

DOI <https://doi.org/10.15407/csc.2022.02.003>  
УДК 004.932

**М.К. ШИЛО**, студент магістратури, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», 03056, м. Київ, просп. Перемоги, 37, Україна, ORCID: <https://orcid.org/0000-0001-5762-2546> maksym.shylo.work@gmail.com

**В.М. КРИГІН**, аспірант, м.н.с. Міжнародний науково-навчальний центр інформаційних технологій та систем НАН та МОН України, 03187, м. Київ, просп. Академіка Глушкова, 40, Україна, ORCID: <https://orcid.org/0000-0002-9000-1685>, valeriy.krygin@gmail.com

## ПЕРЕТВОРЕННЯ ВІДЕОЗАПИСУ З ДОШКИ НА СЛАЙД-ШОУ

*Створено математичну модель перетворення відеозапису з дошки на слайд-шоу. На її основі побудовано інформаційну технологію. Здійснено аналіз схожих робіт. Використано оцінку гомографії та алгоритм максимального потоку Бойкова-Колмогорова як інструменти для одержання панорамних знімків дошки та видалення викладача з відео. Побудовано бінаризовані та знесумлені панорами.*

**Ключові слова:** інтелектуальне оброблення відео, оброблення зображень, стабілізація відео.

### Вступ

Задача створення конспекту лекцій не є новою, а з масовим переходом навчання у віддалений режим актуалізувалася проблема створення якісних лекційних матеріалів, зокрема слайдів, які містять стислу суть занять. У статті описано алгоритм напівавтоматичного створення слайдів на основі відеозапису лекції [1].

За допомогою наведеного алгоритму можна обробляти відео в режимі реального часу, адже кожна його ітерація потребує лише поточний на попередній кадри (рис. 1). Між кожною парою сусідніх кадрів відео розраховується бінарна маска областей, де потенційно є рухомі об'єкти. Ті частини зображення, де не було помічено рух, зберігаються у зображення, яке ми називаємо тут панорамою, тому що це зображення може розширюватися, коли камера рухається навмисно або хитається через небажаний вплив на неї. Отриману панораму ми порівнюємо з тими панорамами, які були отри-

мані на попередніх кроках, і створюємо новий слайд, якщо було помічено достатню кількість змін.

Інформаційна технологія, що реалізує цей алгоритм, працює в напівавтоматичному режимі — вона потребує від користувача введення деяких параметрів. Ці параметри визначають крок, з яким треба обирати кадри для визначення руху, ступінь згладжування маски рухомих об'єктів під час її пошуку, крок між панорамами для перевірки необхідності створення нового слайду та кількість необхідних змін між двома панорамами для створення нового слайду.



Рис. 1. Загальна схема створення слайдів з відео

## Аналіз попередніх робіт

Задача створення слайдів з відеозаписів або фотографій дошки є актуальною вже не перший десяток років, про що свідчать роботи, описані у цьому розділі.

У роботі [2] описується система розпізнавання країв дошки, визначення її позиції та створення слайдів із фотографій, що містять частини дошки, зняті з різних ракурсів. Алгоритм у [3] обробляє відеозапис, виявляє викладача та видаляє його з відео. Автори наголошують на тому, що їхня програма також може локалізувати дошки, які поділено на частини. Цей метод добре працює з різними розмірами відео та умовами зйомки, проте не створює слайдів. У [4] представлено алгоритм, який аналізує відео, виявляє написи та об'єкти на передньому плані, проте не сказано, як поведе себе алгоритм, якщо камера рухатиметься. Робота [5] є найбільш схожою на нашу — в ній представлено метод, який створює слайди з відеозаписів лекцій, проте невідомо, як він поводить себе в умовах, коли камера рухається.

## Вхід алгоритму

Визначмо вхідні дані алгоритму та деякі базові терміни, що використовуються в роботі. Дошкою називатимемо будь-яку пласку поверхню, яку користувач цілеспрямовано знімає. Це може бути крейдяна дошка, маркерна дошка, стіна тощо.

Записами називатимемо ті місця дошки, де відбулася зміна кольору, що закріпилася на відносно тривалий час. Наприклад, якщо в області зйомки камери пройшов студент, це не є зміною кольору на тривалий час, але якщо викладач повісив папірець на дошку і залишив його там, це вважається записом, бо папірець перебуває там протягом певного часу і розташований приблизно у площині дошки.

