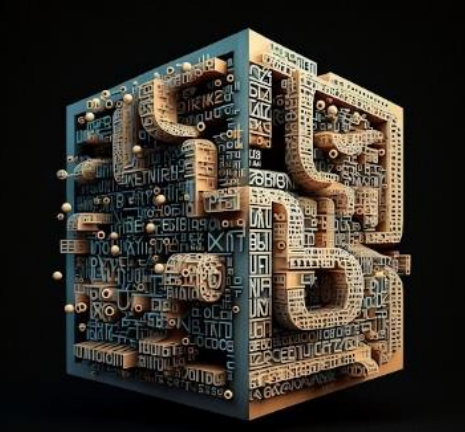


**Міністерство освіти та науки України
Харківський національний економічний
університет імені Семена Кузнеця**



**НАВЧАЛЬНО – НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**
кафедра: Мультимедійних систем і технологій

дисципліна
"ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ"
викладач: доцент, к.т.н.
Токарєв Володимир Володимирович



Тема.

Приклади розрахункових задач з відеоконтенту. RAW-кадр.

Задача_№1

Обчислити розмір одного **RAW**-кадру для цифрового відео, якщо:

-роздільна здатність = 1920×1080 ;

-глибина кольору = 8bit;

-кількість каналів = 3 (RGB).

Відповідь записати у мегабайтах.

Розв'язання:

$$\text{Size} = W \times H \times \text{BitDepth} \times \text{Channels}$$

де:

Size - обсяг одного RAW-кадру;

W - ширина (Width);

H - висота (Height);

BitDepth - глибина кольору;

Channels - кількість каналів.

Підставимо:

$$\text{Size} = 1920 \times 1080 \times 8 \times 3 = 49766400 \text{ bit}$$

Переведемо у байти: $\text{Size} = 49\,766\,400 / 8 = 6\,220\,800 \text{ Byte}$

Переведемо в мегабайти: $\text{Size} = 6\,220\,800 / 1048576 \approx 5.93 \text{ MB}$
(1024×1024)

Задача_№2



Обчислити розмір одного **RAW**-кадру для 4K-відео:

- роздільна здатність = 3840×2160 ;
- глибина кольору = 10bit;
- кількість каналів = 3 (RGB).

Відповідь записати у мегабайтах.

Розв'язання:

$$\text{Size} = W \times H \times \text{BitDepth} \times \text{Channels}$$

де:

Size - обсяг одного RAW-кадру;

W - ширина (Width);

H - висота (Height);

BitDepth - глибина кольору;

Channels - кількість каналів.

Підставимо:

$$\text{Size} = 3840 \times 2160 \times 10 \times 3 = 248832000 \text{bit}$$

Переведемо у байти: $\text{Size} = 248832000 / 8 = 31104000 \text{Byte}$

Переведемо в мегабайти: $\text{Size} = 31104000 / 1048576 = 29.66 \text{MB}$

(1024×1024)

Задача_№3

Обчислити обсяг 1 хвилини цифрового відео без компресії, якщо:

-розмір одного кадру = 6.2MB;

-частота кадрів = 30fps.

Відповідь записати у гігабайтах.

Розв'язання:

$$\text{VideoSize} = \text{FrameSize} \times \text{FPS} \times \text{Time}$$

де:

VideoSize - обсяг відео;

FrameSize - розмір кадру;

FPS - частота кадрів у секунду;

Time – тривалість.

1 хвилина:

$$\text{Time} = 60\text{s}$$

Підставимо:

$$\text{VideoSize} = 6.2 \times 30 \times 60 = 11160\text{MB}$$

$$\text{VideoSize} = 11160/1024 \approx 10.9\text{GB}$$

Задача_№4

У 20 секундах цифрового відео міститься 500 кадрів. Знайти FPS.

Розв'язання:

$$FPS = \frac{N}{T}$$

де:

N — кількість кадрів;

T — час у секундах;

FPS - частота кадрів у секунду.

Підставимо:

$$FPS = \frac{500}{20} = 25$$

Відповідь. FPS = 25.

