy) і вертикальному напрямку (Gx).

Аналіз відповідних кінців контурів. Аналіз відповідних кінців контурів виконується методом знаходження Евклідової відстані між кінцевими точками.

Евклідова відстань - найбільш часто використовувана міра відстані. Вона не є географічною відстанню в багатовимірному просторі і обчислюється таким чином

$$P = \sqrt{\sum_{i=1}^{N} \left(A_j - B_j \right)^2}, \qquad (1)$$

де P – відстань між об'єктами A і B; A_j – значення i - ознаки об'єкта A; B_j – значення i - ознаки об'єкта B.

Природна, з геометричної точки зору, евклідова міра відстані може виявитися безглуздою, якщо ознаки виміряні в різних одиницях. Щоб виправити становище, вдаються до нормування кожної ознаки. Застосування евклідової відстані виправдано в випадках, коли ознаки об'єкта однорідні з фізичним змістом і однаково важливі для класифікації; признаковий простір збігається з геометричним простором.

Структурна реалізація фільтра Калмана. Реалізація фільтра Калмана виконана за наступною схемою (рис.1).

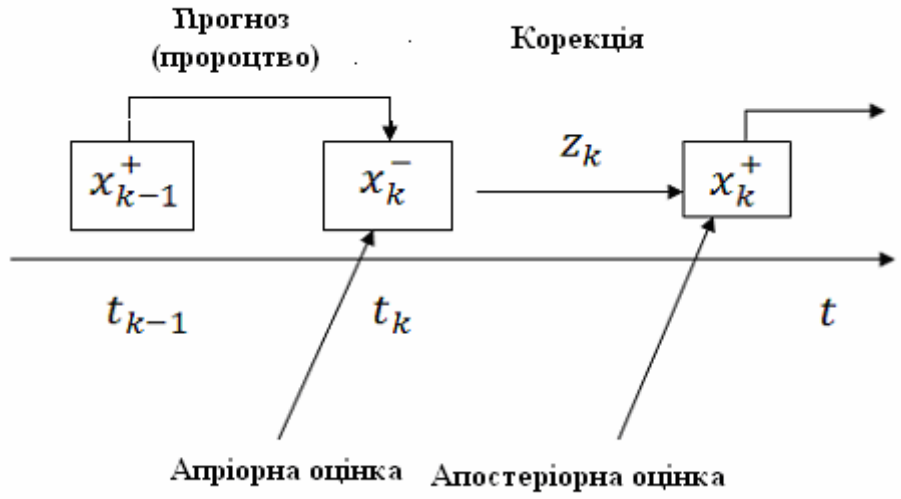


Рис. 1 – Реалізація фільтра Калмана.

Фільтр Калмана працює за системою прогноз-корекція. Припустимо, що в момент часу t_{k-1} отримана оцінка вектора стану системи x^{+}_{k-1} і тепер хочемо отримати оцінку в момент t_k. Для цього будуємо прогноз оцінки x^{-}_{k}, базуючись на x^{+}_{k-1}, отримуємо вимірювання z_k і далі коригуємо оцінку в момент t_k, базуючись на прогнозі і вимірах, і отримуємо остаточну оцінку вектора стану $x^{+}_{k} \cdot x^{-}_{k}$ називають апріорною оцінкою, x^{+}_{k} називають апостеріорною оцінкою.

В OpenCV фільтр Калмана функціонує наступним чином. Для зберігання стану фільтра Калмана використовується структура CvKalman. Фільтр створюється за допомогою функції cvCreateKalman(). Прогнозування відбувається за допомогую функції cvKalmanPredict(), корекція – функцією cvKalmanCorrect(). Ініціалізація відбувається cvReleaseKalman(). Ця структура використовується для стандартного фільтру Калмана.

Практична реалізація відновлення ізогіпсів шляхом калманівської фільтрації. В якості вхідного зображення для фільтрації було взяте зображення вже відновлене статистичним методом (з використанням об'єктної бази даних з навчанням) (рис.2).