Кадром під номером  $i$  шириною  $w$  і висотою  $h$  називаємо відображення з множини  $P = \{1, \dots, w\} \times \{1, \dots, h\}$  координат пікселів у скінченну множину  $C \subset \mathbb{R}$  можливих рівнів яскравості (інтенсивності) пікселів. Відео  $F$  довжиною  $T$  є послідовністю  $(F^i : i = \overline{1, T})$  кадрів

$F^i : P \rightarrow C$ . Той факт, що піксель з координатою  $p \in P$  на кадрі  $F^i$  має інтенсивність  $c \in C$ , позначатимемо  $F_p^i = c$ .

На вхід алгоритму подається відео  $F$ . Необов'язково слід мати одразу всі  $T$  кадри відео — алгоритму достатньо поступового надходження кадрів (кадр з номером  $i < j$  повинен прийти раніше, ніж кадр  $j$ ). В область огляду камери має потрапляти дошка або її частина. Наведений алгоритм не розраховано на відео, що містить декілька дошок, які перебувають не в одній пласкій площині. Камера, що знімає це відео, може рухатися, проте чим довше вона є нерухомою, тим краще: алгоритм не працюватиме, якщо камера рухається постійно. Дошка на відео може перекриватися сторонніми об'єктами, проте бажано, щоб ці об'єкти були рухливими, аби алгоритм виявлення рухомих об'єктів їх помічав.

## Побудова маски рухомих об'єктів

Під час лекцій часто виникають такі ситуації, коли викладач затуляє собою частину дошки з написами — наприклад, для запису нового матеріалу або щоб видалити старі написи з дошки. Іноді студенти просять викладача відійти від дошки, щоб переписати з неї те, що з'явилося на ній лише хвилину тому, проте наша програма дозволяє побачити частину дошки, яку затулив викладач, якщо перед цим перекритий сегмент було добре видно протягом вказаного користувачем часу. Для вирішення цієї задачі ми будемо маску рухомих об'єктів, щоб потім їх видалити. Рухомими об'єктами можуть бути викладач, студенти, а також записи, які щойно з'явилися.

**Маска.** Маскою рухомих об'єктів кадру  $F^i$  називатимемо бінарне зображення  $B^i : P \rightarrow \{0, 1\}$ , де тим пікселям, у яких на відповідному кадрі  $F^i$  було помічено рух, відповідає одиниця, а іншим відповідає нуль. Для обраного користувачем кроку  $s > 0$  на двох кадрах  $F^i$  і  $F^{i+s}$  рухомі об'єкти являють собою підмножину пікселів, колір яких було змінено більше, ніж на певне значення, з урахуванням зміни кольорів у су-

сідніх пікселях. Тобто, якщо рухомий об'єкт складається з одного пікселя, його рух може бути проігноровано залежно від обраних користувачем налаштувань, про які йдеться далі; аналогічно, якщо рухомий об'єкт на кадрі містить нерухомі «дірки» (пікселі, де колір не змінився), вони можуть вважатися частиною рухомого об'єкта. У цьому розділі ми користуємося алгоритмом [6], який використовує алгоритм Бойкова-Колмогорова [7] для пошуку мінімального розрізу на графі, що дозволяє знаходити маску рухомих об'єктів і може налаштовувати міру взаємодії сусідніх пікселів. Для розв'язку цієї задачі можна використовувати алгоритми пошуку оптичного потоку [8, 9], проте нам не вдалося отримати задовільні результати та прийнятну швидкодію за допомогою знайдених рішень. Наразі ми перебуваємо в пошуку такого алгоритму.

Для того, щоб видалити всі рухомі об'єкти, які можуть перекривати дошку, ми беремо кадри  $F^i$  та  $F^{i+s}$  (рис. 2, а та б) та для кожної пари кольорів пікселів  $p$  з однаковими координатами знаходимо модуль  $D_p^i = |F_p^i - F_p^{i+s}|$  різниці інтенсивності (рис. 2, в). Зображення  $D^i$  подаємо на вхід алгоритму Бойкова-Колмогорова.