Задача_№5

Визначити тривалість одного кадру для цифрового відео з частотою 50 fps. Відповідь записати у мілісекундах.

Розв'язання. Період одного кадру (час між кадрами):

$$\Delta t = \frac{1}{FPS}$$

де:

Δt — період одного кадру;

FPS – частота кадрів у секунду.

Підставимо:

$$\Delta t = \frac{1}{50} = 0.02s$$

Переведемо у мілісекунди: $0.02s \times 1000 = 20ms$

Відповідь. Тривалість одного кадру для цифрового відео з частотою 50 fps = 20ms.

Задача_№6

Скільки кадрів міститиме відео тривалістю 3 хвилини при 30fps?

Розв'язання.

$$N = FPS \times T$$

де:

N — кількість кадрів;

T — час у секундах;

FPS - частота кадрів у секунду.

Переведемо 3хв.у секунди: $3m \times 60s = 180s$

Підставимо 3 хв.: $N = 30 \times 180 = 5400$

Відповідь. Відео тривалістю 3 хвилини при 30fps міститиме = 5400 кадрів.

Задача_№7

Монітор має:

- роздільну здатність 1920×1080;
- діагональ 24 дюйми.

Знайти PPI - пікселів на дюйм.

Розв'язання.

$$PPI = \frac{\sqrt{W^2 + H^2}}{Diagonal}$$

де:

W - ширина (Width) у пікселях;

H - висота (Height) у пікселях;

Diagonal - діагональ екрана (в дюймах).

Підставимо:

$$PPI = \frac{\sqrt{1920^2 + 1080^2}}{24} \approx 91.8$$

Відповідь. PPI=92.

Задача_№8

Зображення має ширину 3000px і друкується на ширину 10 дюймів.
Знайти DPI.

Розв'язання. Якщо ви маєте зображення шириною **W_{px}** пікселів і хочете надрукувати його шириною **W_{inch}** дюймів, тоді потрібна щільність:

$$DPI = \frac{W_{px}}{W_{inch}}$$

де:

W_{px} - ширина зображення у пікселях;

W_{inch} - ширина зображення у дюймів;

DPI - щільність точок на дюйм.

Підставимо:

$$DPI = \frac{3000}{10} = 300$$

Відповідь. DPI = 300.

Задача_№9

Скільки можливих кольорів має RGB-зображення при 8bit на канал?

Розв'язання:

$$\mathit{Colors} = 2^{3b}$$

де:

b – кількість біт які виділені на канал;

Підставимо:

$$\mathit{Colors} = 2^{3*8} = 2^{24} = 16777216$$

Відповідь. 16 777 216 кольорів.

Задача_№10

Порівняти 8-bit і 10-bit: у скільки разів 10-bit має більше рівнів на канал?

Розв'язання:

$$\mathit{Ratio} = \frac{2^{10}}{2^8}$$

де:

Ratio – співвідношення;

Підставимо:

$$\mathit{Ratio} = \frac{1024}{256} = 4$$

Відповідь. У 4 рази більше.

Задача_№11

Потрібно створити YouTube Short тривалістю 45 секунд у форматі 1920×1080, 30fps. Скільки кадрів міститиме такий ролик?

Розв'язання:

$$N = FPS \times T$$

де:

N — кількість кадрів;

T — час у секундах;

FPS - частота кадрів у секунду.

Підставимо:

$$N = 30 \times 45 = 1350$$

Відповідь. 1350 кадрів.

Задача_№12

Потрібно оцінити, у скільки разів збільшиться потреба у відеоданих при переході з 1080p 30fps на 4K 60fps, якщо складність сцени не змінюється.

Розв'язання:

$$\text{Bitrate} \propto (W \times H) \times \text{FPS}$$

де:

W - ширина (Width);

H - висота (Height);

FPS - частота кадрів у секунду.

1.Порівняємо кількість пікселів:	1080p: $1920 \times 1080 = 2.07\text{Мп}$
	4K: $3840 \times 2160 = 8.29\text{Мп}$

Відношення: $8.29/2.07 \approx 48.29/2.07 \approx 4$

2.Порівняємо FPS: $60/30 = 2$

3.Загальний коефіцієнт: $4 \times 2 = 8$

Відповідь. Потреба в даних збільшується приблизно у 8 разів.