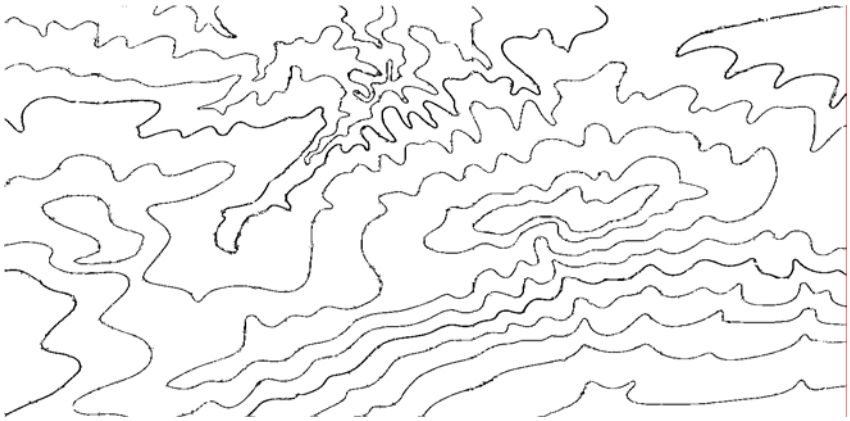


Рис. 2 – Вхідне зображення.

При детальному розгляді видно, що не всі розриви усунені (рис.3).

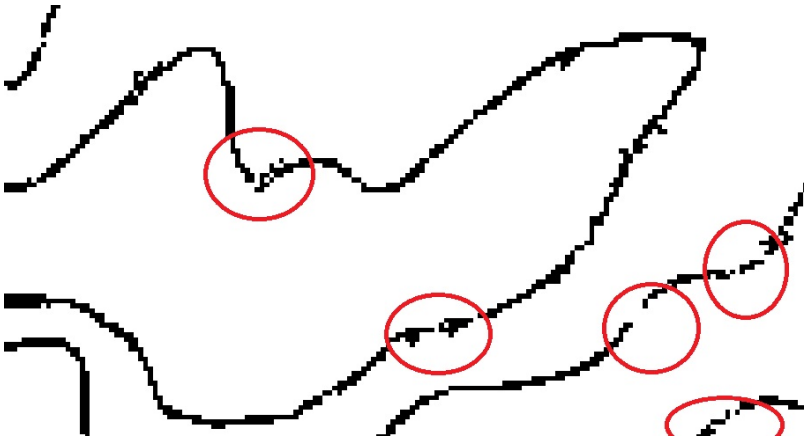


Рис. 3 – Детальний розгляд зображення.

Для аналізу візьмемо частину зображення з найбільшою кількістю розривів (рис.4). Детальний розгляд (рис. 5).

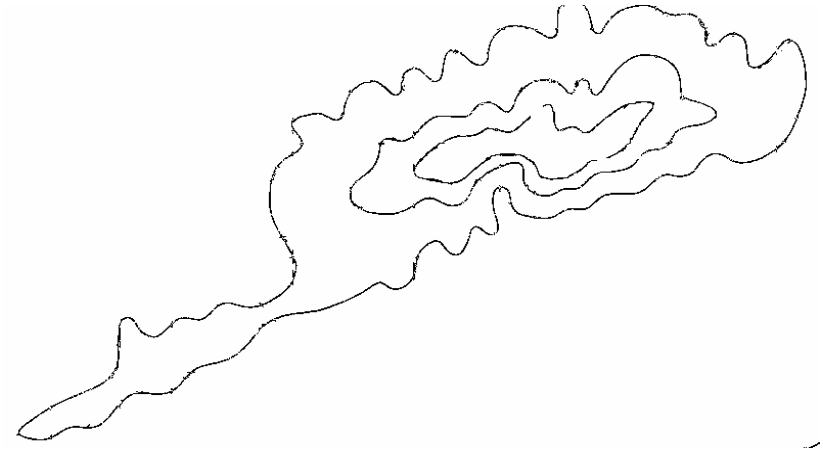


Рис. 4 – Частина зображення для фільтрації.

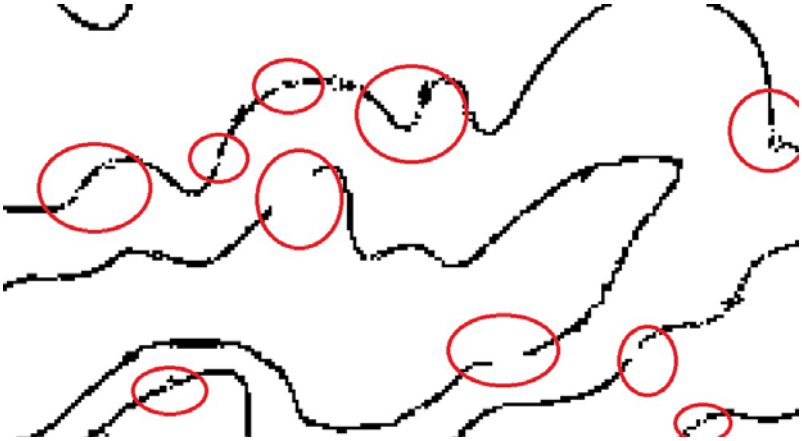


Рис. 5 – Детальний розляд частини зображення.