Треба знайти маску  $B^i : P \rightarrow \{0, 1\}$  рухомих об'єктів для кадрів  $F^i$  і  $F^{i+s}$ . Введемо функції

$$q_p(B_p^i) = \begin{cases} \alpha D_p^i, & \text{якщо } B_p^i = 0, \\ 255 - D_p^i, & \text{якщо } B_p^i = 1, \end{cases} \quad (1)$$

$$g(B_p^i, B_{p'}^i) = \beta |B_p^i - B_{p'}^i|, \quad (2)$$

де  $\alpha$  та  $\beta$  — параметри згладжування маски, що задаються користувачем інформаційної технології. Позначмо множину  $\Gamma \subset P^2$  сусідніх пікселів. У цій роботі сусідніми до пікселя  $p \in P$  вважаються пікселі з множини  $\{(p_{x+1}, p_y), (p_x, p_{y+1})\} \cap P$ . Сформулюймо пошук маски  $B^i$  у вигляді задачі мінімізації виразу

$$E(B_p^i) = \sum_{p \in P} q_p(B_p^i) + \sum_{(p, p') \in \Gamma} g(B_p^i, B_{p'}^i). \quad (3)$$

На виході отримуємо маску  $B_p^i$  рухомих об'єктів (рис. 2, з). Її ми використовуємо, щоб не переносити на фінальне зображення ті пікселі, на яких було помічено рух, адже зміна

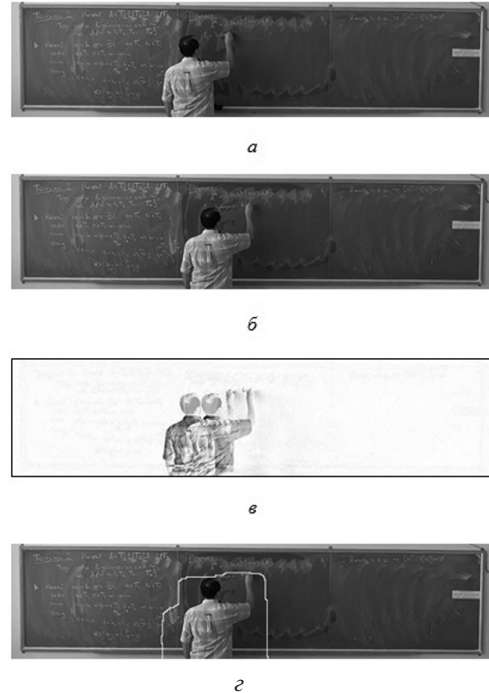


Рис.2. Процес створення маски: а — попередній кадр; б — поточний кадр; в — інвертована різниця попереднього і поточного кадрів; з — маска рухомих об'єктів

яскравості пікселя виникає не лише під час створення напису, а й під час тимчасового затуляння дошки.

**Обробка ключових точок з маскою.** Ключові точки (рис.3,а), що дає нам дескриптор ознак SIFT [10], ми використовуємо для оцінки гомографії, але також видаляємо ті ключові точки, які потрапляють в область маски рухомих об'єктів. У такий спосіб ми зменшуємо шанси того, що замість дошки алгоритм "зачепиться" за одяг викладача або інші рухомі об'єкти. На рис. 3,б можна помітити, що є точки, які насправді не належать рухомих об'єктам кадру  $F^{i+s}$ , але були видалені. Це відбулося тому, що такі точки відповідають точкам з кадру  $F^i$ , які, своєю чергою, лежать у межах маски рухомих об'єктів.

## Пошук відповідності між кадрами

Рух камери не є рідкістю для відеозаписів лекцій. Камера може здригатися, коли її прикріп-

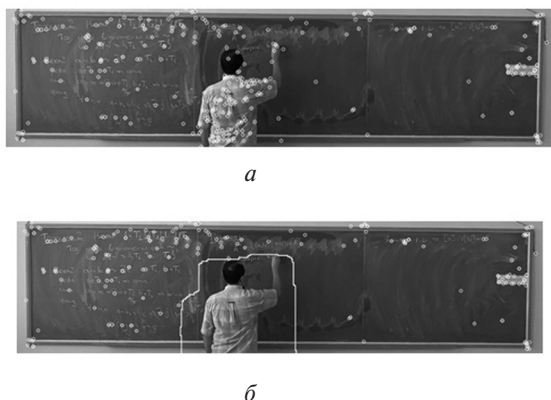


Рис. 3. Результат застосування маски для відповідних точок на кадрі  $F^{i+s}$ : *a* — ключові точки на кадрі; *б* — відповідні ключові точки, що залишились після видалення тих точок, що попали в область маски рухомих об'єктів

лено до столу, де студенти записують лекцію, або від вібрацій підлоги, коли викладач ходить. Камеру може рухати оператор для того, щоб сфокусувати увагу глядачів на певному сегменті дошки. У цьому розділі описано оцінку гомографії за допомогою дескриптора ознак *SIFT*, як одного з кроків подолання згаданих проблем.