Задача_№13.

Потрібно змонтувати скринкаст: 1920×1080, 30fps, 8Mbps, 15хвилин.
Знайти приблизний розмір файлу у МВ.

Розв'язання:

$$\text{Size} = \text{Bitrate} \times \text{Time}$$

де:

Bitrate - обсяг даних який містить 1 секунда відео;

Time – тривалість.

1.Переклад у МВ:

$$\text{МВ} = \text{Mbit} / 8$$

2.Переклад у s:

$$15\text{m} \times 60\text{s} = 900\text{s}$$

3.Підставимо:

$$8\text{Mbps} \times 900\text{s} = 7200\text{Mbit}$$

4.У мегабайтах:

$$7200 / 8 = 900\text{МВ}$$

Відповідь. 900МВ.



Association for
Computing Machinery

Advancing Computing as a Science & Profession

Задача_№14

Одне фото у відеоряді показується **5 секунд**. Відео має 30fps.
Скільки кадрів утворюється на основі цього фото?

Розв'язання:

$$N = T \times FPS$$

де:

N - скільки кадрів у секунду;

FPS - частота кадрів у секунду;

T – час у секундах.

Підставимо:

$$N = 5s \times 30fps = 150$$

Відповідь. 150 кадрів.



Association for
Computing Machinery

Advancing Computing as a Science & Profession

Задача_№15

Є 12 фотографій. Кожна показується по 4 секунди.
Знайти загальну тривалість відеоряду.

Розв'язання:

$$T = \sum t_i$$

де:

T – час у секундах;

t - час у секундах одного фото.

Підставимо:

$$T = 12 \times 4s = 48s$$

Відповідь. 48 секунд.

Задача_№16

Загальний бітрейт відео дорівнює 5Mbps, аудіо займає 192kbps.
Знайти відеобітрейт.

Розв'язання:

$$B_{video} = B_{total} - B_{audio}$$

де:

B_{total} – загальний бітрейт;

B_{audio} – аудіо бітрейт.

Переведемо аудіо в Mbps:

$$192 \text{ Kbps} = 0.192 \text{ Mbps}$$

Підставимо:

$$B_{video} = 5 \text{ Mbps} - 0.192 \text{ Mbps} \approx 4.81 \text{ Mbps}$$

Відповідь. Відеобітрейт $\approx 4.81 \text{ Mbps}$

Задача_№17

Потрібно вмістити **1 годину** відео в обсяг **2 GB**.
Знайти середній загальний бітрейт.

Розв'язання:

$$\textit{Bitrate} = \frac{\textit{Size}}{\textit{Time}}$$

де:

Size – розмір файлу;

Time – тривалість.

1.Обсяг:

$$2\text{GB} = 2048\text{MB}$$

2.У мегабітах:

$$2048 \times 8 = 16384\text{Mbit}$$

3.Часу s:

$$1 \text{ год} = 3600\text{s}$$

Підставимо:

$$\textit{Bitrate} = \frac{16384}{3600} \approx 4.55\text{Mbps}$$

Відповідь. Відеобітрейт $\approx 4.55\text{Mbps}$.



Задача_№18

Обчислити розмір відеофайлу тривалістю **10 хвилин**, якщо бітрейт дорівнює **6Mbps**.

Розв'язання:

$$\text{Size} = \text{Bitrate} \times \text{Time}$$

де:

Bitrate - обсяг даних який містить 1 секунда відео;

Time – тривалість.

1.Переклад у МВ:

$$\text{MB} = \text{Mbit} / 8$$

2.Переклад у s:

$$10 \times 60 = 600\text{s}$$

3.Підставимо:

$$6\text{Mbps} \times 600\text{s} = 3600\text{Mbit}$$

4.У мегабайтах:

$$3600\text{Mbit} / 8 = 450\text{MB}$$

Відповідь. Відеобітрейт \approx 450MB.

Задача_№19

Для того ж кадру 720×576 потрібно забезпечити відображення 16:9.
Знайти PAR.