Застосуємо алгоритм фільтрації на зображенні поетапно.

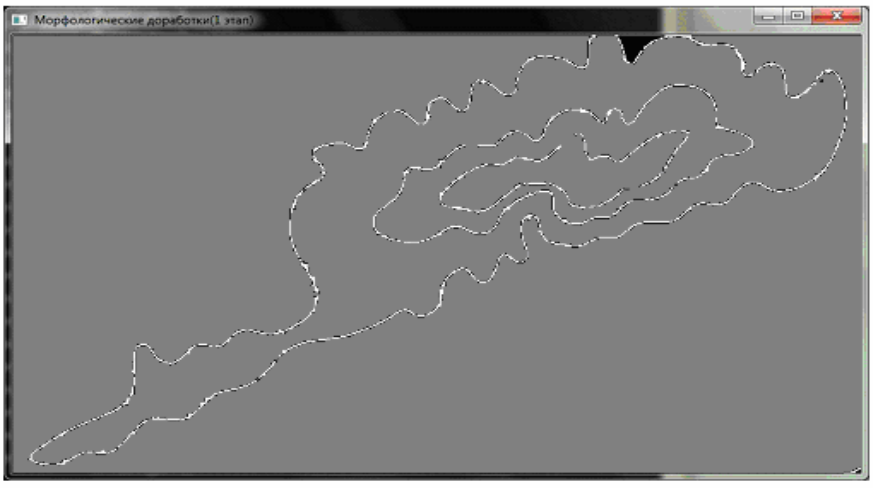


Рис. 6 – Морфологічне доопрацювання зображення.

З рис. 6 видно, що зображення було частково позбавлено шумів та випадкових вкраплень.

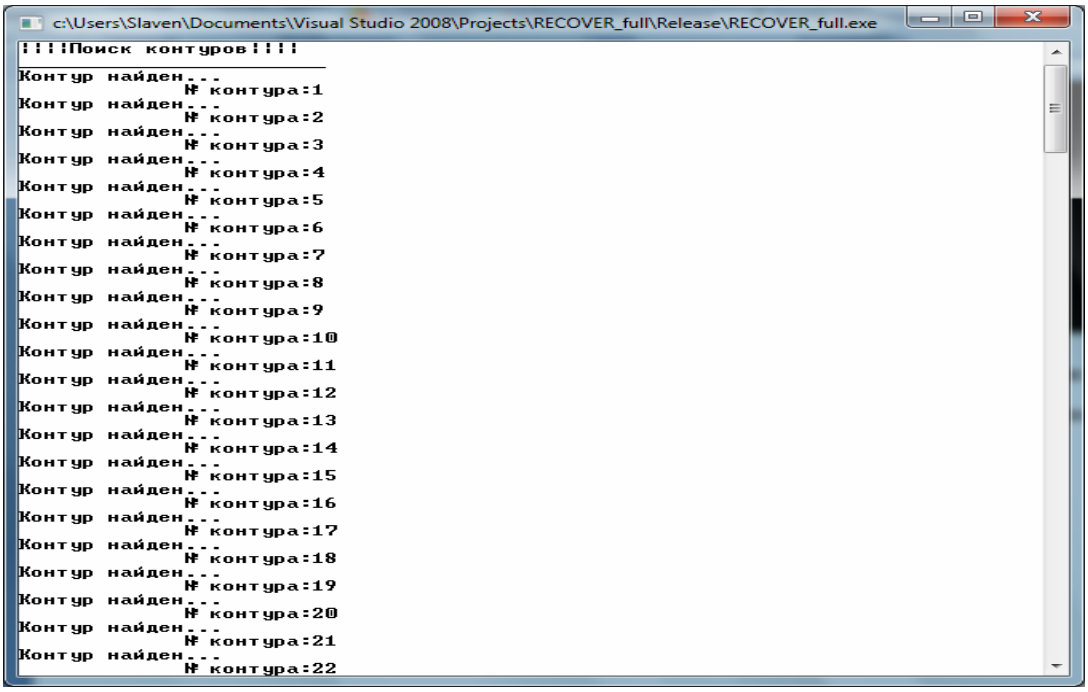


Рис. 7 – Знаходження розірваних контурів.