Оскільки дошка вважається пласкою поверхнею, побудувати відповідність між точками дошки на різних зображеннях можна за допомогою гомографії [11]. Рухи камери під час лекції можуть змінюватися від незначних субпіксельних зсувів до зміщень, внаслідок яких видимою стає частина дошки, яка доти була прихованою. Наша мета — зробити відео, де камера виглядає статичною, а сегменти дошки поступово стають видимими.

Дано два кадри  $F^i$  та  $F^{i+s}$ . Нехай ми маємо набір  $\{(p_j^i, p_j^{i+s}) : j = \overline{1, 4}\}$  пар координат відповідних точок, де  $p_j^i \in R^2 \times \{1\}$  — координата пікселя  $p$  на кадрі номер  $i$ ,  $p_j^{i+s} \in R^2 \times \{1\}$  — відповідна їй координата на кадрі номер  $i+s$ . Відповідними точками є ті, що є проєкціями однієї і тієї ж точки у просторі. Якщо ці точки є точками пласкої дошки, вони пов'язані між собою відношенням

$$H_i \cdot p_j^{i+s} = c \cdot p_j^i, \quad \forall i = \overline{1, 4}. \quad (4)$$

Рівняння (4) можна легко розв'язати відносно невідомої матриці гомографії  $H^i$ . Цю матрицю можна використовувати для компенсації руху камери — ми застосуємо її до координат пікселів кадру номер  $i+s$ , після чого точки зображення, отриманого внаслідок перетворення, мають ті ж координати, що й відповідні їм точки на кадрі під номером  $i$ . Якщо отримані координати не є цілочисельними, їх можна округлити, що не матиме значного негативного впливу на якість результату.

Ми можемо використовувати дескриптори ознак, такі як *SIFT*, *SURF* [12] або *ORB* [13], щоб знайти ключові точки (*feature points*). Ми обрали *SIFT*, оскільки під час експериментів він надав візуально кращі результати, ніж інші алгоритми. Для оцінки гомографії використовували *RANSAC* [14].

Позначмо множину  $M^i$  усіх знайдених відповідних точок між кадрами  $F^i$  і  $F^{i+s}$ . Користувач вводить рівень  $\varepsilon > 0$  дозволеної похибки розрахунків — що менше, то краще, проте тим довше працюватиме алгоритм пошуку гомографії. Проблема, яку ми хочемо розв'язати, має вигляд

$$\sum_{(p^i, p^{i+s}) \in M^i} \left[ \left\| \frac{H^i \cdot p^{i+s}}{(H^i \cdot p^{i+s})_z} - p^i \right\| < \varepsilon \right] \rightarrow \max_{\substack{H \in R^{3 \times 3} \\ \det H \neq 0}}, \quad (5)$$

Ця задача є дуже складною. Загальний підхід полягає у виборі чотирьох випадкових пар з  $M^i$ , розв'язку задачі (4) та перевірки суми (5). Повторюємо процедуру скільки завгодно разів (що більше, то краще, проте конкретніших рекомендацій немає), і обираємо гомографію, за якої виходить найбільша кількість точок, що потрапили в  $\varepsilon$ -окіл відповідних їм точок іншого зображення.

## Створення панорами

Часто бувають ситуації, коли для запису лекції використовується камера з невисокою роздільною здатністю, через що виникає необхідність рухати камеру, тому написи, які були на дошці, перестають бути видимими для глядачів, які не встигли записати шойно представлений матеріал. Рішенням цієї проблеми та

проблем, пов'язаних із тремтінням камери, є побудова панорами (рис. 4), яка створюється шляхом склеювання та накладання кадрів з відео.

Панорамою  $W^i$  називатимемо відображення  $W^i: P_W^i \rightarrow C$  з множини  $P_W^i = \{1, \dots, w^i\} \times \{1, \dots, h^i\}$  пікселів у множину  $C \subset \mathbb{R}$  інтенсивності. Зауважмо, що в кожній панорамі може бути своя ширина  $w^i \geq w$  та висота  $h^i \geq h$ .