Розв'язання:

$$PAR = \frac{DAR \times H}{W}$$

де:

PAR - співвідношення ширини пікселя до висоти пікселя;

DAR – співвідношення сторін зображення на екрані;

W - ширина (Width);

H - висота (Height).

$$DAR = 16/9$$

Підставимо:

$$PAR = \frac{\frac{16}{9} \times 576}{720} = \frac{1024}{720} = 1.422$$

Відповідь. PAR ≈ 1.422.

Задача_№20

Для кадру **720×576** потрібно забезпечити відображення **4:3**.
Знайти PAR.

Розв'язання:

$$PAR = \frac{DAR \times H}{W}$$

де:

PAR - співвідношення ширини пікселя до висоти пікселя;

DAR – співвідношення сторін зображення на екрані;

W - ширина (Width);

H - висота (Height).

$$DAR = 4/3$$

Підставимо:

$$PAR = \frac{\frac{4}{3} \times 576}{720} = \frac{1024}{720} = 1.067$$

Відповідь. PAR \approx 1.067.

Задача_№21

Для пікселя: $R=120$, $G=200$, $B=80$ обчислити компоненту яскравості Y .

Розв'язання:

$$Y = 0.2126R + 0.7152G + 0.0722B$$

Підставимо:

$$Y = 0.2126 \cdot 120 + 0.7152 \cdot 200 + 0.0722 \cdot 80$$

$$Y = 25.512 + 143.04 + 5.776 = 174.328$$

Відповідь. Компонент яскравості $Y \approx 174.328$.

Задача_№22

Необхідно визначити приблизний розмір відеофайлу, якщо, роздільна здатність відео: 1920×1080, частота кадрів: 60fps, середній бітрейт відео: 12Mbps, тривалість відео: 8 хвилин. Потрібно знайти. Розмір відеофайлу в: мегабітах, мегабайтах, гігабайтах.

Розв'язання:

$$\text{Size} = \text{Bitrate} \times \text{Time}$$

де:

Bitrate - обсяг даних який містить 1 секунда відео;

Time – тривалість.

1.Переведемо час у секунди: $8\text{ь} \times 60\text{s} = 480\text{s}$

2.Обчислимо розмір у мегабітах: $12\text{Mbps} \times 480\text{s} = 5760\text{Mbit}$

3.Переведемо у мегабайти: $5760\text{Mbit}/8 = 720\text{MB}$

4.У гігабайтах: $720/1024 = 0.703\text{GB}$

Відповідь. 5760 Mbit, 720MB, 0.7GB.

Задача_№23

Відео має параметри, тривалість: 15 хвилин, розмір файлу: 1.2 GB,
Необхідно знайти середній бітрейт відеопотоку в Mbps.

Розв'язання:

$$\mathit{Bitrate} = \frac{\mathit{Size}}{\mathit{Time}}$$

де:

Size – розмір файлу;

Time – тривалість.

1.Переведемо гігабайти у мегабайти: $1.2\text{GB} \times 1024 = 1228.8\text{MB}$

2.Переведемо мегабайти у мегабіти: $1228.8\text{MB} \times 8 = 9830.4\text{Mbit}$

3.Переведемо час у секунди: $15\text{m} \times 60\text{s} = 900\text{s}$

Підставимо: $9830.4\text{Mbit}/900\text{s} = 10.92\text{Mbit}$

Відповідь. 10.92Mbit.

Задача_№24

Є відеоряд із 18 фотографій. Кожне фото показується по 6 секунд. Відео створюється з частотою 25fps. Необхідно знайти: загальну тривалість ролика, загальну кількість кадрів, кількість кадрів, що відповідає одному фото.

Розв'язання:

$$T = n \times t$$

де:

n — кількість фото;

t — час показу одного фото.

Кількість кадрів для одного фото:

$$N_1 = t \times \text{FPS}$$

Загальна кількість кадрів:

$$N = T \times \text{FPS}$$

Знайдемо загальну тривалість:

$$T = 18 \times 6s = 108s$$

Кількість кадрів для одного фото:

$$N_1 = 6s \times 25\text{fps} = 150$$

Загальна кількість кадрів:

$$N = 108s \times 25\text{fps} = 2700$$

Відповідь. $T=108s$, $N_1=150$, $N=2700$.