З рис.7 видно, що в досліджуваному фрагменті є 22 розірваних контури. Застосуємо фільтр Калмана в комплексі з морфологічними перетвореннями. Результати наведені на рис. 8.

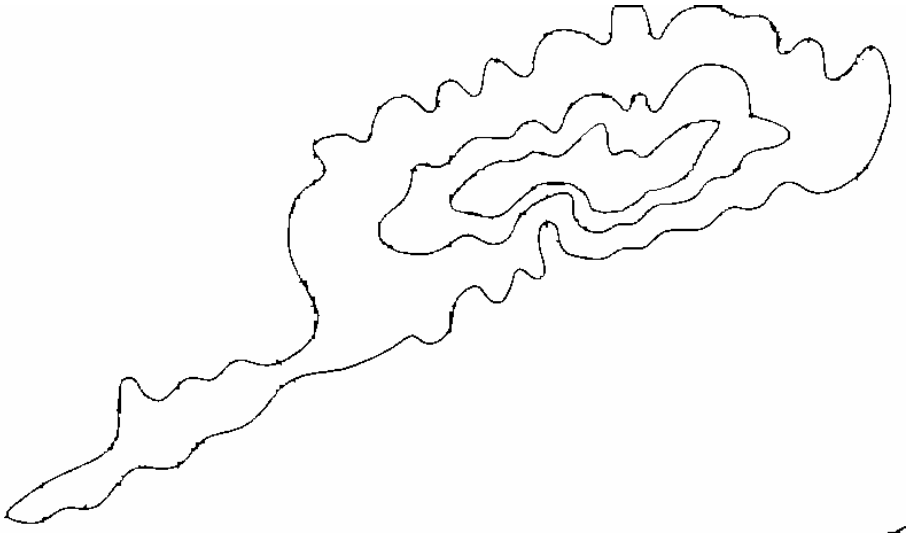


Рис. 8 – Результат калманівської фільтрації.

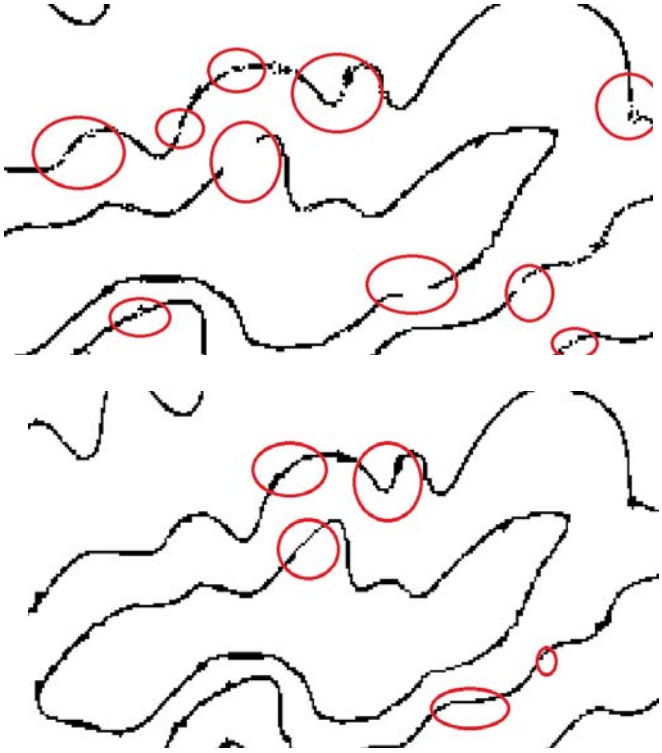
Роздивимося результат фільтрації детально (рис.9). З рисунку видно, що в результаті обробки вхідного зображення розриви були усунуті.

Рис. 9 – Детальний результат фільтрації.

Дослідження ефективності відновлення ізогіпс. Після проведеного дослідження наявно видно, що в результаті відновлення з використанням об'єктної бази даних з навчанням були усунуті далеко не всі розриви ізогіпсів. Зіставимо результати відновлення ізогіпсів з використанням об'єктної база даних з навчанням і з використанням калманівської фільтрації. Результат порівняння наведений на рис. 10.