### Алгоритм: Створення панорами

**Вхід:** Два кадри  $F^i$  та  $F^{i+s}$ , поточна панорама  $W^i, W^1 = F^1$ .

**Вихід:** Панорама  $W^{i+s}$ .

#### Кроки:

1. Знаходимо набір  $M^i$  пар відповідних пікселів між кадрами  $F^i$  і  $F^{i+s}$  і будемо множини  $M^{ii} = \{p^i, p^{i+s} \in M^i : B_{p^i}^i = B_{p^{i+s}}^{i+s} = 0\}$  тих пар відповідних пікселів, координати яких не належать до області рухомих об'єктів.

2. Якщо  $|M^{ii}| < 0,5 \cdot |M^i|$  або  $|M^{ii}| < 4$ , завершуємо алгоритм із результатом  $W^{i+s} = W^i$ .

3. Знаходимо набір  $M_W^i$  пар відповідних пікселів між панорамою  $W^i$  і кадром  $F^{i+s}$  і будемо множини

$$M_W^{ii} = \{p_W^i, p^{i+s} \in M_W^i : \exists p^i \in P : (p^i, p^{i+s}) \in M^{ii}\}.$$

4. На базі множини  $M_W^{ii}$  пар відповідних точок знаходимо матрицю  $H_W^i$  гомографії, що зіставляє пікселі кадру  $F^{i+s}$  та панорами  $W^i$ .

5. Рахуємо координати  $\ell_1^i = H_W^i \cdot (0, 0, 1)^T$ ,  $\ell_2^i = H_W^i \cdot (w-1, 0, 1)^T$ ,  $\ell_3^i = H_W^i \cdot (0, h-1, 1)^T$ ,  $\ell_4^i = H_W^i \cdot (w-1, h-1, 1)^T$  крайніх точок кадру  $F^{i+s}$  після застосування до них матриці  $H_W^i$ .

6. Для визначення множини  $P_W^i$  нової панорами  $W^{i+s}$  рахуємо величини  $x_{\min}^i = \min_{j=1,4} \frac{(\ell_j^i)_x}{(\ell_j^i)_z}$ ,

$$x_{\max}^i = \max_{j=1,4} \frac{(\ell_j^i)_x}{(\ell_j^i)_z}, \quad y_{\min}^i = \min_{j=1,4} \frac{(\ell_j^i)_y}{(\ell_j^i)_z}, \quad y_{\max}^i = \max_{j=1,4} \frac{(\ell_j^i)_y}{(\ell_j^i)_z}.$$

Позначмо  $P_W^{i+s} = \{1, \dots, \max(w^i, x_{\max}^i - x_{\min}^i)\} \times \{1, \dots, \max(h^i, y_{\max}^i - y_{\min}^i)\}$ .

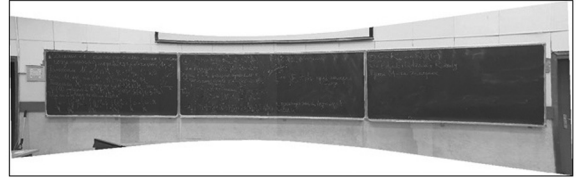


Рис. 4. Приклад отриманої панорами з автоматичним видаленням викладача (Посилання на оригінальне відео: [https://youtu.be/\\_pt7x37urx4](https://youtu.be/_pt7x37urx4))

7. Будемо нову панораму  $W^{i+s}$ . Для зручності позначмо обернену матрицю  $H' = (H_W^i)^{-1}$ , числа  $x'_{\min} = \min(x'_{\min}, 0)$ ,  $y'_{\min} = \min(y'_{\min}, 0)$  і відображення  $f^i: p \rightarrow \left( \left| \frac{(H' \cdot p)_x}{(H' \cdot p)_z} \right|, \left| \frac{(H' \cdot p)_y}{(H' \cdot p)_z} \right| \right)$ , що

перетворює координати з панорами до пікселів кадру за допомогою гомографії. Інтенсивність у пікселі  $p$  панорами  $W^{i+s}$  визначається за формулою

$$W^{i+s}(p) = \begin{cases} F^{i+s}(f(p)), & f(p) \in P, \\ W^i(p + (x'_{\min}, y'_{\min})), & p + (x'_{\min}, y'_{\min}) \in P, \\ 0, & p + (x'_{\min}, y'_{\min}) \notin P. \end{cases}$$

### Створення слайдів із відео

Після видалення викладача та суміщення кадрів ми створюємо слайди у вигляді коротких нотаток відеозапису лекції. Ми почергово беремо пари кадрів панорами, які необов'язково є сусідніми — для збільшення чутливості алгоритму створення нових слайдів можна використовувати кадри, різниця між індексами яких є більшою, тому що змін на дошці між сусідніми кадрами може не бути.