Задача_№25

Потрібно порівняти два відеоформати за обсягом піксельних даних в секунду: формат А: 1280×720, 30fps. Формат В: 3840×2160, 60fps. Необхідно визначити: скільки пікселів обробляється за секунду у кожному форматі, у скільки разів формат В “важчий” за формат А.

Розв’язання:

$$\textit{Bitrate} \propto (W \times H) \times \textit{FPS}$$

де:

W - ширина (Width);

H - висота (Height);

FPS - частота кадрів у секунду.

$$\text{Формат А:} \quad (1280 \times 720) \times 30\text{fps} = 27648000$$

$$\text{Формат В:} \quad (3840 \times 2160) \times 60\text{fps} = 497664000$$

$$\text{Порівняння форматів:} \quad 497664000/27648000 = 18$$

Відповідь. Формат А = 27 648 000px/s, Формат В = 497664000px/s,
Формат В приблизно у 18 разів важчий за Формат А.

Чим відрізняється готовий кадр від RAW-кадру

RAW-кадр і готовий кадр — це два різні стани одного й того самого зображення в цифровому відео або цифровій фотографії.

1. RAW — (от англ. raw — «сирий», «необроблений»), це сирі дані з сенсора.

2. Готовий кадр — це вже оброблене зображення, придатне для перегляду, кодування, передачі або монтажу.

Розглянемо докладніше кожен вид кадру цифрового відео

RAW-кадр — це набір первинних цифрових даних, які сенсор камери зчитав зі сцени до повноцінної візуальної обробки.

Готовий кадр — це результат обробки **RAW**-даних.

Головна різниця в одній фразі RAW-кадр містить максимум первинної інформації, а готовий кадр містить уже інтерпретоване і візуально підготовлене зображення.

Чим відрізняється готовий кадр від RAW-кадру



RAW-кадр — це як негатив у плівковій фотографії.

Готовий кадр — це вже надрукована фотографія.

RAW-кадр це - більше даних більше можливостей редагування «пласке» зображення.

Готовий кадр це - гарний вигляд малий розмір готовність до показу. Де виникає **RAW**-кадр, а де — готовий кадр. Повний тракт можна записати так:

Scene → **Lens** → **Sensor** → **ADC** → **RAW** → **ISP** → **Ready Frame**

де:

- після сенсора та **АЦП** з'являється **RAW**-кадр;
- після **ISP** з'являється готовий кадр.

Що саме містить **RAW**-кадр

RAW-кадр це не “звичайна картинка”, яку зручно одразу дивитися. Він містить:

- сирі значення сенсора;
- часто мозаїчний **Bayer**-патерн;
- ширший динамічний діапазон;
- більше інформації про світло і колір;
- мінімум або відсутність фінальної корекції.

Тобто **RAW**-кадр:

- ще не має повного RGB у кожному пікселі;
- ще не має правильного балансу білого;
- ще може містити шум;
- ще не адаптований до людського сприйняття і дисплея.

Що саме містить **готовий кадр**

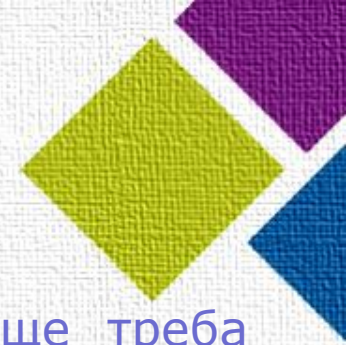
Готовий кадр уже має:

- повний колір у кожному пікселі;
- відкоригований баланс білого;
- скориговану яскравість і контраст;
- зменшений шум;

• **готовий колірний простір;**

• **формат, зручний для кодування або відображення.**





Тобто **готовий кадр**:

- вже можна показувати на екрані;
- вже можна стискати кодеком;
- вже можна вставляти у відеопотік.

RAW-кадр має більше "сирого потенціалу", але ним ще треба правильно скористатися.