а)

б)



а) відновлення з використанням бази даних, б) з використанням фільтра Калмана.

Рис. 10 – Порівняння ефективності відновлення.

З графіка видно, що відновлення ізогіпс за допомогою калманівської фільтрації є значно ефективнішим способом, ніж відновлення з використанням бази даних, що навчається. Це пояснюється тим, що не треба витрачати час на оновлення бази даних. Тому процес відновлення можна зробити повністю автоматизованим. Мінуси калманівської фільтрації полягають в тому, що її ефективність напряму залежить від якості вхідного зображення. Цю залежність можна побачити на рис. 12.

З отриманих даних побудуємо графік ефективності.

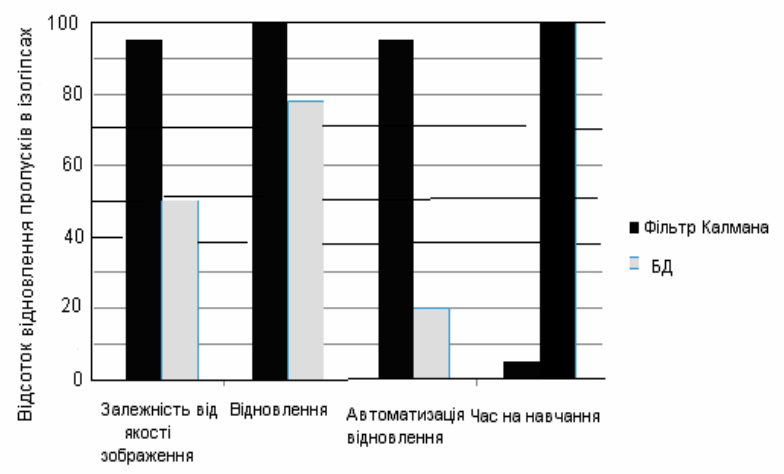


Рис. 11 – Оцінка ефективності відновлення.

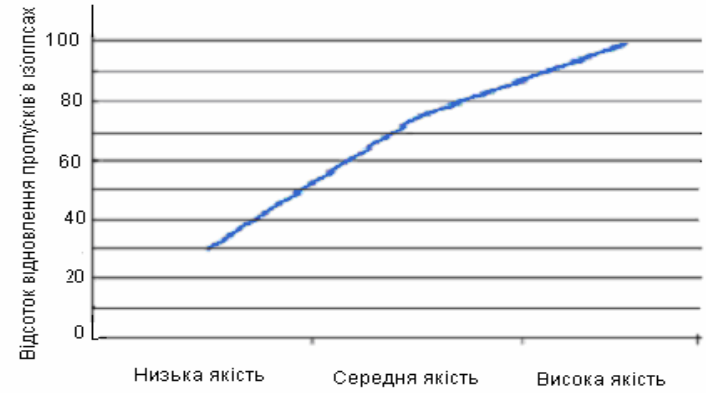


Рис. 12 – Залежність калманівської фільтрації від якості вхідного зображення.

Висновки. Дослідження показали, що використання фільтра Калмана значно спрощує процес відновлення ліній рівних висот – ізогіпс та автоматизує його.

Список літератури

1. *Bradsky G., Kaehler A.* Learning OpenCV - O'Reilly, 2008. - С. 1.

2. OpenCV шаг за шагом. Обработка изображения – морфологические преобразования. [Электронный ресурс] – Режим доступу: http://robocraft.ru/blog/computervision/319.html.

Реализация фильтра Калмана для восстановления изогипс при построении карт цифрового рельефа. Лимонов О.С., Перелыгин Б.В., Пустовит Т.М.

В статье исследован вариант реализации фильтра Калмана для восстановления равных высот - изогипс.

Ключевые слова: *фильтр Калмана, изогипсы, карты цифрового рельефа, OpenC, алгоритм восстановления.*

Calman filter for reconstruction for isohypses reconstruction in maps of digital relief creation. Limonov A.C., Pereligin B.V., Pustovit T.M.

In article abilities of Calman filter realization for *isohypses reconstruction in maps of digital relief creation are investigated.*

Key words: *Calman filter, isohypses, maps of digital relief, OpenCV, reconstruction algorithm.*