### Порівняння слайдів

Ми використовуємо оператор Лапласа [15] для знаходження перепадів яскравості та зменшення рівня шуму. У наших експериментах саме цей оператор надав кращі результати, ніж інші диференціальні оператори

$$L(x, y) = \nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}. \quad (6)$$

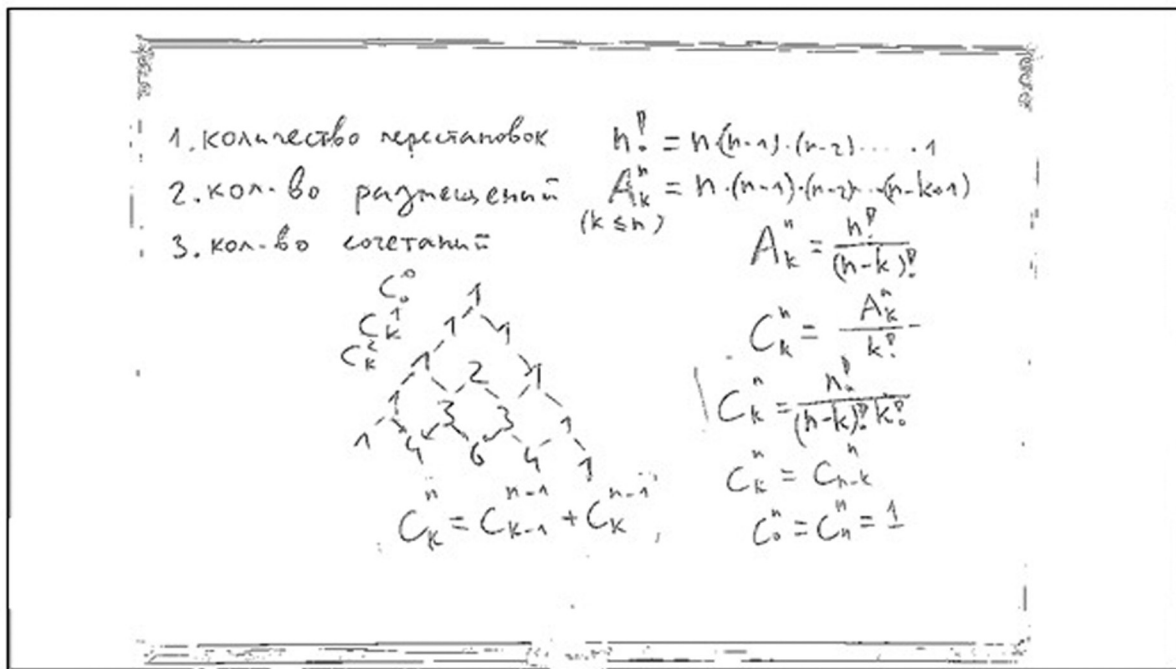


Рис. 5. Слайд після бінаризації Оцу (Посилання на оригінальне відео: <https://youtu.be/JXFz3gbnRMM>)

Для одержання маски всіх написів на дошці застосовуємо метод бінаризації Оцу [15]. Це — швидкий алгоритм, який, на наш погляд, дає прийнятну якість бінаризації для створення слайдів (рис. 5).

Не кожна панорама є новим слайдом. Коли зміни на дошці є незначними (наприклад, лектор дописав формулу), треба лише оновити наявний слайд. Якщо ж зміни є значними (наприклад, лектор стер попередні записи), треба створити новий слайд і працювати з ним. Щоб мінімізувати недоліки маски й отримувати слайди з новою інформацією на дошці, ми виділяємо нові написи за допомогою оператора Лапласа.