Готовий кадр — уже фінальний результат. **RAW**-кадр часто містить дані з матриці **Байєра**, наприклад:

$$\begin{matrix} G & R \\ B & G \end{matrix}$$

Тобто кожен сенсорний елемент бачить лише частину кольору.

У **готовому кадрі** після Дебайєризації кожен піксель уже має повні значення **RGB**.

Дебайєризація або Demosaicing — це критично важливий етап цифрової обробки сигналів, що полягає у відновленні повноколірного зображення з "сирих" даних **RAW**, отриманих з одноматричного сенсора камери. Оскільки більшість сучасних цифрових сенсорів не можуть фіксувати всі три основні кольори **RGB** у кожному пікселі одночасно, вони використовують спеціальний колірний фільтр. Найпоширенішим є фільтр Байєра.

Типовий сенсор камери складається з фотодіодів, які чутливі лише до інтенсивності світла, але не до його спектрального складу (кольору). Щоб отримати колір, поверх сенсора накладається Color Filter Array-CFA. У структурі фільтра Байєра:

- 50% пікселів покриті зеленими фільтрами (оскільки людське око найбільш чутливе до зеленого спектра, що визначає яскравість та деталізацію);

- 25% — червоними;

- 25% — синіми.

В результаті кожен фізичний піксель на етапі захоплення світла "знає" значення лише одного кольору. Demosaicing — це математичний процес інтерполяції, який "вгадує" два відсутні колірні компоненти для кожного пікселя, аналізуючи значення сусідніх комірок.

Demosaicing — це міст між фізичним захопленням фотонів сенсором та цифровим кольоровим зображенням, яке ми бачимо на екрані. Від ефективності цього процесу безпосередньо залежить реальна роздільна здатність камери (яка завжди трохи нижча за кількість фізичних мегапікселів через необхідність інтерполяції).

Приклад.

Зйомка при жовтому світлі лампи.

- **RAW**-кадр може виглядати жовтуватим, але його можна гнучко скоригувати;

- **Готовий кадр** уже буде “нормалізований”, якщо камера правильно визначила баланс білого.

Формально що відбувається між **RAW**-кадром і **готовим кадром** можна записати так:

$$\text{Ready Frame} = \text{ISP}(\text{RAW})$$

Готовий кадр є результатом функції процесора сигналів зображення (ISP), застосованої до сирих даних (RAW).

1.RAW - аргумент функції. Це вихідні дані («сировина»), зняті безпосередньо з сенсора камери. Вони містять лише значення інтенсивності світла для кожного пікселя, накритими фільтром **Байєра**. Це ще не зображення, а набір електричних сигналів, оцифрованих АЦП (аналого-цифровим перетворювачем).

2.ISP Оператор/Функція. **Image Signal Processor** - Процесор сигналів зображення. У цій формулі **ISP** виступає як складна функція, що містить у собі десятки алгоритмів. Це «чорна скринька».

3.Ready Frame – Результат. Фінальний масив пікселів у стандартному колірному просторі (наприклад, **RGB**), готовий для відображення на дисплеї або стиснення у відеофайл.

Щоб знак рівності у формулі став можливим, **ISP** виконує послідовність операцій, яку можна представити як композицію підфункцій:

- відновлення кольору (те, що ми розбирали вище);
- корекція колірної температури;
- усунення цифрових шумів;
- приведення кольорів до реалістичного вигляду та корекція яскравості під людський зір;
- посилення контурної різкості.

Ця формула підкреслює дві критичні концепції цифрового відео.