Слайд — це та панорама  $W^i$ , на якій відображено стан дошки напередодні стану з великою кількістю змін. Ми порівнюємо між собою не кожную панораму, а пропускаємо вказану користувачем кількість  $q$  панорам. Оскільки розміри слайду  $W^i$  і панорами  $W^{i+s-q}$  можуть бути різними, треба враховувати це у процесі їх порівняння. Для цього розраховуємо гори-

зонтальний зсув  $x_{\min}^{i'} = \sum_{j=1}^q \min(x_{\min}^{i+s-j}, 0)$  і вертикальний зсув  $y_{\min}^{i'} = \sum_{j=1}^q \min(y_{\min}^{i+s-j}, 0)$ . Користувач вказує кількість змін  $t \in [0; 1]$  між слайдом та панорамою, яка є вирішальною для прийняття рішення щодо створення нового слайду. Нехай  $L^i$  та  $L^{i+s-q}$  — панорами після обробки оператором Лапласа панорам  $W^i$  та  $W^{i+s-q}$  відповідно. Якщо виконується нерівність

$$\sum_{(x,y) \in P^i} |L^i(x,y) - L^{i+s-q}(x - x_{\min}^{i'}, y - y_{\min}^{i'})| > t \cdot |P^{i+s-q}|,$$

бінаризовану  $W^i$  додаємо до списку слайдів, а в іншому разі переходимо до перевірки слайду  $W^{i+s-q}$ .

### Висновки та подальша робота

У цій роботі описано принцип роботи інформаційної технології, яка в режимі реального часу перетворює відеозапис лекції на панорамне слайд-шоу. В експериментах, які здійснювалися на обчислювальній техніці з 16 ГБ опе-

ративної пам'яті та процесором *Intel® Core™ i7-8550U* з тактовою частотою 1,8 ГГц, обробка одного кадру розміром 1280 × 720 пікселів в одному потоці в середньому триває три секунди, що є добрим проміжком між сусідніми кадрами для запобігання пропускам цілі під час пошуку руху.

Оскільки кожен крок алгоритму потребує лише попередніх кадрів і поточного, його можна використовувати для оброблення відео лек-

ції під час трансляції за допомогою звичайного ноутбука. Подальшими кроками є автоматичне детектування дошки, розбиття її на сектори, розпізнавання написаного на дошці тексту та формул, а також можливість створювати слайди за допомогою камери, що постійно рухається, зокрема в умовах часткої зміни фокусної відстані, що на поточний момент дуже ускладнює процес напіваавтоматичного створення слайдів.

## REFERENCES

1. Patent Ukrainy № H04N7/00, G02B27/00, G06T7/215. Sposib peretvorennya videozapysu z doshky u slayd-shou / V.M Krygin. № u202100547; zavavl. 10.02.2021; opubl. 01.09.2021, Byul № 35. 4 p. [Патент України № H04N7/00, G02B27/00, G06T7/215. Спосіб перетворення відеозапису з дошки у слайд-шоу / В.М Кригін. № u202100547; заявл. 10.02.2021; опубл. 01.09.2021, Бюл № 35. 4 с.]
2. Zhang, Z., He, L.W., 2007. White board scanning and image enhancement. *Digital Signal Processing*, 17(2), pp. 414–432. [Online] Available at: <<https://doi.org/10.1016/j.dsp.2006.05.006>, <http://www.sciencedirect.com/science/article/pii/S1051200406000595>> [Accessed 27 May. 2021].
3. Gonzalez, A., Suh, B., Choi, E., 2012. Whiteboard disclosure using back ground subtraction and object tracking.
4. He, L.W., Zhang, Z., 2004. Real-time white board capture and processing using a video camera for teleconferencing. Tech. Rep. MSR-TR-2004-91, [Online] Available at: <<https://www.microsoft.com/en-us/research/publication/real-time-whiteboard-capture-and-processing-using-a-video-camera-for-teleconferencing/>> [Accessed 27 May. 2021].
5. Davila, K., Zanibbi, R., 2017. “White board video summarization via spatio-temporal conflict minimization”. 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 01, pp. 355–362. <https://doi.org/10.1109/ICDAR.2017.66>.
6. Pavlyuk, A.D, Krygin, V.M, Tkach, V.M., 2021. “Vidstezhennya periodychnoho rukhu ob yektiv zi statychnoyi kamery na prykladi videoihor”. *Teoretychni i prykladni problemy fizyky, matematyky ta informatyky*. 2021. Т 2. № 19. S. 135–137. [Online] Available at: <<https://drive.google.com/file/d/1V9zQB2Jvgcaw3o7eEkzMmviLRrhrHbpD/view/>> [Accessed 14 Dec. 2021]. [Павлюк А. Д, Кригін В. М, Ткач В. М Відстеження періодичного руху об'єктів зі статичної камери на прикладі відеоігор. Теоретичні і прикладні проблеми фізики, математики та інформатики. 2021. Т 2. № 19. С. 135–137. URL: <https://drive.google.com/file/d/1V9zQB2Jvgcaw3o7eEkzMmviLRrhrHbpD/view>. (дата звернення: 14 грудня 2021 р.).
7. Boykov, Y., Kolmogorov, V. 2004. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no 9, pp. 1124–1137, doi:10.1109/TPAMI.2004.60.
8. Lucas, B. D., Kanade, T., 1981. “An iterative image registration technique with an application to stereo vision”. Vol. 81. *Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2 (IJCAI'81)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA pp. 674–679.
9. Farneback, G., 2003. “Two-Frame Motion Estimation Based on Polynomial Expansion”. Bigun J., Gustavsson T. (eds) *Image Analysis*. SCIA 2003. Lecture Notes in Computer Science, vol 2749. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/3-540-45103-X\\_50](https://doi.org/10.1007/3-540-45103-X_50).
10. Lowe, D.G., 2004. “Distinctive image features from scale-invariant key points”. *Int. J. Comput. Vision* 60(2), pp. 91–110. <https://doi.org/10.1023/B:VISI.0000029664.99615.94>
11. Hartley, R., Zisserman, A., 2004. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edn. <https://doi.org/10.1017/CBO9780511811685>.
12. Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L., 2008. “Speeded-up robust features (SURF)”. *Computer vision and image understanding*, 110(3), pp. 346–359, <https://doi.org/10.1016/j.cviu.2007.09.014>.
13. Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. “ORB: An efficient alternative to SIFT or SURF”. In *International conference on computer vision*. pp. 2564–2571. IEEE Computer Society, <http://dblp.uni-trier.de/db/>

- conf/icc/icc2011.html# RubleeRKB11.
14. *Fischler, M.A., Bolles, R.C.*, 1981. "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography". *Commun. ACM*, 24(6), pp. 381-395. <https://doi.org/10.1145/358669.358692>.
  15. *Van der Walt, S., Sch nberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N.* et. all, 2014. Scikit-image: image processing in Python. *PeerJ*, 2, e453. <https://doi.org/10.7717/peerj.453>.

Received 14.12.2021

*M.K. Shylo*, master student, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Peremogy ave., 37, Kyiv, 03056, Ukraine, maksym.shylo.work@gmail.com

*V.M. Krygin*, Ph.D. student, junior researcher, International Research and Training Center for Information Technologies and Systems of the NAS of MES of Ukraine, Glushkov ave., 40, Kyiv, 03187, Ukraine, valeriy.krygin@gmail.com

#### CREATING SLIDES FROM VIDEO LECTURE

**Introduction.** Video recordings of lectures are no longer a rarity in the conditions of distance learning. Videos may be in an inconvenient format for students or contain different artefacts due to compression, camera quality, and other factors. It is useful to have a presentation of the study material, which contains only the text from the board because such a view of the material is most like the compendium.

**Purpose.** The item of our work is creating an algorithm for obtaining panorama slides without a teacher from video lecture. To develop this algorithm, we use the Boykov-Kolmogorov Max-flow algorithm for obtaining a mask of moving objects. In some video lectures, the camera is moved so that the teacher is in sight, so some of the recordings are not visible. To do this, we make frames stitching to get a panorama. To reduce duplication of slides due to camera shaking, we perform video-stabilization as a preprocessing step, then replace the pixels using the mask. Finally, we create slides by comparing changes in the frames, binarizing and denoising.

**Methods.** We used the Boykov-Kolmogorov Max-flow algorithm, SIFT, Laplace operator and Otsu binarization for developing information technology.

**Results.** As a result, we get panorama slides with the extracted text or drawings from the board, which will help to simplify the creation of e-learning materials for both new and existing lecture recordings. Teachers will also be able to quickly provide lecture material to students even if they teach several complex subjects.

**Conclusion.** We have developed an algorithm and implemented information technology for obtaining slides from video-lecture. Now, a teacher can give students a short content of video material. The next steps are automatic detection of the board, division of the board into sections, and recognition of the text and formulas written on the board.

**Keywords:** *intellectual video processing, images processing, video stabilization.*