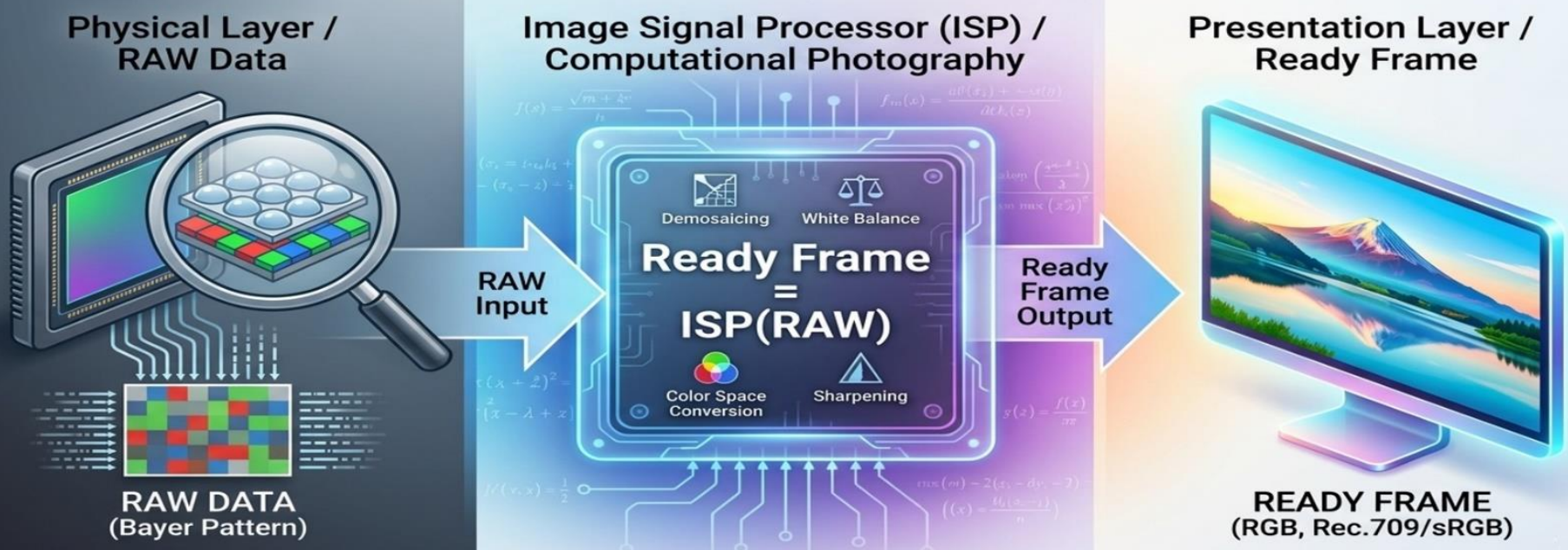
1.Незворотність. Після того, як функція **ISP** відпрацювала і видала **Ready Frame** (особливо якщо це 8-бітний **JPEG** або стиснене відео), значна частина метаданих **RAW** втрачається. Не можна ефективно виконати зворотну операцію:

$$\mathbf{RAW} = \mathbf{ISP}^{-1} (\mathbf{Ready\ Frame})$$

2.Детермінізм. Якість «Готового кадру» залежить не лише від «якості» **RAW** (сенсора/об'єктива), а й від досконалості алгоритмів **ISP**. Саме тому різні смартфони з однаковими сенсорами видають абсолютно різну картинку.

Ця формула визначає перехід від апаратного рівня захоплення даних (**Physical Layer**) до програмно-візуального представлення контенту (Presentation Layer). Вона ілюструє роль обчислювальної фотографії як ключового посередника у формуванні сучасного цифрового сигналу».

Digital Video Pipeline: RAW to Ready Frame



Приклад_1.

Фото людини на вулиці. Сцена - людина стоїть на сонці.

RAW-кадр може містити:

- дуже яскраве небо;
- тіні на обличчі;
- нейтральні, але "сірі" кольори;
- великий запас для редагування.

Готовий кадр після обробки може містити:

- шкіра має приємніший колір;
- небо має нормальний контраст;
- тіні трохи підняті;
- шум мінімальний.



ПОРІВНЯННЯ КАДРІВ: RAW ТА ГОТОВИЙ КАДР

1. RAW-КАДР

Дуже яскраве небо (вибите)

Глибокі тіні на обличчі (без деталей)

Нейтральні, сірі кольори



Великий запас для редагування



2. ГОТОВИЙ КАДР (Після Обробки)

Приємний, здоровий колір шкіри

Небо з нормальним контрастом та деталями хмар

Тіні розм'якшені (виявлені деталі)

Шум мінімальний (чиста картинка)



Обробка



Приклад_2.

Фотографія на нічній вулиці. Сцена — людина йде повз яскраву вивіску неонових магазину.

RAW-кадр може містити:

- дуже багато цифрового шуму (особливо в тінях);
- розмите обличчя через довгу витримку (**motion blur**);
- темні, «засмічені» неонові кольори;
- видимі артефакти стиснення в темних зонах (якщо **RAW** стиснений).

Готовий кадр після обробки може містити:

- ефективне шумозаглушення (**denoising**) із збереженням деталей;
- підвищену різкість (**sharpening**), яка «заморожує» рух;
- чисті, насичені, «світящі» неонові відтінки;
- вирівняний баланс білого для складного освітлення.

ПОРІВНЯННЯ КАДРІВ (НІЧ): ТА ГОТОВИЙ КАДР

1. RAW-КАДР (НІЧ)



Величезна кількість цифрового шуму (зерно)

Motion blur на обличчі (розмите)

Тьмяні та «засмічені» кольори

Великий запас для видалення шуму



2. ГОТОВИЙ КАДР (Після Обробки)



Чітке обличчя та деталі шкіри

Чисті та насичені неонові відтінки

Різкість відновлена (заморожено рух)

Мінімальний шум (чиста картинка)

Розширений діапазон, чиста гістограма



Обробка

Зазвичай користувач бачить лише готовий кадр.

Pipeline послідовність дій:

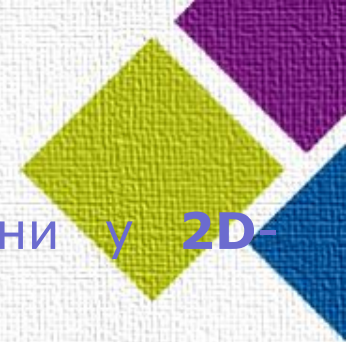
- сенсор;
- ISP;
- HDR;
- шумозаглушення;
- процес збільшення чіткості;
- готове відео.

Цифровий пристрій часто дуже активно “покрощує” кадр автоматично.

Що таке **Pipeline**

Pipeline - конвеєр у контексті **цифрового відео** - це не просто послідовність дій, а архітектурний принцип організації обчислень, де завдання розбивається на окремі етапи, що виконуються послідовно або паралельно.

Pipeline - це модель обробки даних, у якій вихідний результат одного етапу є вхідними даними для наступного. Головна мета конвеєра - забезпечити безперервний потік обробки інформації, мінімізуючи прості обчислювальних потужностей.



А. Графічний конвеєр

Процес перетворення математичного опису **3D**-сцени у **2D**-зображення на екрані.

Input: Координати вершин, текстури, джерела світла.

Етапи:

1. Трансформація геометрії.
2. Освітлення.
3. Растеризація (перетворення векторів у пікселі).
4. Фрагментне затінення (обчислення кольору пікселя).

Output: Готовий кадр (Ready Frame).

Б. Конвеєр кодування

Процес стиснення відеопотоку наприклад, у кодек H.264 або HEVC.

Етапи:

Оцінка руху: Пошук рухомих об'єктів між кадрами.

Перетворення (DCT): Дискретне косинусне перетворення.

Квантування: Зниження точності для зменшення обсягу даних.

Ентропійне кодування: Фінальне бінарне стиснення.

Ключові характеристики **Pipeline**

1.Latency – Затримка. Час, необхідний для повного проходження одного елемента через усі етапи конвеєра.

2.Throughput - Пропускна здатність. Кількість елементів (кадрів), які виходять із конвеєра за одиницю часу (наприклад, **60FPS**).

3.Ефективний Pipeline дозволяє мати високий **Throughput**, навіть якщо **Latency** окремого кадру є значною. Це досягається за рахунок того, що поки один кадр записується на диск, наступний уже проходить колірну корекцію, а третій — рендериться.

4.Pipeline - це фундамент цифрового відео. Без нього неможливий ані **real-time** рендеринг у відеоіграх, ані трансляція відео високої чіткості в мережі, оскільки саме конвеєрна архітектура дозволяє обробляти гігабайти даних у реальному часі.